__/__/__

Module-2: Data Link Layer

* Error Control

* Types of errors:

- **Single bit error:**

In single bit error, only one bit of given data unit is changed from 1 to 0 or 0 to 1

eg:

1	0	0	1	0

Sent

1	1	0	1	0

Received

0 changed to 1

- **Burst error:**

Two or more bits in the data unit have changed from 0 to 1 or 1 to 0.

The bits need not be consecutive.

The length of burst is taken from first corrupted bit to the last corrupted bit.

eg:

~~bit~~ burst length = 6

0	1	0	0	1	1	0

Sent

corrupted bits

0	0	1	0	0	1	1

Received

* Redundancy:

- The central concept in error detection and correction is redundancy.
- To detect / Correct errors, we need to add few extra bits.
- These bits are added by sender and removed by receiver

__/__/__

* Error detection v/s correction.

- Correction of errors is more difficult than identifying errors.
- In error detection, we only see if an error has occurred or no.
- In error correction, we need to know the total no. of bits that are corrupted and their location in ~~measure~~ message.
- Two main types of error correction:
  - Forward Error Correction : Receiver tries to guess message using redundant bits.
  - Retransmission:

    Receiver detects an error and asks sender to resend the message.

* Coding:

- Redundancy can also be achieved through various coding schemes.
- The sender adds redundant bits ~~to~~ and creates a new relationship between redundant bits and original data bits.
- The receiver checks the relationship b/w them to detect errors.
- Coding can also be divided into 2 parts:
  - Block Coding:

In block coding, we divide our message into blocks, each of k-bits also called as dataword. We add 'r' redundant bits to it to make the length $n = k+r$. This n-bit blocks are called codewords.

10101
11110
00011

__/__/__

- Hamming code/ Distance (Take XOR)
- Central concept for error control is hamming distance.
- Hamming distance between 2 words (of same size) is the number of differences b/w corresponding bits.
- Distance between two words x, y can be given as d(x,y)
- Minimum hamming distance is smallest hamming distance between all possible pairs of set

- Hamming Code for Error detection
  $d_{min} = s+1$ ($s \equiv$ errors)
- Min. Hamming distance for Error correction
  $d_{min} = 2t+1$ ($t \equiv$ errors)
  For error correction, d_{min} must be odd.
- Linear Block Code = Hamming Code
- Single Parity Check Code
- a k-bit word is changed to a n-bit word, where $n = k+1$
- The extra bit added is called parity bit, is selected to make total no. of 1's odd / even.
- Here the min. Hamming distance will be $d_{min} = 2$ which means it is a single-bit-error detecting code; it cannot correct any error.
  [Make data word have even no. of 1's]

eg: Sender sends dataword 1011. Codeword formed is 10111. We examine 5 cases:

(1) No error occurs: received codeword is 10111. Syndrome is 0. Dataword created is 1011

(2) One single-bit error changes a_1: received codeword is 10011. Syndrome is 1. No dataword created.

See Hamming C(7,4) related sums.

_/_/_

) One single-bit error changes r0: received codeword is 10110. Syndrome is 1. No ~~codeword~~ dataword formed.

4) An error changes r0 and second error changes a3: Received codeword is 00110. Syndrome is 0. Produced dataword is 0011. Parity checkers cannot count even number of errors. The errors cancel each other out and give syndrome value 0.

5) 3 bits a3, a2, a1 are changed by errors: Received code word is 01011. Syndrome is 1. No dataword formed.

* Cyclic Codes

- They are special linear block codes.
- If a codeword is shifted cyclically, result is another codeword.

* Cyclic Redundancy Check

- Subset of cyclic codes used for user detection.
- Used in LANS, WANs.

eg: Division in CRC encoder

Divisor: 1011, Dividend: 1001

```
             1010  ---> Quotient
       1011 | 1001000
              1011
              ----
              0100
              0000
              ----
               1000
               1011
               ----
               0110
               0000
               ----
                110
               0110 ---> Remainder
```

codeword: 1001110

Dividend + Remainder.

__/__/__

CRC Division with polynomial.

