Error Control: CRC & Checksum - Part II

Complete Course on Computer Networks - Part II

Ravindrababu Ravula • Lesson 3 • Feb 22, 2021

Computer Networks

Error Control Methods PART 1

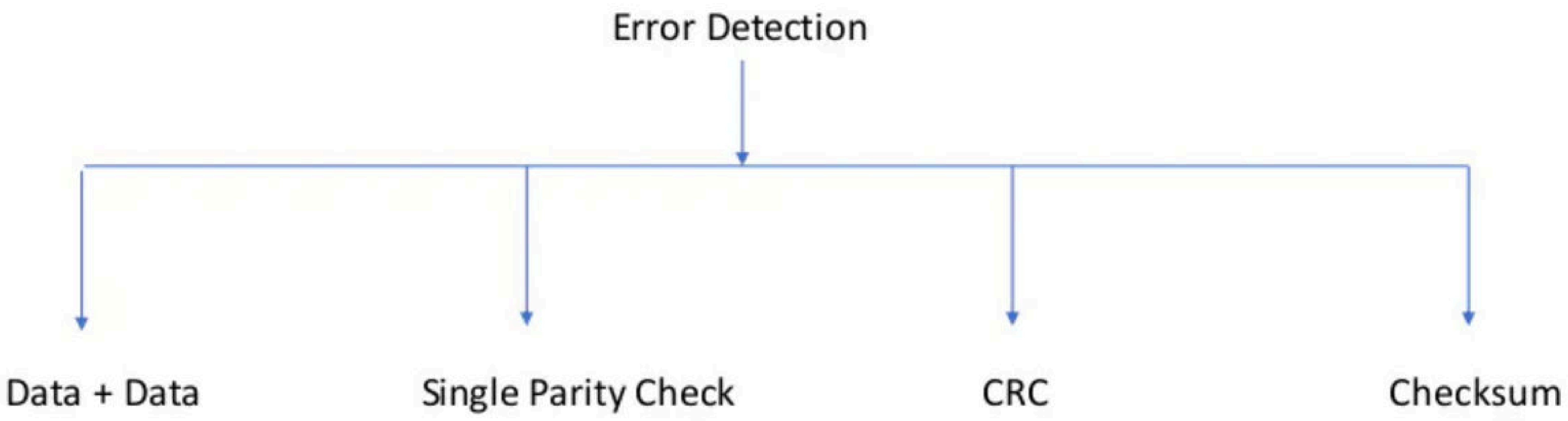
Error Handling Methods

Error Detection

Error detection is a technique that is used to check if any error occurred in the data during the transmission.

Error Correction

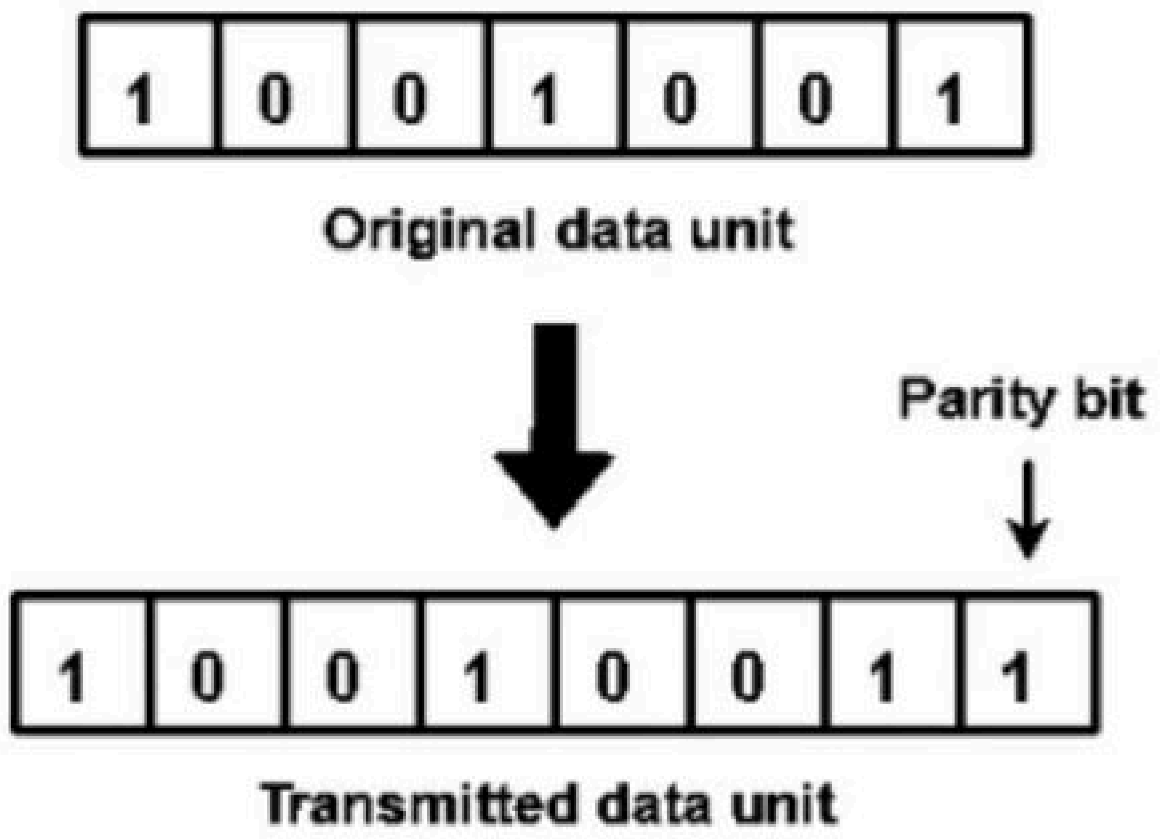
Error Correction is a technique that is used to correct error occurred in the data by its own during the transmission.


Error Detection
Data + Data
Single Parity Check
CRC
Checksum

Single Parity Check-

In this technique,

- One extra bit called as **parity bit** is sent along with the original data bits.
- Parity bit helps to check if any error occurred in the data during the transmission.



Limitation-

- This technique can not detect an even number of bit errors (two, four, six and so on).
- If even number of bits flip during transmission, then receiver can not catch the error.

Cyclic Redundancy Check-

- Cyclic Redundancy Check (CRC) is an error detection method.
- It is based on binary division.

