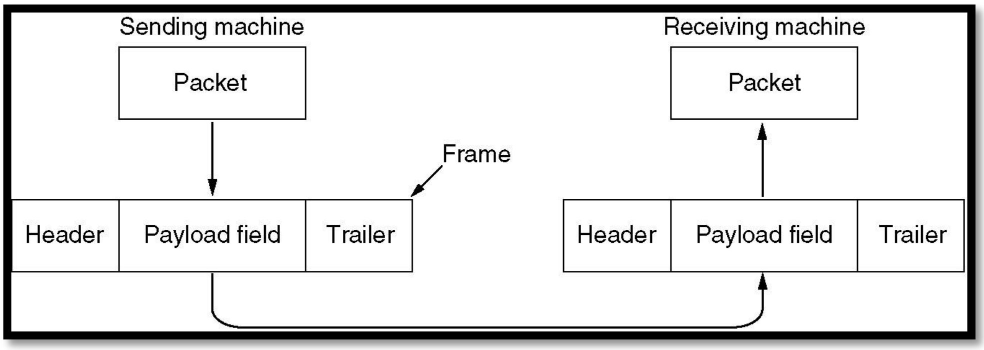
Data Link Layer

Data-link layer has the responsibility of transferring datagram from one **node to physically adjacent node** over a link. The data link layer is divided into two sublayers: **Logical Link Control (LLC)** and **Media Access Control (MAC)**. The LLC sublayer manages communications between devices over a single link of a network. The MAC sublayer governs protocol access to the physical network medium.

Main Services Provided by Data link layer

- **Framing**

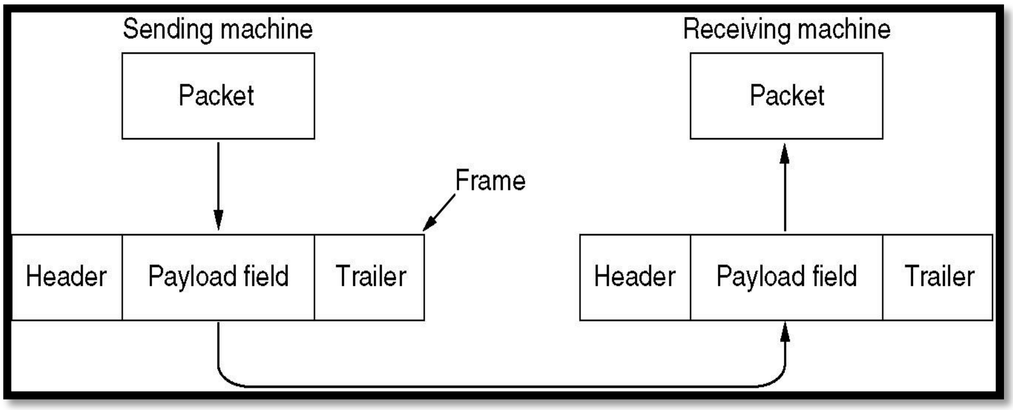


- **Error Control**
- **Flow Control**

Functions of the Data Link Layer

- Provide service interface to the network layer
- Dealing with transmission errors
- Regulating data flow (Slow receivers not swamped by fast senders)

Relationship between Packets and Frames



Data Link Protocols

Elementary Data Link Protocols

The main job of elementary data link layer protocols is to receive packets from network layer, create the frame and send it to physical layer, or vice versa. These are some elementary data link layer protocols:

An Unrestricted Simplex Protocol (SP)	one direction transmitted data
A Simplex Stop-and-Wait Protocol(SWP)	flooding control
A Simplex Protocol for a Noisy Channel(SPN)	limit send and receive between sender and receiver, capacities are limited

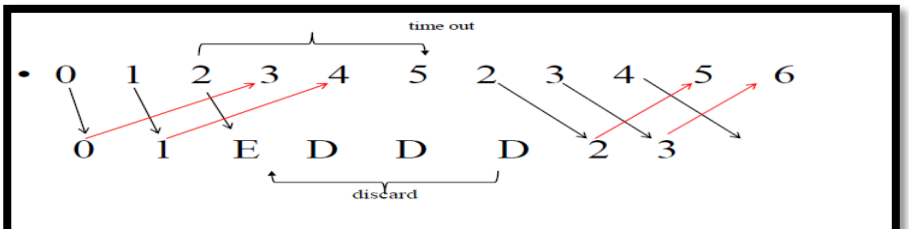
Sliding Window Protocols

The next three protocols are bidirectional protocols that belong to a class called sliding window protocols.

A One-Bit Sliding Window Protocol(SWP)	1- assign variable 2- define frame 3- accept frame
A Protocol Using Go Back N protocol	Discarding &Buffering
A Protocol Using Selective Repeat (SRP)	accept and buffer delay and effected frame) without ACK

Go Back N Protocol

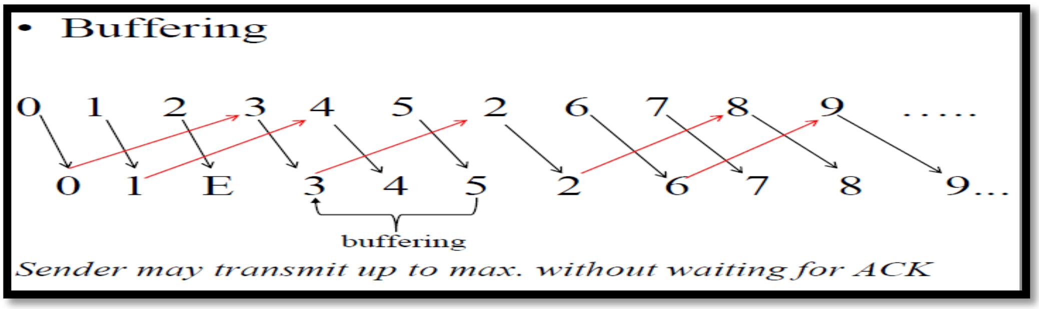
If there is one frame k missing, the receiver simply discards all subsequent frames k+1, k+2…., sending no acknowledgments. So the sender will retransmit frames from k onwards. This can be a waste of bandwidth.



Discarding operation

Selective Repeat Protocol (SRP)

Another strategy is to re-send only the ones that are actually lost or damaged. The receiver buffers all the frames after the lost one. When the sender finally noticed the problem (e.g. no ack for the lost frame is received within time-out limit), the sender retransmits the frame in question.



Sender may transmit up to max. without waiting for ACK

PPP – Point *to* Point Protocol

- Carry network data of **any** network layer protocol at **the same time**
- Error detection (**no** correction)
- has a very simple mechanism for **error control(** A CRC field is used to **detect** errors)
- Does **not provide** flow control
- Connection life, signal link, negotiator

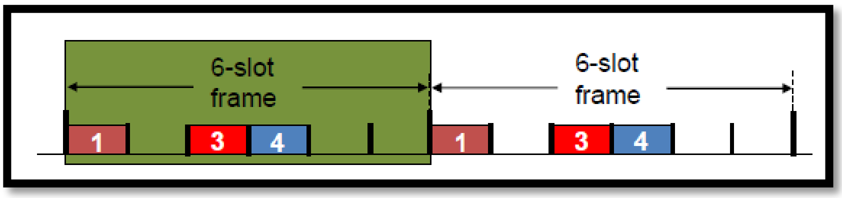
Media Access control Protocols

There are three broad classes of MAC protocols, these are:

