Question# 1.

Part (a)

Given:

- Round-trip time (RTT) = 100 ms = 0.1 s
- Packets sent per RTT = 5
- Packet size = 1500 bytes

Calculation:

- Data sent per RTT = 5×1500=7500**---------(1.5 marks)**
- Transmission rate = 7500/0.1 = 75000 bytes/s**------(1.5 marks)**

Part (b)

Similarities between OSI and TCP/IP reference models------ (1.5 marks)

1. **Layered architecture:**
   Both models use a layered approach to organize network communication. Each layer has a specific responsibility, which helps in modular design, easier troubleshooting, and standardization.
2. **Separation of services and protocols**
   In both models, each layer provides services to the layer above it while relying on the services of the layer below. This abstraction allows protocols to be developed and updated independently.

Differences between OSI and TCP/IP reference models ------- (1.5 marks)

1. **Number of layers:**
   - The **OSI model** has **7 layers** (Physical, Data Link, Network, Transport, Session, Presentation, Application).
   - The **TCP/IP model** typically has **4 layers** (Link, Internet, Transport, Application), where several OSI layers are combined.
2. **Model purpose and adoption:**
   - The **OSI model** is primarily a **conceptual and teaching model** developed by ISO and is not directly tied to specific protocols.
   - The **TCP/IP model** is a **practical, protocol-oriented model** developed around real-world protocols (such as IP, TCP, and UDP) and is widely implemented on the Internet.

Question# 2.

Part (a)

C=Blog2 (l +SNR)

$C=2\chi10^6(1+63)$

Maximum achievable rate= C= 12 Mbps**.-------(2.5 marks)**

$R=2B\log_2 L$

To find the number of signal levels LLL, set R=12R = 12R=12 Mbps (Shannon limit):

$12\times10^6=2\times2\times10^6\log_2 L$ **-------(2.5 marks)**

L = 8

Part (b)

Solution:

Four digital channels are multiplexed using time-division multiplexing (TDM) without synchronization bits. Two of the channels operate at a bit rate of 200 kbps each, while the remaining two channels operate at a bit rate of 150 kbps each. In every TDM frame, 4 bits are taken from each of the first two channels and 3 bits are taken from each of the remaining two channels.

(i) Size of one TDM frame

The size of a TDM frame is equal to the total number of bits collected from all channels in one frame. Since 4 bits are taken from each of the first two channels and 3 bits are taken from each of the other two channels, the frame size is calculated as follows:

4 + 4 + 3 + 3 = 14 bits

Therefore, the size of one TDM frame is **14 bits**.

(ii) Appropriate frame rate

The frame rate must be selected such that it can support the bit rates of all channels. For the 200 kbps channels, 4 bits are transmitted per frame, so the required frame rate is:

200,000 ÷ 4 = 50,000 frames per second

For the 150 kbps channels, 3 bits are transmitted per frame, so the required frame rate is:

150,000 ÷ 3 = 50,000 frames per second

Since all channels require the same frame rate, the appropriate frame rate is **50,000 frames per second**.

(iii) Duration of one frame

The duration of one frame is the reciprocal of the frame rate. It is calculated as:

1 ÷ 50,000 = 0.00002 seconds

This is equal to **20 microseconds (20 µs)**.
Hence, the duration of one TDM frame is **20 µs**.

(iv) Overall data rate of the multiplexed TDM signal

The overall data rate of the multiplexed signal is obtained by multiplying the frame size by the frame rate:

14 × 50,000 = 700,000 bits per second

This is equal to **700 kbps**.
The result also matches the sum of the individual channel data rates (200 + 200 + 150 + 150 = 700 kbps).

Question# 3.

Part (a)

Given:

Original message at the sender:
1001 1100 1010 0011

Received message at the receiver:
1001 1100 1010 0111

Word size: **4 bits**

Checksum Generation at the Transmitter

First, the 16-bit message is divided into 4-bit words:

- Word 1: 1001
- Word 2: 1100
- Word 3: 1010
- Word 4: 0011

These words are added using **one's complement arithmetic**.

Step-by-step addition:

1001 + 1100 = 1 0101
The carry is added back:
0101 + 1 = 0110

0110 + 1010 = 1 0000
Carry is added back:
0000 + 1 = 0001

0001 + 0011 = 0100

The final sum is **0100**.

Now, the **one's complement** of the sum is taken to generate the checksum:

One's complement of 0100 = **1011**

Therefore, the **checksum generated at the transmitter is 1011**, and the transmitted data consists of the original message followed by this checksum.

Checksum Verification at the Receiver

The receiver divides the received message into 4-bit words and includes the checksum:

- Word 1: 1001
- Word 2: 1100
- Word 3: 1010
- Word 4: 0111
- Checksum: 1011

Now, all words are added using one's complement arithmetic.

1001 + 1100 = 1 0101
Carry added back:
0101 + 1 = 0110

0110 + 1010 = 1 0000
Carry added back:
0000 + 1 = 0001

0001 + 0111 = 1000

1000 + 1011 = 1 0011
Carry added back:
0011 + 1 = 0100

The final sum obtained at the receiver is **0100**.

Error Analysis and Conclusion

According to the Internet checksum method:

- If the final sum after adding all words and the checksum is **all 1s**, the transmission is error free.
- If the final sum is **not all 1s**, an error is detected.

Since the final sum at the receiver is **0100**, which is **not equal to 1111**, this indicates that an error occurred during transmission.