Dividend: $x^6 + x^3 \equiv 1001000$

Divisor: $x^3 + x + 1 \equiv 1011$

```
              1010111
        ______________
  1011 | 1001000000
         1011
         ______
         00100
          0000
          _____
            1000
            1011
            ____
            0110
            0000
            ____
             1100
             1011
             ____
              1110
              1011
              ____
               1010
               1011
               ____
               0001
```

Data word : Dividend + Remainder

= 1001000001

Standard Polynomials

CRC-8: $x^8 + x^2 + x + 1$ ⟶ ATM Header

100000111

CRC-10: $x^{10} + x^9 + x^5 + x^4 + x^2 + 1$ ⟶ ATM AAL

11000110101

CRC-16 ⟶ HDLC

CRC-32 ⟶ LANS

__/__/___

Checksum

Internet checksum: Traditionally Internet has used a 16-bit check sum

Sender's side:

Message divided into 16-bit words.

The value of checksum word is initially set to zero.

The words including checksum are added using one's compliment

The sum is complemented and becomes checksum.

Check sum is sent in with data.

Receiver's Side:

Message and checksum are received.

Message is divided into 16-bit words.

All words are added using one's complement addition.

Sum is complemented and becomes checksum.

If value of checksum is 0, message accepted ; else rejected.

Data link layer

Data link ~~&~~ layer is divided into two sub-layers:

- Logical link control (LLC sublayer) → flow and error control
- Media Access Control (MAC sublayer) → multiple access resolution.

Responsible for moving data packets from one interface to another within the shared channel.

Controls frame synchronisation, error checking and flow control.

Framing:

Datalink layer needs to pack bits into framing so that they can be distinguished from one another.

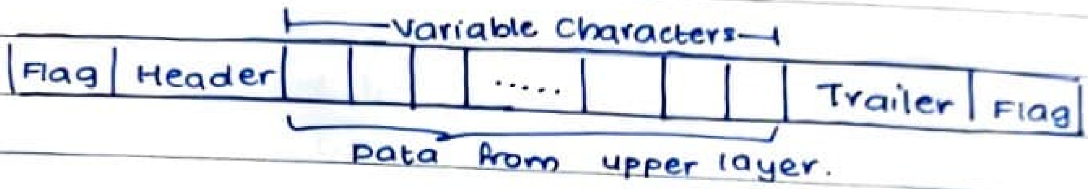
(1) Fix sized Framing: Size itself acts as a delimiter

(2) Variable sized Framing: Need to define end of one frame and start of next

- Character (Byte) oriented Faming
- Bit oriented Farming.

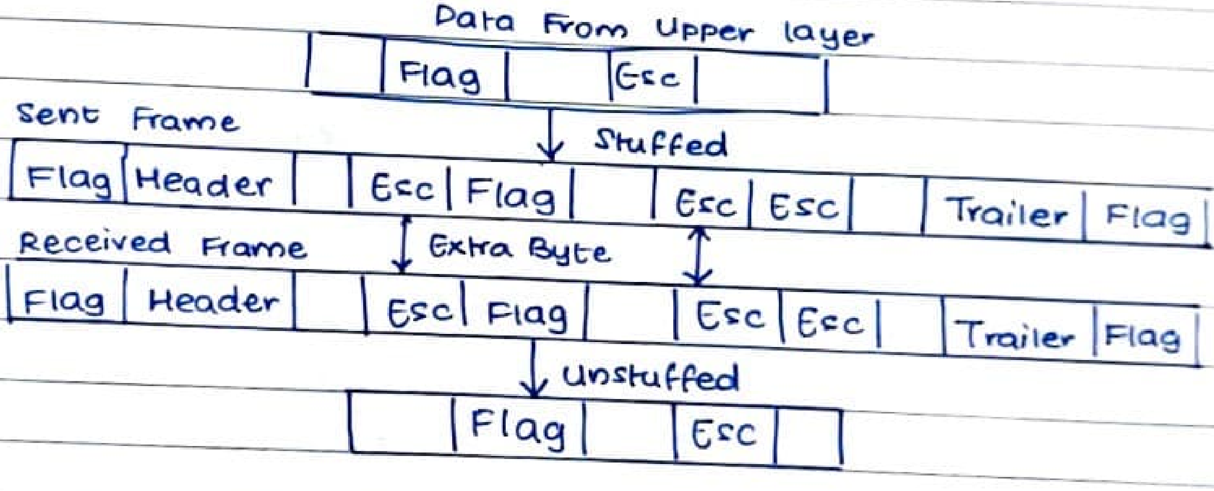
* Character Oriented Framing.