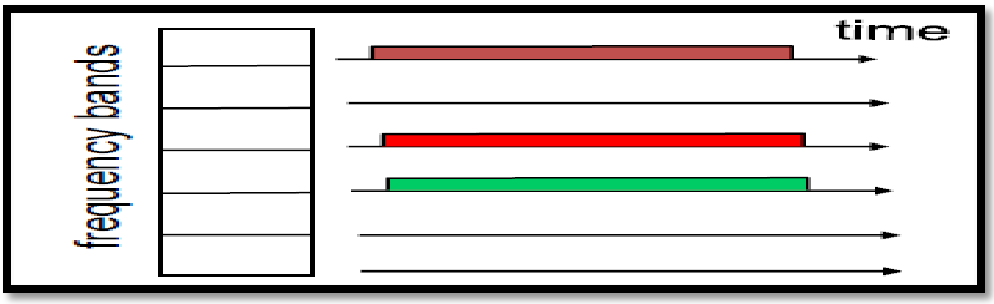
1. ***Channel Partitioning***
   - divide channel into smaller "pieces" (time slots, frequency, code)
   - allocate piece to node for exclusive use
2. ***"taking turns"***
   - nodes take turns, but nodes with more to send can take longer turns
3. ***Random Access***
   - channel not divided, allow collisions
   - "recover" from collisions

Channel partitioning MAC protocols

1. **Time Division Multiple Access** (TDMA)
   - Access to channel in "rounds"
   - Each station gets fixed length slot (length = pkt trans time) in each round
   - Unused slots go idle



2. **Frequency Division Multiple Access (FDMA)**
    - Channel spectrum divided into frequency bands
    - Each station assigned fixed frequency band
    - Unused transmission time in frequency bands go idle



Taking Turns

1. **Polling**
    - master node “invites” slave nodes to transmit in turn
    - typically used with “dumb” slave devices

Access Method

Whenever multiple users have unregulated access to a single line, there is a danger of **signals overlapping and destroying each other**. Such overlaps, which turn the signals into unusable noise, are called **collisions**.

As traffic increases on a multiple access link, so do collisions. A LAN therefore needs **a mechanism to coordinate traffic, minimize the number of collisions that occur, and maximize the number of frames that are delivered successfully**.

The access mechanism used in an Ethernet is called ***Carrier Sense Multiple Access (CSMA)*** standardized in IEEE 802.3, and then this protocol developed to ***carrier sense multiple access with collision detection (CSMA/CD)***

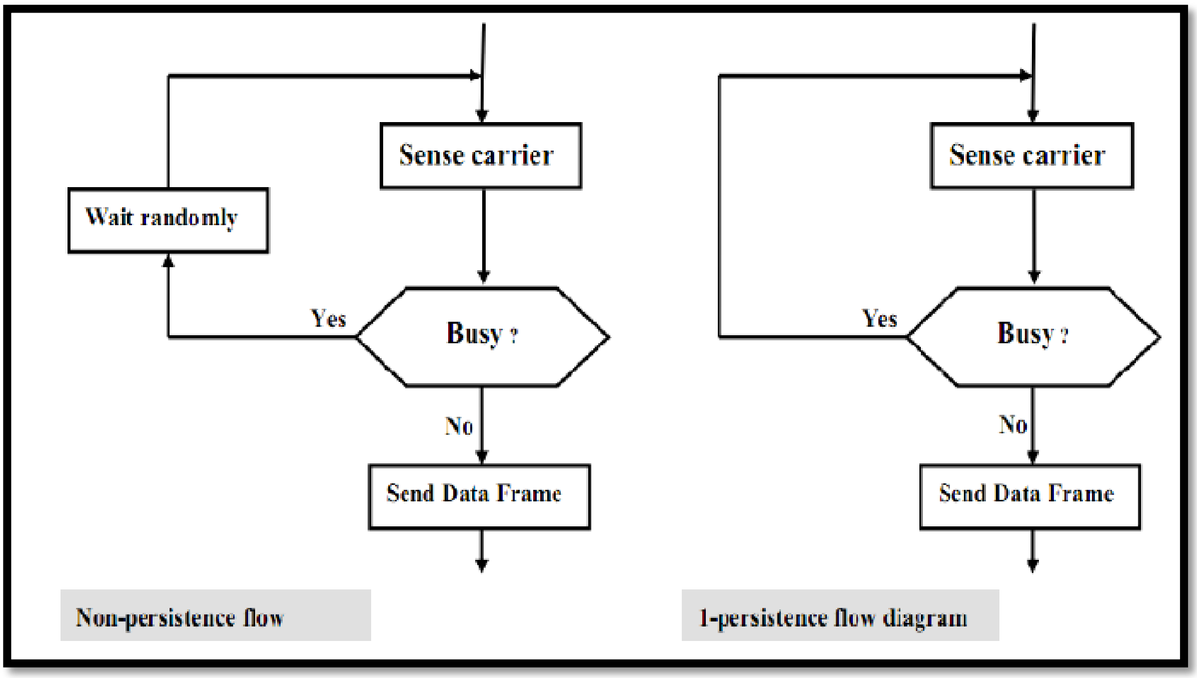
- **CSMA (carrier sense multiple access)**
  CSMA: listen before transmit
    - If channel sensed idle: transmit entire frame
    - if channel sensed busy, defer transmission

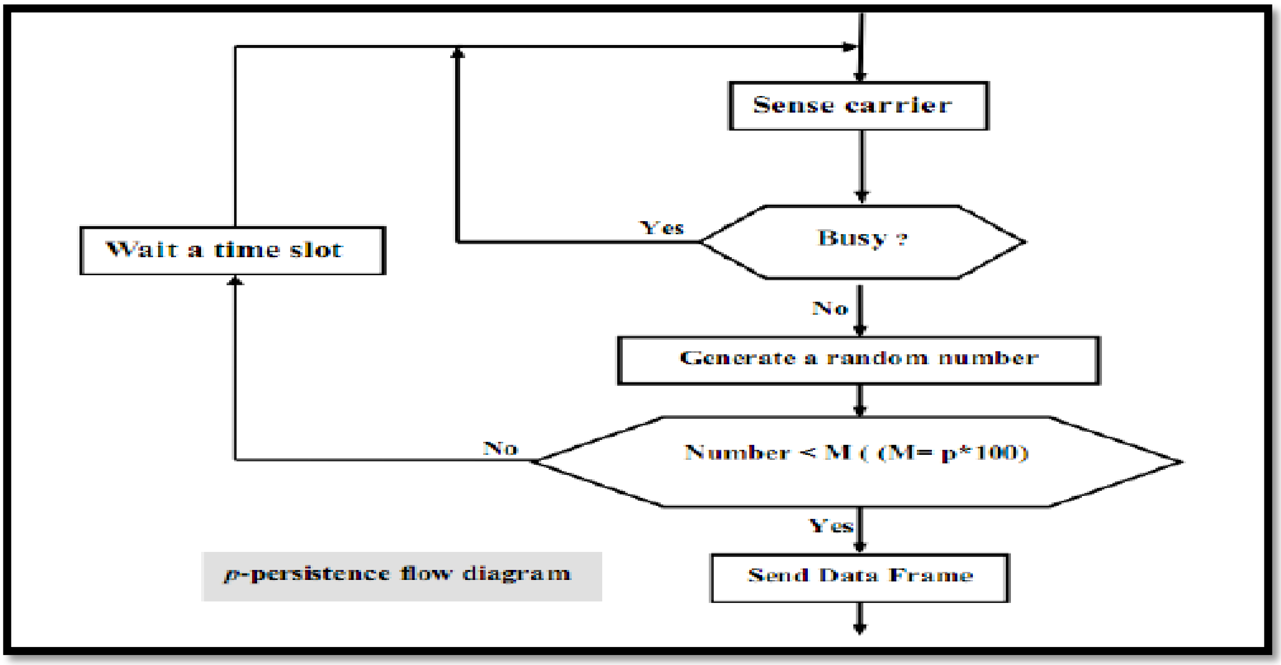
- **CSMA/CD (collision detection)**
  CSMA/CD: carrier sensing, deferral as in CSMA
    - collisions detected within short time
    - colliding transmissions aborted, reducing channel wastage