Cyclic Generator-

Data to be sent : 1 0 1 1 0 1 1
CRC generator: 1 1 0 1

CRC generator is 4 bits
There for sender appends 3 bits of 0's to the data

Note: if CRCG= n bits then bits to be appended in data is (n-1) 0's

SENDER'S SIDE

Appended 0's

$$
\begin{array}{r|l}
1\,1\,0\,1 & 1\,0\,1\,1\,0\,1\,1\,\boxed{0\,0\,0} \\
 & 1\,1\,0\,1 \\
\hline
 & 0\,1\,1\,0\,0\,1\,1\,0\,0\,0
\end{array}
$$

Go on applying XOR

Appended 0's

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

Go on applying XOR

Appended 0's

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

Go on applying XOR

Appended 0's

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

Go on applying XOR

CRC

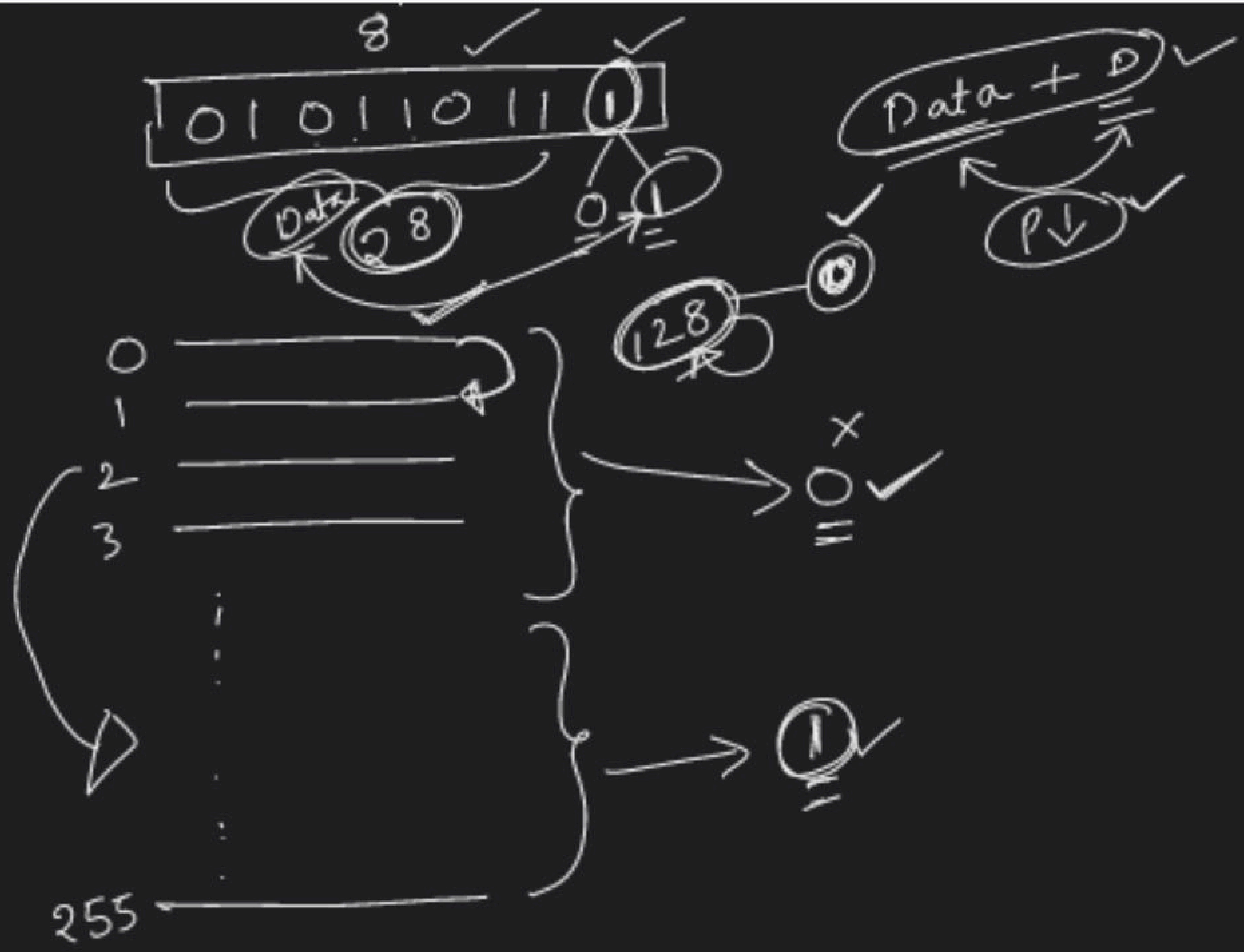
DATA SENT : 1 0 1 1 0 1 1 0 0 1

RECEIVER'S SIDE

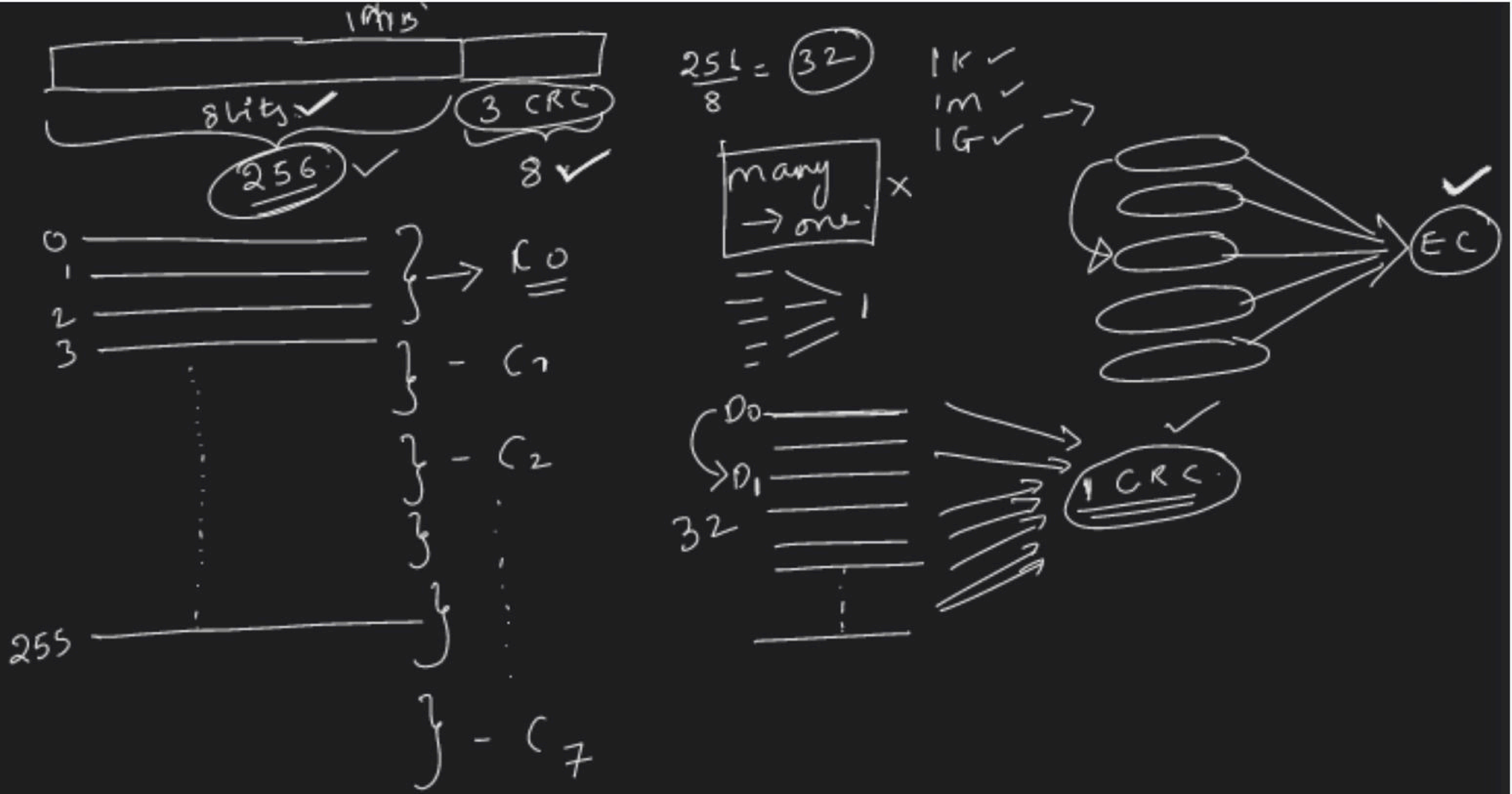
```
1101 ) 1 0 1 1 0 1 1 0 0 1
       1 1 0 1
       -------------------
       0 1 1 0 0 1 1 0 0 1
         1 1 0 1
       -------------------
         0 0 0 1 1 1 0 0 1
               1 1 0 1
       -------------------
          0 0 0 0 0 1 1 0 1
                      1 1 0 1
       -------------------
          0 0 0 0 0 0 0 0 0
```