- In Character (Byte) oriented Framing, the data to be Carried is 8-bit.
- Header and Trailer are also multiples of 8 bit.
- To separate one from the next, flag is added to the start and end of frame.



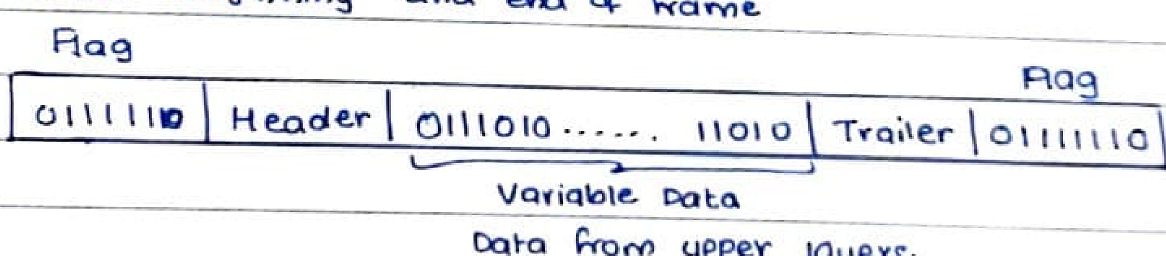
* Byte Stuffing and unstuffing

Byte stuffing means adding one extra byte before the flag or escape characters.



* Bit oriented Framing:

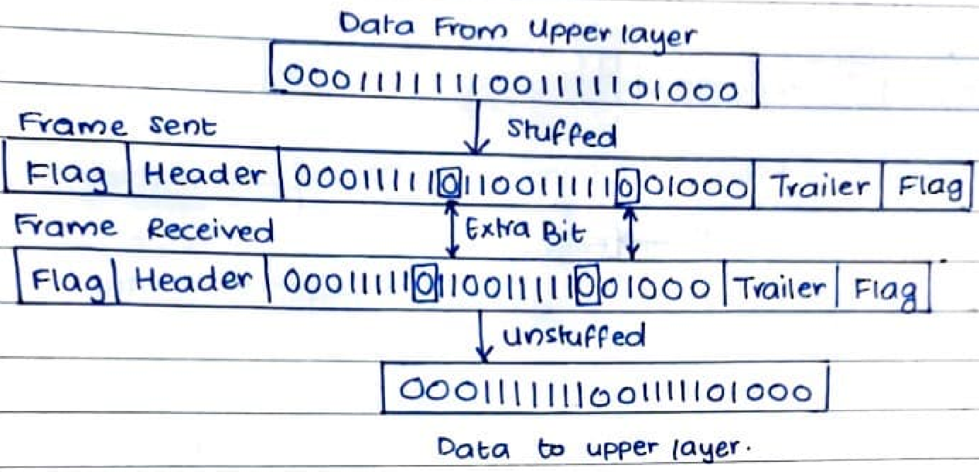
- A special 8-bit pattern flag 01111110 is used as delimiter to define beginning and end of frame



_/_/_

★ • Bit Stuffing and unstuffing

- In a bit stuffing if a zero and five consecutive 1's are occured, add an extra zero.
- The extra bit is too stuffed but it is later removed by receiver.



★ Flow Control

- One of the major responsibilities of the data link layer is the flow and error control. Collectively these functions are called as data link controls.
- Flow control refers to a set of procedures that restrict the amount of data that the sender can send.
- Flow of data must not overwhelm the receiving end.
- Receiving data has limited speed at which it can process incoming data and limited storage for storing incoming data.