Carrier Sense: when a station in an Ethernet network has data to transmit ,it first see the network if it is use by other stations this is carrier sense .**under three case of persistence strategy**

Non-persistence	senses the line if it is idle it sends immediately if the line busy ,it waits a **random time** then sense the line again this method reduce the chance of collision but it also reduced network efficiency.
1-persistence	After the station finds the line idle it sends its data immediately (with probability 1) this method increases the chance of collision.
p-persistence	**After the station finds the line idle it may transmit or no**. here the probability of sending is defined by P and probability of refusing is (1-P) for example if p=0.3 then station sends with 30% of the time and refusing 70% of the time. The station generates a **random number** between 1 and 100 if the number generated is less than 30 the station sends its data else it waits one slot time before sensing the medium again this method reduces the chance of collision and increasing network efficiency.

Jam Signal: when system detected a collision it immediately stop transmitting data and starts sending this signal any system received packet must discard this packet and should not attempt to transmit any data until network has cleared.





p-persistence flow diagram

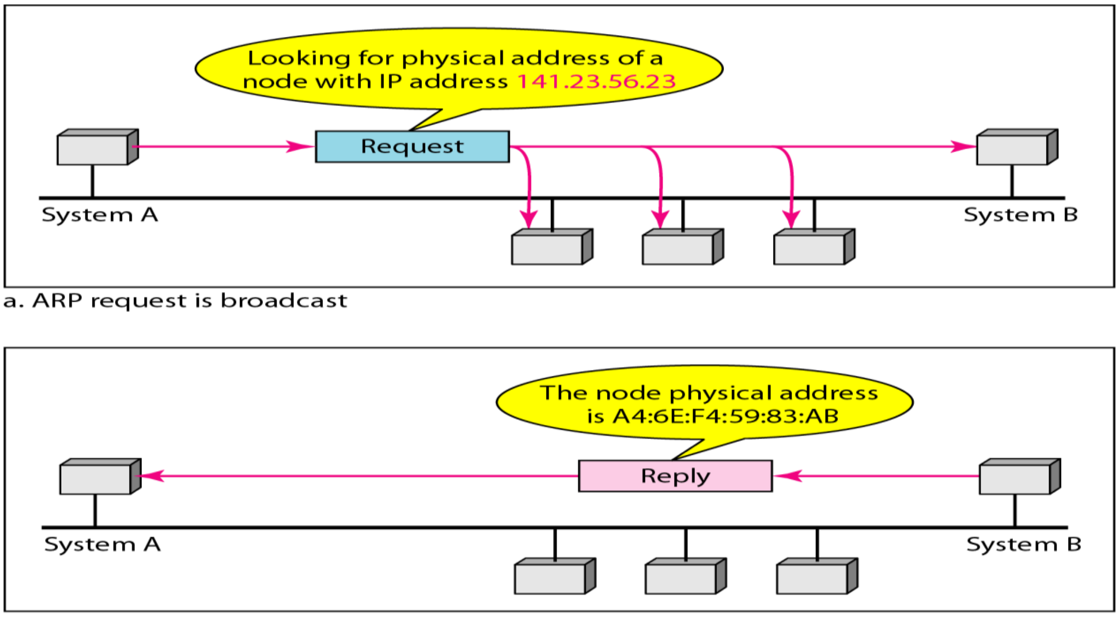
Address Resolution Protocol (ARP)

The delivery of a packet to a host or a router requires two levels of addressing: **logical** and **physical**.

- ARP Maps IP addresses to MAC addresses
- ARP **Request** is a broadcast but ARP **reply** is Unicast.
- **ARP tables** contain the MAC and IP addresses of other devices on the network

ARP Operation:

1. A wants to send datagram to B
   - B’s MAC address not in A’s ARP table.
2. A broadcasts ARP query packet, containing B's IP address
   - dest MAC address = FF-FF-FF-FF-FF-FF
3. all nodes on LAN receive ARP query
4. B receives ARP packet, replies to A with its (B's) MAC address
   - frame sent to A’s MAC address (unicast)


Looking for physical address of a node with IP address 141.23.56.23
Request
System A
System B
The node physical address is A4:6E:F4:59:83:AB
Reply
System A
System B

a. ARP request is broadcast

b. ARP reply is unicast

Layer 2 Tunneling Protocol (L2TP)

- Is an **extension** of the Point-to-Point Tunneling Protocol (PPTP).
- **Used by** an Internet service provider (ISP) to enable the operation of a **virtual private network** (VPN) over the Internet.
- The goal of a Virtual Private Network (VPN) is to **provide private communications within the public Internet Infrastructure**

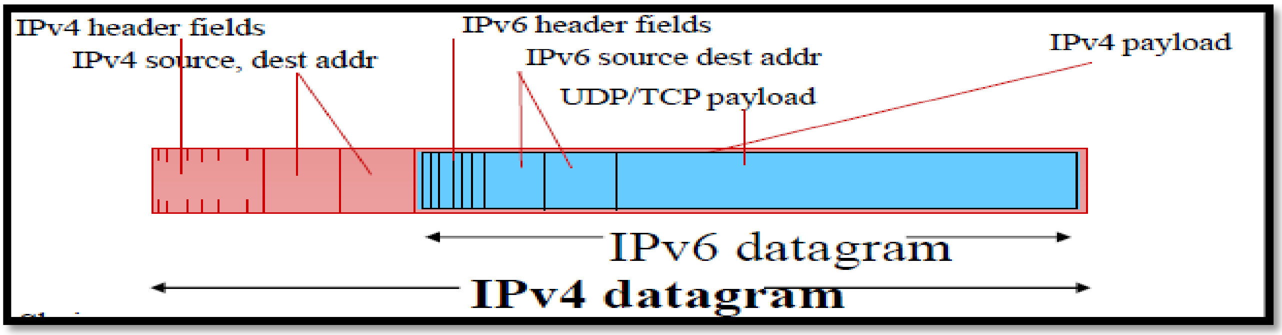
Why is there a need for VPN?

- **Internet has insufficient security mechanisms**
- IP packets are not authenticated or encrypted
- Users with access to network can read content of IP traffic

