# Ethernet MAC frame (802.3 MAC frame)

| | | |
|---|---|---|
| Header | Data or Payload | Trailer or CRC |

| | | |
|---|---|---|
| Destination Address (6 byte) | Source Address (6 byte) | Length or Type (2 byte) |

Header Length = 14 byte

Trailer Length = 4 byte

46 byte $\leq$ Data or Payload Length $\leq$ 1500 byte

Therefore

64 byte $\leq$ Frame Length $\leq$ 1518 byte

# Type Of Casting

From the Destination Address ( 6 byte ) we come to know about type of casting.

**Unicasting** ( one to one ) Least significant bit of first byte is 0

**Multicasting** ( one to many ) Least significant bit of first byte is 1

Information is send by one sender and is recieved by particular set of recievers connected to network

**Broadcasting** ( one to all ) All 6 bytes of Destination Address are 1 i.e 48 1's in Destination Address.

Information is send by one sender and is recieved by all recievers connected to network

Q. The Destination Address is given in Hex notation as following. Give the type of casting

1st byte

① 4A : 30 : 10 : 21 : 10 : 1A — Unicasting

01001010 — Least Significant bit

1st byte

② 47 : 20 : 1B : 2E : 08 : EE — Multicasting

01000111

③ FF : FF : FF : FF : FF : FF — Broadcasting

# Ethernet

## CSMA / CD

CS ( Channel Sense ) When any station wants to trans its data to other station (s), it senses the channel.

MA ( Multiple Access ) Suppose that the channel is free and two or more than 2 stations sensed the channel, they get it free so they transfer their packets. This is called Multiple Access, Since multiple stations are accessing the channel simentaneously

CD ( Collision Detection ) At the situation of multiple access of channel collision occur while transferring data packet. The JAM signal is sent by the station nearby to collison in both the directions that collison has occured and stop sending packet

# Backoff Algorithm

This Algorithm is used for giving waiting time to stations that are involved in collision.

$$\text{Waiting Time} = \underbrace{K}_{0 \text{ to } 2^{n}-1} \times 51.2\ \mu\text{sec}$$

$n$ — Collision number.

If a packet is involved in collision for 1st time then $n = 1$.

If same packet is involved in collision for 2nd time then $n = 2$.

The drawback of Backoff Algorithm is Capture Effect

## Capture Effect

Suppose two stations A, B are there. Suppose both of them send 1st packet after sencing the channel and getting it free. Then collision occured, detected using Backoff Algo.

Since packet $1_A$, packet $1_B$ is involved 1st time in collision so values of K are 0, 1 for both A, B

| A | B | |
|---|---|---|
| 0 | 0 | ✗ |
| 0 | 1 | B will wait |
| 1 | 0 | A will wait |
| 1 | 1 | ✗ |

Let me suppose that B will wait so A will send its 1st packet.

After this both will sense channel and get free. So A will send its 2nd packet, B will send its 1st packet. Again collision occur but this time it is between $2_A$, $1_B$

For packet $1_B$ it is involved in collision 2nd time so k are 0, 1, 2, 3

| A | B | |
|---|---|---|
| 0 | 0 | X |
| 0 | 1 | |
| 0 | 2 | |
| 0 | 3 | |
| 1 | 0 | |
| 1 | 1 | X |
| 1 | 2 | |
| 1 | 3 | |

Possibility for A to send = $\frac{5}{8}$

Possibility for B to send = $\frac{1}{8}$

For next time the possibility for B to send decreases more. Hence we can say that whole of the channel is captured by A, i.e the station which gets the waiting time Zero for first time captures the channel

- Error
  - Bit Error
    - Single bit Error: Only 1 bit is changed
    - Multiple bit Error or Burst Error: 2 or more bits have changed
  - Packet Error: Sliding Window protocol is used for Packet Error Control
    - Selective Repeat ARQ
    - Go-Back-N ARQ

