Module 2:

Data Link Layer and Medium Access Sub Layer

Error Detection & Correction

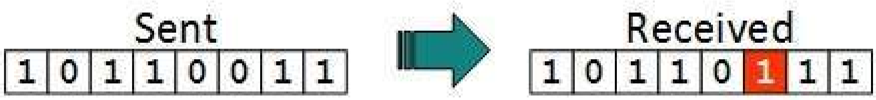
There are many reasons such as noise, crosstalk etc., which may help data to get corrupted during transmission. The upper layers work on some generalized view of network architecture and are not aware of actual hardware data processing. Hence, the upper layers expect error-free transmission between the systems. Most of the applications will not function expectedly if they receive erroneous data. Applications such as voice and video may not be that affected and with some errors they may still function well.

Data-link layer uses some error control mechanism to ensure that frames (data bit streams) are transmitted with certain level of accuracy. But to understand how errors is controlled, it is essential to know what types of errors may occur.

Types of Errors

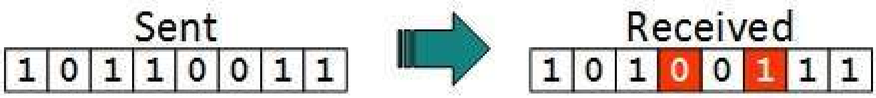
There may be three types of errors:

- **Single bit error**



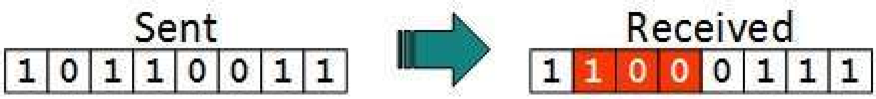
In a frame, there is only one bit, anywhere though, which is corrupt.

- **Multiple bits error**



Frame is received with more than one bits in corrupted state.

- **Burst error**



Frame contains more than1 consecutive bits corrupted.

Error control mechanism may involve two possible ways:

- Error detection
- Error correction

Error Detection Method

A condition when the receiver's information does not match with the sender's information. During transmission, digital signals suffer from noise that can introduce errors in the binary bits travelling from sender to receiver. That means a 0 bit may change to 1 or a 1 bit may change to 0.

Error Detecting Codes (Implemented either at Data link layer or Transport Layer of OSI Model
Whenever a message is transmitted, it may get scrambled by noise or data may get corrupted. To avoid this, we use error-detecting codes which are additional data added to a given digital message to help us detect if any error has occurred during transmission of the message.

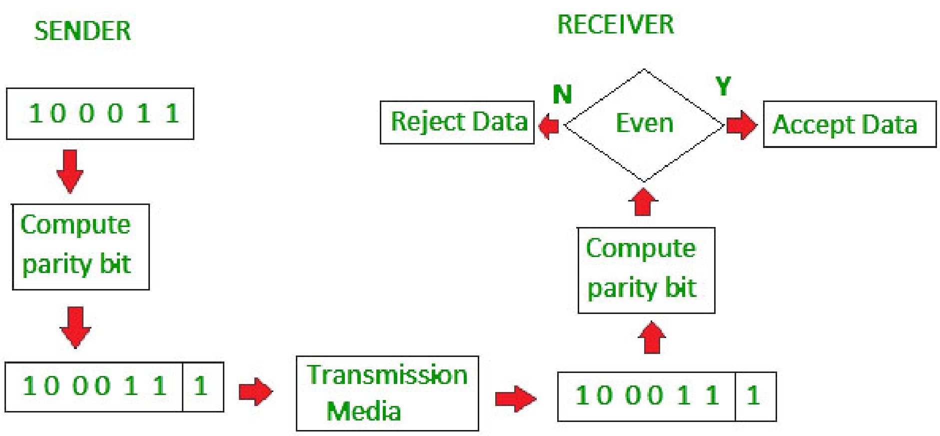
Basic approach used for error detection is the use of redundancy bits, where additional bits are added to facilitate detection of errors.
Some popular techniques for error detection are:

1. Simple Parity check
2. Two-dimensional Parity check
3. Checksum
4. Cyclic redundancy check (CRC)
5 Longitudinal Redundancy Check (LRC)

1. Simple Parity check:
Blocks of data from the source are subjected to a check bit or parity bit generator form, where a parity of :

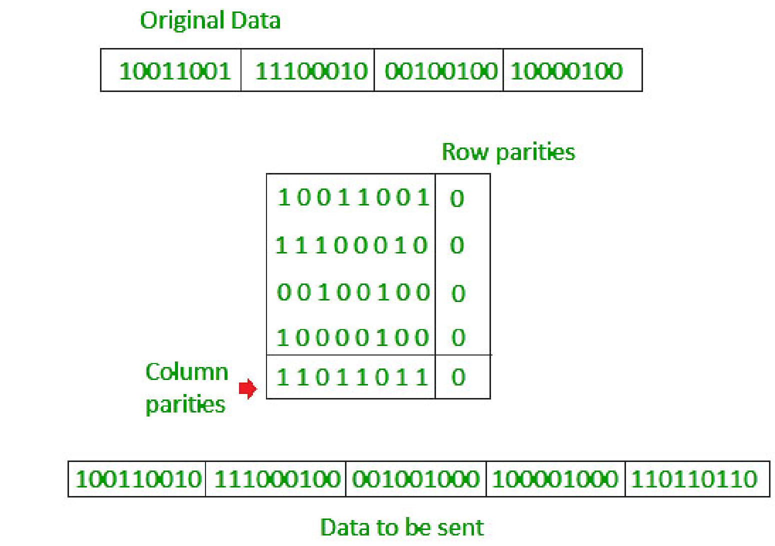
- 1 is added to the block if it contains odd number of 1's, and
- 0 is added if it contains even number of 1's

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



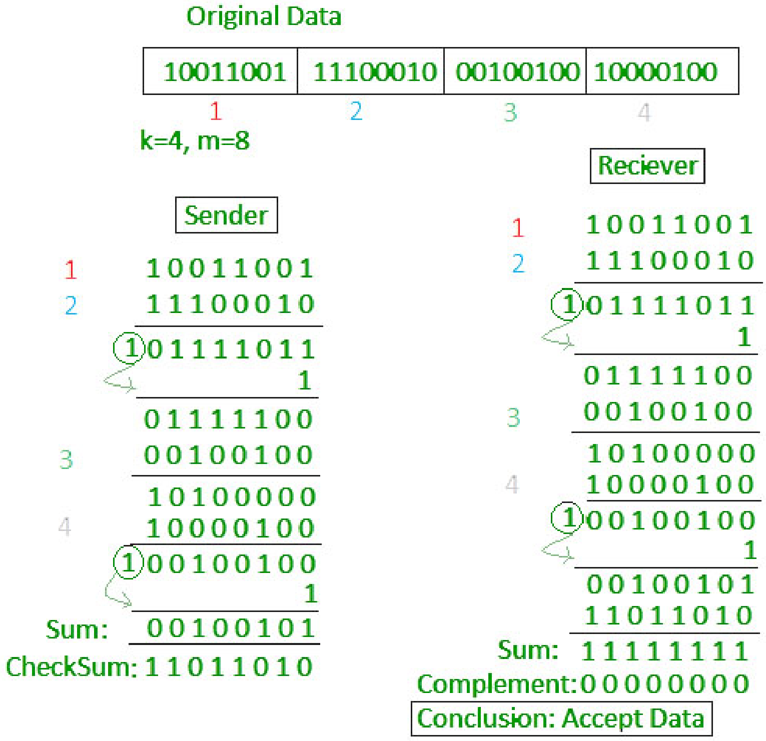
2. Two-dimensional Parity check

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.



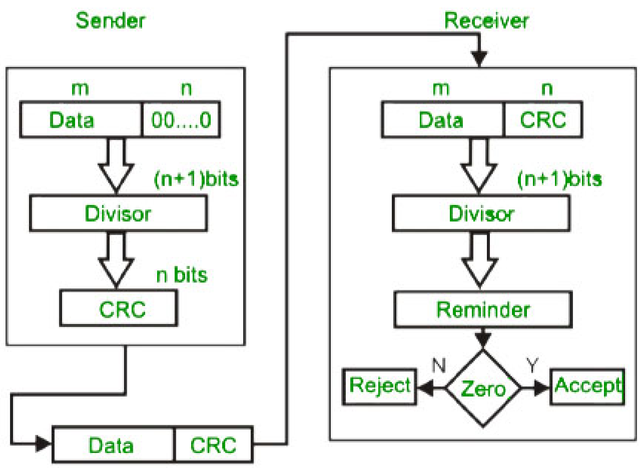
3. Checksum

- In checksum error detection scheme, the data is divided into k segments each of m bits.
- In the sender's end the segments are added using 1's complement arithmetic to get the sum. The sum is complemented to get the checksum.
- The checksum segment is sent along with the data segments.
- 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.

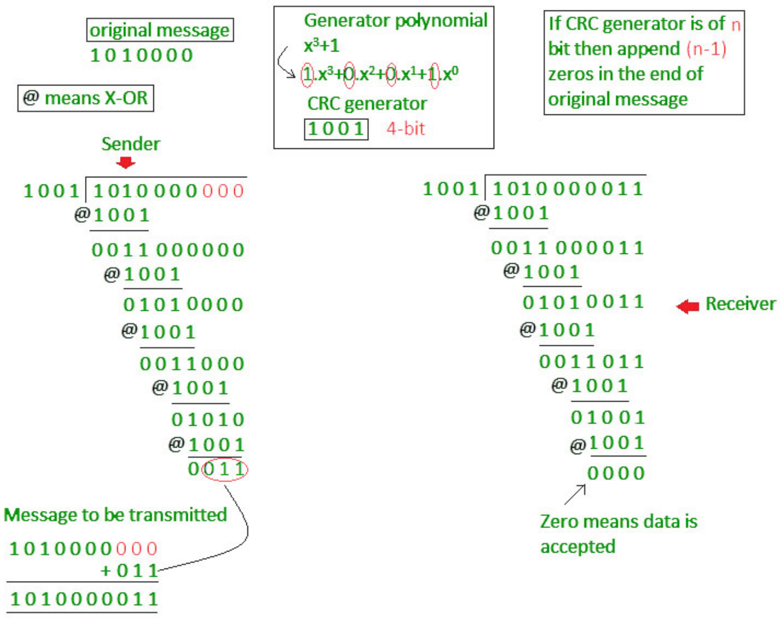


4. Cyclic redundancy check (CRC)

- Unlike 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 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.



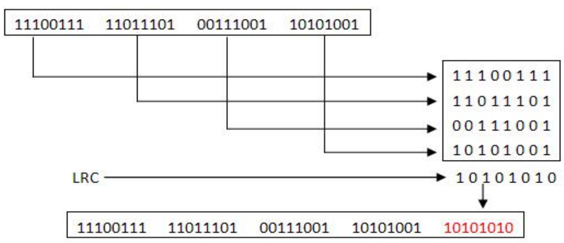
Example :



*5. **Longitudinal Redundancy Check:***

In longitudinal redundancy method, a BLOCK of bits are arranged in a table format (in rows and columns) and we will calculate the parity bit for each column separately. The set of these parity bits are also sent along with our original data bits.

Longitudinal redundancy check is a bit-by-bit parity computation, as we calculate the parity of each column individually.This method can easily detect burst errors and single bit errors and it fails to detect the 2 bit errors occurred in same vertical slice.



Original data message including LRC

Error Correction Method

Error Correction codes are used to detect and correct the errors when data is transmitted from the sender to the receiver.

Error Correction can be handled in two ways:

- **Backward error correction:** Once the error is discovered, the receiver requests the sender to retransmit the entire data unit.
- **Forward error correction:** In this case, the receiver uses the error-correcting code which automatically corrects the errors.

A single additional bit can detect the error but cannot correct it.

For correcting the errors, one has to know the exact position of the error. For example, If we want to calculate a single-bit error, the error correction code will determine which one of seven bits is in error. To achieve this, we have to add some additional redundant bits.

Suppose r is the number of redundant bits and d is the total number of the data bits. The number of redundant bits r can be calculated by using the formula:

$2^r >= d+r+1$

The value of r is calculated by using the above formula. For example, if the value of d is 4, then the possible smallest value that satisfies the above relation would be 3.

To determine the position of the bit which is in error, a technique developed by R.W Hamming is Hamming code which can be applied to any length of the data unit and uses the relationship between data units and redundant units.

Hamming Code

Parity bits: The bit which is appended to the original data of binary bits so that the total number of 1s is even or odd.

Even parity: To check for even parity, if the total number of 1s is even, then the value of the parity bit is 0. If the total number of 1s occurrences is odd, then the value of the parity bit is 1.