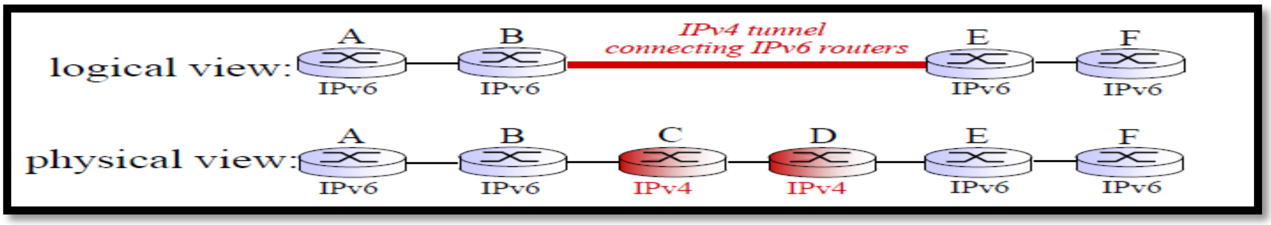
Transition from IPv4 to IPv6

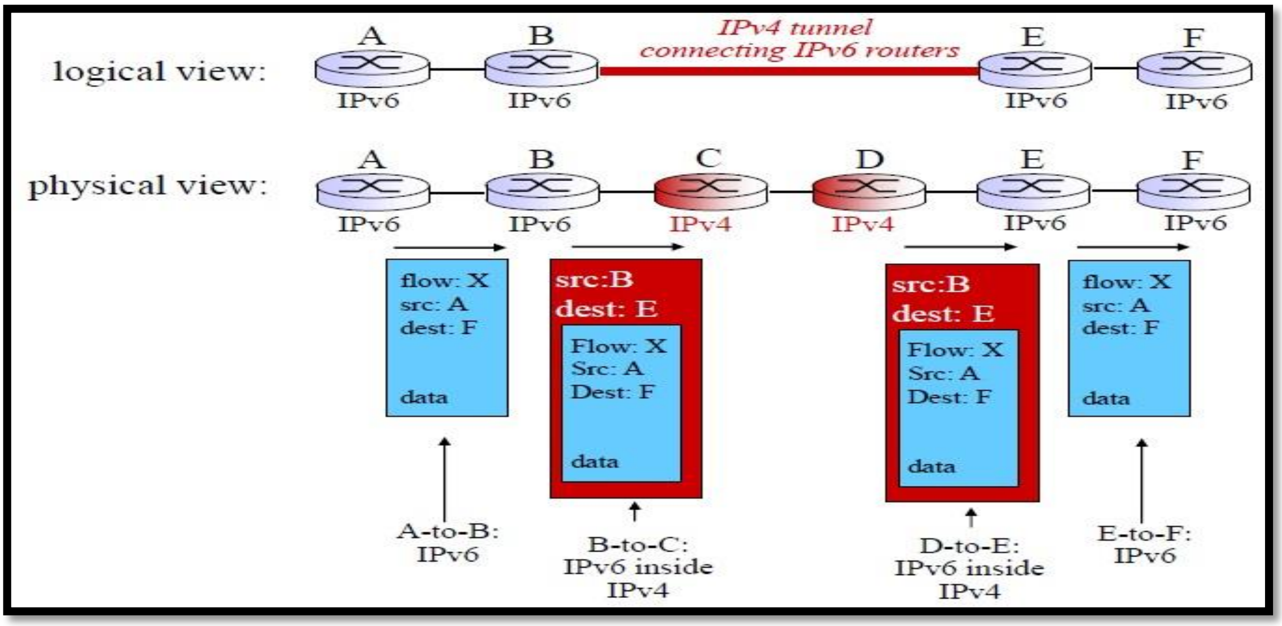
- Not all routers can be upgraded simultaneously
- How will network operate with mixed IPv4 and IPv6 routers?

Answer: this can be done by Tunneling: IPv6 datagram carried as *payload* in IPv4 datagram among IPv4 routers



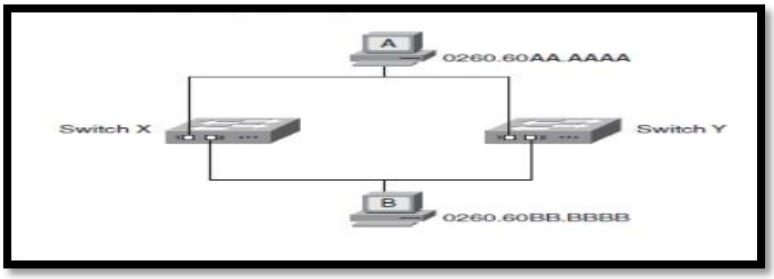
Tunneling





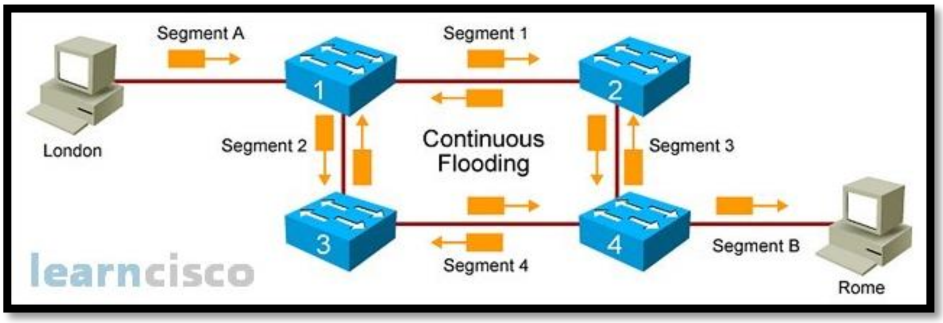
Switching Loops

Redundancy in a network, such as that shown in Figure below, is **desirable** so that communication can still take place if a link or device fails. **For example**, if switch X in this figure stopped functioning, devices A and B could still communicate through switch Y. However, in a switched network, **redundancy can cause problems which is called switching loop**.



When a switching loop is introduced into the network, a destructive **broadcast storm** will develop **within seconds**. ***There are three types of problem occur due to switch looping, these are:***

1. **Broadcast storm** occurs if a broadcast frame is sent on the network.
2. Devices can **receive multiple copies of the same frame**.in redundant topologies.
3. The **MAC address table can change rapidly and contain wrong information**.



To overcome these problems, you must have a way to logically disable part of the redundant network for regular traffic while maintaining redundancy for the case when an error occurs. Spanning Tree Protocol (STP) does just that.

Spanning Tree Protocol(STP)

Spanning Tree Protocol (STP) was developed to prevent the broadcast storms caused by switching loops. Spanning Tree Protocol (STP) is a Layer 2 protocol that runs on **bridges and switches**. The main purpose of STP is to ensure that you do not create loops when you have redundant paths in your network.

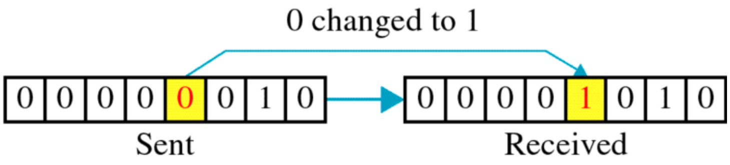
Types of Errors (Detection and Correction)

Networks must be able to transfer data from one device to another with complete accuracy. **Data can be corrupted during transmission**. For reliable communication, **errors must be detected and corrected**. Error detection and correction are implemented either at the **data link layer** or the **transport layer** of the OSI model.

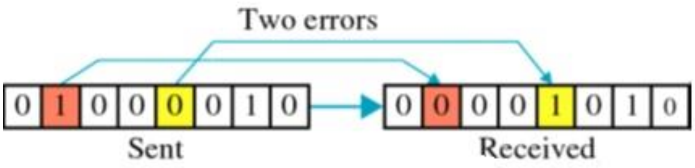
Type Of Errors

There are three types of error, these are:

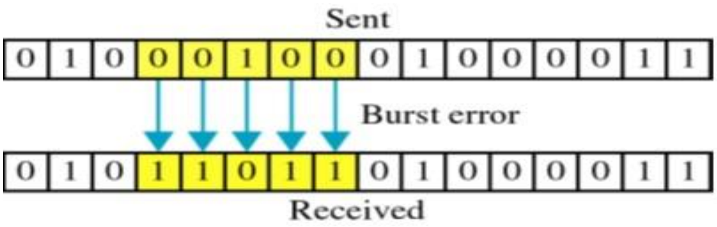
1. **Single Bit Error**



2. **Multiple Error**



3. **Burst Error:** means that two or more bits in the data unit have changed from 1 to 0 or from 0 to 1. **Burst errors do not necessarily mean that the errors occur in consecutive bits**, the length of the burst is measured from the first corrupted bit to the last corrupted bit. Some bits in between may not have been corrupted.



