Computer Network

Data-Link Layer

Lecture : 8/9

Gaurav Raj

TCP/IP

TCP/IP Layer	Hardware	Software/Protocols
Application	None	HTTP, FTP, SMTP, POP3, IMAP, DNS, SSH
Transport	None	TCP, UDP
Internet	Routers	IP (IPv4/v6), ICMP, IGMP, ARP, RARP Routing(DVR(RIP), LSR(OSPF), BGP)
Data Link	Switches, Bridges, NICs	Ethernet (MAC framing), Wi-Fi (802.11 MAC), PPP, Frame Relay, HDLC
Physical	Cables (fiber, coaxial, twisted pair), Hubs, Repeaters, Connectors (RJ-45), Amplifier	ONLY physical standards (IEEE 802.3 for wiring, IEEE 802.11 PHY for Wi-Fi)

Data-Link Layer

Responsibility
Framing
Error Detection
Error Recovery
Flow Control
Access Control
Addressing
Link Management
Framing and Encapsulation

Cyclic Redundancy Check : Error Control

Term	Meaning
Data/Message	Binary string to be transmitted
Generator (G)	A predetermined binary number (like a polynomial)
Divisor	Same as generator
CRC or Remainder	The extra bits added to message for error detection
Transmitted Frame	Message + CRC

Cyclic Redundancy Check : Error Control

Conceptual Steps in CRC

1. **Append (n−1) Zeros** to the data, where n = length of the generator.

2. **Divide** the new data by the generator using **modulo-2 division** (XOR instead of subtraction).

3. The **remainder** becomes the **CRC**.

4. The sender **appends** this CRC to the original data.

5. The receiver **divides** the received data (message + CRC) by the same generator.

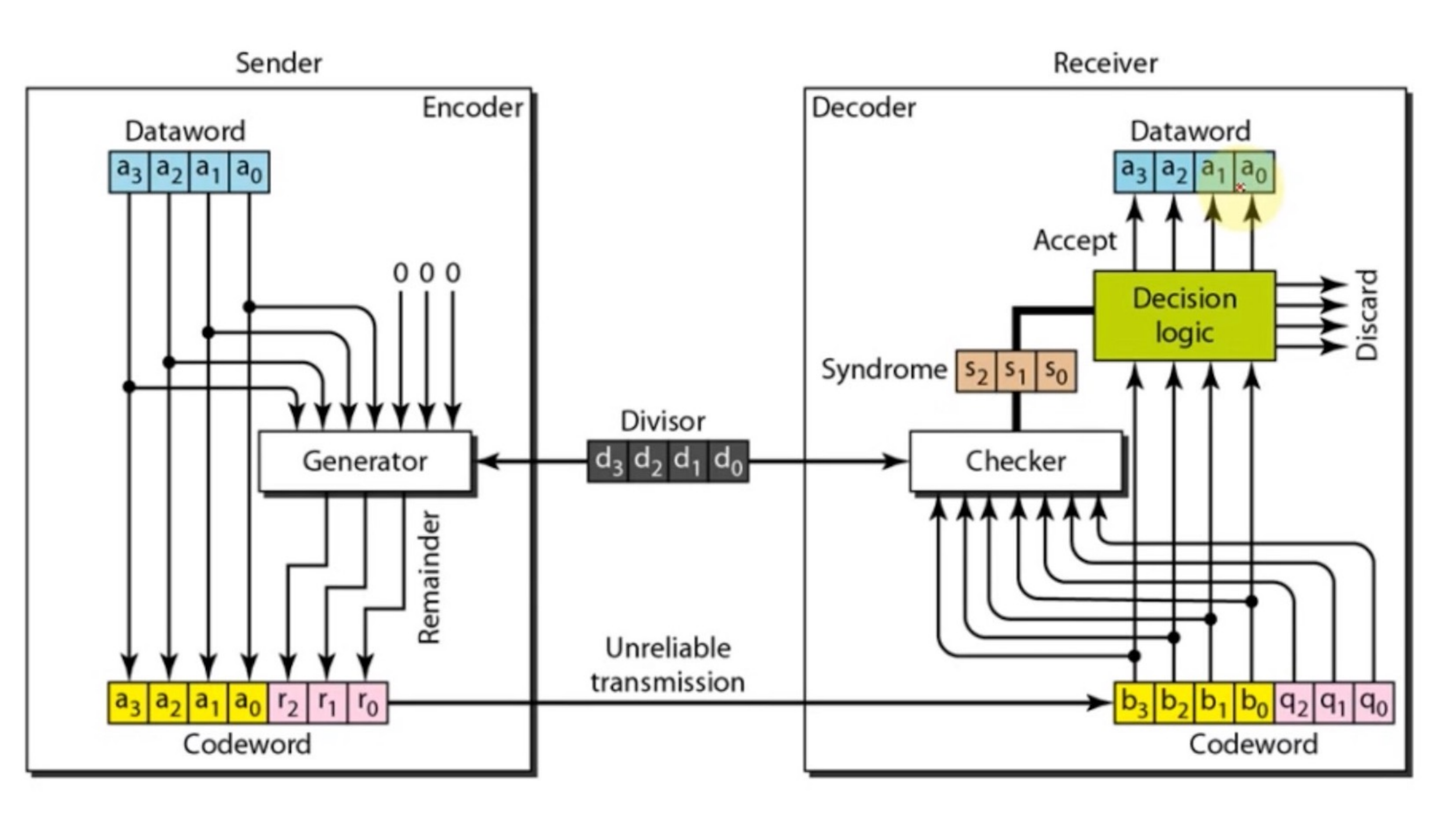
    1. If remainder = 0 ⇒ **No Error**
    2. Else ⇒ **Error Detected**