Burst error length or length of Burst error

The length from the 1st corrupted bit to the last corrupted bit

Sent 0100010001000011

Recieved 0100111010110011

Burst error length = 8 bits not 4 bits

## Hamming Distance

Hamming Distance b/w two codewords is the number of differences b/w corresponding bits

e.g

$d(000, 011) = 2$

## Minimum Hamming Distance

Minimum Hamming Distance is the smallest Hamming distance b/w all possible pairs in set of codewords

e.g

Codewords

00000
01011
10101
11110

$d(00000, 01011) = 3$

$d(00000, 10101) = 3$

$d(00000, 11110) = 4$

$d(01011, 10101) = 4$

$d(01011, 11110) = 3$

$d(10101, 11110) = 3$

$\boxed{d_{min} = 3}$

To guarantee detection of upto s errors

$d_{min} = s + 1$

To guarantee correction of upto s errors

$d_{min} = 2s + 1$

Imp

Block Coding

on

k bits are there in datawords

r redundant bits

k+r bits are there in codewords

$2^k$ datawords

$2^{k+r}$ codewords

$2^{k+r} - 2^k$ are not used for message transfer or are illegal or are invalid.

**Q** A 4B / 5B block coding is used

databits codebits

$2^4$ distinct datawords

$2^5$ distinct codewords

$2^5 - 2^4$ codewords are illegal

**Q** Let

$k = 2$

$\therefore$ Datawords are

00
01
10
11

$r = 1$

Codewords ( using even parity )

000
011
101
110

$d(000, 011) = 2$
$d(000, 101) = 2$
$d(000, 110) = 2$
$d(011, 101) = 2$
$d(011, 110) = 2$
$d(101, 110) = 2$

$d_{min} = 2$

For guarantee correction of s errors

$d_{min} = 2s + 1$

$2 = 2s + 1$

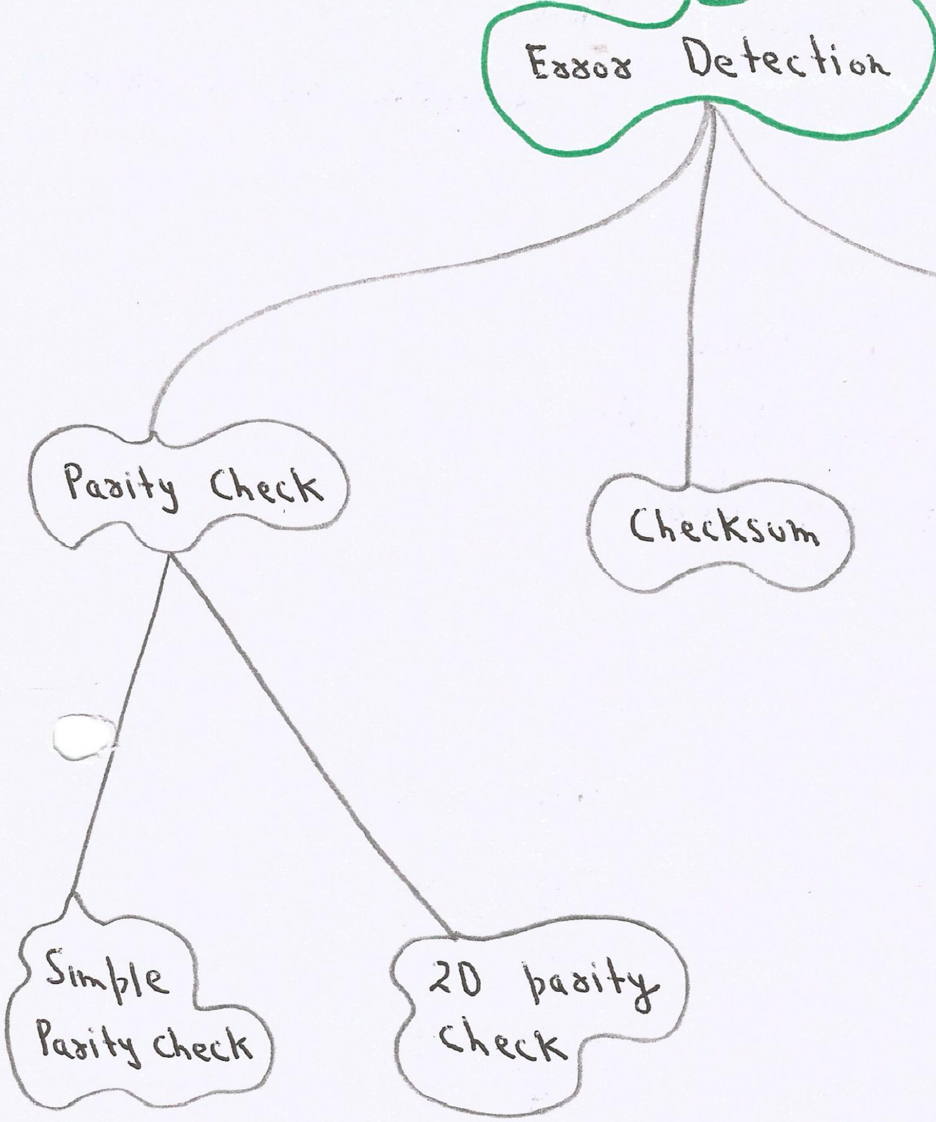
$s = \frac{1}{2}$ $\therefore$ No error can be corrected

