

## Data Link Layer

Data Link Layer is second layer of OSI Layered Model. This layer is one of the most complicated layers and has complex functionalities and liabilities. Data link layer hides the details of underlying hardware and represents itself to upper layer as the medium to communicate.

Data link layer works between two hosts which are directly connected in some sense. This direct connection could be point to point or broadcast. Systems on broadcast network are said to be on same link. The work of data link layer tends to get more complex when it is dealing with multiple hosts on single collision domain.

Data link layer is responsible for converting data stream to signals bit by bit and to send that over the underlying hardware. At the receiving end, Data link layer picks up data from hardware which are in the form of electrical signals, assembles them in a recognizable frame format, and hands over to upper layer.

Data link layer has two sub-layers:

- **Logical Link Control:** It deals with protocols, flow-control, and error control
- **Media Access Control:** It deals with actual control of media

### Functionality of Data-link Layer

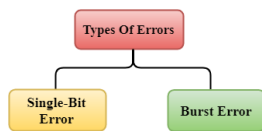
Data link layer does many tasks on behalf of upper layer. These are:

- **Framing**  
Data-link layer takes packets from Network Layer and encapsulates them into Frames. Then, it sends each frame bit-by-bit on the hardware. At receiver's end, data link layer picks up signals from hardware and assembles them into frames.
- **Addressing**  
Data-link layer provides layer-2 hardware addressing mechanism. Hardware address is assumed to be unique on the link. It is encoded into hardware at the time of manufacturing.
- **Synchronization**  
When data frames are sent on the link, both machines must be synchronized in order for transfer to take place.
- **Error Control**  
Sometimes signals may have encountered a problem in transition and the bits are flipped. These errors are detected and attempted to recover actual data bits. It also provides an error reporting mechanism to the sender.
- **Flow Control**  
Stations on same link may have different speed or capacity. Data-link layer ensures flow control that enables both machines to exchange data on same speed.
- **Multi-Access**  
When a host on the shared link tries to transfer the data, it has a high probability of collision. Data-link layer provides a mechanism such as CSMA/CD to equip capability of accessing a shared media among multiple systems.

## Error Detection

When data is transmitted from one device to another device, the system does not guarantee whether the data received by the device is identical to the data transmitted by another device. An Error is a situation when the message received at the receiver end is not identical to the message transmitted.

### Types Of Errors

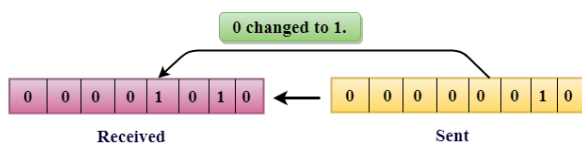


Errors can be classified into two categories:

- Single-Bit Error
- Burst Error

### Single-Bit Error:

The only one bit of a given data unit is changed from 1 to 0 or from 0 to 1.



In the above figure, the message which is sent is corrupted as single-bit, i.e., 0 bit is changed to 1.

Keep Watching

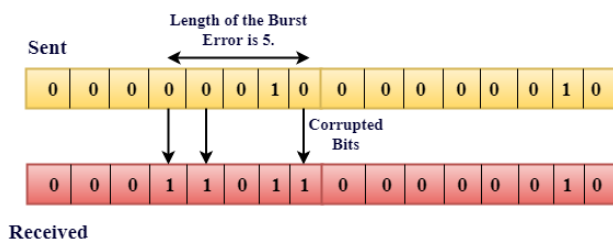
**Single-Bit Error** does not appear more likely in Serial Data Transmission. For example, Sender sends the data at 10 Mbps, this means that the bit lasts only for 1  $\mu$ s and for a single-bit error to occurred, a noise must be more than 1  $\mu$ s.

Single-Bit Error mainly occurs in Parallel Data Transmission. For example, if eight wires are used to send the eight bits of a byte, if one of the wire is noisy, then single-bit is corrupted per byte.

### Burst Error:

The two or more bits are changed from 0 to 1 or from 1 to 0 is known as Burst Error.

The Burst Error is determined from the first corrupted bit to the last corrupted bit.



The duration of noise in Burst Error is more than the duration of noise in Single-Bit.

Burst Errors are most likely to occur in Serial Data Transmission.

The number of affected bits depends on the duration of the noise and data rate.