Go on applying XOR

CRC IS 0, DATA RECEIVED IS RIGHT!



8
0 1 0 1 1 0 1 1 1
Data
128
0
1
Data + D
P↓
128
0
1
2
3
255
×
0
1



1MB
8 bits
256
3 CRC
8
0
1
2
3
255
C0
C1
C2
C7
256/8 = 32
1K
1M
1G
many → one
X
1
EC
D0
D1
32
1 CRC

D 8
EC 8
256
256
0
1
2
3
4
255
256
D
G
gn func
one-one
Data → Data → CRC
CRC
CS.
one-one mapp.
Data + Data
NT → 2
1.5 X
2.0
D1 → D
ED
D10
100%
many
100
10
Data
EC
many → 1
D + D
100%
2^100 → 2^10
Bit err
Burst err
EC
D + D
ECC
Data
Data
Data

Computer Networks

Error Control Methods PART 2

CHECKSUM

CHECK SUM

To verify — Addition

Adding

8 bit-CS
16 bit-CS
32 bit-CS

S
OS
CS
CRC
PQ
CN

8 | 8 | 8 bit | 8 bits

DATA TO BE SENT

SUPPOSE WE ARE USING 8 BIT CHECKSUM

D + CS

00001000	00001100	00001010	00010000	00000011	00000001	00000010	00000111
8	8	8	8	8	8	8	8

Not

ENCODE EACH 8 BITS INTO A DECIMAL NUMBER

00001000	00001100	00001010	00010000	00000011	00000001	00000010	00000111
8	12	10	32	3	1	2	7