For guarantee detection of s errors

$d_{min} = s + 1$

$2 = s + 1$

$s = 1$ $\therefore$ guarantee the detection upto 1 error




## Simple Parity Check

( guarantees the detection of 1 error)

can detect odd number of errors

Data units

1100111 1011101 0111001 0101001

Sent as

11001111 10111011 01110010 01010011

4 extra bits

2 D Parity Check

(guarantees the detection of 3 errors)

Table of 4 में जो number आते हैं उतने error particular जगह पर detect नहीं होंगे

$4 \times 1 = 4$

$\frac{4}{2} = 2$

rows

4 errors detect नहीं होंगे अगर दो error किसी row में होंगे और दो किसी दूसरी row में same position पर होंगे।

बाकी जगह 4 errors भी detect हो जाएंगे

$4 \times 2 = 8$

$\frac{8}{2} = 4$

$\frac{8}{4} = 2$

$\frac{8}{6} = X$

$4 \times 3 = 12$

$\frac{12}{2} = 6$

$\frac{12}{4} = 3$ odd X

$\frac{12}{6} = 2$

$\frac{12}{8} = X$

$\frac{12}{10} = X$

Data Units

1100111 1011101 0111001 0101001

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

Sent as

11001111 10111011 01110010 01010011 010101

12 extra bits (1+1+1+1+8)

# Checksum

16 bit checksum is used in Internet

## Sender Side :-

> Data unit is divided into equal segment of n bits

> The segments are added in such way that resultant number is also n bits long.

> The resultant n bit no is then complemented using 1's Complement. This is known as checksum Then this checksum is inserted at the end of data unit.

## Reciever Side :-

> Codeword is divided into equal segment of n bits

> The segments are added in such way that the resultant number is also n bits long.

> The resultant n bit no is complemented using 1's complement. The checksum which is now formed if contains all Zeros then it means no error is there

Let the data to be send is

7, 11, 12, 6

1. 7, 11, 12, 6

   7 → 0111, 11 → 1011, 12 → 1100, 6 → 0110

2. The segments are added in such a way that resultant number is also n bit long.

```
  1
  1 0 1
  0 1 1 1
  1 0 1 1
  1 1 0 0
  0 1 1 0
  -------
  0 1 0 0
      1 0
  -------
  0 1 1 0
```

or

$7 + 11 + 12 + 6 = 36$

$(36)_{10} = (100100)_2$

```
= 0 1 0 0
      1 0
  -------
  0 1 1 0
```

3. Take complement of using 1's compliment

   1's compliment of 0110 is 1001 = 9

   checksum

7, 11, 12, 6, 9

## Reciever

(2)

$$
\begin{array}{r}
1 \\
1\ 0\ 1 \\
0\,1\,1\,1 \\
1\,0\,1\,1 \\
1\,1\,0\,0 \\
0\,1\,1\,0 \\
1\,0\,0\,1 \\
\hline
1\,1\,0\,1 \\
1\,0 \\
\hline
1\,1\,1\,1
\end{array}
$$

$7 + 11 + 12 + 6 + 9 = 45$

or $(45)_{10} = (101101)_2$

$$
\begin{array}{r}
= 1101 \\
10 \\
\hline
1111
\end{array}
$$

(3)

Complemint of $1111 = 0000$

It means no error is there.

# Performance Of Checksum

1. If data gets corrupted in such a way that the sum of data before corruption and after corruption remains the same then that error will not be detected.

e.g 7, 11, 12, 6, 9 Sum = 45

$\downarrow$ gets corrupted and changed to

7, 9, 14, 4, 11 Sum = 45

This will not be detected.