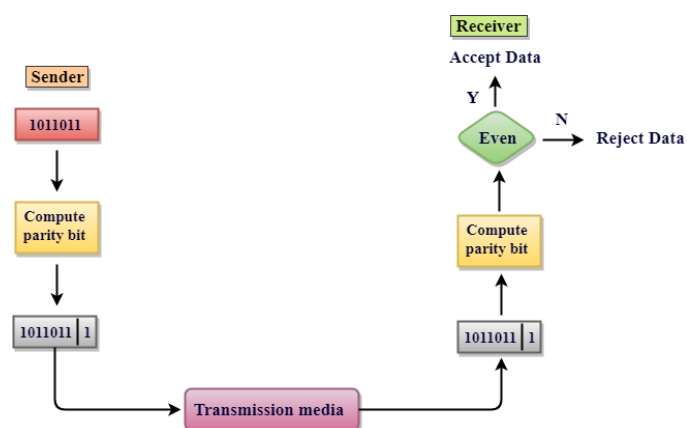
## Error Detecting Techniques:

The most popular Error Detecting Techniques are:

- Single parity check
- Two-dimensional parity check
- Checksum
- Cyclic redundancy check

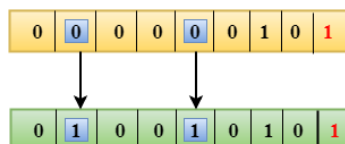
### Single Parity Check

- Single Parity checking is the simple mechanism and inexpensive to detect the errors.
- In this technique, a redundant bit is also known as a parity bit which is appended at the end of the data unit so that the number of 1s becomes even. Therefore, the total number of transmitted bits would be 9 bits.
- If the number of 1s bits is odd, then parity bit 1 is appended and if the number of 1s bits is even, then parity bit 0 is appended at the end of the data unit.
- At the receiving end, the parity bit is calculated from the received data bits and compared with the received parity bit.
- This technique generates the total number of 1s even, so it is known as even-parity checking.



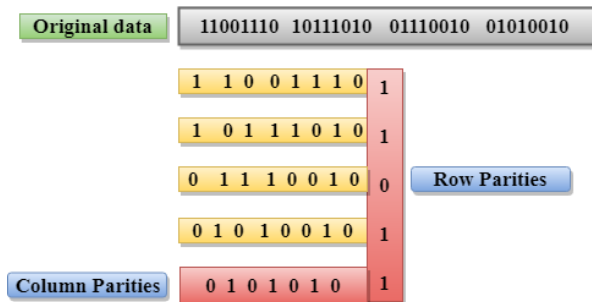
### Drawbacks Of Single Parity Checking

- It can only detect single-bit errors which are very rare.
- If two bits are interchanged, then it cannot detect the errors.



### Two-Dimensional Parity Check

- Performance can be improved by using **Two-Dimensional Parity Check** which organizes the data in the form of a table.
- Parity check bits are computed for each row, which is equivalent to the single-parity check.
- In Two-Dimensional Parity check, a block of bits is divided into rows, and the redundant row of bits is added to the whole block.
- At the receiving end, the parity bits are compared with the parity bits computed from the received data.



### Drawbacks Of 2D Parity Check

- If two bits in one data unit are corrupted and two bits exactly the same position in another data unit are also corrupted, then 2D Parity checker will not be able to detect the error.
- This technique cannot be used to detect the 4-bit errors or more in some cases.

### Checksum

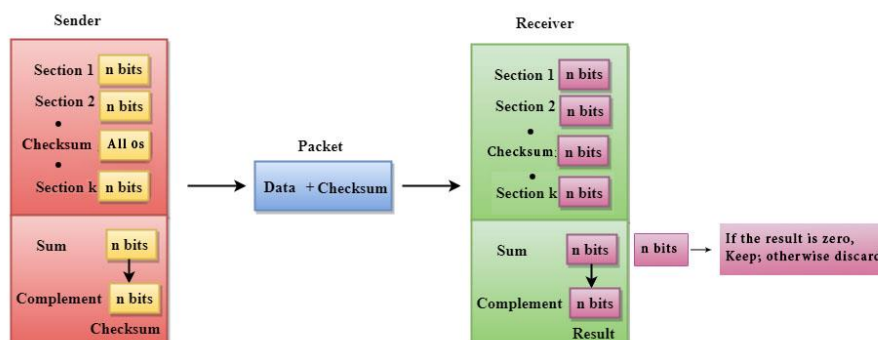
A Checksum is an error detection technique based on the concept of redundancy.