ADDING ALL WE GET 75

CHECKSUM = -75

75

75

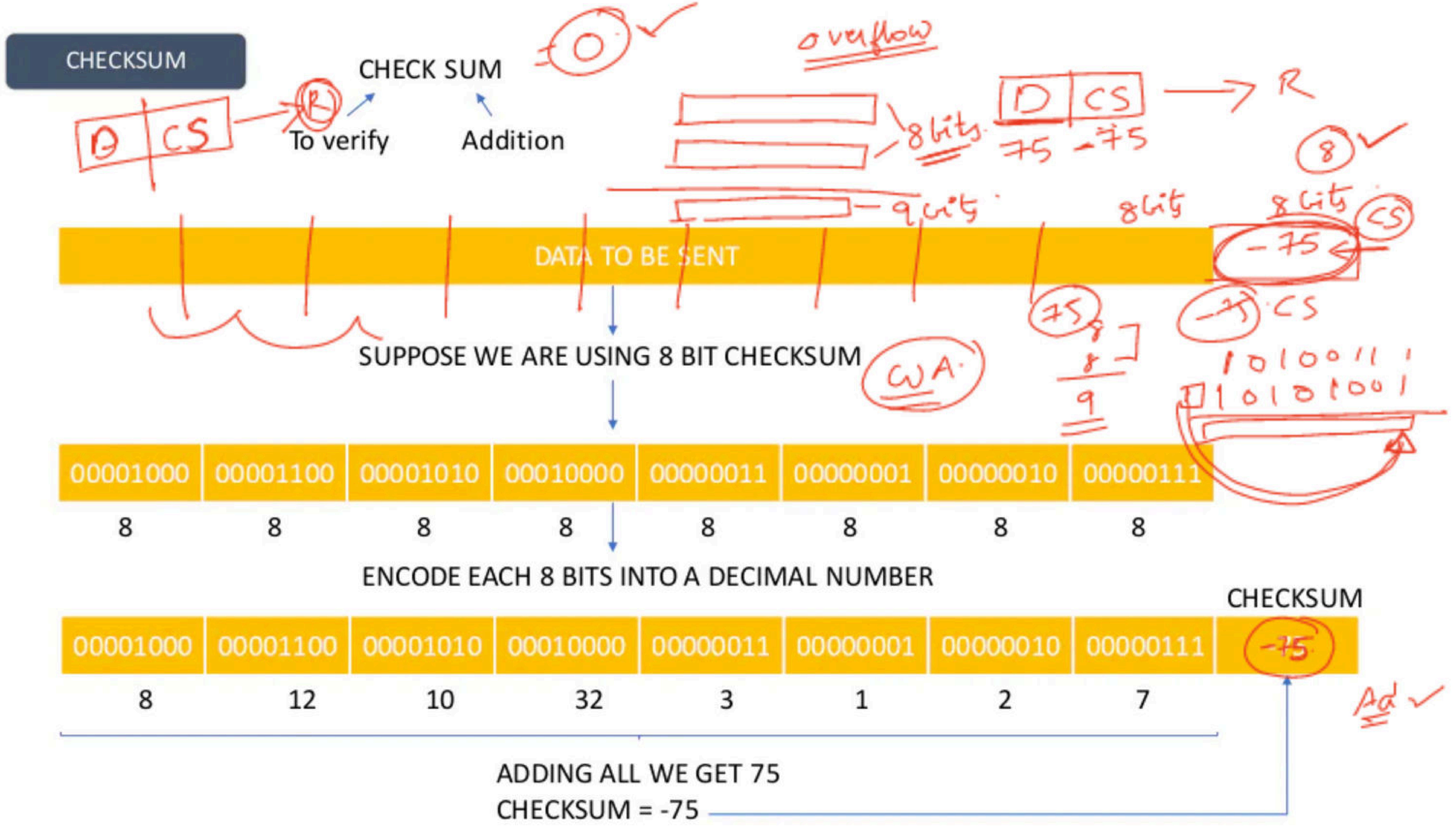
1's

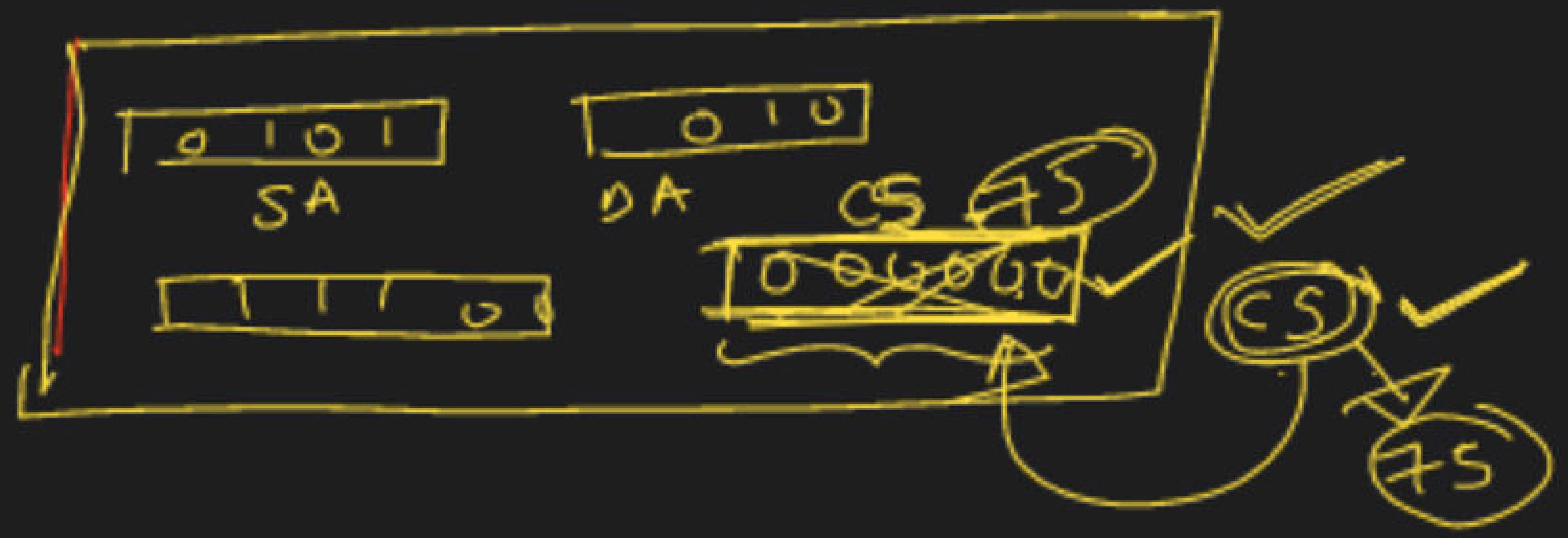
-75

CS

Sum = 75

checksum = -75


CHECKSUM
CHECK SUM
To verify
Addition
D
CS
R
overflow
8 bits
9 bits
75
-75
DATA TO BE SENT
SUPPOSE WE ARE USING 8 BIT CHECKSUM
WA
00001000
00001100
00001010
00010000
00000011
00000001
00000010
00000111
8
8
8
8
8
8
8
8
ENCODE EACH 8 BITS INTO A DECIMAL NUMBER
CHECKSUM
00001000
00001100
00001010
00010000
00000011
00000001
00000010
00000111
-75
8
12
10
32
3
1
2
7
ADDING ALL WE GET 75
CHECKSUM = -75