Odd Parity: To check for odd parity, if the total number of 1s is even, then the value of parity bit is 1. If the total number of 1s is odd, then the value of parity bit is 0.

Algorithm of Hamming code:

- An information of 'd' bits are added to the redundant bits 'r' to form d+r.
- The location of each of the (d+r) digits is assigned a decimal value.
- The 'r' bits are placed in the positions 1,2,.....2^{k-1}.
- At the receiving end, the parity bits are recalculated. The decimal value of the parity bits determines the position of an error.

Relationship b/w Error position & binary number.

Error Position	Binary Number
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Let's understand the concept of Hamming code through an example:

Suppose the original data is 1010 which is to be sent.

Total number of data bits 'd' = 4

Number of redundant bits r : $2^r >= d+r+1$

$2^r >= 4+r+1$

Therefore, the value of r is 3 that satisfies the above relation.

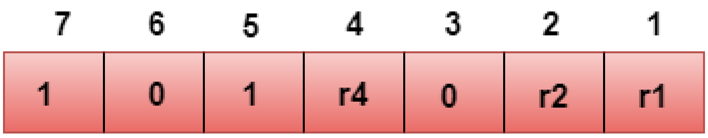
Total number of bits = d+r = 4+3 = 7;

Determining the position of the redundant bits

The number of redundant bits is 3. The three bits are represented by r1, r2, r4. The position of the redundant bits is calculated with corresponds to the raised power of 2. Therefore, their corresponding positions are **1, 2^1, 2^2**.

1. The position of r1 = 1
2. The position of r2 = 2
3. The position of r4 = 4

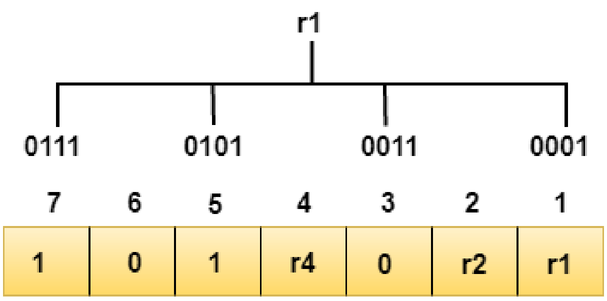
Representation of Data on the addition of parity bits:



Determining the Parity bits

Determining the r1 bit

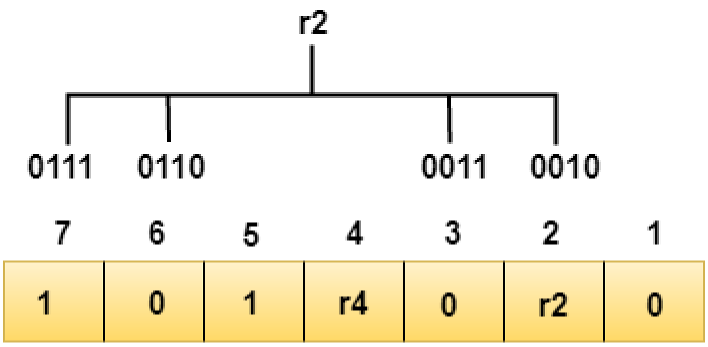
The r1 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the first position.



We observe from the above figure that the bit positions that includes 1 in the first position are 1, 3, 5, 7. Now, we perform the even-parity check at these bit positions. The total number of 1 at these bit positions corresponding to r1 is **even, therefore, the value of the r1 bit is 0**.

Determining r2 bit

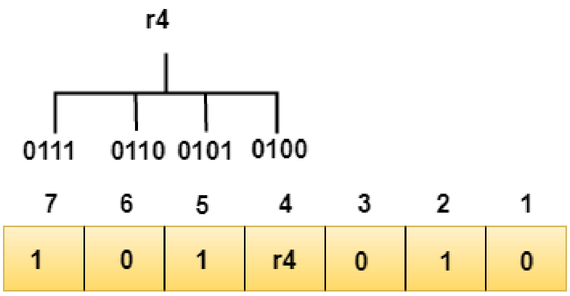
The r2 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the second position.



We observe from the above figure that the bit positions that includes 1 in the second position are **2, 3, 6, 7**. Now, we perform the even-parity check at these bit positions. The total number of 1 at these bit positions corresponding to r2 is **odd, therefore, the value of the r2 bit is 1**.

Determining r4 bit

The r4 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the third position.



We observe from the above figure that the bit positions that includes 1 in the third position are **4, 5, 6, 7**. Now, we perform the even-parity check at these bit positions. The total number of 1 at these bit positions corresponding to r4 is **even, therefore, the value of the r4 bit is 0**.

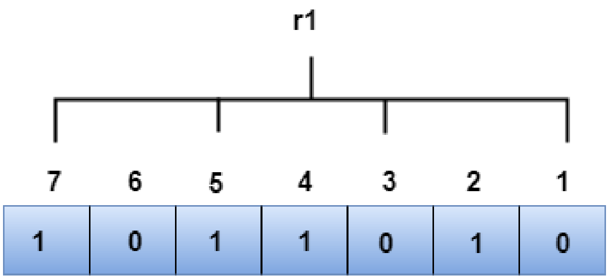
Data transferred is given below:

7	6	5	4	3	2	1
1	0	1	0	0	1	0

Suppose the 4th bit is changed from 0 to 1 at the receiving end, then parity bits are recalculated.

R1 bit

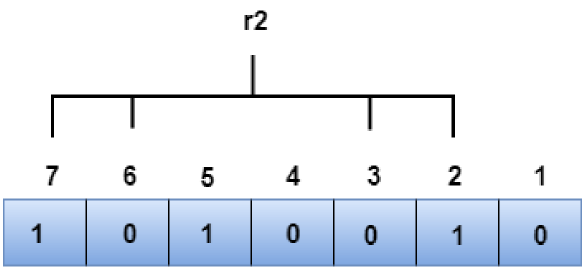
The bit positions of the r1 bit are 1,3,5,7



We observe from the above figure that the binary representation of r1 is 1100. Now, we perform the even-parity check, the total number of 1s appearing in the r1 bit is an even number. Therefore, the value of r1 is 0.

R2 bit

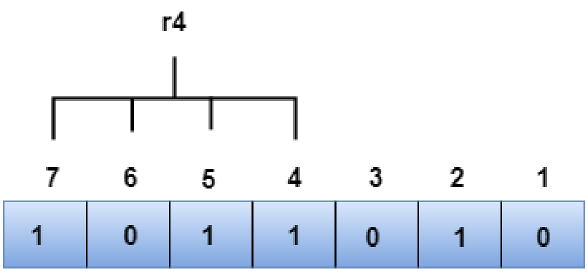
The bit positions of r2 bit are 2,3,6,7.



We observe from the above figure that the binary representation of r2 is 1001. Now, we perform the even-parity check, the total number of 1s appearing in the r2 bit is an even number. Therefore, the value of r2 is 0.

R4 bit

The bit positions of r4 bit are 4,5,6,7.



We observe from the above figure that the binary representation of r4 is 1011. Now, we perform the even-parity check, the total number of 1s appearing in the r4 bit is an odd number. Therefore, the value of r4 is 1.

- *The binary representation of redundant bits, i.e., r4r2r1 is 100, and its corresponding decimal value is 4. Therefore, the error occurs in a 4th bit position. The bit value must be changed from 1 to 0 to correct the error.*

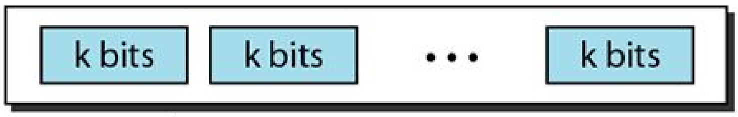
Block coding:

Block coding helps in error detection and re-transmission of the signal. It is normally referred to as mB/nB coding as it replaces each m-bit data group with an n-bit data group (where n>m). Thus, its adds extra bits (redundancy bits) which helps in synchronization at receiver's and sender's end and also providing some kind of error detecting capability.
It normally involves three steps: division, substitution, and combination. In the division step,a sequence of bits is divided into groups of m-bits. In the substitution step, we substitute an m-bit group for an n-bit group. Finally, the n-bit groups are combined together to form a stream which has more bits than the original bits.

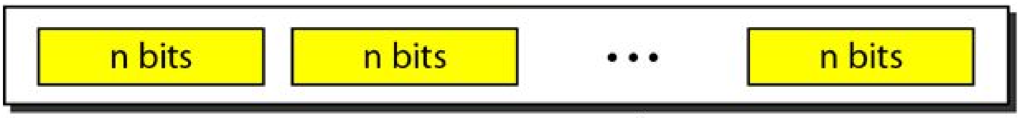
In block coding, we divide our message into blocks, each of k bits, called data words. We add r redundant bits to each block to make the length n = k + r. The resulting n-bit blocks are called code words.

For example, we have a set of data words, each of size k, and a set of code words, each of size of n. With k bits, we can create a combination of 2k data words, with n bits; we can create a combination of 2n code words. Since n > k, the number of possible code words is larger than the number of possible data words.

The block coding process is one-to-one; the same data word is always encoded as the same code word. This means that we have 2n-2k code words that are not used. We call these code words invalid or illegal. The following figure shows the situation.



2^k Datawords, each of k bits



2^n Codewords, each of n bits (only 2^k of them are valid)

Error Detection

If the following two conditions are met, the receiver can detect a change in the original code word by using Block coding technique.

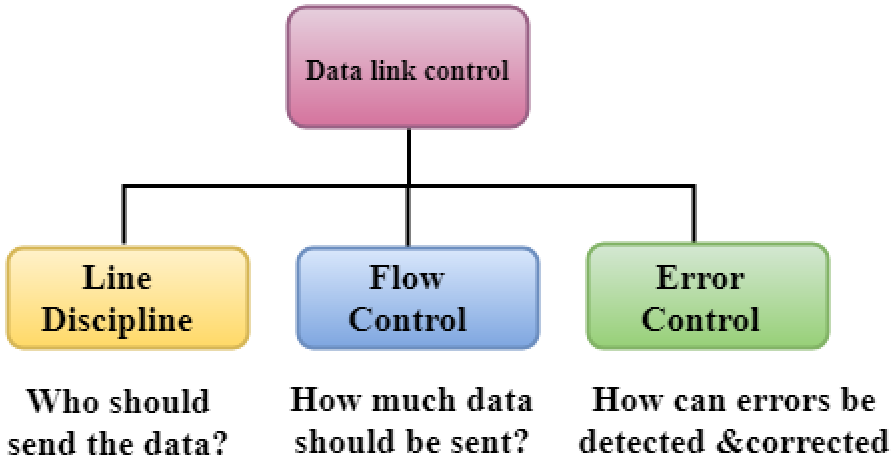
1. The receiver has (or can find) a list of valid code words.
2. The original code word has changed to an invalid one.

Data Link Controls

Data Link Control is the service provided by the Data Link Layer to provide reliable data transfer over the physical medium. For example, In the half-duplex transmission mode, one device can only transmit the data at a time. If both the devices at the end of the links transmit the data simultaneously, they will collide and leads to the loss of the information. The Data link layer provides the coordination among the devices so that no collision occurs.

The Data link layer provides three functions:

- Line discipline
- Flow Control
- Error Control



Line Discipline

- Line Discipline is a functionality of the Data link layer that provides the coordination among the link systems. It determines which device can send, and when it can send the data.

Line Discipline can be achieved in two ways:

- ENQ/ACK
- Poll/select

END/ACK

END/ACK stands for Enquiry/Acknowledgement is used when there is no wrong receiver available on the link and having a dedicated path between the two devices so that the device capable of receiving the transmission is the intended one.

END/ACK coordinates which device will start the transmission and whether the recipient is ready or not.

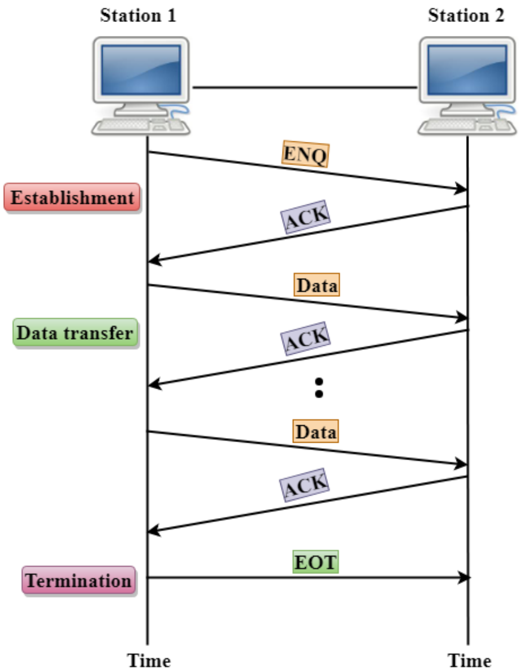
Working of END/ACK

The transmitter transmits the frame called an Enquiry (ENQ) asking whether the receiver is available to receive the data or not.