It is divided into two parts:

#### Checksum Generator

A Checksum is generated at the sending side. Checksum generator subdivides the data into equal segments of  $n$  bits each, and all these segments are added together by using one's complement arithmetic. The sum is complemented and appended to the original data, known as checksum field. The extended data is transmitted across the network.

Suppose  $L$  is the total sum of the data segments, then the checksum would be  $?L$



1. The Sender follows the given steps:
2. The block unit is divided into  $k$  sections, and each of  $n$  bits.
3. All the  $k$  sections are added together by using one's complement to get the sum.
4. The sum is complemented and it becomes the checksum field.
5. The original data and checksum field are sent across the network.

### Checksum Checker

A Checksum is verified at the receiving side. The receiver subdivides the incoming data into equal segments of  $n$  bits each, and all these segments are added together, and then this sum is complemented. If the complement of the sum is zero, then the data is accepted otherwise data is rejected.

1. The Receiver follows the given steps:
2. The block unit is divided into  $k$  sections and each of  $n$  bits.
3. All the  $k$  sections are added together by using one's complement algorithm to get the sum.
4. The sum is complemented.
5. If the result of the sum is zero, then the data is accepted otherwise the data is discarded.

### Cyclic Redundancy Check (CRC)

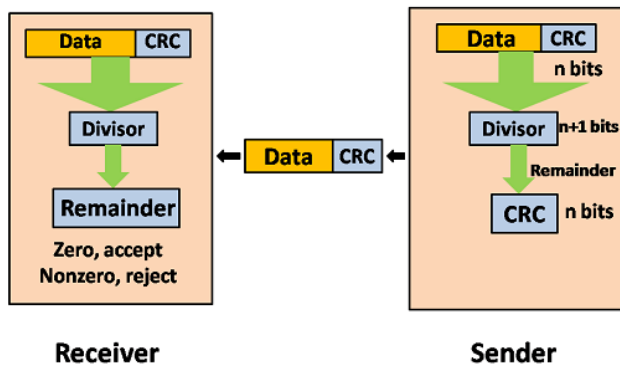
CRC is a redundancy error technique used to determine the error.

**Following are the steps used in CRC for error detection:**

- In CRC technique, a string of  $n$  0s is appended to the data unit, and this  $n$  number is less than the number of bits in a predetermined number, known as division which is  $n+1$  bits.
- Secondly, the newly extended data is divided by a divisor using a process is known as binary division. The remainder generated from this division is known as CRC remainder.
- Thirdly, the CRC remainder replaces the appended 0s at the end of the original data. This newly generated unit is sent to the receiver.
- The receiver receives the data followed by the CRC remainder. The receiver will treat this whole unit as a single unit, and it is divided by the same divisor that was used to find the CRC remainder.

If the resultant of this division is zero which means that it has no error, and the data is accepted.

If the resultant of this division is not zero which means that the data consists of an error. Therefore, the data is discarded.

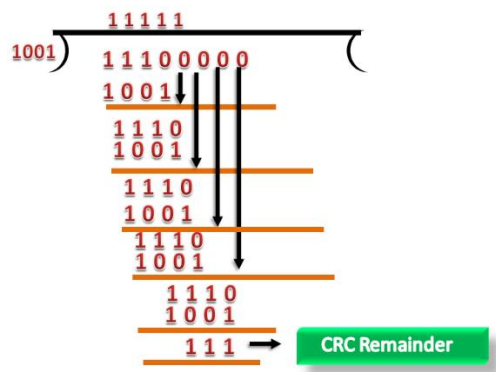


Let's understand this concept through an example:

**Suppose the original data is 11100 and divisor is 1001.**

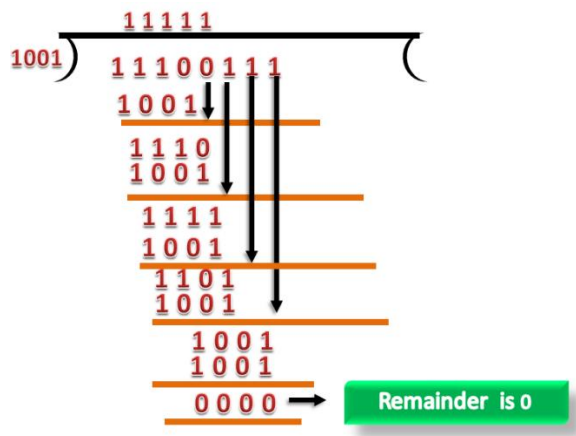
### CRC Generator

- A CRC generator uses a modulo-2 division. Firstly, three zeroes are appended at the end of the data as the length of the divisor is 4 and we know that the length of the string 0s to be appended is always one less than the length of the divisor.
- Now, the string becomes 11100000, and the resultant string is divided by the divisor 1001.
- The remainder generated from the binary division is known as CRC remainder. The generated value of the CRC remainder is 111.
- CRC remainder replaces the appended string of 0s at the end of the data unit, and the final string would be 11100111 which is sent across the network.



### CRC Checker

- The functionality of the CRC checker is similar to the CRC generator.
- When the string 11100111 is received at the receiving end, then CRC checker performs the modulo-2 division.
- A string is divided by the same divisor, i.e., 1001.
- In this case, CRC checker generates the remainder of zero. Therefore, the data is accepted.



## Error Correction

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:

$$2^r \geq d + r + 1$$

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  $1, 2, \dots, 2^{k-1}$ .
- 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:**  $2^r \geq d+r+1$

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

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<sup>1</sup>, 2<sup>2</sup>**.

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:

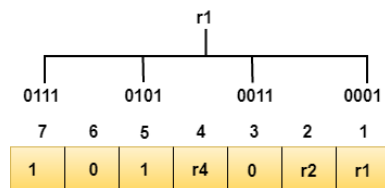
7	6	5	4	3	2	1
1	0	1	r4	0	r2	r1

### Determining the Parity bits



## Determining the r1 bit

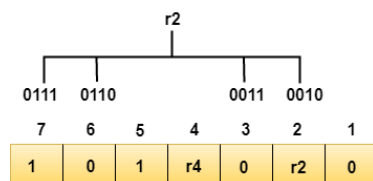
The r1 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the first position.



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

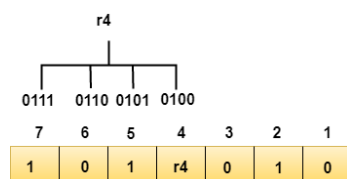
The r2 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the second position.



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:

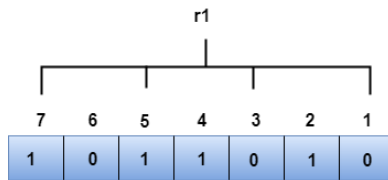
7	6	5	4	3	2	1
1	0	1	0	0	1	0

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

---

## R1 bit

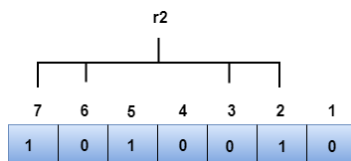
The bit positions of the r1 bit are 1,3,5,7



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

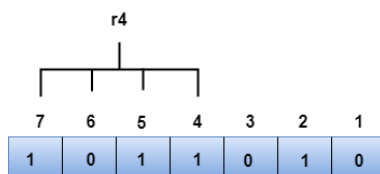
The bit positions of r2 bit are 2,3,6,7.



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.



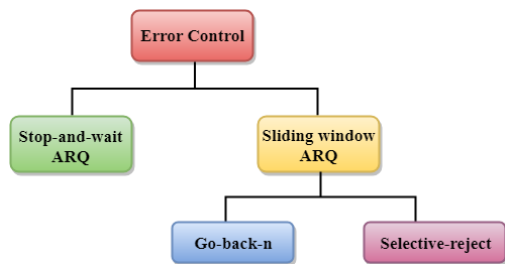
We observe from the above figure that the binary representation of r4 is 1011. Now, we perform the even-parity check, the total number of 1s appearing in the r4 bit is an odd number. Therefore, the value of r4 is 1.

- The binary representation of redundant bits, i.e.,  $r_4r_2r_1$  is 100, and its corresponding decimal value is 4. Therefore, the error occurs in a 4<sup>th</sup> bit position. The bit value must be changed from 1 to 0 to correct the error.