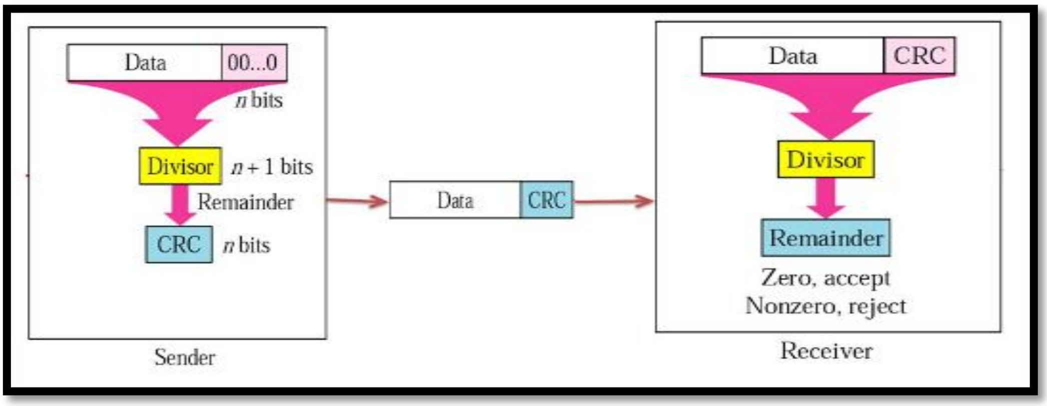
Error Detection

Error detection means to decide whether the received data is correct or not without having a copy of the original message. **Error detection uses the concept of redundancy**, which means **adding extra bits for detecting errors at the destination**. There are four types of redundancy checks are used in data communications, these are:

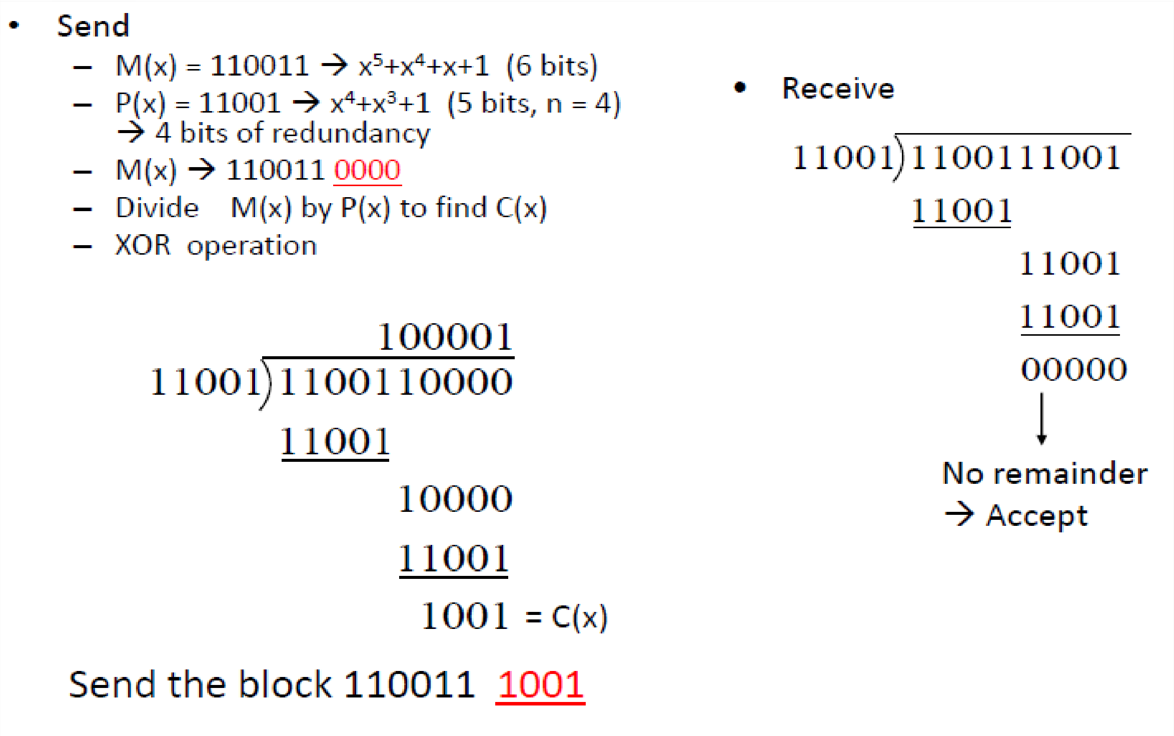
1. **Vertical Redundancy Check (VRC).**
2. **Longitudinal Redundancy Check (LRC).**
3. **Cyclic Redundancy Check (CRC)**
4. **Checksum**

Cyclical Redundancy Check (CRC)

The ***most powerful redundancy technique***, unlike the VRC and LRC, CRC is based on binary division.



Example 1:



- Send
  - M(x) = 110011 → x^5+x^4+x+1 (6 bits)
  - P(x) = 11001 → x^4+x^3+1 (5 bits, n = 4) → 4 bits of redundancy
  - M(x) → 110011 0000
  - Divide M(x) by P(x) to find C(x)
  - XOR operation

```
              100001
       11001)1100110000
             11001
                 10000
                 11001
                  1001 = C(x)
```

Send the block 110011 1001

- Receive

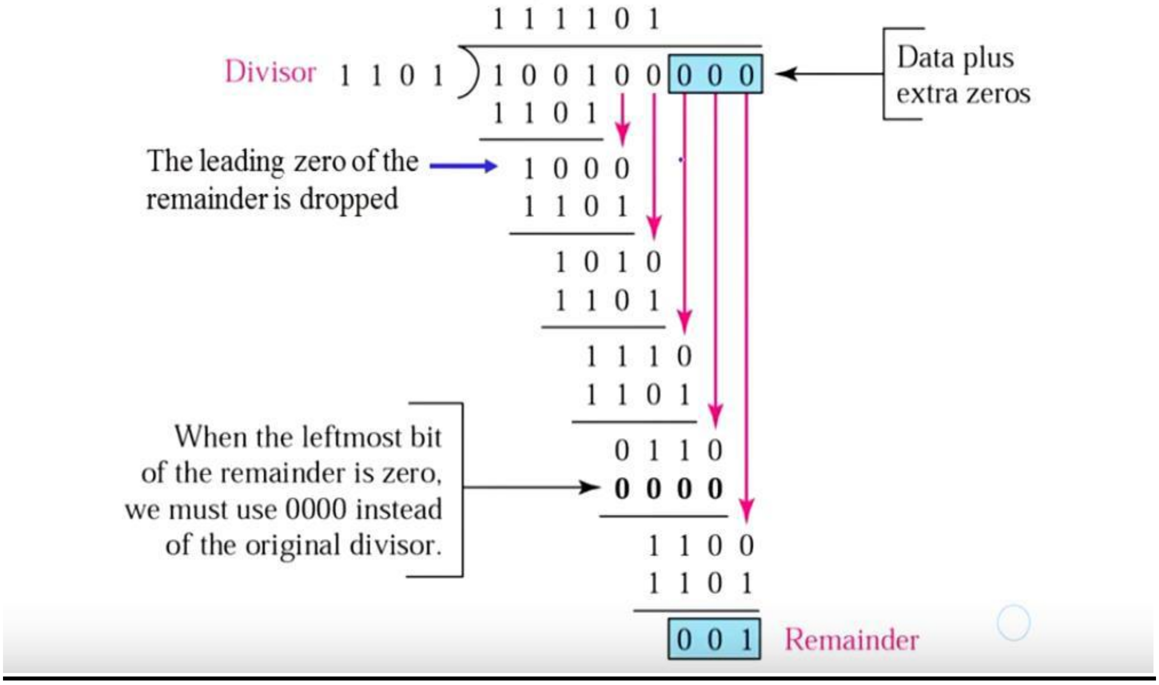
```
11001)1100111001
        11001
            11001
            11001
            00000
              ↓
```