The receiver responses either with the positive acknowledgement(ACK) or with the negative acknowledgement(NACK) where positive acknowledgement means that the receiver is ready to receive the transmission and negative acknowledgement means that the receiver is unable to accept the transmission.

Following are the responses of the receiver:

- If the response to the ENQ is positive, the sender will transmit its data, and once all of its data has been transmitted, the device finishes its transmission with an EOT (END-of-Transmission) frame.
- If the response to the ENQ is negative, then the sender disconnects and restarts the transmission at another time.
- If the response is neither negative nor positive, the sender assumes that the ENQ frame was lost during the transmission and makes three attempts to establish a link before giving up.



Poll/Select

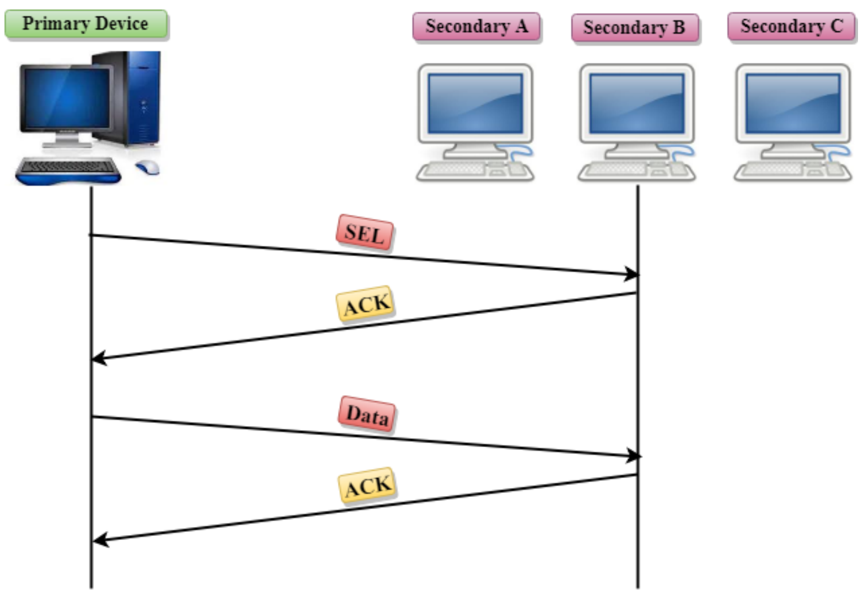
The Poll/Select method of line discipline works with those topologies where one device is designated as a primary station, and other devices are secondary stations.

Working of Poll/Select

- In this, the primary device and multiple secondary devices consist of a single transmission line, and all the exchanges are made through the primary device even though the destination is a secondary device.
- The primary device has control over the communication link, and the secondary device follows the instructions of the primary device.
- The primary device determines which device is allowed to use the communication channel. Therefore, we can say that it is an initiator of the session.
- If the primary device wants to receive the data from the secondary device, it asks the secondary device that they anything to send, this process is known as polling.
- If the primary device wants to send some data to the secondary device, then it tells the target secondary to get ready to receive the data, this process is known as selecting.

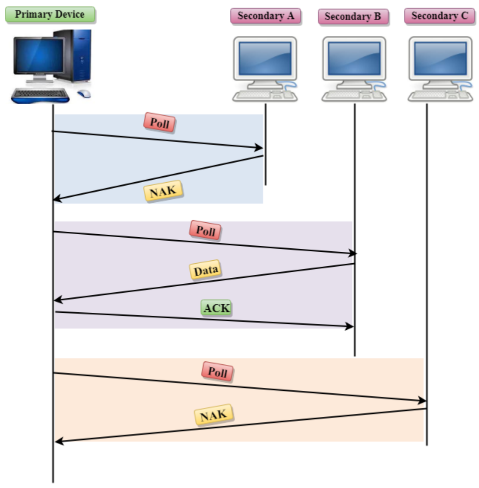
Select

- The select mode is used when the primary device has something to send.
- When the primary device wants to send some data, then it alerts the secondary device for the upcoming transmission by transmitting a Select (SEL) frame, one field of the frame includes the address of the intended secondary device.
- When the secondary device receives the SEL frame, it sends an acknowledgement that indicates the secondary ready status.
- If the secondary device is ready to accept the data, then the primary device sends two or more data frames to the intended secondary device. Once the data has been transmitted, the secondary sends an acknowledgement specifies that the data has been received.



Poll

- The Poll mode is used when the primary device wants to receive some data from the secondary device.
- When a primary device wants to receive the data, then it asks each device whether it has anything to send.
- Firstly, the primary asks (poll) the first secondary device, if it responds with the NACK (Negative Acknowledgement) means that it has nothing to send. Now, it approaches the second secondary device, it responds with the ACK means that it has the data to send. The secondary device can send more than one frame one after another or sometimes it may be required to send ACK before sending each one, depending on the type of the protocol being used.



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

- In the Stop-and-wait method, the sender waits for an acknowledgement after every frame it sends.
- When acknowledgement is received, then only next frame is sent. The process of alternately sending and waiting of a frame continues until the sender transmits the EOT (End of transmission) frame.

Advantage of Stop-and-wait

The Stop-and-wait method is simple as each frame is checked and acknowledged before the next frame is sent.

Disadvantage of Stop-and-wait

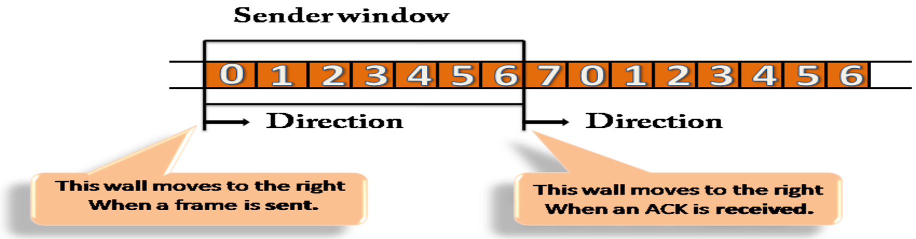
Stop-and-wait technique is inefficient to use as each frame must travel across all the way to the receiver, and an acknowledgement travels all the way before the next frame is sent. Each frame sent and received uses the entire time needed to traverse the link.

Sliding Window

- The Sliding Window is a method of flow control in which a sender can transmit the several frames before getting an acknowledgement.
- In Sliding Window Control, multiple frames can be sent one after the another due to which capacity of the communication channel can be utilized efficiently.
- A single ACK acknowledge multiple frames.
- Sliding Window refers to imaginary boxes at both the sender and receiver end.
- The window can hold the frames at either end, and it provides the upper limit on the number of frames that can be transmitted before the acknowledgement.
- Frames can be acknowledged even when the window is not completely filled.
- The window has a specific size in which they are numbered as modulo-n means that they are numbered from 0 to n-1. For example, if n = 8, the frames are numbered from 0,1,2,3,4,5,6,7,0,1,2,3,4,5,6,7,0,1........
- The size of the window is represented as n-1. Therefore, maximum n-1 frames can be sent before acknowledgement.
- When the receiver sends the ACK, it includes the number of the next frame that it wants to receive. For example, to acknowledge the string of frames ending with frame number 4, the receiver will send the ACK containing the number 5. When the sender sees the ACK with the number 5, it got to know that the frames from 0 through 4 have been received.

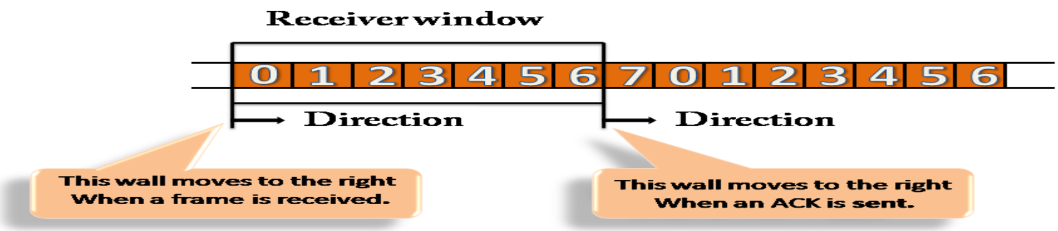
Sender Window

- 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

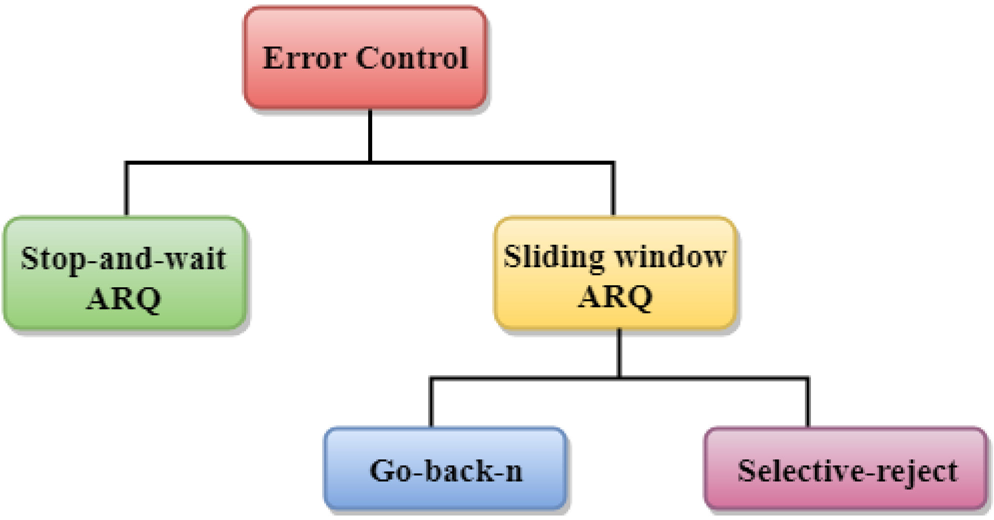
- At the beginning of transmission, the receiver window does not contain n frames, but it contains n-1 spaces for frames.
- When the new frame arrives, the size of the window shrinks.
- The receiver window does not represent the number of frames received, but it represents the number of frames that can 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.



Error Control

Error Control is a technique of error detection and retransmission.

Categories of Error Control:



Stop-and-wait ARQ

Stop-and-wait ARQ is a technique used to retransmit the data in case of damaged or lost frames.

This technique works on the principle that the sender will not transmit the next frame until it receives the acknowledgement of the last transmitted frame.

Four features are required for the retransmission:

- The sending device keeps a copy of the last transmitted frame until the acknowledgement is received. Keeping the copy allows the sender to retransmit the data if the frame is not received correctly.
- Both the data frames and the ACK frames are numbered alternately 0 and 1 so that they can be identified individually. Suppose data 1 frame acknowledges the data 0 frame means that the data 0 frame has been arrived correctly and expects to receive data 1 frame.
- If an error occurs in the last transmitted frame, then the receiver sends the NAK frame which is not numbered. On receiving the NAK frame, sender retransmits the data.
- It works with the timer. If the acknowledgement is not received within the allotted time, then the sender assumes that the frame is lost during the transmission, so it will retransmit the frame.

Two possibilities of the retransmission:

- **Damaged Frame:** When the receiver receives a damaged frame, i.e., the frame contains an error, then it returns the NAK frame. For example, when the data 0 frame is sent, and then the receiver sends the ACK 1 frame means that the data 0 has arrived correctly, and transmits the data 1 frame. The sender transmits the next frame: data 1. It reaches undamaged, and the receiver returns ACK 0. The sender transmits the next frame: data 0. The receiver reports an error and returns the NAK frame. The sender retransmits the data 0 frame.

- **Lost Frame:** Sender is equipped with the timer and starts when the frame is transmitted. Sometimes the frame has not arrived at the receiving end so that it can be acknowledged neither positively nor negatively. The sender waits for acknowledgement until the timer goes off. If the timer goes off, it retransmits the last transmitted frame.