0101
SA
010
DA
CS
75
CS
75

CRCG = 1101

Data = 1010110111 10 → Reception

1) C

2) NC

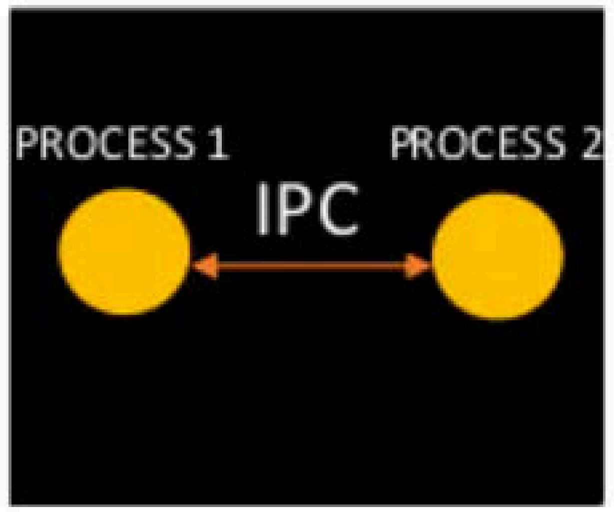
Corruption

$\boxed{\text{OSI}}$

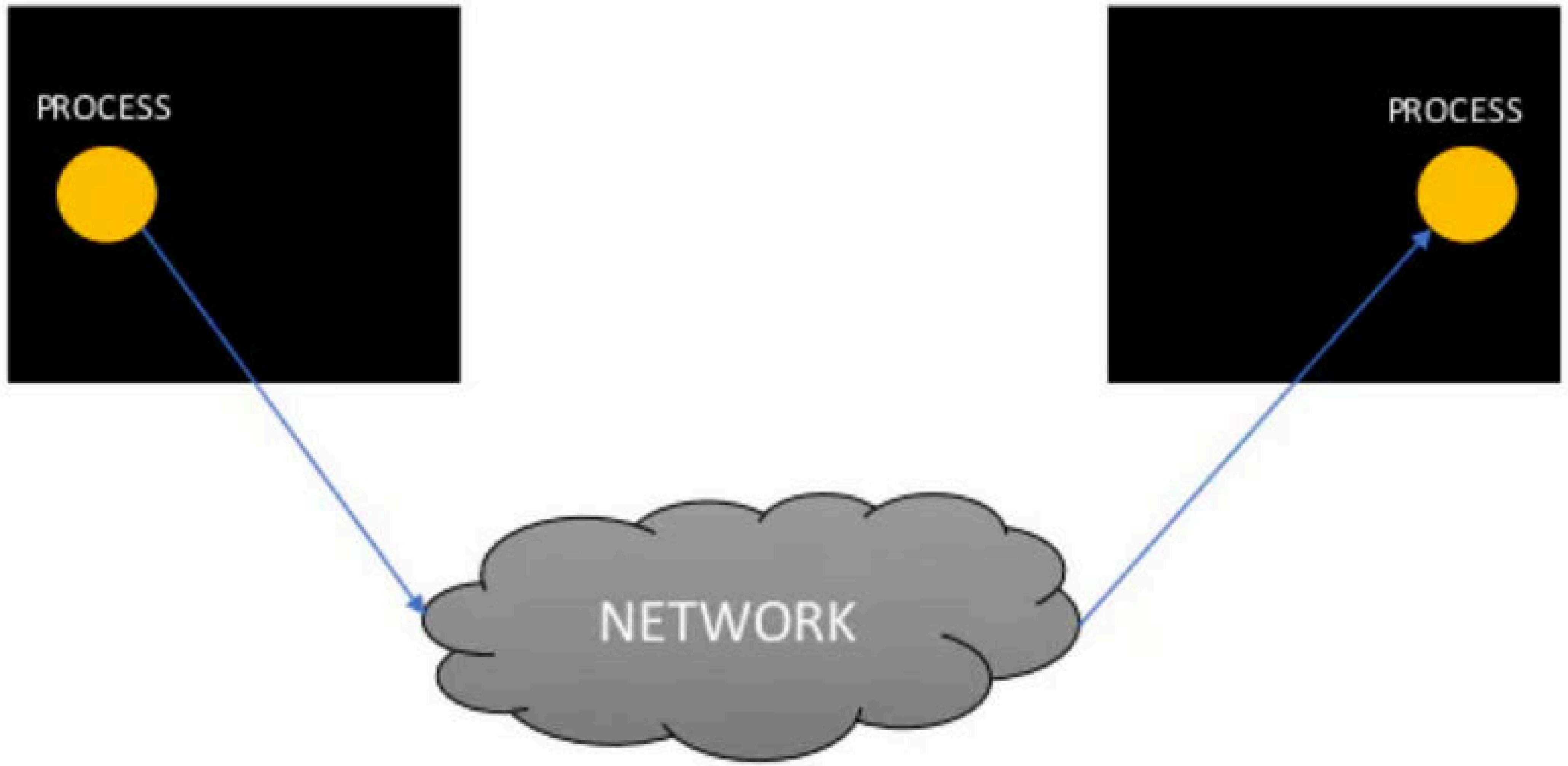
Computer Networks

ISO – OSI LAYERS

Communication between processes of same host



Communication between processes of different hosts



Functions

Mandatory

- Error Control
- Flow Control
- Access Control
- Multiplexing and Demultiplexing
- Addressing and many more……

Optional

- Encryption and Decryption
- Checkpointing
- Routing (We can use Flooding)

There are certain models that provide functionalities like OSI, TCP/IP,ATM,IEEE

ISO – OSI MODEL

ISO stands for International organization of Standardization.

OSI is Open System Interconnection and the model is commonly known as OSI model.

The ISO-OSI model is a seven layer architecture. It defines seven layers or levels in a complete communication system. They are:

- Application Layer
- Presentation Layer
- Session Layer
- Transport Layer
- Network Layer
- Datalink Layer
- Physical Layer

ADVANTAGES OF LAYERING

- DIVIDE AND CONQUER
- ENCAPSULATION
- ABSTRACTION
- TESTING

To Remember the Layers

Application Layer	ALLIGATOR
Presentation Layer	PET
Session Layer	STEVE'S
Transport Layer	TOUCH
Network Layer	NOT
Datalink Layer	DO
Physical Layer	PLEASE

Computer Networks

Physical Layer

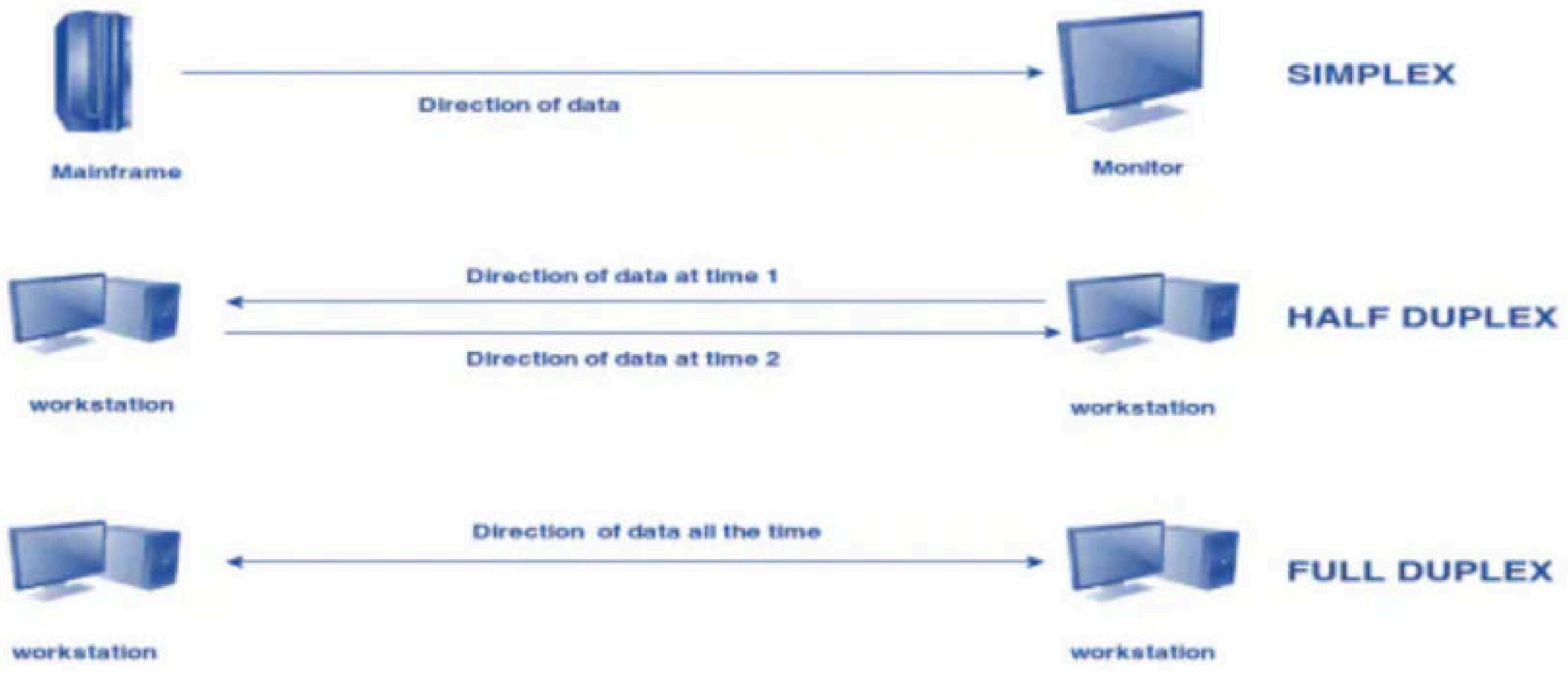
Functions of Physical Layer

1.) Physical Layer is electrical, mechanical, procedural and functional characteristics of physical links

It depends upon the type of links we are using to communicate.