★ Protocols

- Noiseless Channels:

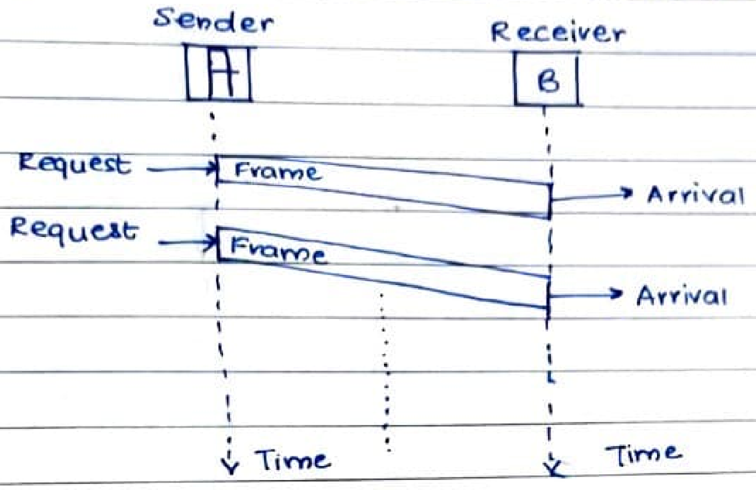
For an ideal channel where ~~there~~ there are no corrupted frames, we have 2 protocols:

- Simplest Protocol
- Stop - Wait Protocol.

(1) Simplest Protocol

- Cannot be used in real life - No flow and error control
- Receiver will never be overwhelmed with incoming frames.
- Receiver can handle frames in negligible time.
- No. of events at sender site = No. of events at receiver site.

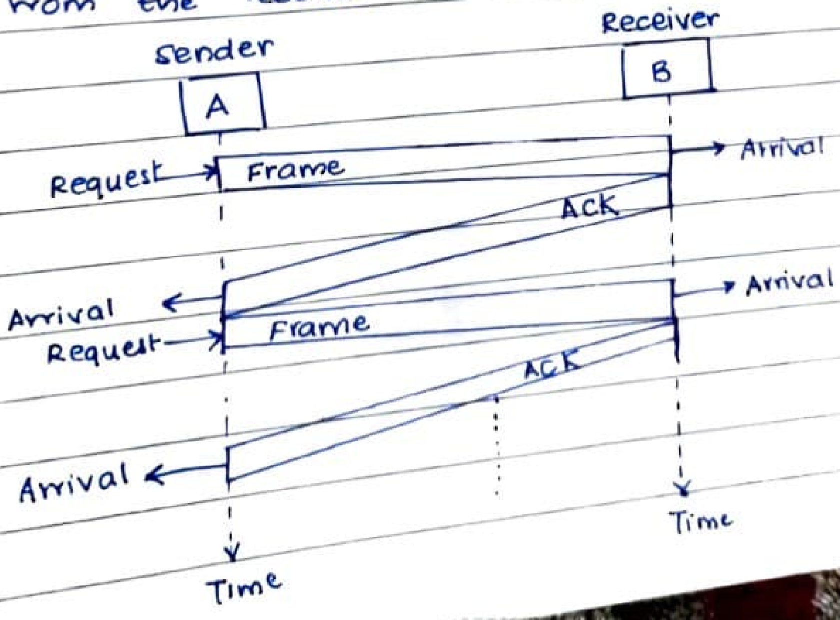
eg:



(2) Stop-wait Protocol

- If frames arrive faster, they are processed. Frames must be kept in store until their use.
- To prevent the receiver's side to get overwhelmed, the receiver should notify the sender to slow down. (ACF)
- The sender sends in one frame, waits to receive confirmation from the receiver and then sends next frame.