Part (b)

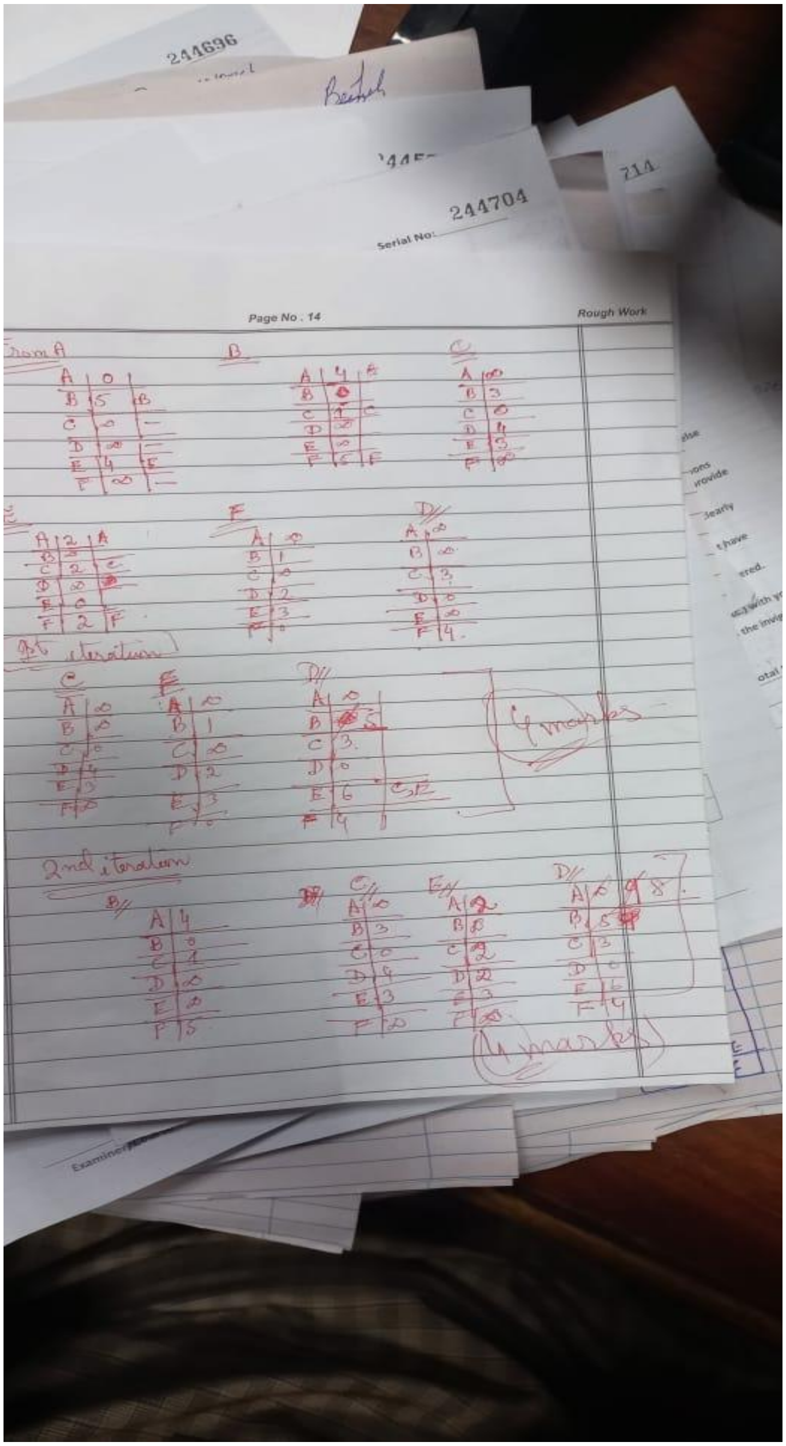
Given Received Data:

0110 0111 1100 1111 0111 1101

After bit-destuffing

0110 0111 1101 1110 1111 11

Question# 4 (a).



Part (b)

The binary representation of the given address is 11001101 00010000 00100101 00100I11. If we set 32- 28 rightmost bits to 0, we get 11001101 000100000100101 0010000 or **205.16.37.32.**

The binary representation of the given address is 11001101 0001000000100101 00100111. If we set 32- 28 rightmost bits to 1, we get 11001101 00010000 0010010100101111 or **205.16.37.47**

Question# 5.

Part (a)

a. Does not request any special action
b. Connection termination
c. Acknowledges a FIN and requests connection termination.

iv.

TCP Connection Termination Diagram

```pgsql
Client                                      Server
  |                                           |
  |  FIN ----------------------------->      |   (Step 1: Client initiates termination)
  |                                           |
  |  ACK <-----------------------------      |   (Step 2: Server acknowledges FIN)
  |                                           |
  |                                           |
  |                  FIN ------------->      |   (Step 3: Server initiates termination)
  |                                           |
  |  ACK <-----------------------------      |   (Step 4: Client acknowledges FIN)
  |                                           |
Connection Closed                      Connection Closed
```

Part (a)

1. Generic Domains (gTLDs)

These domains classify organizations by their purpose or type.

Examples:

- **.com** – Commercial organizations
- **.edu** – Educational institutions
- **.org** – Non-profit organizations
- **.gov** – Government organizations
- **.mil** – Military organizations

---

2. Country Code Domains (ccTLDs)

These domains represent specific countries or geographical regions and consist of two-letter codes.

Examples:

- **.pk** – Pakistan
- **.us** – United States
- **.uk** – United Kingdom
- **.in** – India

---

3. Inverse Domain (Reverse Domain)

This domain is used for **reverse DNS lookups**, where an IP address is mapped back to a domain name instead of mapping a name to an IP address.

Example:

- **in-addr.arpa** (for IPv4 addresses)

Labels in the Domain Name

For the DNS name:

`voyager.fhda.edu`

The domain name consists of **3 labels**, separated by dots:

1. **voyager** – Host or subdomain name

2. **fhda** – Organization or institutional domain
3. **edu** – Generic top-level domain (gTLD)

Total number of labels = 3