Cyclic Redundancy Check : Error Control

Message (M): 0000 (All 4-bit)

Generator: **G(x) = 1011 ($x^3 + x + 1$)**

Cyclic Redundancy Check : Error Control



1. A computer network uses polynomials over GF(2) for error checking with 8 bits as information bits and uses x^3+x+1 as the generator polynomial to generate the check bits. In this network, the message 01011011 is transmitted as

GATE:2017

A. 01011011010
B. 01011011011
C. 01011011101
D. 01011011100

Cyclic Generator: **$G(x)$ = 1011 ($x^3 + x + 1$)**

Dataword	Codeword	Dataword	Codeword
0000	0000000	1000	1000101
0001	0001011	1001	1001110
0010	0010110	1010	1010011
0011	0011101	1011	1011000
0100	0100111	1100	1100010
0101	0101100	1101	1101001
0110	0110001	1110	1110100
0111	0111010	1111	1111111

Let's Test $G(x)=x3+x+1=1011$.

$G(x) \mid (x^n+1)$ This is equivalent to saying:

We want to find the **smallest value of n** such that:

What is the **order** of $G(x)$ in $GF(2)$?

Let's compute powers of x mod G(x)

We reduce each **x^k modulo $G(x)$** and see when we first get 1 again. We'll use polynomial division mod 2:

k	X^k mod $G(x)$
1	x
2	x^2
3	$X^3 \bmod G(x) = x + 1$
4	$x(x+1) = x^2 + x$
5	$x(x^2 + x) = x^3 + x^2 \equiv (x + 1) + x^2 = x^2 + x + 1.$
6	$x \cdot (x^2+x+1) = x^3+x^2+x \equiv (x+1)+x^2+x = x^2+1$
7	$X^7 = x \cdot (x^2 + 1) = x^3 + x \equiv (x+1) + x = 1$
8	$X^7 \bmod G(x) =1$

So, the **order of x mod $G(x)$ is 7**

Yes, $G(x)=x^3+x+1$ is a **cyclic generator** for a **(7, 4)** cyclic code, because it divides x^7+1 over $GF(2)$.

Cyclic Redundancy Check : Error Control

Type of Change	Detected by CRC?	Why?
Single-bit flip	Yes	Changes polynomial, changes remainder
Multiple random bit flips	Yes (usually)	Remainder likely changes
Burst errors (small)	Yes	Covered by degree of generator
Cyclic rotation of bits	No	Rotated version still divisible by generator
Cleverly crafted errors	No	If difference is a multiple of generator

CRCs are **not cryptographic hashes**. They're **fast error-checking tools**, not meant for security or uniqueness.

When You SHOULD Use a Cyclic Generator (i.e., Generator that makes codewords cyclic)

Use Case	**Reason**
You want closed behaviour under bit rotation	Good for applications where cyclic shifts might occur (like rotating disk errors).
Better error detection	Some cyclic codes (like Hamming, BCH, Reed-Solomon) have **strong error detection and correction** capabilities.

When You Should AVOID Using a Cyclic Generator (e.g., in typical CRC)

Use Case	**Reason**
Standard CRC for error detection in packets/files	CRC is **not meant** to be closed under rotation. In fact, cyclic shifts are often **not detectable** by CRC.
Security or anti-manipulation use	If rotation creates another valid codeword, **CRC fails to detect it**, which is dangerous.