Characteristics

- Used in Connection-oriented communication.
- It offers error and flows 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

Useful Terms:

- **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**) = 2* Propagation Delay
- TimeOut (**TO**) = 2* RTT
- Time To Live (**TTL**) = 2* TimeOut. (Maximum TTL is 180 seconds)

<u>Simple Stop and Wait</u>

Sender:

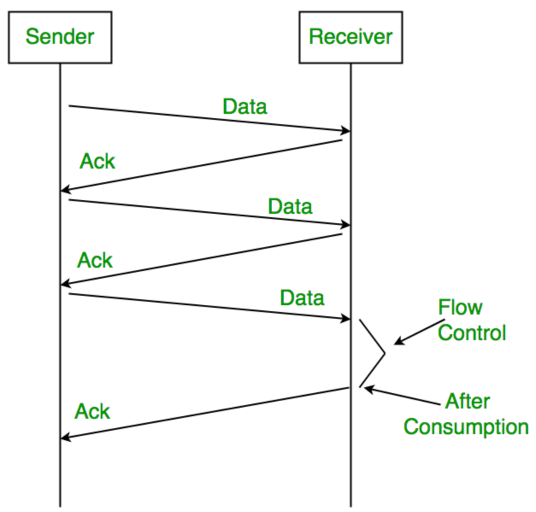
Rule 1) Send one data packet at a time.

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

Receiver:

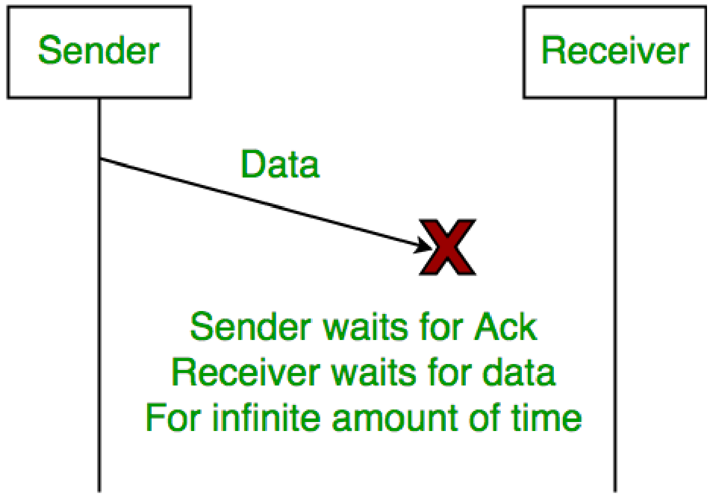
Rule 1) Send acknowledgement after receiving and consuming a data packet.

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

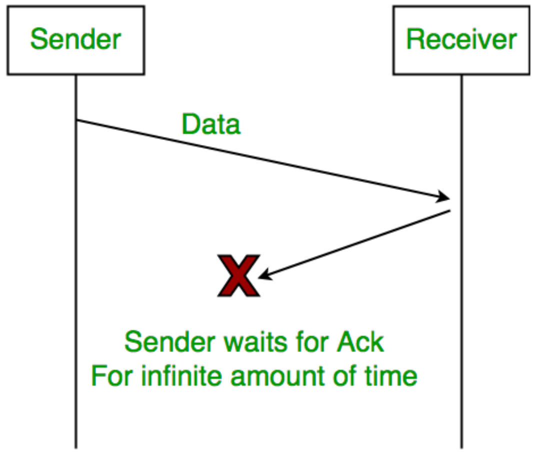


Problems :

1. Lost Data



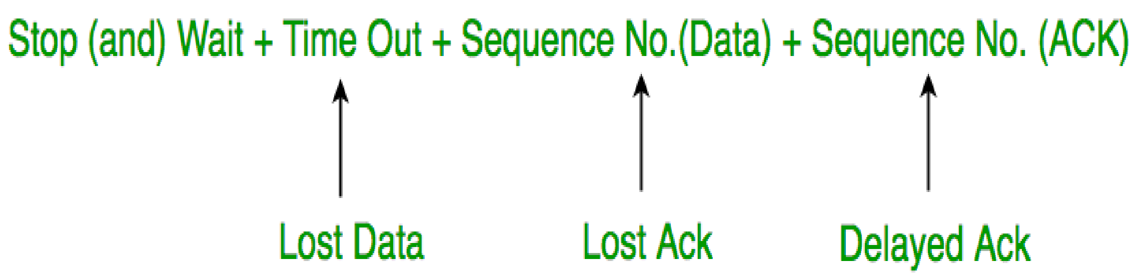
2. Lost Acknowledgement:



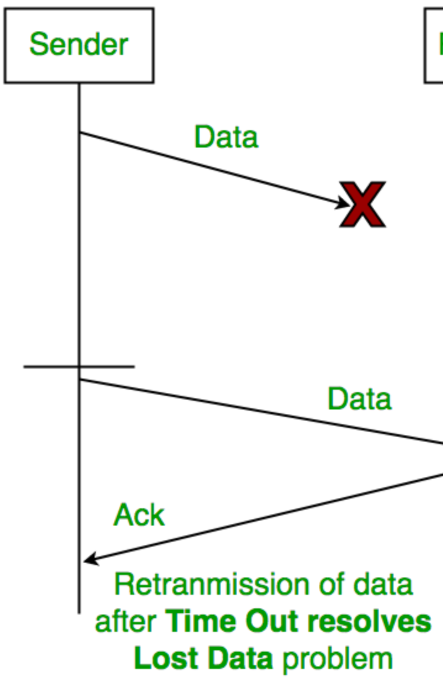
3. Delayed Acknowledgement/Data: After a timeout on the sender side, a long-delayed acknowledgement might be wrongly considered as acknowledgement of some other recent packet.

Stop and Wait for ARQ (Automatic Repeat Request)

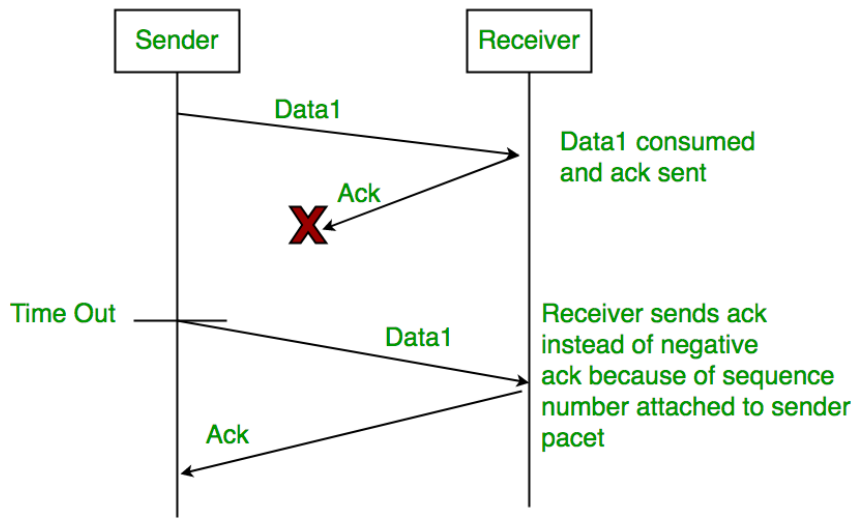
The above 3 problems are resolved by Stop and Wait for ARQ (Automatic Repeat Request) that does both error control and flow control.



1. Time Out:



2. Sequence Number (Data)



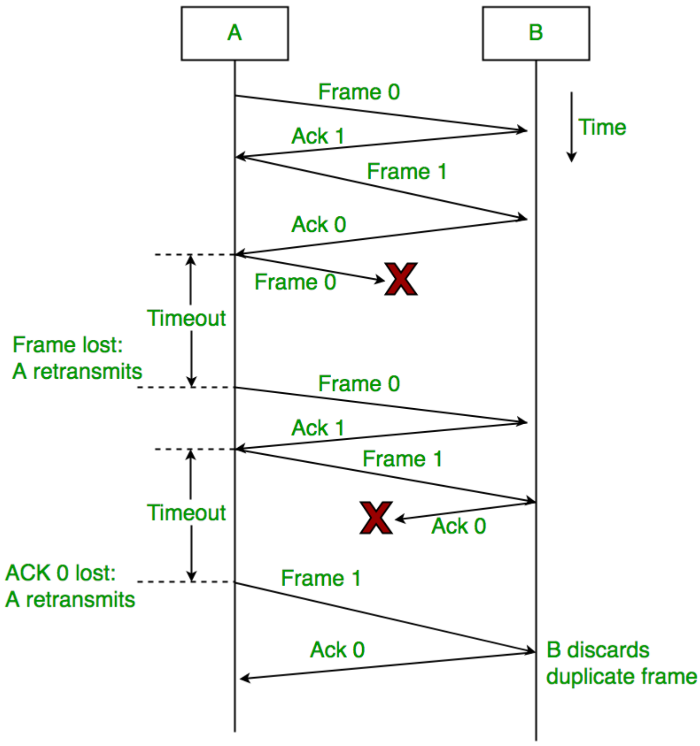
3. Delayed Acknowledgement:
This is resolved by introducing sequence numbers for acknowledgement also.

Working of Stop and Wait for ARQ:

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

2) 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.



Characteristics of Stop and Wait ARQ:

- 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 acknowledgements 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

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.

Sliding Window ARQ

Sliding Window ARQ is a technique used for continuous transmission error control. 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.

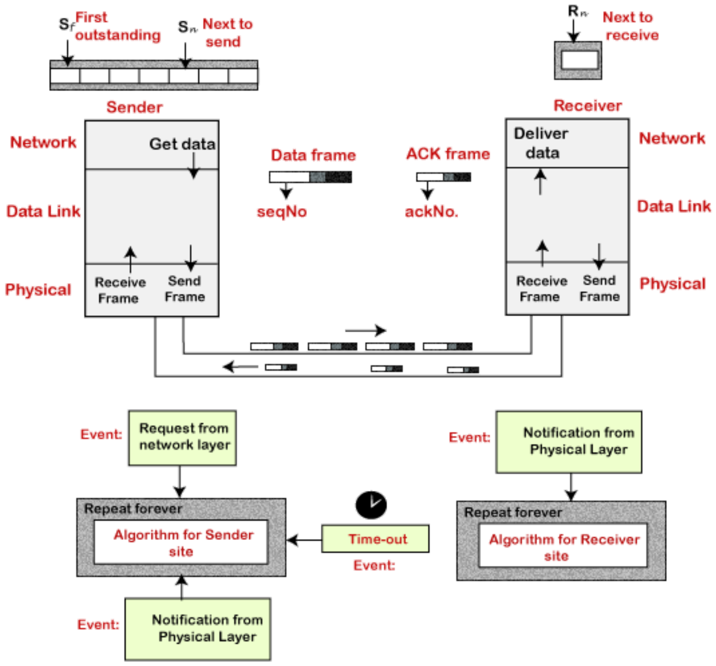
Three Features used for retransmission:

- In this case, the sender keeps the copies of all the transmitted frames until they have been acknowledged. Suppose the frames from 0 through 4 have been transmitted, and the last acknowledgement was for frame 2, the sender has to keep the copies of frames 3 and 4 until they receive correctly.
- The receiver can send either NAK or ACK depending on the conditions. The NAK frame tells the sender that the data have been received damaged. Since the sliding window is a continuous transmission mechanism, both ACK and NAK must be numbered for the identification of a frame. The ACK frame consists of a number that represents the next frame which the receiver expects to receive. The NAK frame consists of a number that represents the damaged frame.
- The sliding window ARQ is equipped with the timer to handle the lost acknowledgements. Suppose then n-1 frames have been sent before receiving any acknowledgement. The sender waits for the acknowledgement, so it starts the timer and waits before sending any more. If the allotted time runs out, the sender retransmits one or all the frames depending upon the protocol used.

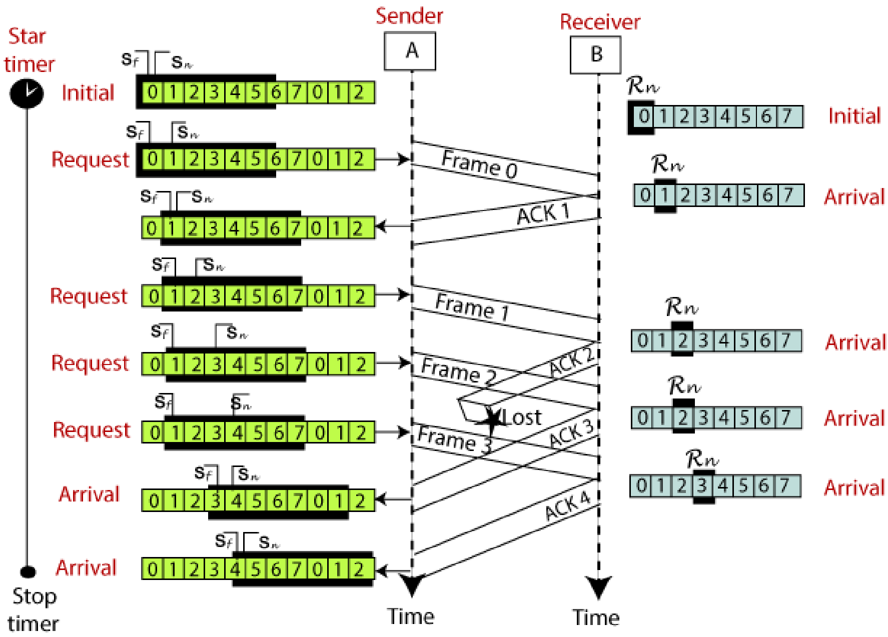
Two protocols used in sliding window ARQ:

- **Go-Back-n ARQ:** In Go-Back-N ARQ protocol, if one frame is lost or damaged, then it retransmits all the frames after which it does not receive the positive ACK.
- 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.