2. If the data gets corrupted in such a way that the values of several words are incremented but total increment is multiple of $2^n - 1$ then that error will not be detected

$n$ — bits

$\boxed{n = 4}$ in question discussed so

$2^n - 1 = 15$ $\therefore$ increment of 15, 30, 45, ..... will not be detected.

CRC — Checksum

Cyclic Redundancy

<u>Let</u>

Dataword :- 1001

Divisor :- 1011

(even parity)

```
             1010
        ___________
 1011  )  1001000
          1011
          ----
          x0100
           0000
           ----
           x1000
            1011
            ----
            x0110
             0000
             ----
             x110   — CRC
```

length = 4

∴ append 3 zeros

तो remainder में तीन digits आए है उनके तीन Zeros की जगह डाल दो

i.e 1001000

becomes

1001110 — codeword for CRC

This codeword is send by sender

**Reciever**

```
          1010
      ---------
1011 ) 1001110
       1011
       -------
       X0101
        0000
        ------
        X1011
         1011
         -----
         X0000
          0000
          -----
          X000
```

means codeword is correct and accepted

If the error is there i.e the codeword gets changed then the remainder Zero नहीं आएगा

Sometimes instead of giving string of 0's and 1's the polynomial is given.

Polynomial notation

Odd coeffecient means 1

Even coeffecient means 0

e.g

$$P(x) = x^3 + x^2 + 1$$

means

$$1x^3 + 1x^2 + 0x^1 + 1x^0$$

$$= 1101$$

$$P(x) = 4x^4 + x^3 + 3x^2 + 1$$

means

$$4x^4 + 1x^3 + 3x^2 + 0x^1 + 1x^0$$

$$= 01101$$

# Latency

Latency is calculated for each and every router. Latency is made of four components

Queuing Delay

Processing Delay

Transmission Delay

Propagation Delay

## Queuing Delay

Before processing the packet i.e before issuing the packet to router it is placed in a queue. Queuing delay depends upon queue size (buffer size) and varies from 0 to $\infty$. For 1st packet in queue it is zero. For the packet which do not get space in buffer it is $\infty$.

## Processing Delay

It is the time taken by the router to process the packet header to extract the destination Address, searching routing table to decide the outgoing route. Processing delay depends upon Router processing speed not on packet size.

## Transmission Delay

The amount of time it is reqd to push all the bits of packet into wire (link)

$$\text{Transmission Delay for Packet} = \frac{\text{Packet size}}{\text{Bandwidth}}$$

## Propagation Delay

$$\text{Propagation Delay} = \frac{\text{Distance b/w two concecutive routers}}{\text{Propagation speed}}$$

## Latency

Latency = Queuing Delay + Processing Delay + Transmission Delay + Propagation Delay

## Round Trip Time ( RTT , RTD )

RTT is the time it takes for packet to reach destination plus time it takes for acknowledgement to come back to source. RTT is also known as ping time or MIN ACK waiting time

## Time Out

Time Out = 2 x RTT

MAX ACK waiting time

## TTL

TTL = 2 x Time Out

# RTT calculation

Assume

Queving Delay, Processing Delay, $(TD)_{ack} = 0$

## System having 2 routers

From A to B (packet)


**At A**

$(\cancel{\text{Queving Delay}})_{packet} + (\cancel{\text{Processing Delay}})_{packet} + (TD)_{packet} + (PD)_{packet}$

**At B**

$(\cancel{\text{Queving Delay}})_{packet} + (\cancel{\text{Processing Delay}})_{packet}$

From B to A (ack)

**At B**

$(\cancel{\text{Queving Delay}})_{ack} + (\cancel{\text{Processing Delay}})_{ack} + (\cancel{TD})_{ack} + (PD)_{ack}$

**At A**

$(\cancel{\text{Queving Delay}})_{ack} + (\cancel{\text{Processing Delay}})_{ack}$

RTT = From A to B (packet) + From B to A (ack)

$$= (TD)_{packet} + (PD)_{packet} + (PD)_{ack}$$

$$(PD)_{packet} = (PD)_{ack} = PD$$

$$\boxed{RTT = (TD)_{packet} + 2PD}$$

System having n routers

$$RTT = (n-1)(TD)_{packet} + 2(n-1)PD$$