If it is a copper wire then messages will be converted into electrical signals.
If link is an optical fibre then messages will be converted to light signals.
In case of Wireless communication, messages are sent into form of Electro Magnetic Waves.

2.) Physical Layer also includes the Transmission Mode – Simplex / Duplex



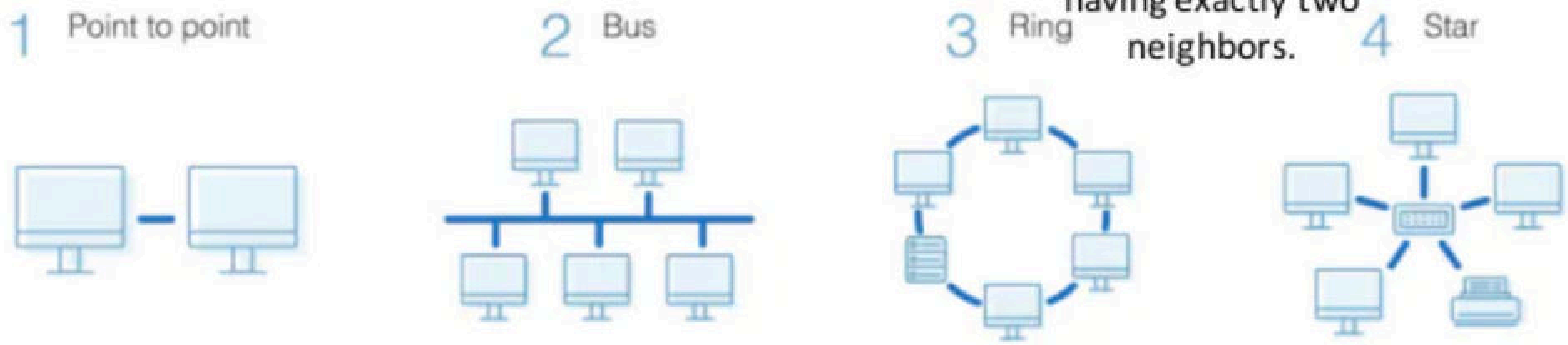
3.) Physical Layer also deals with Topologies

Point to Point topology is the simplest topology which connects two nodes directly together with a common link.

A bus topology orients all the devices on a network along a single cable running in a single direction from one end of the network to the other

Ring topology is where nodes are arranged in a circle (or ring). The data can travel through the ring network in either one direction or both directions, with each device having exactly two neighbors.

A star topology, the most common network topology, is laid out so every node in the network is directly connected to one central hub via coaxial, twisted-pair, or fiber-optic cable. Acting as a server, this central node manages data transmission—as information sent from any node on the network has to pass through the central one to reach its destination—and functions as a repeater, which helps prevent data loss.



Each node in a star topology is directly connected to the central hub, a tree topology has a parent-child hierarchy to how the nodes are connected.

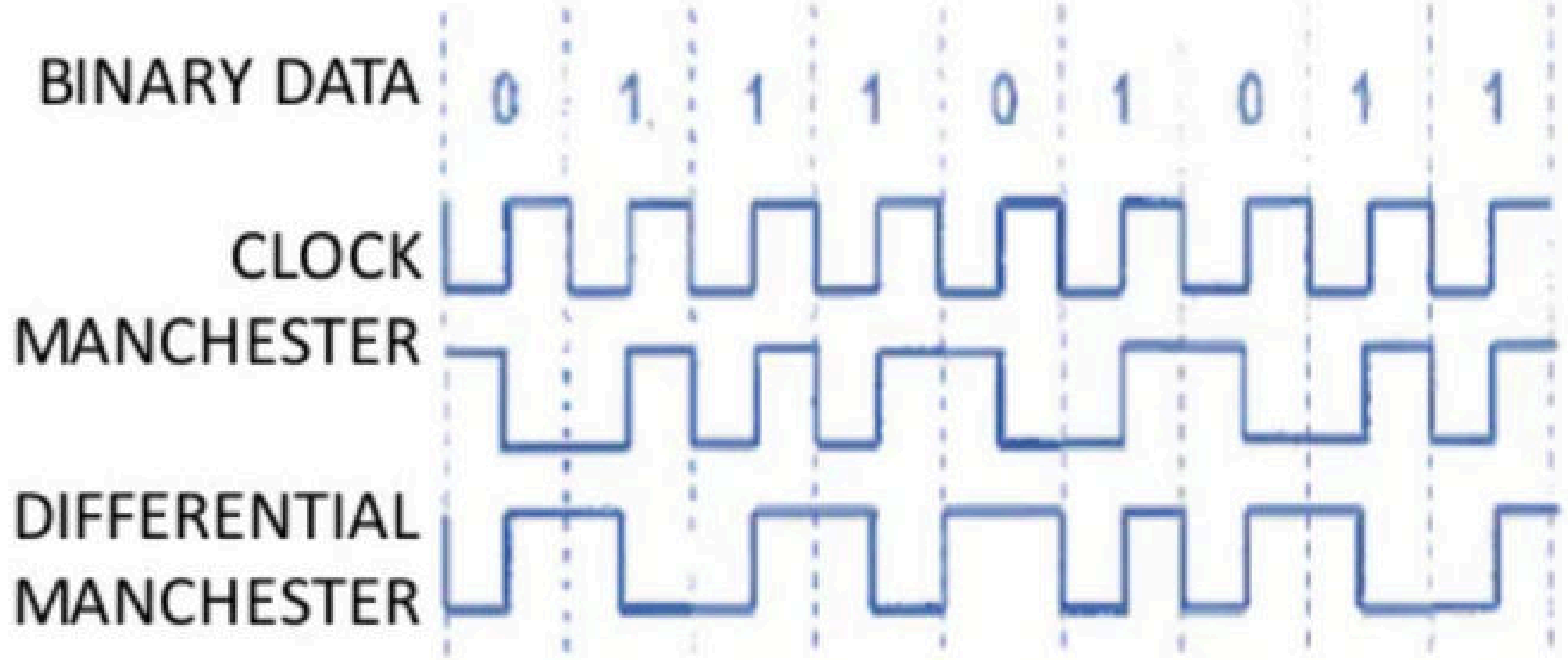
A mesh topology is a network setup where each computer and network device is interconnected with one another.