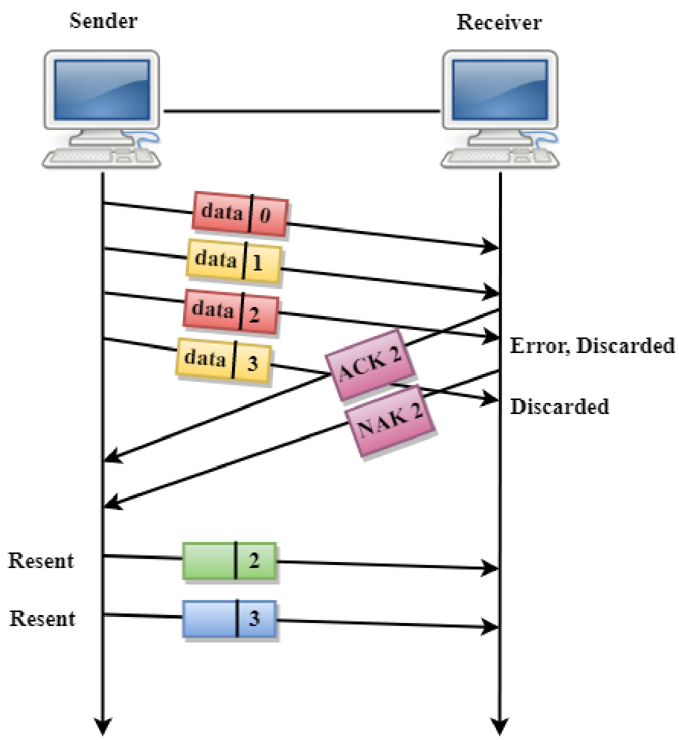


- The example of Go-Back-N ARQ is shown below in the figure.



Three possibilities can occur for retransmission:

- **Damaged Frame:** When the frame is damaged, then the receiver sends a NAK frame.



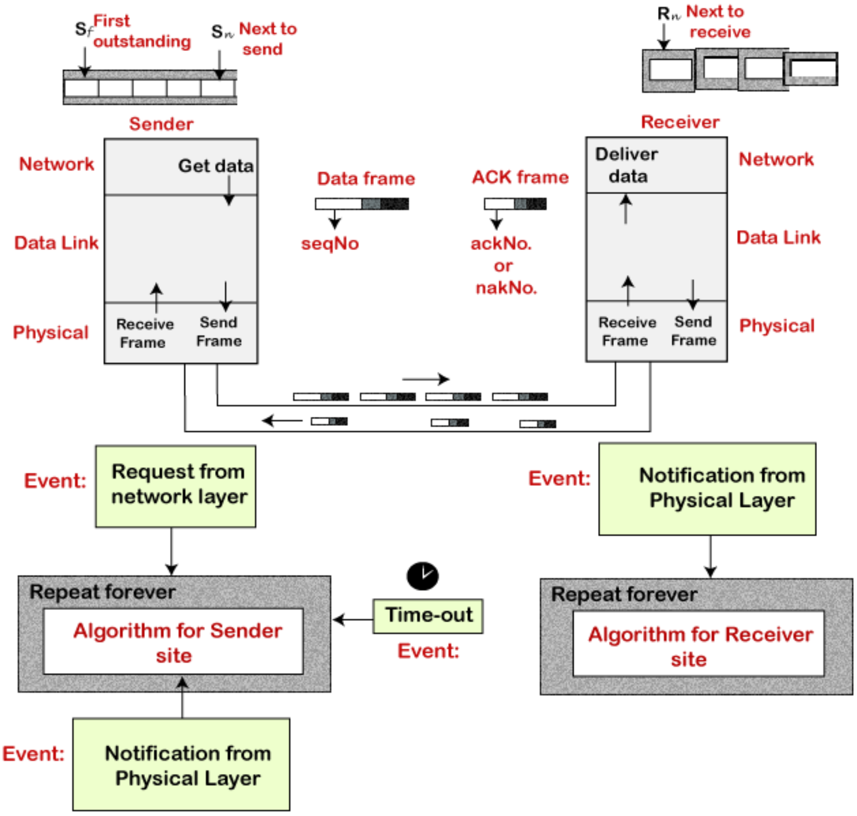
In the above figure, three frames have been transmitted before an error discovered in the third frame. In this case, ACK 2 has been returned telling that the frames 0,1 have been received successfully without any error. The receiver discovers the error in data 2 frame, so it returns the NAK 2 frame. The frame 3 is also discarded as it is transmitted after the damaged frame. Therefore, the sender retransmits the frames 2,3.

- **Lost Data Frame:** In Sliding window protocols, data frames are sent sequentially. If any of the frames is lost, then the next frame arrive at the receiver is out of sequence. The receiver checks the sequence number of each of the frame, discovers the frame that has been skipped, and returns the NAK for the missing frame. The sending device retransmits the frame indicated by NAK as well as the frames transmitted after the lost frame.
- **Lost Acknowledgement:** The sender can send as many frames as the windows allow before waiting for any acknowledgement. Once the limit of the window is reached, the sender has no more frames to send; it must wait for the acknowledgement. If the acknowledgement is lost, then the sender could wait forever. To avoid such situation, the sender is equipped with the timer that starts counting whenever the window capacity is reached. If the acknowledgement has not been received within the time limit, then the sender retransmits the frame since the last ACK.

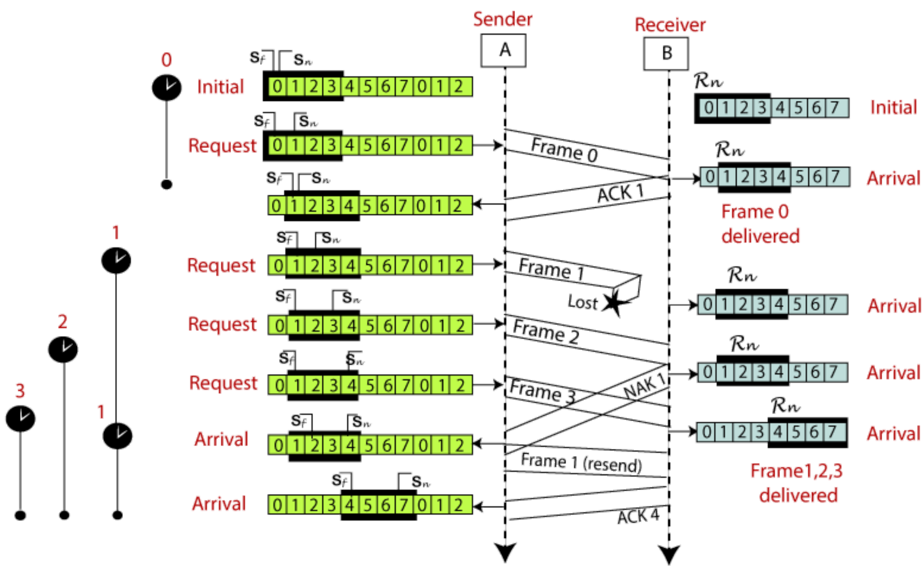
Selective-Reject ARQ

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.

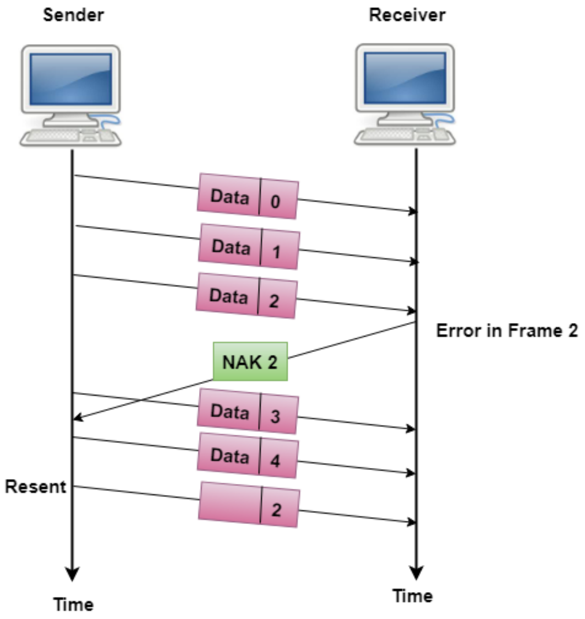


The example of the Selective Repeat ARQ protocol is shown below in the figure.



Features of Selective-Reject ARQ

- Selective-Reject ARQ technique is more efficient than Go-Back-n ARQ.
- In this technique, only those frames are retransmitted for which negative acknowledgement (NAK) has been received.
- The receiver storage buffer keeps all the damaged frames on hold until the frame in error is correctly received.
- The receiver must have an appropriate logic for reinserting the frames in a correct order.
- The sender must consist of a searching mechanism that selects only the requested frame for retransmission.



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.

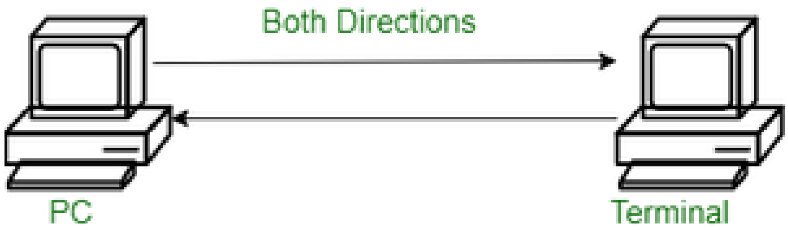
Piggybacking:

In this article, we will cover the overview of networking communication and mainly focus on the concept of piggybacking in networks. And we will also discuss the advantages and disadvantages of using piggybacking in networks. Finally, we will see the conclusion. Let's discuss it one by one.

Networking Communication:

Sliding window algorithms are methods of flow control for network data transfer. The data link layer uses a sender to have more than one acknowledgement packet at a time, which improves network throughput. Both the sender and receiver maintain a finite-size buffer to hold outgoing and incoming packets from the other side. Every packet sends by the sender must be acknowledged by the receiver. The sender maintains a timer for every packet sent, and any packet unacknowledged at a certain time is resent. The sender may send a whole window of packets before receiving an acknowledgement for the first packet in the window. This results in higher transfer rates, as the sender may send multiple packets without waiting for each packet's acknowledgement. The receiver advertises a window size that tells the sender not to fill up the receiver buffers.

Efficiency can also be improved by making use of full-duplex transmission. **Full Duplex transmission** is a two-way directional communication simultaneously. It provides better performance than simple and half-duplex transmission modes.



Full-duplex transmission

Solution 1 –

One way to achieve full-duplex transmission is to have two separate channels with one for forwarding data transmission and the other for reverse data transfer (to accept). But this will almost completely waste the bandwidth of the reverse channel.

Solution 2(Piggybacking) –

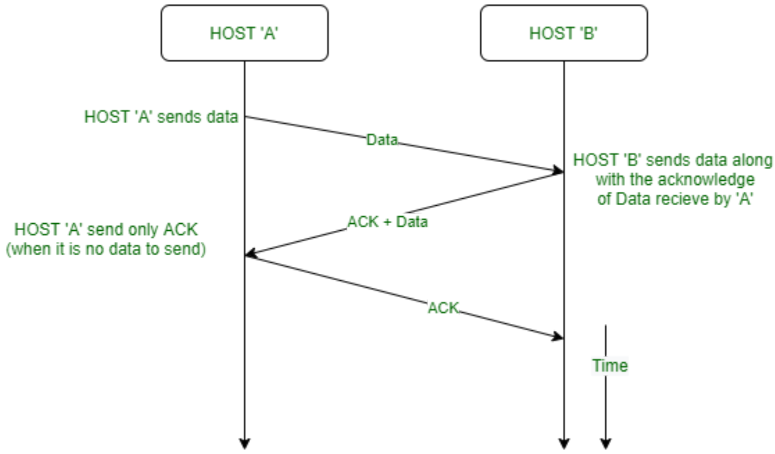
A preferable solution would be to use each channel to transmit the frame (front and back) both ways, with both channels having the same capacity. Assume that A and B are users. Then the data frames from A to B are interconnected with the acknowledgement from A to B. and can be identified as a data frame or acknowledgement by checking the sort field in the header of the received frame.