## 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.

### Sliding Window ARQ

SlidingWindow ARQ is a technique used for continuous transmission error control.

### Three Features used for retransmission:

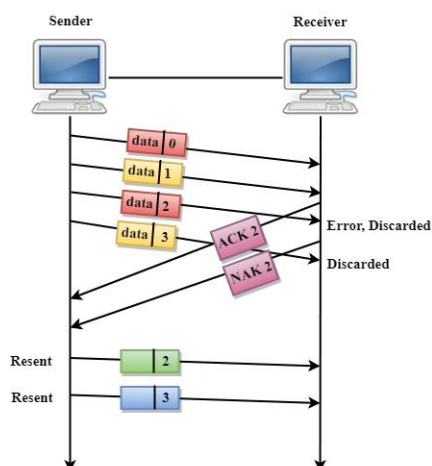
- In this case, the sender keeps the copies of all the transmitted frames until they have been acknowledged. Suppose the frames from 0 through 4 have been transmitted, and the last acknowledgement was for frame 2, the sender has to keep the copies of frames 3 and 4 until they receive correctly.
- The 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.

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.

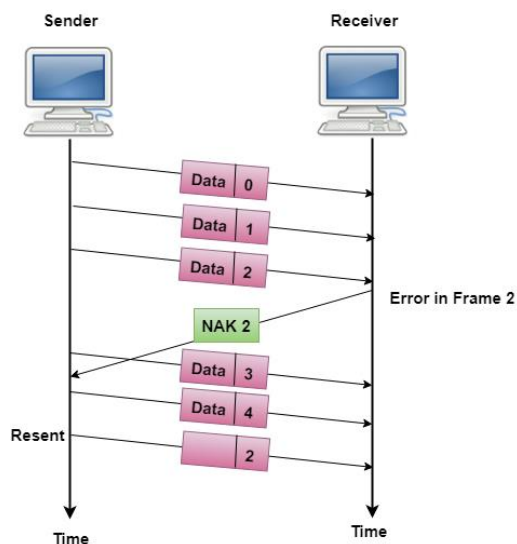
- **Lost Data Frame:** In Sliding window protocols, data frames are sent sequentially. If any of the frames is lost, then the next frame arrive at the receiver is out of sequence. The receiver checks the sequence number

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-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.



### Multiple Access Protocols in Computer Network

The Data Link Layer is responsible for transmission of data between two nodes. Its main functions are-

- Data Link Control
- Multiple Access Control



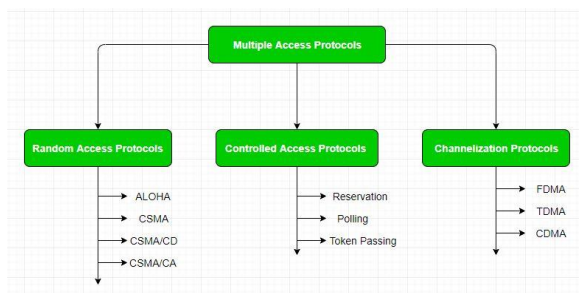
### Data Link control –

The data link control is responsible for reliable transmission of message over transmission channel by using techniques like framing, error control and flow control. For Data link control refer to – [Stop and Wait ARQ](#)

### Multiple Access Control –

If there is a dedicated link between the sender and the receiver then data link control layer is sufficient, however if there is no dedicated link present then multiple stations can access the channel simultaneously. Hence multiple access protocols are required to decrease collision and avoid crosstalk. For example, in a classroom full of students, when a teacher asks a question and all the students (or stations) start answering simultaneously (send data at same time) then a lot of chaos is created( data overlap or data lost) then it is the job of the teacher (multiple access protocols) to manage the students and make them answer one at a time.

Thus, protocols are required for sharing data on non dedicated channels. Multiple access protocols can be subdivided further as –



**1. Random Access Protocol:** In this, all stations have same superiority that is no station has more priority than another station. Any station can send data depending on medium's state( idle or busy). It has two features:

1. There is no fixed time for sending data
2. There is no fixed sequence of stations sending data

The Random access protocols are further subdivided as:

**(a) ALOHA –** It was designed for wireless LAN but is also applicable for shared medium. In this, multiple stations can transmit data at the same time and can hence lead to collision and data being garbled.