Hybrid topology is an integration of two or more different topologie s to form a resultant topology

4.) Encoding -

Encoding is a method of converting a stream of data bits into a predefined code. 1- To provide a predictable pattern that can be recognized by both the sender and the received. 2- To distinguish data bits from control bits and provide better media error detection. 3- To provide codes for control purposes such as identifying the beginning and end of a frame.

Signaling, the Physical layer must generate the electrical, optical, or wireless signals that represent the "1" and "0" on the media.



Baud rate = 2*bit rate

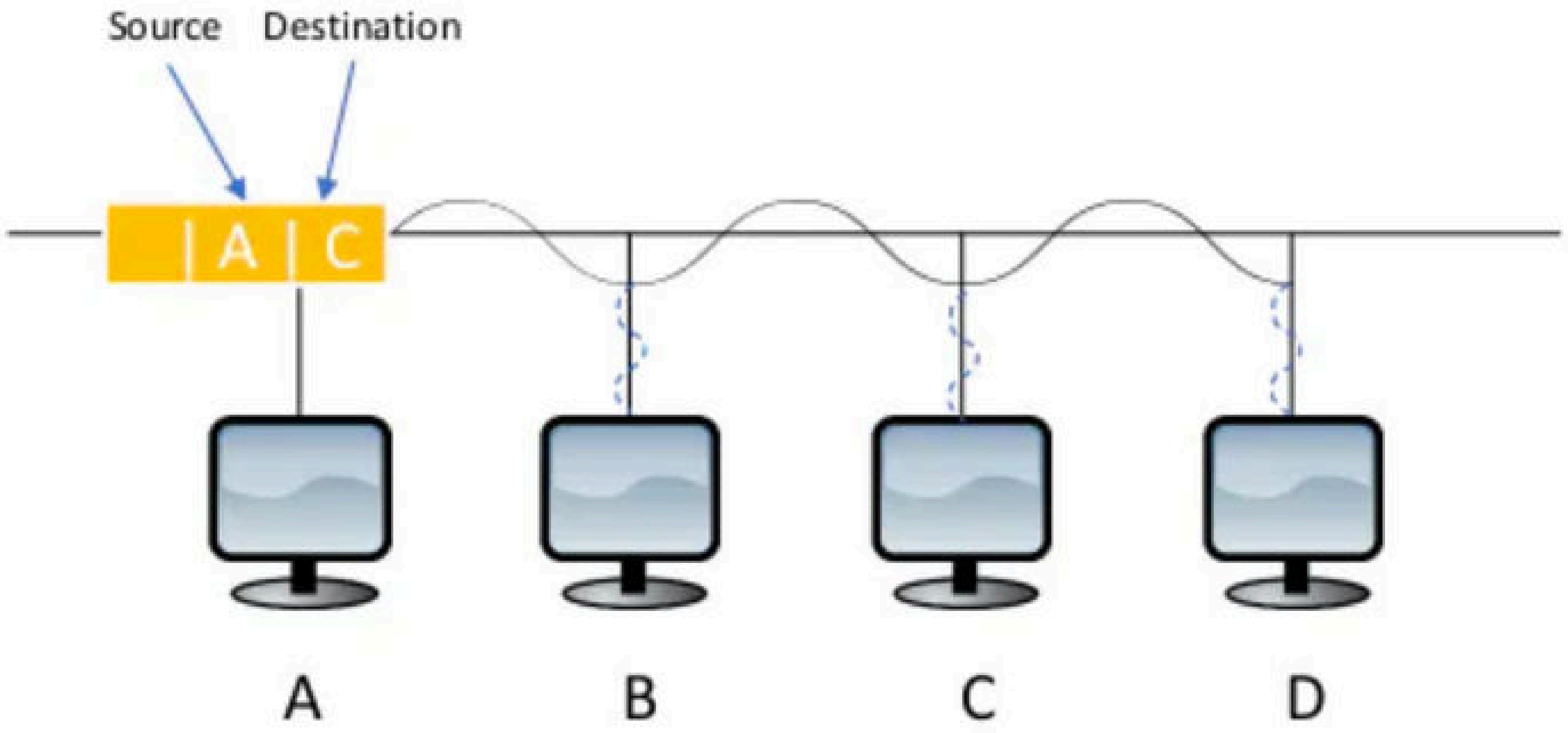
Computer Networks

Data Link Layer

Functions of DLL

- Error Control
- Flow Control
- Access Control
- Framing
- Physical Addressing

FRAMING

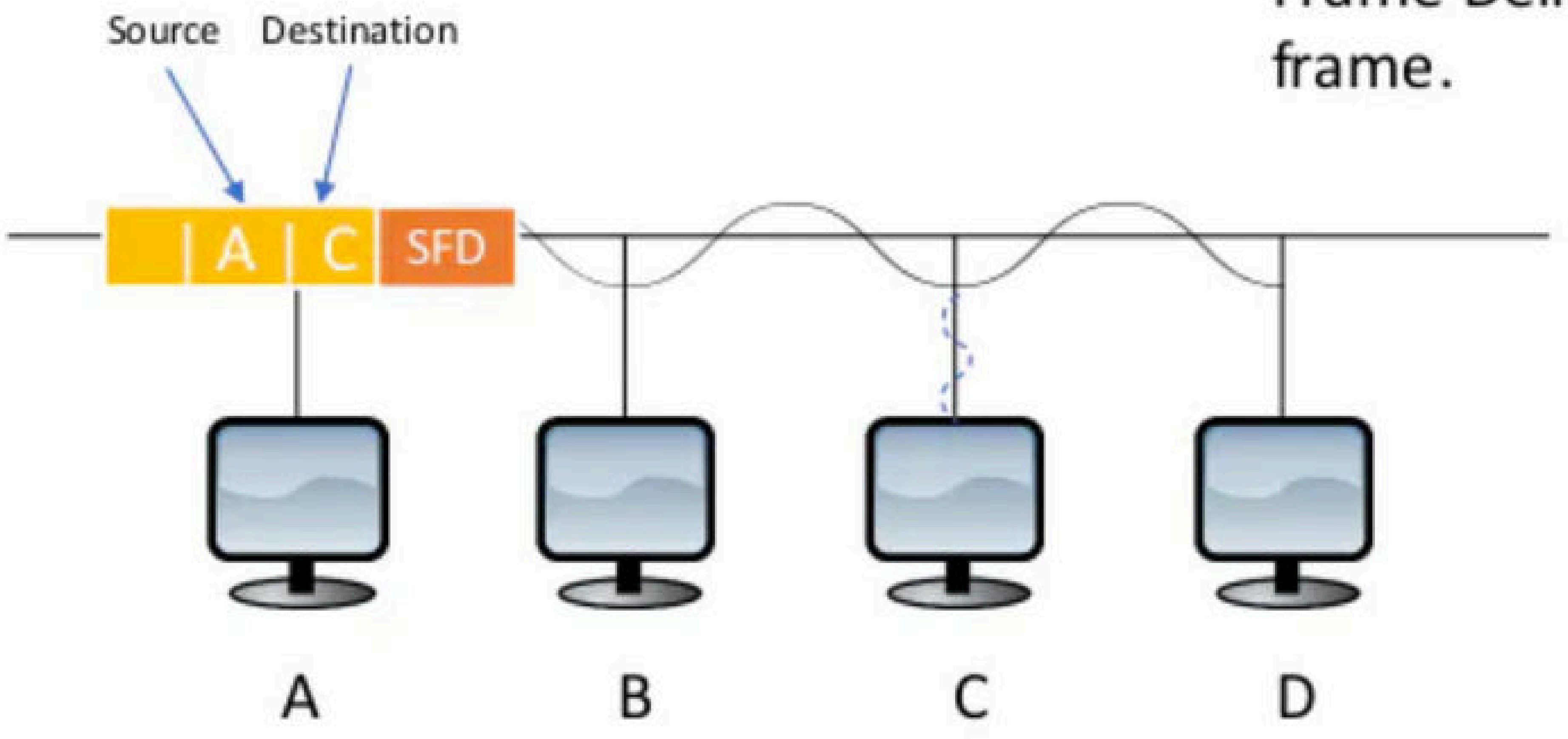


Suppose A wants to send a message to C.
It will send a frame which includes A and C as Source and Destination respectively.
The message would received by every node connected to the link but accepted by the one in the destination address of the message.
The question is,
When should any node look for a message?
Also, all the must only check the beginning of the Frame and see whether that frame is for them or no,
How will they know the beginning of the frame?

FRAMING

Start Frame Delimiter

It is a 1 byte field which is always set to 10101011. The last two bits "11" indicate the end of Start Frame Delimiter and marks the beginning of the frame.

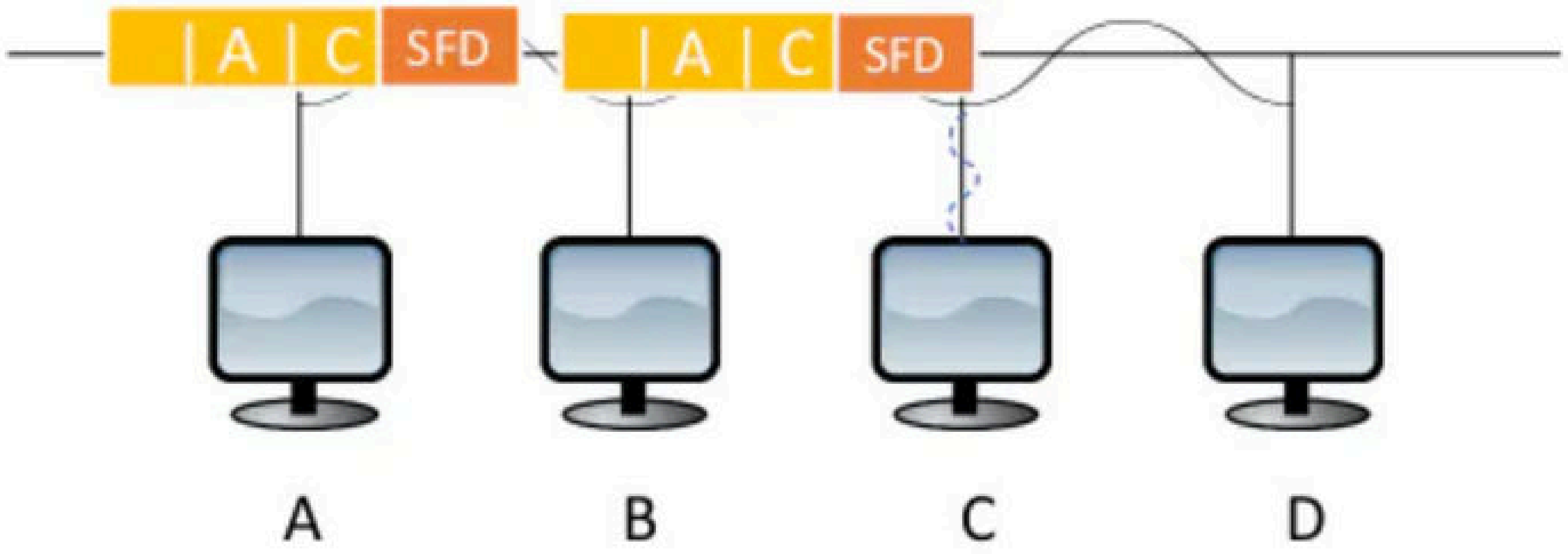


- The above two fields are added by the physical layer and represents the physical layer header.
- Sometimes, Start Frame Delimiter (SFD) is considered to be a part of Preamble.
- That is why, at many places, Preamble field length is described as 8 bytes.

SFD will be added at the beginning of the frame, So that the hosts will come to know that a data packet has arrived and they have to check the destination address that is after the SFD

FRAMING

Suppose after sending this frame,