Technique	Error Detection Capability	Complexity
Parity Bit	Detect single-bit error	Low
Checksum	Detects multiple errors but weak	Low
CRC	Detects burst errors(many bits)	Moderate

1. Consider the generator polynomial $G(x)=x^3+x+1$. A message $M(x)=11000$ (i.e., 5 bits) is to be transmitted using CRC. Determine the CRC check bits (remainder) and the final transmitted bit stream.

[MCQ]

(A) 001

(B) 110

(C) 111

(D) 101

2. A CRC scheme uses generator x^4+x+1 (i.e., 10011). If the message is 111000, what are the CRC bits and the transmitted data?

[MCQ]

- **A** 1001
- **B** 1011
- **C** 0001
- **D** 1101

3. A sender wants to transmit the following three 8-bit binary data words using checksum-based error detection:

Word 1: 11001001 **[NAT]**
Word 2: 01101101
Word 3: 10011011

Compute the 8-bit checksum that should be sent by the sender.

4. At the receiver’s end, the received words are:

Word 1: 11000001 **[NAT]**
Word 2: 01101101
Word 3: 10011011
Checksum: ________

Fill in the correct checksum value from QN:5 and verify if any error is detected at the receiver side.

5. A 2-D even parity scheme is used to detect errors in a data block consisting of **4 rows × 7 bits**. The sender constructs the 2-D parity matrix by first adding one **row parity bit** at the end of each row and then one **column parity bit** at the bottom of each column, including the parity bits.

The resulting **5×8 matrix** is sent over a noisy channel.

At the receiver's end, the received matrix is as follows: **[NAT]**

Row 1: 1 0 1 1 0 1 0 | 0
Row 2: 0 1 0 1 1 1 0 | 0
Row 3: 1 1 1 0 1 0 1 | 1
Row 4: 0 0 0 1 0 0 1 | 0
----
0 0 0 1 0 0 0 | 1

(a) Has any error occurred during transmission?

(b) If yes, identify the bit position (row, column) where the single-bit error occurred.

Hamming Distance(error detection and error correction)

It is used in digital communication and memory systems.

Error Detection & Correction Capability

- Detectable Errors:
  Up to two-bit errors can be detected.
- Correctable Errors:
  Only one-bit error can be corrected.

Formulas to Remember

1. Number of Parity Bits (r):
$2^r \geq m+r+1$,where m is the number of data bits.

2. Total Codeword Length (n):

$$n = m + r$$

3. Positions of Parity Bits:
Parity bits are placed at positions that are powers of 2:

$$2^0, 2^1, 2^2, \ldots$$

Parity Check Equation:
Each parity bit covers specific data bits based on its binary representation.

Hamming Distance(error detection and error correction)

Interpreting the Formula

•**To detect up to d errors**, we need a **minimum Hamming distance** of **d+1**.

•**To correct up to d errors**, we need a **minimum Hamming distance** of **2d+1**.

Why Does Standard Hamming Code Detect 2 Errors and Correct 1?

Hamming code has a **minimum Hamming distance of 3**:

•If **1-bit flips**, it moves the received word 1 step away from the original → **correctable**.

•If **2 bits flip**, the word is now **2 steps away** and might resemble another valid word → **detectable but not correctable**.

•If **3 bits flip**, it can turn into another valid codeword → **not even detectable**.

Thus, standard **Hamming(7,4) or Hamming(11,7) codes can correct only 1-bit error and detect up to 2-bit errors**.

standard Hamming code is constructed with **minimum Hamming distance = 3**.

Code Type	**1-bit Error**	**2-bit Error**	**More than 2**
Standard Hamming	Corrects	Detects	Fails
SEC-DED (Hamming + P)	Corrects	Detects	Fails
BCH, Reed-Solomon	Can Correct	Can Correct	Possible

Hamming Distance(error detection and error correction)

Step 1: Consider a Valid Hamming Codeword
We will use **Hamming(7,4)**, where 4 data bits are encoded into a **7-bit codeword**.
Let's say the **valid codeword** generated is:
C=0110011. This follows all parity rules and is a **valid codeword**.

A Single-Bit Flip (Correctable Error)
If **only 1-bit flips**, let's say the **3rd bit flips from 1 to 0**:

Position	1	2	3	4	5	6	7
Valid C	0	1	1	0	0	1	1
Received	0	1	**0**	0	0	1	1

Hamming Distance(error detection and error correction)

A Two-Bit Flip (Detectable but NOT Correctable)

Now, suppose **two bits flip** instead:

The **3rd bit flips from 1 to 0**.