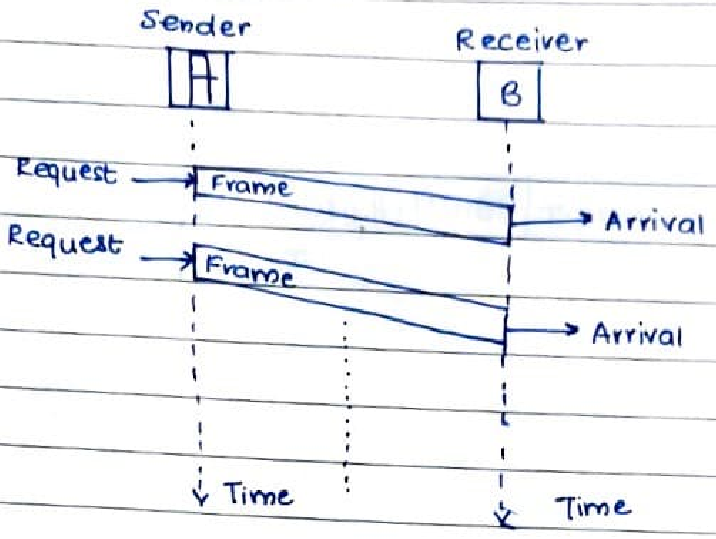
eg:



__/__/__

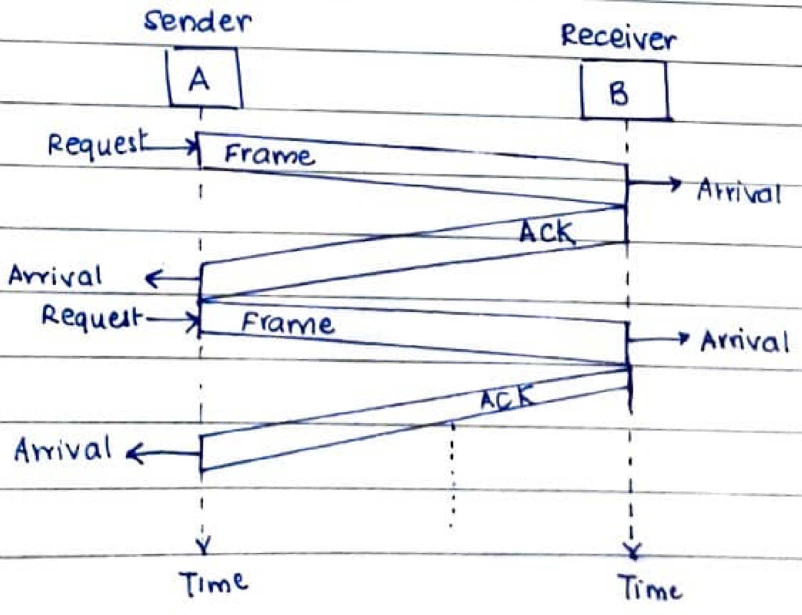
(1) Simplest Protocol

- Cannot be used in real life - No flow and error control
- Receiver will never be overwhelmed with incoming frames.
- Receiver can handle frames in negligible time.
- No. of events at sender site = No. of events at receiver site.



(2) Stop - Wait Protocol

- If frames arrive faster, they are processed. Frames must be kept in store until their use.
- To prevent the receiver's side to get overwhelmed, the receiver should notify the sender to slow down. (ACK)
- The sender sends in one frame, waits to receive confirmation from the receiver and then sends next frame.



__/__/__

- Noisy channel

• Although ~~Soft~~ stop and wait protocol shows us the idea of how to add a predecessor to flow control, noiseless channels are non-existent. Noisy channels have error control and flow control.

(1) Stop and wait Automatic Repeat Request
(2) Go-back-N Automatic Repeat Request
(3) Selective Repeat Automatic Repeat Request.

(1) Stop and wait ARQ

- Error control in stop and wait ARQ is done by keeping copy of sent frame and retransmitting the frame when time expires.
- We use sequence numbers to number frames. Numbers are based on modulo-2 arithmetic.

• Check eg:

- Efficiency: Inefficient if channel is thick and long.

Bandwidth-delay Product = Bandwidth x Round trip delay

$$\text{Link utilization} = \frac{\text{No. of bits sent by system}}{\text{Bandwidth delay}}$$

- Pipelining: No pipelining.

Pipelining improves efficiency

We need to wait for a frame to reach and get acknowledged before the next frame is sent

(2) Go-Back-N ARQ

- One can send several frames before receiving acknowledgement.
- System keeps copy of frames till the acknowledgement does not arrive
- Sequence numbers are in range of 0 to 2^m-1.