No remainder → Accept

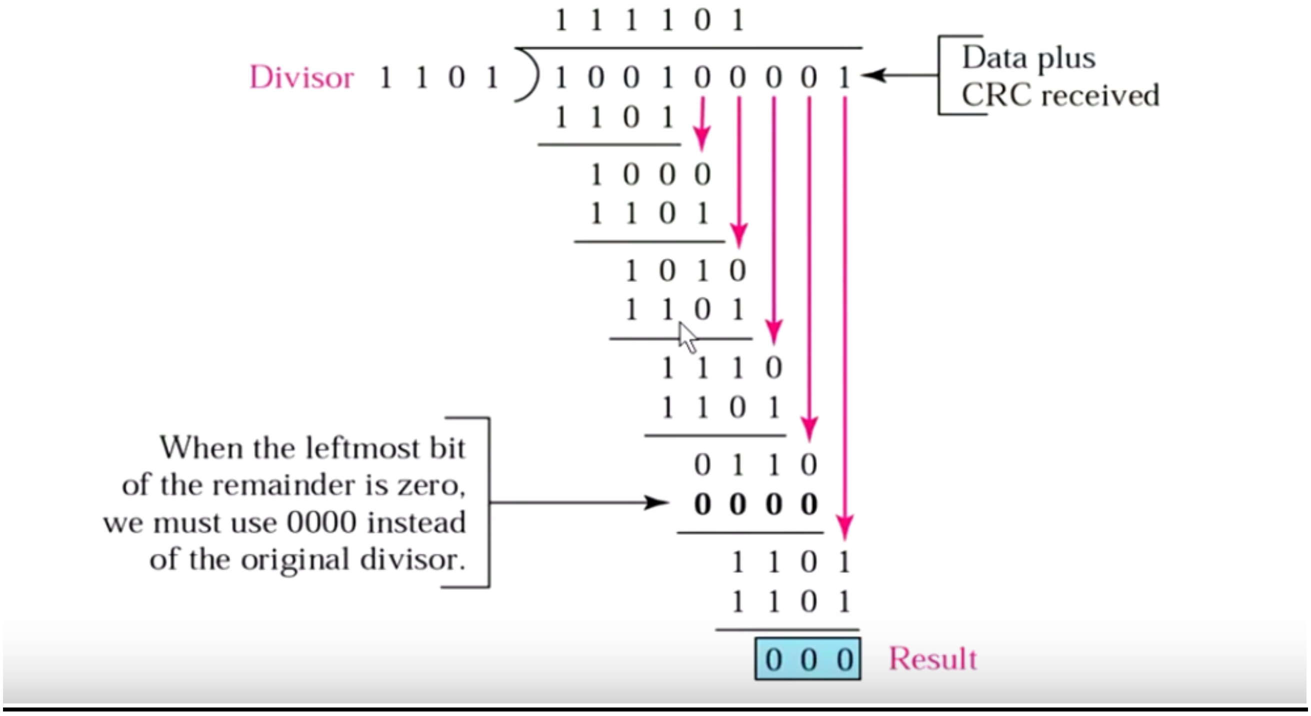
Example 2: **Define CRC and use it to determine the message to be sent T(x):**

Given:
G(x)= 1101
M(x)= 1001

At the sender



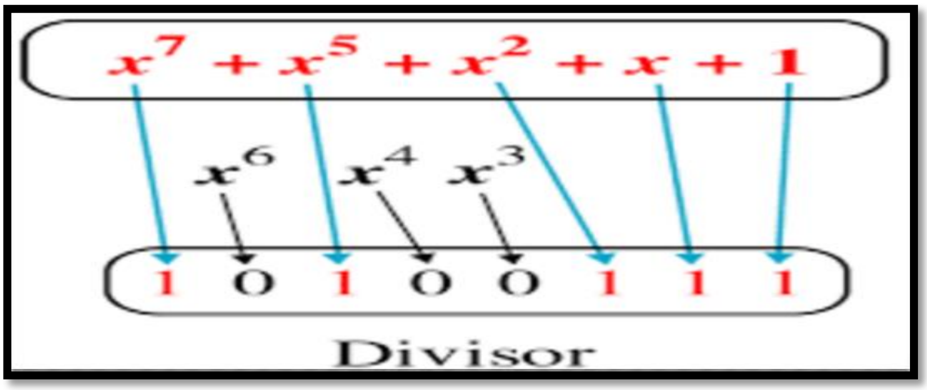
At the receiver



Polynomial Calculation

To calculate the devisor we can use the polynomial as shown below:

- The polynomial **should not be devisable by X**.
- The polynomial **should be devisable by X+1**.