The **6th bit flips from 1 to 0**

Position	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Valid	**0**	**1**	**1**	**0**	**0**	**1**	**1**
Error	**0**	**1**	**0**	**0**	**0**	**0**	**1**

A 3-bit error (Another Valid Codeword)

Position	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Valid	**0**	**1**	**1**	**0**	**0**	**1**	**1**
Error	**1**	**0**	**0**	**0**	**0**	**1**	**1**

This **happens to be another valid codeword** because it still satisfies parity rules! The receiver **won't even realize there was an error** because it **looks like a correct codeword**. (1^{st}, 2^{nd},3^{rd} bit gets flipped)

Hamming Distance(error detection and error correction)

Determine the Number of Parity Bits

Given **7-bit data(1101001) (m=7)**, find r = ?

Arrange Data and Parity Bits

The total number of bits in the Hamming code will be:

n = m + r = 7 + 4 = 11

Place **parity bits at positions**: 1, 2, 4, 8

Position	1	2	3	4	5	6	7	8	9	10	11
Bit Type	P1	P2	D1	P3	D2	D3	D4	P4	D5	D6	D7

Position	1	2	3	4	5	6	7	8	9	10	11
Bit	**P1**	**P2**	**1**	**P3**	**1**	**0**	**1**	**P4**	**0**	**0**	**1**

Hamming Distance(error detection and error correction)

1	**0**	**0**	**0**	**1**
2	**0**	**0**	**1**	**0**
3	**0**	**0**	**1**	**1**
4	**0**	**1**	**0**	**0**
5	**0**	**1**	**0**	**1**
6	**0**	**1**	**1**	**0**
7	**0**	**1**	**1**	**1**
8	**1**	**0**	**0**	**0**
9	**1**	**0**	**0**	**1**
10	**1**	**0**	**1**	**0**
11	**1**	**0**	**1**	**1**
12	**1**	**1**	**0**	**0**

Hamming Distance(error detection and error correction)

Calculate Parity Bits

Parity bits are calculated based on even parity.

- **P1 (Covers positions: 1, 3, 5, 7, 9, 11)** P1=Parity(1,1,1,0,1) = 0
- **P2 (Covers positions: 2, 3, 6, 7, 10, 11)** P2=Parity(1,0,1,0,1) = 1
- **P3 (Covers positions: 4, 5, 6, 7)** P3=Parity(1,0,1) = 0
- **P4 (Covers positions: 8, 9, 10, 11)** P4=Parity(0,0,1) =1

Position	1	2	3	4	5	6	7	8	9	10	11
Bit	0	1	1	0	1	0	1	1	0	0	1

Position	1	2	3	4	5	6	7	8	9	10	11
Bit	P1	P2	1	P3	1	0	1	P4	0	0	1

Hamming Distance(error detection and error correction)

Position	1	2	3	4	5	6	7	8	9	10	11
Bit	0	1	1	0	1/0	0	1	1	0	0	1

2. Consider a binary code that consists of only four valid code words as given below:

00000,01011,10101,11110

Let the minimum Hamming distance of the code be p and the maximum number of erroneous bits that can be corrected by the code be q. Then the values of p and q are

GATE:2017

A. p=3, q=1
B. p=3, q=2
C. p=4, q=2
D. p=4, q=1

3. Assume that a 12-bit Hamming codeword consisting of 8-bit data and 4 check bits is $\mathbf{d_8 d_7 d_6 d_5 c_8 d_4 d_3 d_2 c_4 d_1 c_2 c_1}$, where the data bits and the check bits are given in the following tables:

Data Bits							
D8	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**
1	**1**	**0**	**X**	**0**	**1**	**0**	**1**

Check bits			
C8	**C4**	**C2**	**C1**
Y	**0**	**1**	**0**

Which one of the following choices gives the correct values of x and y?

GATE: 2021

A. X is 0, y is 0
B. X is 0, y is 1
C. X is 1, y is 0
D. X is 1, y is 1

Ethernet Frame Format (IEEE 802.3)

Preamble: 7B	SFD: 1 B	Destination MAC: 6B	Source MAC: 6B	Length: 2B	Data: (46 – 1500) B	FCS (CRC): 4B

Field	**Size**	**Description**
Destination MAC	6 Bytes	Receiver's address
Source MAC	6 Bytes	Sender's address
Length/Type	2 Bytes	Indicates either data length (if ≤ 1500) or Ethertype (if ≥ 0x0600)
Payload	46–1500 Bytes	Actual data being sent
FCS (CRC)	4 Bytes	Error detection using CRC-32

Ethernet Frame Format (IEEE 802.3)

Preamble: 7B	SFD: 1 B	Destination MAC: 6B	Source MAC: 6B	Length: 2B	Data: (46 – 1500) B	FCS (CRC): 4B

Preamble (7 Bytes)

- Value: 10101010 repeated
- Purpose: Allows the receiver to **synchronize clock** with sender before actual data begins.
- Not considered part of the "actual Ethernet frame" as per IEEE 802.3.