One more improvement can be made. When a data frame arrives, the receiver waits does not send the control frame (acknowledgement) back immediately. The receiver waits until its network layer moves to the next data packet.

Acknowledgement is associated with this outgoing data frame. Thus the acknowledgement travels along with the next data frame.

Definition of Piggybacking :

*This technique in which the outgoing acknowledgement is delayed temporarily is called **piggybacking**.*



As we can see in the figure, we can see with piggybacking, a single message (ACK + DATA) over the wire in place of two separate messages. Piggybacking improves the efficiency of the bidirectional protocols.

Advantages of piggybacking:

1. The major advantage of piggybacking is the better use of available channel bandwidth. This happens because an acknowledgement frame needs not to be sent separately.
2. Usage cost reduction
3. Improves latency of data transfer

Disadvantages of piggybacking:

1. The disadvantage of piggybacking is the additional complexity.
2. If the data link layer waits long before transmitting the acknowledgement (block the ACK for some time), the frame will rebroadcast.

NOTE – To avoid the delay and rebroadcast of frame transmission, piggybacking uses a very short duration timer.

Random Access Protocol

The data link layer is used in a computer network to transmit the data between two devices or nodes. It divides the layer into parts such as **data link control** and the **multiple access resolution/protocol**. The upper layer has the responsibility to flow control and the error control in the data link layer, and hence it is termed as **logical of data link control**. Whereas the lower sub-layer is used to handle and reduce the collision or multiple access on a channel. Hence it is termed as media access control or the multiple access resolutions.

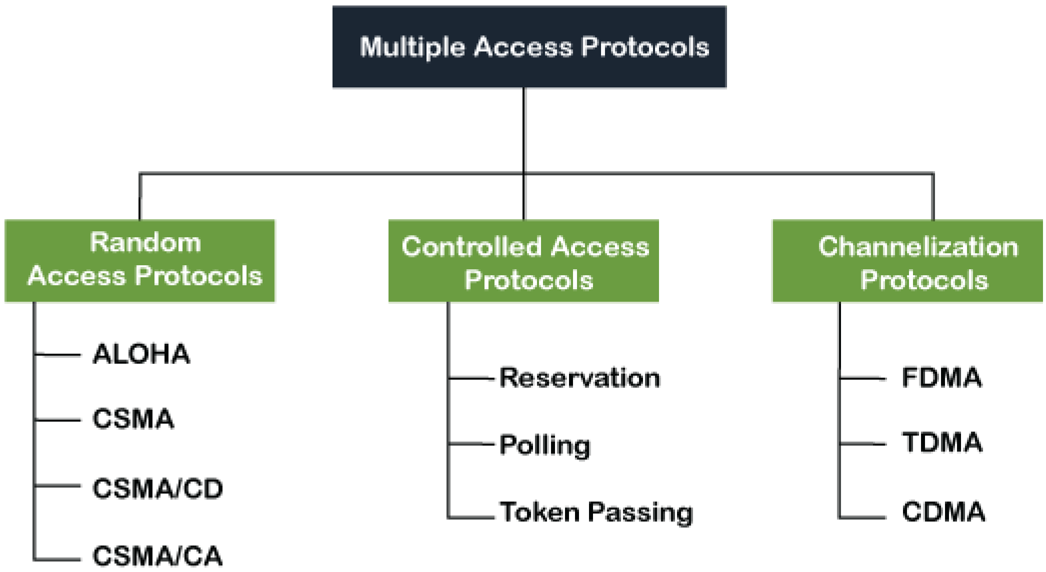
A data link control is a reliable channel for transmitting data over a dedicated link using various techniques such as framing, error control and flow control of data packets in the computer network.

What is a multiple access protocol?

When a sender and receiver have a dedicated link to transmit data packets, the data link control is enough to handle the channel. Suppose there is no dedicated path to communicate or transfer the data between two devices. In that case, multiple stations access the channel and simultaneously transmits the data over the channel. It may create collision and cross talk. Hence, the multiple access protocol is required to reduce the collision and avoid crosstalk between the channels.

For example, suppose that there is a classroom full of students. When a teacher asks a question, all the students (small channels) in the class start answering the question at the same time (transferring the data simultaneously). All the students respond at the same time due to which data is overlap or data lost. Therefore it is the responsibility of a teacher (multiple access protocol) to manage the students and make them one answer.

Following are the types of multiple access protocol that is subdivided into the different process as:



A. Random Access Protocol

In this protocol, all the station has the equal priority to send the data over a channel. In random access protocol, one or more stations cannot depend on another station nor any station control another station. Depending on the channel's state (idle or busy), each station transmits the data frame. However, if more than one station sends the data over a channel, there may be a collision or data conflict. Due to the collision, the data frame packets may be lost or changed. And hence, it does not receive by the receiver end.

Following are the different methods of random-access protocols for broadcasting frames on the channel.

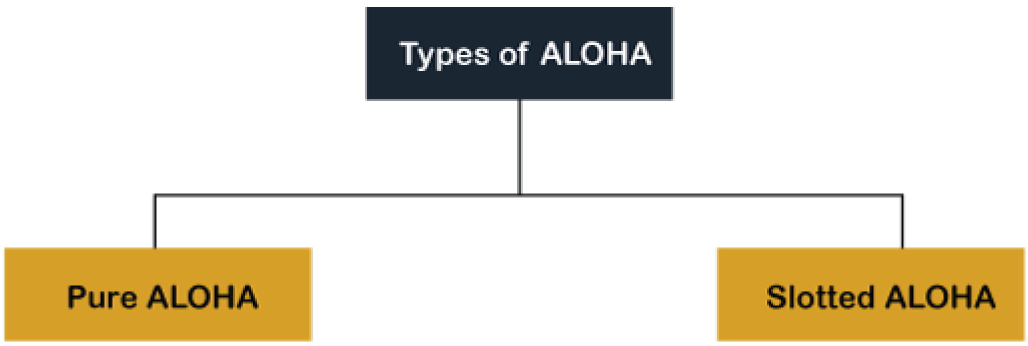
- Aloha
- CSMA
- CSMA/CD
- CSMA/CA

ALOHA Random Access Protocol

It is designed for wireless LAN (Local Area Network) but can also be used in a shared medium to transmit data. Using this method, any station can transmit data across a network simultaneously when a data frameset is available for transmission.

Aloha Rules

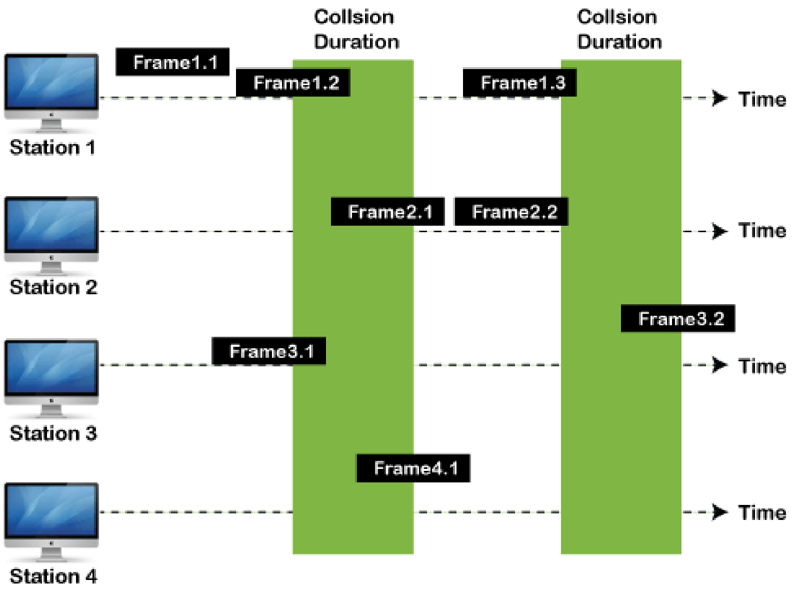
1. Any station can transmit data to a channel at any time.
2. It does not require any carrier sensing.
3. Collision and data frames may be lost during the transmission of data through multiple stations.
4. Acknowledgment of the frames exists in Aloha. Hence, there is no collision detection.
5. It requires retransmission of data after some random amount of time.



Pure Aloha

Whenever data is available for sending over a channel at stations, we use Pure Aloha. 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. When any station transmits the data frame to a channel, the pure Aloha waits for the receiver's acknowledgment. If it does not acknowledge the receiver end within the specified time, the station waits for a random amount of time, called the backoff time (Tb). And the station may assume the frame has been lost or destroyed. Therefore, it retransmits the frame until all the data are successfully transmitted to the receiver.

1. The total vulnerable time of pure Aloha is 2 * Tfr.
2. Maximum throughput occurs when G = 1/ 2 that is 18.4%.
3. Successful transmission of data frame is S = G * e ^ - 2 G.



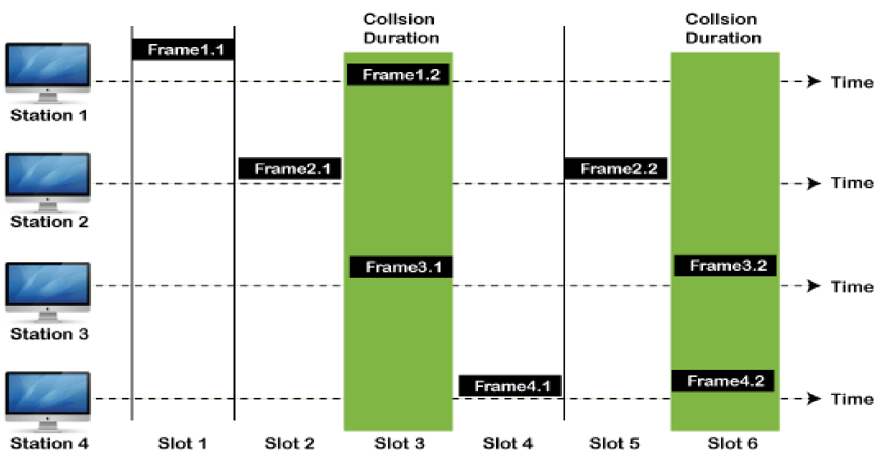
Frames in Pure ALOHA

As we can see in the figure above, there are four stations for accessing a shared channel and transmitting data frames. Some frames collide because most stations send their frames at the same time. Only two frames, frame 1.1 and frame 2.2, are successfully transmitted to the receiver end. At the same time, other frames are lost or destroyed. Whenever two frames fall on a shared channel simultaneously, collisions can occur, and both will suffer damage. If the new frame's first bit enters the channel before finishing the last bit of the second frame. Both frames are completely finished, and both stations must retransmit the data frame.

Slotted Aloha

The slotted Aloha is designed to overcome the pure Aloha's efficiency because pure Aloha has a very high possibility of frame hitting. 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. And if the stations are unable to send data to the beginning of the slot, the station will have to wait until the beginning of the slot for the next time. However, the possibility of a collision remains when trying to send a frame at the beginning of two or more station time slot.

1. Maximum throughput occurs in the slotted Aloha when G = 1 that is 37%.
2. The probability of successfully transmitting the data frame in the slotted Aloha is S = G * e ^ - 2 G.
3. The total vulnerable time required in slotted Aloha is Tfr.



Frames in Slotted ALOHA

Difference between Pure Aloha and Slotted Aloha

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%.

CSMA (Carrier Sense Multiple Access)

It is a **carrier sense multiple access** based on media access protocol to sense the traffic on a channel (idle or busy) before transmitting the data. It means that if the channel is idle, the station can send data to the channel. Otherwise, it must wait until the channel becomes idle. Hence, it reduces the chances of a collision on a transmission medium.

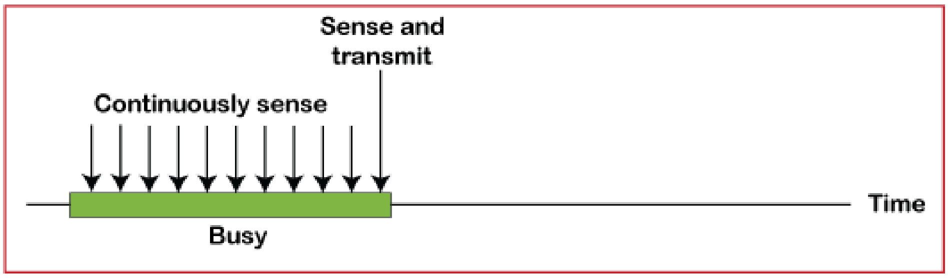
CSMA Access Modes

1-Persistent: In the 1-Persistent mode of CSMA that defines each node, first sense the shared channel and if the channel is idle, it immediately sends the data. Else it must wait and keep track of the status of the channel to be idle and broadcast the frame unconditionally as soon as the channel is idle.