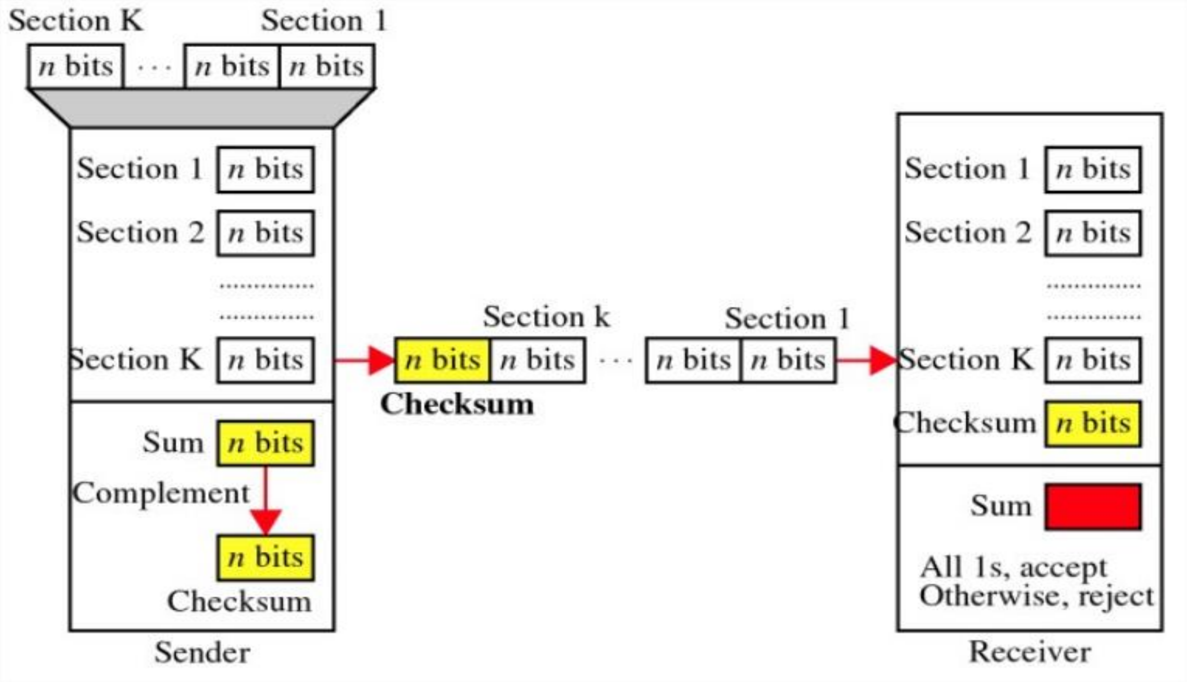


Checksum Technique

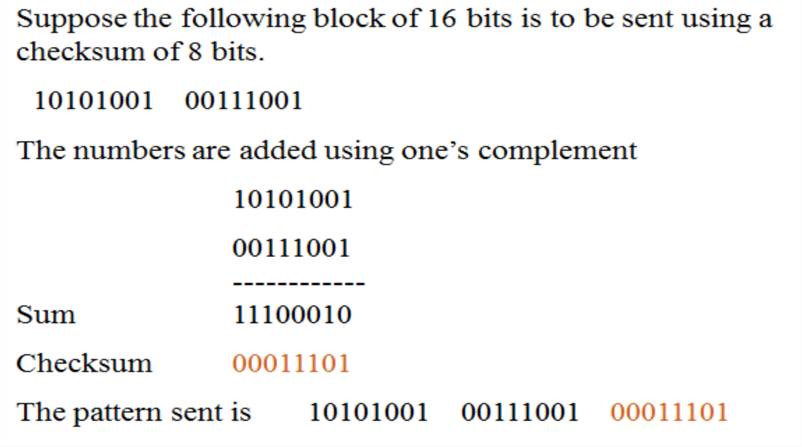
The error detection method used by the ***higher layer protocols***. Like other methods, it depends on the concept of redundancy.

- **At the Sender Side**
  1. The unit is **divided** in to **k** sections, each of **n bits**.
  2. All sections are added using **one's complement** to get the sum in such a way that the total is also **n bits** long.
  3. The **sum is then complemented** and **becomes the checksum**.
  4. The **checksum** is sent with the data.

- **At the receiver Side**
  1. The received data is divided in to **k** sections, each of **n bits**.
  2. All sections are **added using one's complement** to get the sum in such a way that the total is also **n bits** long.
  3. The **sum is then complemented**.
  4. If the **result is zero**, the **data are accepted**, otherwise they are **rejected**.



Example:

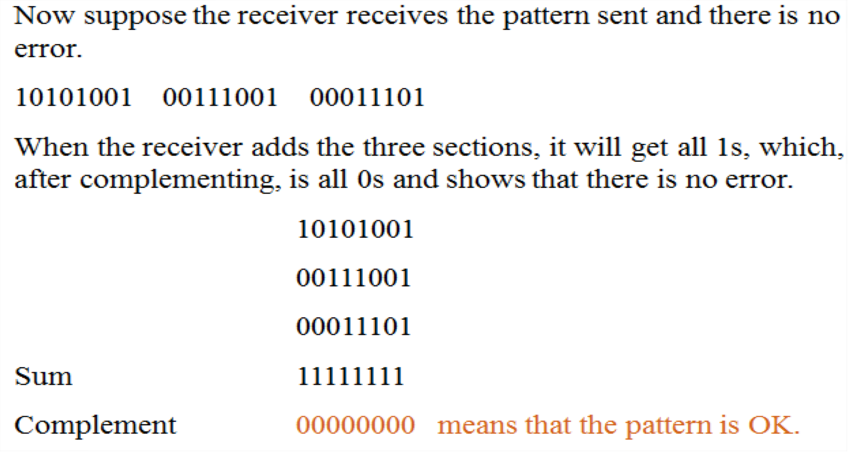


Suppose the following block of 16 bits is to be sent using a checksum of 8 bits.

10101001 00111001

The numbers are added using one's complement

```
                 10101001
                 00111001
                 ------------
Sum              11100010
Checksum         00011101
The pattern sent is      10101001    00111001    00011101
```

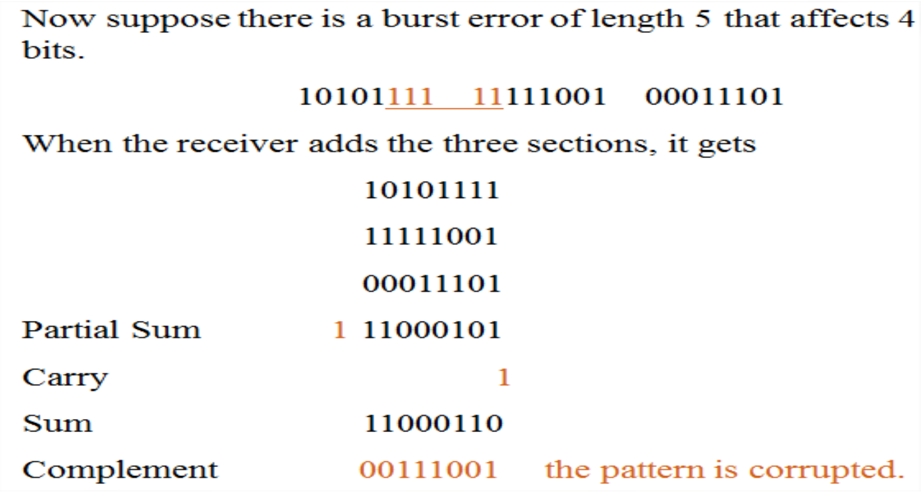


Now suppose the receiver receives the pattern sent and there is no error.

10101001 00111001 00011101

When the receiver adds the three sections, it will get all 1s, which, after complementing, is all 0s and shows that there is no error.

```
                 10101001
                 00111001
                 00011101
Sum              11111111
Complement       00000000   means that the pattern is OK.
```



Now suppose there is a burst error of length 5 that affects 4 bits.

10101111 11111001 00011101

When the receiver adds the three sections, it gets

	10101111	
	11111001	
	00011101	
Partial Sum	1 11000101	
Carry	1	
Sum	11000110	
Complement	00111001	the pattern is corrupted.

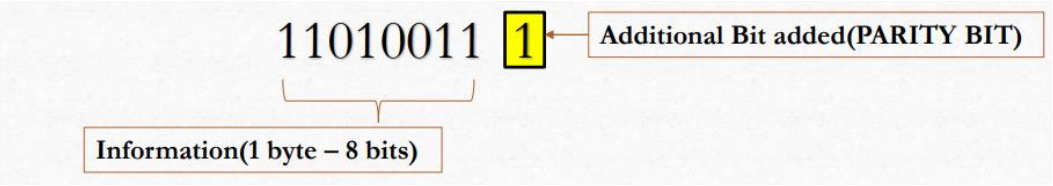
Parity Check Method

1. The Simplest method Available - it's a linear, systematic block code
2. Two Methods of Parity Check are there:

- Single Parity Check(VRC)
- Two Dimensional Parity Check(LRC)

Single Parity Check (VRC)

1. In Single parity check, a parity bit is added to every data unit so that the total number of 1s is even or odd.


11010011
1
Additional Bit added(PARITY BIT)
Information(1 byte – 8 bits)

2. There are **two ways** to generate a Single parity bit: One is called **Even parity (total number of 1's transmitted must be even)** and the other is **Odd parity (**total number of 1's transmitted must be odd)

How Is The Even Parity Bit Generated?

Total number of '1's should be EVEN.

If the byte that we want to transmit is (10101101) for example
Then
Step 1: count the number of 1's in the byte (answer is 5)
Step 2: compute the parity value
Since the total number of 1's is 5 then the even parity bit is 1 and the value will be (101011011)
If the number of 1's is already even the parity bit will be 0.

How Is The ODD Parity Bit Generated?