- **Pure Aloha:**

When a station sends data it waits for an acknowledgement. If the acknowledgement doesn't come within the allotted time then the station waits for a random amount of time called back-off time ( $T_b$ ) and re-sends the data. Since different stations wait for different amount of time, the probability of further collision decreases.

Vulnerable Time =  $2 \times$  Frame transmission time

Throughput =  $G \exp\{-2 \times G\}$

Maximum throughput = 0.184 for  $G=0.5$

- **Slotted Aloha:**

It is similar to pure aloha, except that we divide time into slots and sending of data is allowed only at the beginning of these slots. If a station misses out the allowed time, it must wait for the next slot. This reduces the probability of collision.

Vulnerable Time = Frame transmission time

$$\text{Throughput} = G \exp\{-*G\}$$

Maximum throughput = 0.368 for G=1

For more information on ALOHA refer – [LAN Technologies](#)

**(b) CSMA** – Carrier Sense Multiple Access ensures fewer collisions as the station is required to first sense the medium (for idle or busy) before transmitting data. If it is idle then it sends data, otherwise it waits till the channel becomes idle. However there is still chance of collision in CSMA due to propagation delay. For example, if station A wants to send data, it will first sense the medium. If it finds the channel idle, it will start sending data. However, by the time the first bit of data is transmitted (delayed due to propagation delay) from station A, if station B requests to send data and senses the medium it will also find it idle and will also send data. This will result in collision of data from station A and B.

CSMA access modes-

- **1-persistent:** The node senses the channel, if idle it sends the data, otherwise it continuously keeps on checking the medium for being idle and transmits unconditionally (with 1 probability) as soon as the channel gets idle.
- **Non-Persistent:** The node senses the channel, if idle it sends the data, otherwise it checks the medium after a random amount of time (not continuously) and transmits when found idle.
- **P-persistent:** The node senses the medium, if idle it sends the data with p probability. If the data is not transmitted ((1-p) probability) then it waits for some time and checks the medium again, now if it is found idle then it send with p probability. This repeat continues until the frame is sent. It is used in Wifi and packet radio systems.
- **O-persistent:** Superiority of nodes is decided beforehand and transmission occurs in that order. If the medium is idle, node waits for its time slot to send data.

**(c) CSMA/CD** – Carrier sense multiple access with collision detection. Stations can terminate transmission of data if collision is detected. For more details refer – [Efficiency of CSMA/CD](#)

**(d) CSMA/CA** – Carrier sense multiple access with collision avoidance. The process of collisions detection involves sender receiving acknowledgement signals. If there is just one signal (its own) then the data is successfully sent but if there are two signals (its own and the one with which it has collided) then it means a collision has occurred. To distinguish between these two cases, collision must have a lot of impact on received signal. However it is not so in wired networks, so CSMA/CA is used in this case.

CSMA/CA avoids collision by:

1. **Interframe space** – Station waits for medium to become idle and if found idle it does not immediately send data (to avoid collision due to propagation delay) rather it waits for a period of time called Interframe space or IFS. After this time it again checks the medium for being idle. The IFS duration depends on the priority of station.
2. **Contention Window** – It is the amount of time divided into slots. If the sender is ready to send data, it chooses a random number of slots as wait time which doubles every time medium is not found idle. If the medium is found busy it does not restart the entire process, rather it restarts the timer when the channel is found idle again.
3. **Acknowledgement** – The sender re-transmits the data if acknowledgement is not received before time-out.

## 2. Controlled Access:

In this, the data is sent by that station which is approved by all other stations. For further details refer – [Controlled Access Protocols](#)

## 3. Channelization:

In this, the available bandwidth of the link is shared in time, frequency and code to multiple stations to access channel simultaneously.

- **Frequency Division Multiple Access (FDMA)** – The available bandwidth is divided into equal bands so that each station can be allocated its own band. Guard bands are also added so that no two bands overlap to avoid crosstalk and noise.
- **Time Division Multiple Access (TDMA)** – In this, the bandwidth is shared between multiple stations. To avoid collision time is divided into slots and stations are allotted these slots to

transmit data. However there is a overhead of synchronization as each station needs to know its time slot. This is resolved by adding synchronization bits to each slot. Another issue with TDMA is propagation delay which is resolved by addition of guard bands.