A or some other station sends one more frame then C should know the end of the First frame i.e when it has to stop reading.



FRAMING

FIXED LENGTH

Here the size of the frame is fixed and so the frame length acts as delimiter of the frame. Consequently, it does not require additional boundary bits to identify the start and end of the frame.

VARIABLE LENGTH

Here, the size of each frame to be transmitted may be different. So additional mechanisms are kept to mark the end of one frame and the beginning of the next frame.

Two ways to define frame delimiters in variable sized framing are :

•**Length Field** – Here, a length field is used that determines the size of the frame. It is used in Ethernet (IEEE 802.3).
•**End Delimiter** – Here, a pattern is used as a delimiter to determine the size of frame. It is used in Token Rings. If the pattern occurs in the message, then two approaches are used to avoid the situation –

- **Character-Stuffing** – A byte is stuffed in the message to differentiate from the delimiter. This is also called character-oriented framing.
- **Bit – Stuffing** – A pattern of bits of arbitrary length is stuffed in the message to differentiate from the delimiter. This is also called bit – oriented framing.

GATE 2004 IT

In a data link protocol, the frame delimiter flag is given by 0111. Assuming that bit stuffing is employed, the transmitter sends the data sequence 01110110 as

A. 01101011

B. 011010110

C. 011101100

D. 0110101100

In the data link layer, bits stuffing is employed then bit stuffing is done using the flag delimiter. If there is a flag of n bits then we will compare the data sequence with the flag and for every n-1 bits matched found, a bit 0 is stuffed in the data sequence.

Thus using the above logic,

Delimiter flag: 0111

Data sequence: 01110110

So, for a flag of 4 bits we will compare data sequence with a pattern of 3 bits, i.e., 011.

0 1 1 0 1 0 1 1 0 0

In the above pattern the underlined bits are found matched. Hence, 0 in italics is stuffed. Thus resulting in the data sequence as 0110101100