Total number of ‘1’s should be odd

. If the byte that we want to transmit is (10101100) for example

Then

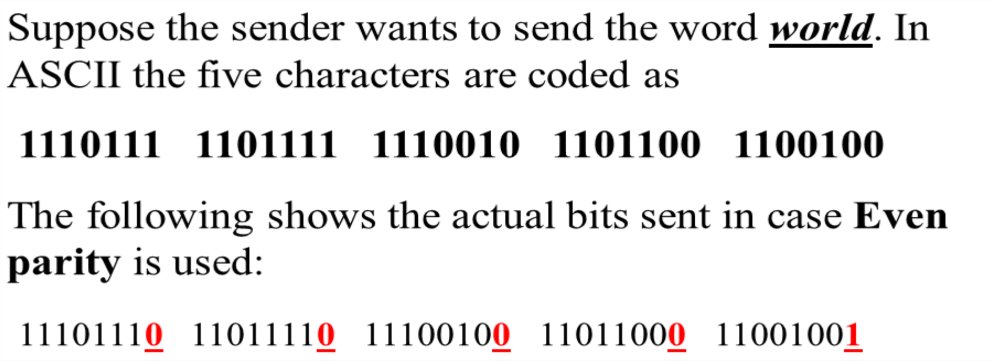
Step 1: count the number of 1’s in the byte (answer is 4)

Step 2: compute the parity value

Since the total number of 1’s is 4 then the odd parity bit is 1 and the value will be (101011001)

If the number of 1’s is already odd the parity bit will be 0.

Examlp1:

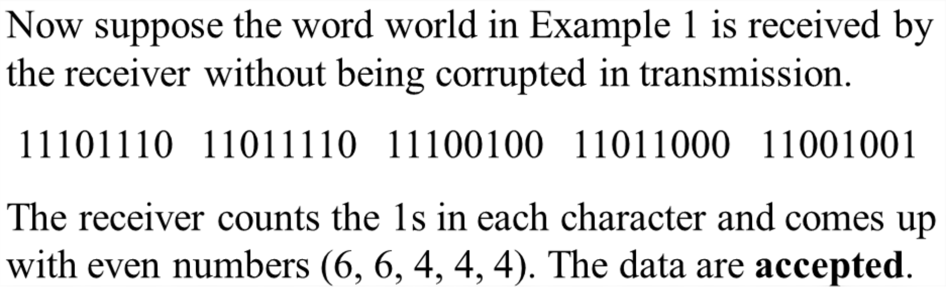


Suppose the sender wants to send the word ***world***. In ASCII the five characters are coded as

1110111 1101111 1110010 1101100 1100100

The following shows the actual bits sent in case **Even parity** is used:

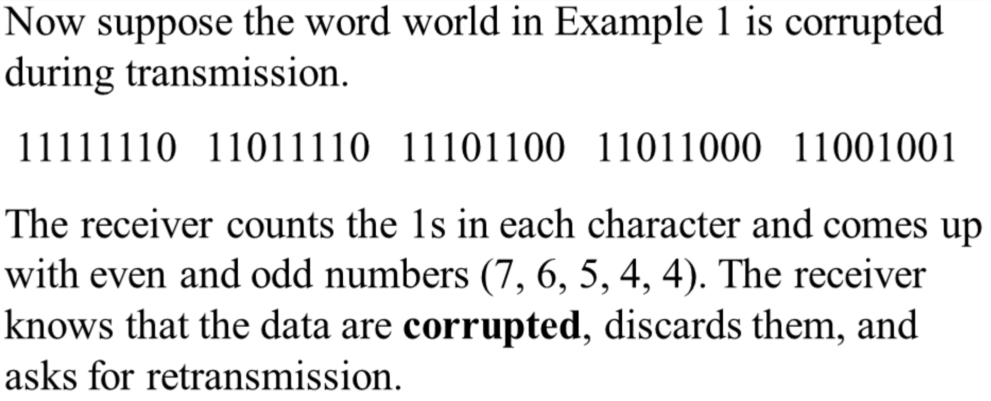
1110111**0** 1101111**0** 1110010**0** 1101100**0** 1100100**1**



Now suppose the word world in Example 1 is received by the receiver without being corrupted in transmission.

11101110 11011110 11100100 11011000 11001001

The receiver counts the 1s in each character and comes up with even numbers (6, 6, 4, 4, 4). The data are **accepted**.



Now suppose the word world in Example 1 is corrupted during transmission.

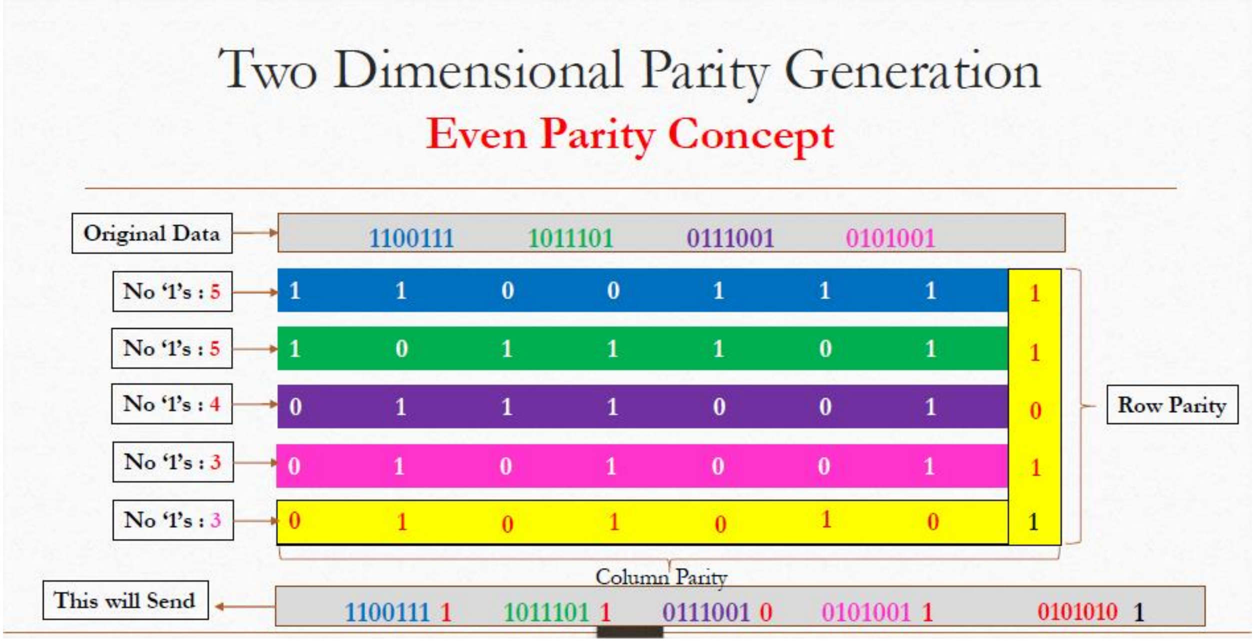
11111110 11011110 11101100 11011000 11001001

The receiver counts the 1s in each character and comes up with even and odd numbers (7, 6, 5, 4, 4). The receiver knows that the data are **corrupted**, discards them, and asks for retransmission.

Two Dimensional Parity Check (LRC)

In two-dimensional parity check, a block of bits is divided into rows and a redundant row of bits is added to the whole block.

Even Parity Concept



Two Dimensional Parity Generation

Even Parity Concept

Original Data → 1100111 1011101 0111001 0101001

No '1's : 5	1	1	0	0	1	1	1	1
No '1's : 5	1	0	1	1	1	0	1	1
No '1's : 4	0	1	1	1	0	0	1	0
No '1's : 3	0	1	0	1	0	0	1	1
No '1's : 3	0	1	0	1	0	1	0	1

Row Parity

Column Parity

This will Send ← 1100111 1 1011101 1 0111001 0 0101001 1 0101010 1