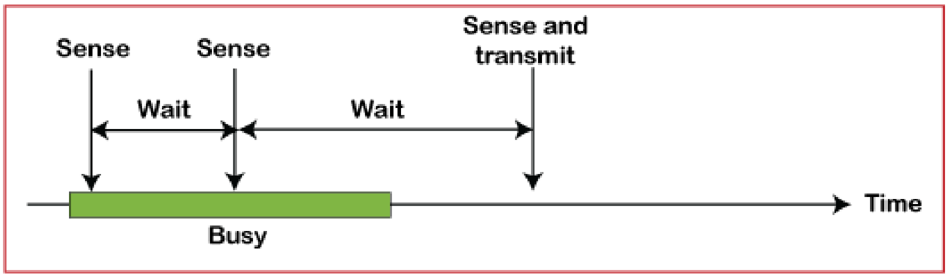
Non-Persistent: It is the access mode of CSMA that defines before transmitting the data, each node must sense the channel, and if the channel is inactive, it immediately sends the data. Otherwise, the station must wait for a random time (not continuously), and when the channel is found to be idle, it transmits the frames.

P-Persistent: It is the combination of 1-Persistent and Non-persistent modes. The P-Persistent mode defines that each node senses the channel, and if the channel is inactive, it sends a frame with a **P** probability. If the data is not transmitted, it waits for a (**q = 1-p probability**) random time and resumes the frame with the next time slot.

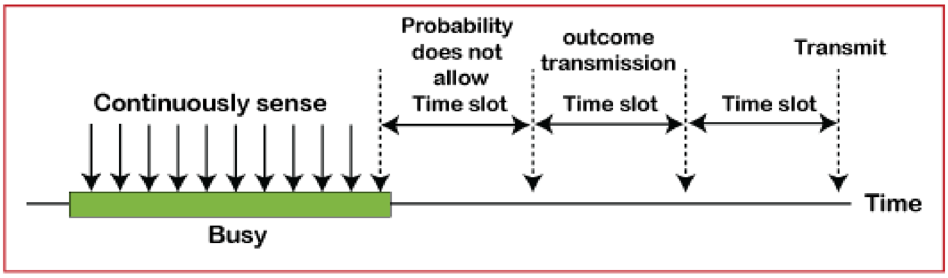
O- Persistent: It is an O-persistent method that defines the superiority of the station before the transmission of the frame on the shared channel. If it is found that the channel is inactive, each station waits for its turn to retransmit the data.



a. 1-persistent



b. Nonpersistent



c. p-persistent

CSMA/ CD

It is a **carrier sense multiple access/ collision detection** network protocol to transmit data frames. The CSMA/CD protocol works with a medium access control layer. Therefore, it first senses the shared channel before broadcasting the frames, and if the channel is idle, it transmits a frame to check whether the transmission was successful. If the frame is successfully received, the station sends another frame. If any collision is detected in the CSMA/CD, the station sends a jam/ stop signal to the shared channel to terminate data transmission. After that, it waits for a random time before sending a frame to a channel.

Carrier Sense Multiple Access with Collision Detection (CSMA/CD) is a network protocol for carrier transmission that operates in the Medium Access Control (MAC) layer. It senses or listens whether the shared channel for transmission is busy or not, and defers transmissions until the channel is free. The collision detection technology detects collisions by sensing transmissions from other stations. On detection of a collision, the station stops transmitting, sends a jam signal, and then waits for a random time interval before retransmission.

Algorithms

The algorithm of CSMA/CD is:

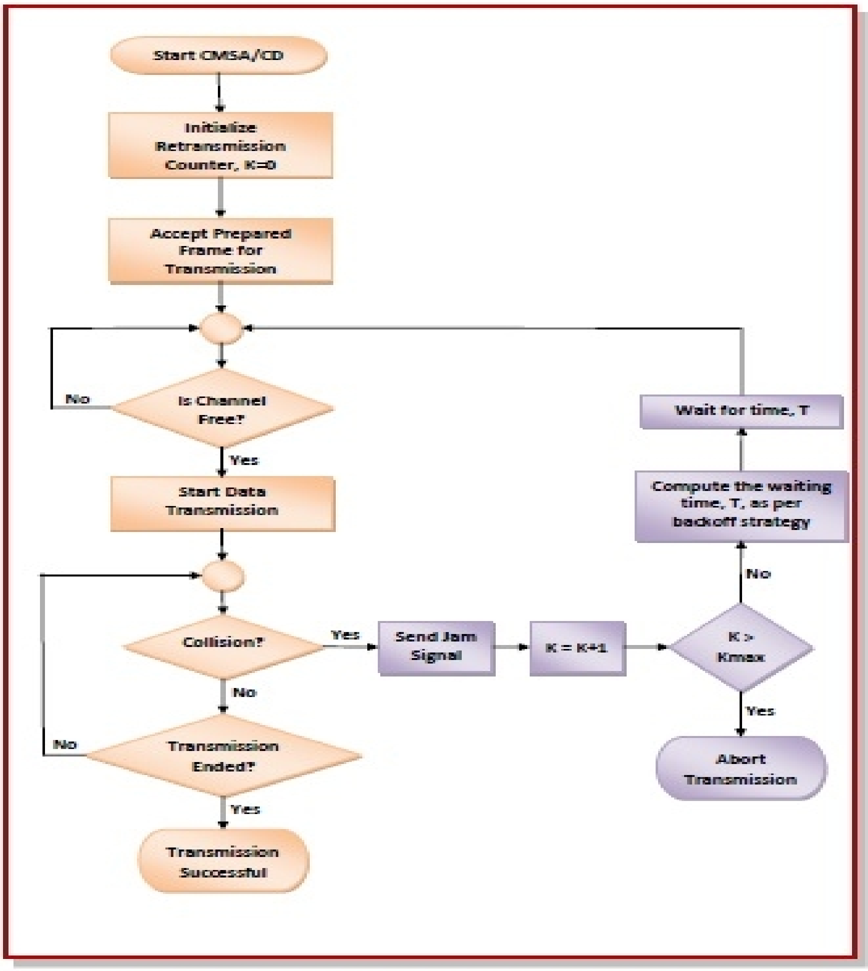
- When a frame is ready, the transmitting station checks whether the channel is idle or busy.
- If the channel is busy, the station waits until the channel becomes idle.

- If the channel is idle, the station starts transmitting and continually monitors the channel to detect collision.
- If a collision is detected, the station starts the collision resolution algorithm.
- The station resets the retransmission counters and completes frame transmission.

The algorithm of Collision Resolution is:

- The station continues transmission of the current frame for a specified time along with a jam signal, to ensure that all the other stations detect collision.
- The station increments the retransmission counter.
- If the maximum number of retransmission attempts is reached, then the station aborts transmission.
- Otherwise, the station waits for a backoff period which is generally a function of the number of collisions and restart main algorithm.

The following flowchart summarizes the algorithms:



- Though this algorithm detects collisions, it does not reduce the number of collisions.
- It is not appropriate for large networks performance degrades exponentially when more stations are added.

Advantages of CSMA CD:

1. It is used for collision detection on a shared channel within a very short time.
2. CSMA CD is better than CSMA for collision detection.
3. CSMA CD is used to avoid any form of waste transmission.
4. When necessary, it is used to use or share the same amount of bandwidth at each station.

5. It has lower CSMA CD overhead as compared to the CSMA CA.

Disadvantage of CSMA CD

1. It is not suitable for long-distance networks because as the distance increases, CSMA CD' efficiency decreases.
2. It can detect collision only up to 2500 meters, and beyond this range, it cannot detect collisions.
3. When multiple devices are added to a CSMA CD, collision detection performance is reduced.

CSMA/ CA

It is a **carrier sense multiple access/collision avoidance** network protocol for carrier transmission of data frames. It is a protocol that works with a medium access control layer. When a data frame is sent to a channel, it receives an acknowledgment to check whether the channel is clear. If the station receives only a single (own) acknowledgments, that means the data frame has been successfully transmitted to the receiver. But if it gets two signals (its own and one more in which the collision of frames), a collision of the frame occurs in the shared channel. Detects the collision of the frame when a sender receives an acknowledgment signal.

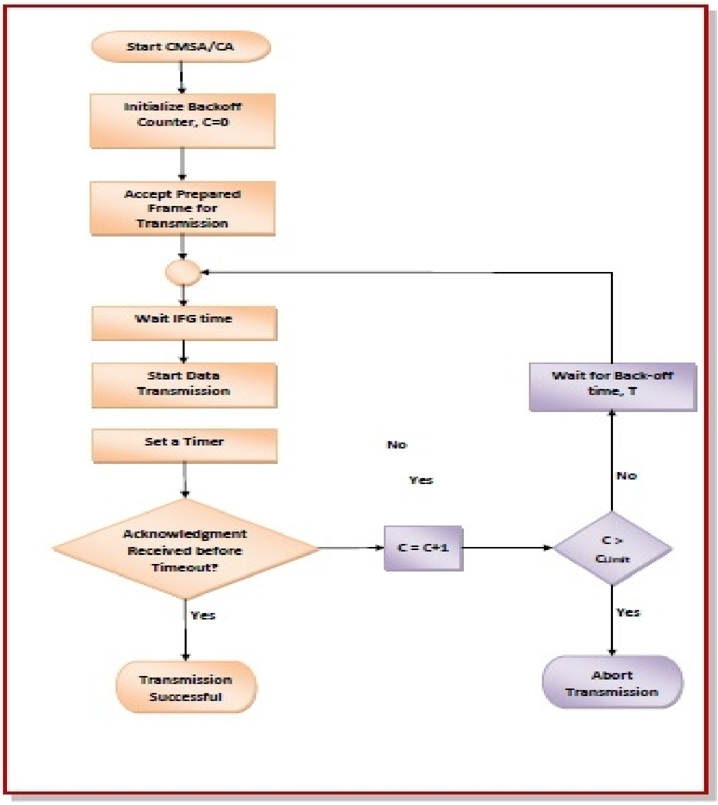
Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is a network protocol for carrier transmission that operates in the Medium Access Control (MAC) layer. In contrast to CSMA/CD (Carrier Sense Multiple Access/Collision Detection) that deals with collisions after their occurrence, CSMA/CA prevents collisions prior to their occurrence.

Algorithm

The algorithm of CSMA/CA is:

- When a frame is ready, the transmitting station checks whether the channel is idle or busy.
- If the channel is busy, the station waits until the channel becomes idle.
- If the channel is idle, the station waits for an Inter-frame gap (IFG) amount of time and then sends the frame.
- After sending the frame, it sets a timer.
- The station then waits for acknowledgement from the receiver. If it receives the acknowledgement before expiry of timer, it marks a successful transmission.
- Otherwise, it waits for a back-off time period and restarts the algorithm.

The following flowchart summarizes the algorithms:



Advantage and Disadvantage of CSMA CA

Advantage of CSMA CA

1. When the size of data packets is large, the chances of collision in CSMA CA is less.
2. It controls the data packets and sends the data when the receiver wants to send them.
3. It is used to prevent collision rather than collision detection on the shared channel.
4. CSMA CA avoids wasted transmission of data over the channel.
5. It is best suited for wireless transmission in a network.
6. It avoids unnecessary data traffic on the network with the help of the RTS/ CTS extension.

The disadvantage of CSMA CA

1. Sometime CSMA/CA takes much waiting time as usual to transmit the data packet.
2. It consumes more bandwidth by each station.
3. Its efficiency is less than a CSMA CD.

Following are the methods used in the CSMA/ CA to avoid the collision:

Interframe space: In this method, the station waits for the channel to become idle, and if it gets the channel is idle, it does not immediately send the data. Instead of this, it waits for some time, and this time period is called the **Interframe** space or IFS. However, the IFS time is often used to define the priority of the station.

Contention window: In the Contention window, the total time is divided into different slots. When the station/ sender is ready to transmit the data frame, it chooses a random slot number of slots as **wait time**. If the channel is still busy, it does not restart the entire process, except that it restarts the timer only to send data packets when the channel is inactive.

Acknowledgment: In the acknowledgment method, the sender station sends the data frame to the shared channel if the acknowledgment is not received ahead of time.