SFD – Start Frame Delimiter (1 Byte)

- Value: 10101011
- Marks the **end of preamble** and **start of frame**.
- Also **not** part of the frame when calculating size (used only by PHY layer for alignment).

Ethernet Frame Format (IEEE 802.3)

Preamble:	SFD:	Destination	Source	Length:	Data:	FCS (CRC):
7B	1 B	MAC: 6B	MAC: 6B	2B	(46 – 1500) B	4B

Ethernet Frame Size Calculation

➤ **Minimum Frame Size: 64 Bytes**

•Includes: 6 + 6 + 2 + 46 + 4 = 64 Bytes

•If payload < 46 bytes, **padding is added** to meet the minimum.

➤ **Maximum Frame Size: 1518 Bytes**

•Includes: 6 + 6 + 2 + 1500 + 4 = 1518 Bytes

•**Does not include preamble (7B) or SFD (1B)**

Ethernet Frame Format (IEEE 802.3)

Preamble:	SFD:	Destination	Source	Length:	Data:	FCS (CRC):
7B	1 B	MAC: 6B	MAC: 6B	2B	(46 – 1500) B	4B

MTU (Maximum Transmission Unit)

- **Definition**: Maximum amount of data the network layer can pass to the data link layer in one frame.
- **Value for Ethernet**: 1500 bytes
- This is **only the payload** part — does **not** include header or CRC.

➤ Ethernet frame size with MTU:

- Payload: 1500 B
- Header (Dest + Src + Type): 14 B
- FCS: 4 B

➡️ **Total**: 1518 bytes (max frame size)

➡️ With preamble + SFD: 1518 + 8 = 1526 bytes (**on the wire**, not counted by MAC layer)

Ethernet Frame Format (IEEE 802.3)

Preamble: 7B	SFD: 1 B	Destination MAC: 6B	Source MAC: 6B	Length: 2B	Data: (46 – 1500) B	FCS (CRC): 4B

Length Field (2 Bytes)

•Appears **after Source MAC** and **before Payload**.
•Indicates the **length of the payload (data)** in **bytes**, not the full frame.
•**Range**: 0 – 1500 (decimal)

Important:

•If this field's value is **≤ 1500**, it is treated as a **length field** → tells the **number of bytes in the payload**.

Not in Syllabus

•If it's **≥ 1536 (0x0600)**, it is interpreted as a **Type field** (Ethertype — used in Ethernet II).

Ethernet Frame Format (IEEE 802.3)

Preamble:	SFD:	Destination	Source	Length:	Data:	FCS (CRC):
7B	1 B	MAC: 6B	MAC: 6B	2B	(46 – 1500) B	4B

Type	First Byte Pattern	Purpose
Unicast	LSB = 0	Single destination
Multicast	LSB = 1	Group destination
Broadcast	All bits 1	All devices on LAN

Preamble: 7B	SFD: 1 B	Destination MAC: 6B	Source MAC: 6B	Length: 2B	Data: (46 – 1500) B	FCS (CRC): 4B

Question	Answer
Is preamble + SFD part of Ethernet frame size?	No (not counted in 64–1518 bytes)
Is preamble + SFD transmitted?	Yes, by PHY layer
Does MTU include headers?	No, MTU = payload only (max 1500B)
Max frame size (excluding preamble)?	1518 bytes
Max size “on the wire”?	1526 bytes (with 8B preamble/SFD)

Q1. What type of address is FF:FF:FF:FF:FF:FF?

A. Unicast
   B. Multicast
   C. Broadcast

Q2. What type of address is 01:00:5E:00:00:FB?

A. Unicast
   B. Multicast
   C. Broadcast

B. Q3. Is the MAC address 00:1A:2B:3C:4D:5E unicast, multicast, or broadcast?

Answer:

Q4. Classify the MAC address 33:33:00:00:00:16

Q5. Is the MAC address 02:AB:CD:EF:12:34 unicast or multicast?

Q6. What type of MAC address is FF:00:00:00:00:00?

Thank You