For more details refer – [Circuit Switching](#)

- **Code Division Multiple Access (CDMA)** – One channel carries all transmissions simultaneously. There is neither division of bandwidth nor division of time. For example, if there are many people in a room all speaking at the same time, then also perfect reception of data is possible if only two person speak the same language. Similarly, data from different stations can be transmitted simultaneously in different code languages.

Difference between FDMA, TDMA and CDMA

### 1. Frequency Division Multiple Access (FDMA) :

FDMA is a type of channelization protocol. In this bandwidth is divided into various frequency bands. Each station is allocated with band to send data and that band is reserved for particular station for all the time which is as follows :

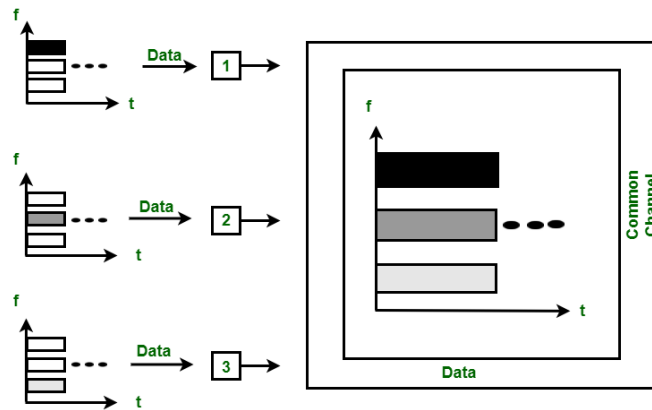


Figure – FDMA

The frequency bands of different stations are separated by small band of unused frequency and that unused frequency bands are called as guard bands that prevents the interference of stations. It is like access method in data link layer in which [data link layer](#) at each station tells its physical layer to make a band pass signal from the data passed to it. The signal is created in the allocated band and there is no physical multiplexer at the physical layer.

### 2. Time Division Multiple Access (TDMA) :

TDMA is the channelization protocol in which bandwidth of channel is divided into various stations on the time basis. There is a time slot given to each station, the station can transmit data during that time slot only which is as follows :

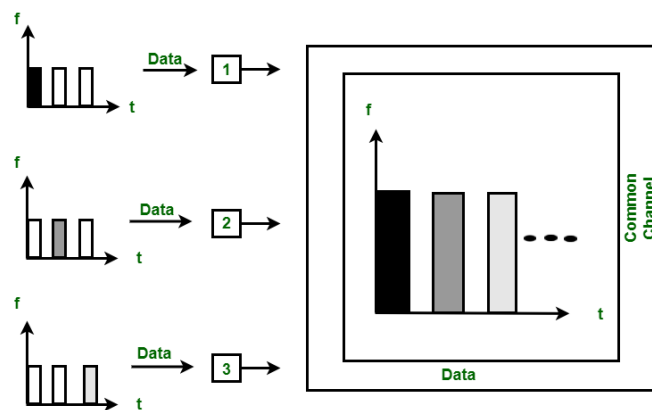


Figure – TDMA



Each station must be aware of its beginning of time slot and the location of the time slot. TDMA requires synchronization between different stations. It is a type of access method in the data link layer. At each station, the data link layer tells the station to use the allocated time slot.

### 3. Code Division Multiple Access (CDMA) :

In CDMA, all the stations can transmit data simultaneously. It allows each station to transmit data over the entire frequency all the time. Multiple simultaneous transmissions are separated by unique code sequences. Each user is assigned with a unique code sequence.

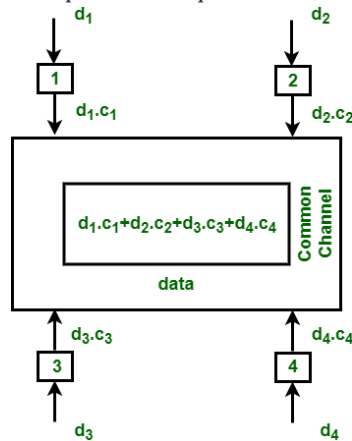


Figure – TDMA

In the above figure, there are 4 stations marked as 1, 2, 3 and 4. Data assigned with respective stations as  $d_1$ ,  $d_2$ ,  $d_3$  and  $d_4$  and the code assigned with respective stations as  $c_1$ ,  $c_2$ ,  $c_3$  and  $c_4$ .