B. Controlled Access Protocol

It is a method of reducing data frame collision on a shared channel. In the controlled access method, each station interacts and decides to send a data frame by a particular station approved by all other stations. It means that a single station cannot send the data frames unless all other stations are not approved. It has three types of controlled access: **Reservation, Polling**, and **Token Passing**.

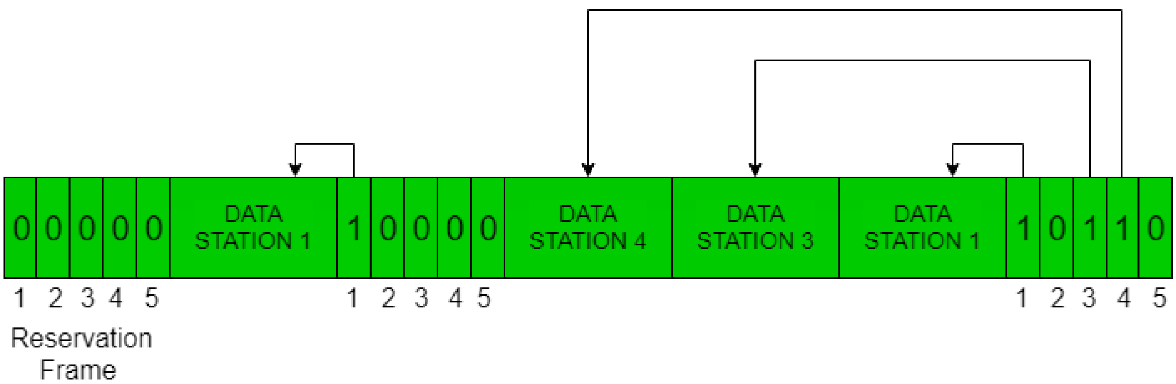
In controlled access, the stations seek information from one another to find which station has the right to send. It allows only one node to send at a time, to avoid collision of messages on shared medium. The three controlled-access methods are:

1. Reservation
2. Polling
3. Token Passing

Reservation

- In the reservation method, a station needs to make a reservation before sending data.
- The timeline has two kinds of periods:
    1. Reservation interval of fixed time length
    2. Data transmission period of variable frames.
- If there are M stations, the reservation interval is divided into M slots, and each station has one slot.
- Suppose if station 1 has a frame to send, it transmits 1 bit during the slot 1. No other station is allowed to transmit during this slot.
- In general, i th station may announce that it has a frame to send by inserting a 1 bit into i th slot. After all N slots have been checked, each station knows which stations wish to transmit.
- The stations which have reserved their slots transfer their frames in that order.
- After data transmission period, next reservation interval begins.
- Since everyone agrees on who goes next, there will never be any collisions.

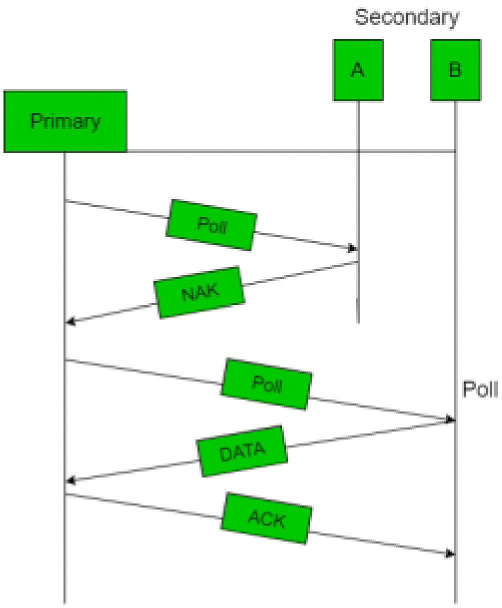
The following figure shows a situation with five stations and a five-slot reservation frame. In the first interval, only stations 1, 3, and 4 have made reservations. In the second interval, only station 1 has made a reservation.

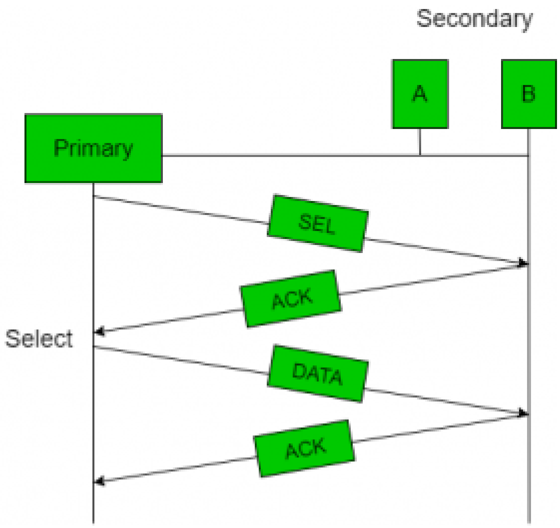


Polling

- Polling process is similar to the roll-call performed in class. Just like the teacher, a controller sends a message to each node in turn.
- In this, one acts as a primary station(controller) and the others are secondary stations. All data exchanges must be made through the controller.

- The message sent by the controller contains the address of the node being selected for granting access.
- Although all nodes receive the message but the addressed one responds to it and sends data, if any. If there is no data, usually a “poll reject”(NAK) message is sent back.
- Problems include high overhead of the polling messages and high dependence on the reliability of the controller.

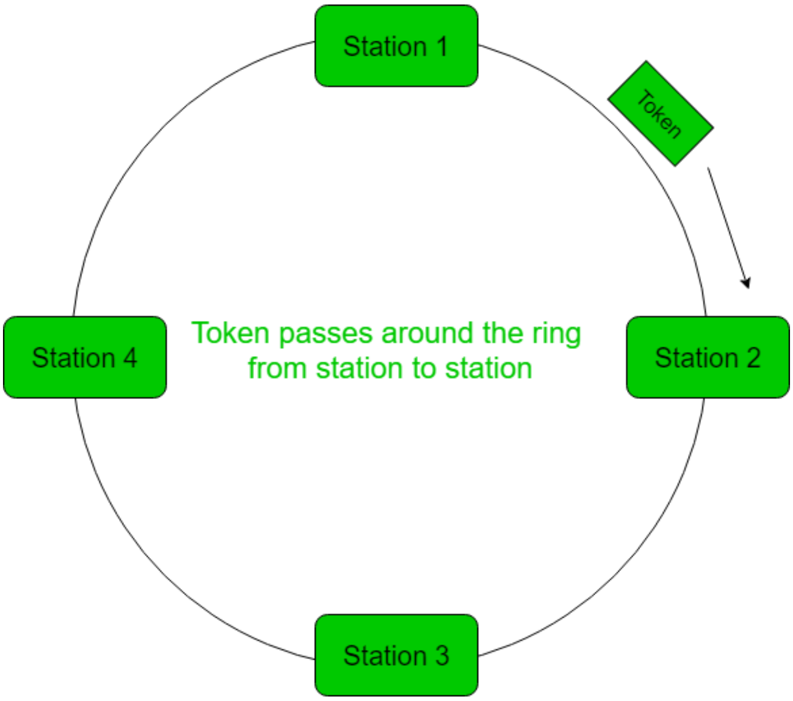




Efficiency Let T_{poll} be the time for polling and T_t be the time required for transmission of data. Then,
Efficiency = $T_t/(T_t + T_{poll})$

Token Passing

- In token passing scheme, the stations are connected logically to each other in form of ring and access to stations is governed by tokens.
- A token is a special bit pattern or a small message, which circulate from one station to the next in some predefined order.
- In Token ring, token is passed from one station to another adjacent station in the ring whereas incase of Token bus, each station uses the bus to send the token to the next station in some predefined order.
- In both cases, token represents permission to send. If a station has a frame queued for transmission when it receives the token, it can send that frame before it passes the token to the next station. If it has no queued frame, it passes the token simply.
- After sending a frame, each station must wait for all N stations (including itself) to send the token to their neighbours and the other N – 1 stations to send a frame, if they have one.
- There exists problems like duplication of token or token is lost or insertion of new station, removal of a station, which need be tackled for correct and reliable operation of this scheme.



Performance Performance of token ring can be concluded by 2 parameters:-

1. **Delay**, which is a measure of time between when a packet is ready and when it is delivered. So, the average time (delay) required to send a token to the next station = a/N.
2. **Throughput**, which is a measure of the successful traffic.

Throughput, $S = 1/(1 + a/N)$ for $a<1$

and

$S = 1/\{a(1 + 1/N)\}$ for $a>1$.

where N = number of stations

$a = T_p/T_t$

(T_p = propagation delay and T_t = transmission delay)

C. Channelization Protocols

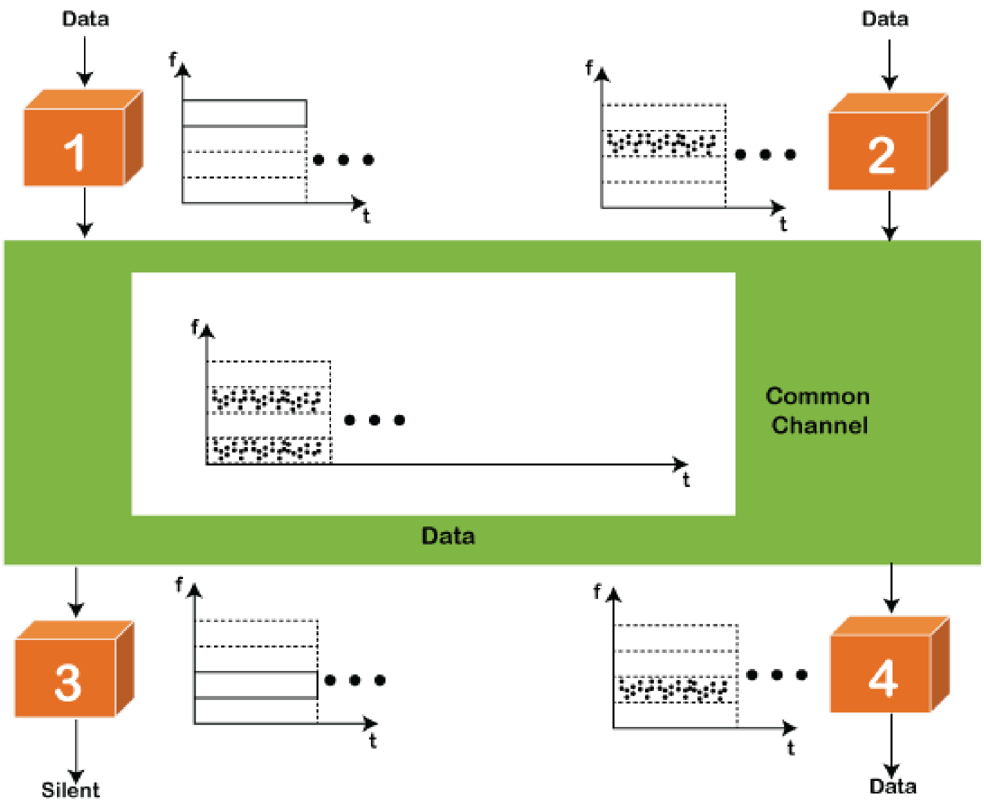
It is a channelization protocol that allows the total usable bandwidth in a shared channel to be shared across multiple stations based on their time, distance and codes. It can access all the stations at the same time to send the data frames to the channel.

Following are the various methods to access the channel based on their time, distance and codes:

1. FDMA (Frequency Division Multiple Access)
2. TDMA (Time Division Multiple Access)
3. CDMA (Code Division Multiple Access)

FDMA

It is a frequency division multiple access (**FDMA**) method used to divide the available bandwidth into equal bands so that multiple users can send data through a different frequency to the subchannel. Each station is reserved with a particular band to prevent the crosstalk between the channels and interferences of stations.



TDMA

Time Division Multiple Access (**TDMA**) is a channel access method. It allows the same frequency bandwidth to be shared across multiple stations. And to avoid collisions in the shared channel, it divides the channel into different frequency slots that allocate stations to transmit the data frames. The same **frequency** bandwidth into the shared channel by dividing the signal into various time slots to transmit it. However, TDMA has an overhead of synchronization that specifies each station's time slot by adding synchronization bits to each slot.

CDMA

The code division multiple access (CDMA) is a channel access method. In CDMA, all stations can simultaneously send the data over the same channel. It means that it allows each station to transmit the data frames with full frequency on the shared channel at all times. It does not require the division of bandwidth on a shared channel based on time slots. If multiple stations send data to a channel simultaneously, their data frames are separated by a unique code sequence. Each station has a different unique code for transmitting the data over a shared channel. For example, there are multiple users in a room that are continuously speaking. Data is received by the users if only two-person interact with each other using the same language. Similarly, in the network, if different stations communicate with each other simultaneously with different code language.