__/__/____

- Sliding Window : Defines range of sequence numbers. For a sender window, it can slide one or more slots at a time.
For a receiver's window, it can slide only one slot at a time, when correct frame arrives.
- see eg:

(3) Selective Repeat ARQ
- Inefficient for a noisy link.
- Size of send window and receive window are same => 2^{m-1}
- Frame has higher probability of damage, means resending frames. This uses up bandwidth and slows down transmission.
- When there is only one frame, the damaged frame is resent. making the receiver process complex.

see eg:

* Multiple Access
- The Media Access control sublayer of data link layer is responsible for multiple access resolution.
- Multiple access protocol has 3 protocols: Random Access, ~~Conc~~ Controlled Access and Channelization Protocols

* Random Access Protocols:
- At each instance, a station has data to send uses a procedure defined by a protocol to make a decision whether to send or not.
- Each station has right to medium without being controlled
- It may lead to access conflict and damaging of frames
- Also called contention based access.

__/__/__

(1) ALOHA Network

- When a station sends data, another station may also do the same. The data from two stations may collide and become garbled.

• Pure ALOHA :

- The idea is that a station sends in a frame whenever it has a frame to send.
- Since, there is only channel to share, there is the possiblity that frames might collide from different stations.
- Even if there is one collision, both frames will get destroyed.
- Resend the frames that are destroyed
- If there is a timeout, resend the frame.
- When time out passes, each station waits for some random amount of time before it resends the frame.
- Randomness avoids more collisions.
  - Back off Time $(T_b) = \dfrac{\text{Distance b/w stations}}{\text{Speed of signal}}$
  - Aloha throughput follows Poisson Distribution:

    Throughput for Pure Aloha ~~SwG~~ $(S) = G \times e^{-2G}$

    $G \equiv$ average number of frames requested

    Maximum Throughput $(S_{max}) = 8.14$ at $G = 1/2$

• Slotted ALOHA:

- Slotted ALOHA was invented to improve efficiency of Pure ALOHA.
- Here, stations can send frames only at their beginning of time slot.
- Throughput $(S) = G \times e^{-G}$

  Max throughpout = ~~0.384~~ 0.368 at $G = 1$

/ /

CSMA - Carrier Sense Multiple Access

Reduces Collision but cannot avoid it completely.

Requires each station check state of medium before sending

Possibility of collision still exists because of propogation ~~time~~ delay when station sends a frame, it still takes time for the first bit to reach station and sense it.

CSMA/CD : Carrier Sense Multiple Access with Collision Detection

- Station monitors channel when sending a frame to see if it transmission was successful.
- Need a restriction on frame size, but must be able to detect collision.
- Augments algorithm to handle collision.
- Before sending last bit, must detect a collision because once entire frame is sent, station does not keep copy of frame.
- Frame Transmission $T_{fr} = 2 * TP$

  TP = Propogation Time.

CSMA/CA : Carrier Sense Multiple Access with Collision Avoidance

- Used in a network where collision cannot be detected.
- Collision can be avoided using three strategies:
  - Inter frame space
  - Contention window
  - acknowledgement

__/__/__

* Controlled Access
- Stations consult one another to find which station has right to send.
- ▲ Three common methods:
  • Reservation:
  - Station needs to make a reservation before sending data.
  - Time is divided in intervals
  - If N stations, N reservation mini-slots.

  • Polling Method
  - Polling works when one station is decided to be primary station and other stations are secondary stations.
  - It is upto primary station to determine which station can use the channel.

  • Token Passing
  - stations are in a logical ring
  - For each station there is a predecessor and successor.
  - Predecessor passes right to access to station to the current station.
  - Special packet called token circulates throughout.

* Channelisation
- similar to multiplexing.
- Three schemes: → Allows multiple users to utilise same band width.
  (1) Frequency Division Multiple Access. : Data link layer access method
  (2) Time Division Multiple Access
  (3) Code Division Multiple Access.
  - each channel separated by code.
  - Each station has a unique chip sequence
    eg: [+1 +1 +1 +1] : orthogonal vectors
    Silent is always [0 0 0 0]