Chapter 2 Problem Solution:

Q2-1: First Principle of Protocol Layering for Bidirectional Communication

The **first principle** states that for communication to be **bidirectional**, **each layer must perform two opposite tasks:**

- **Sending**: The layer at the sender's side adds a **header** (encapsulation).
- **Receiving**: The corresponding layer at the receiver's side removes the **header** (decapsulation).

Q2-2: TCP/IP Layers Involved in a Link-Layer Switch

A **link-layer switch** operates at the **Data Link Layer (Layer 2)**, meaning it primarily deals with:

- **Physical Layer (Layer 1)** – It transmits data as electrical signals.
- **Data Link Layer (Layer 2)** – It uses **MAC addresses** to forward frames.

Q2-3: Layers Involved in a Router (with 3 Links)

A router connects **three different networks**, so it is involved in the following layers:

- **Physical Layer (Layer 1)** – Connects to physical media.
- **Data Link Layer (Layer 2)** – Processes MAC addresses for each network.
- **Network Layer (Layer 3)** – Routes packets based on IP addresses.

Each of the **three links** requires a **separate Layer 2 entity (MAC address and frame format)** but shares a **single Layer 3 entity (IP forwarding).**

Q2-4: Identical Objects in the Application Layer

At the **Application Layer**, the **message (data)** remains identical at both the sender and receiver sites. This is because the **application layer protocols (HTTP, FTP, etc.)** define how the message should be interpreted.

Q2-5: Units of Data Sent or Received at Different Layers

- **Application Layer → Message**
- **Transport Layer → Segment (TCP) / Datagram (UDP)**
- **Network Layer → Packet / Datagram**
- **Data Link Layer → Frame**
- **Physical Layer → Bits**

Q2-6: Which Data Unit is Encapsulated in a Frame?

A **Packet (from the Network Layer)** is encapsulated inside a **Frame (Data Link Layer).**

Q2-7: Which Data Unit is Decapsulated from a User Datagram?

A **Segment (from the Transport Layer)** is decapsulated from a **User Datagram (UDP).**

Q2-8: Which Data Unit Contains an Application-Layer Message + Layer 4 Header?

A **Segment (TCP) or Datagram (UDP)** consists of:

- **Application-layer message (data)**
- **Transport-layer header (TCP/UDP header)**

Q2-9: Application-Layer Protocols

Some common **Application Layer** protocols include:

- **HTTP (Hypertext Transfer Protocol)** – Web browsing
- **SMTP (Simple Mail Transfer Protocol)** – Email sending
- **FTP (File Transfer Protocol)** – File transfers
- **DNS (Domain Name System)** – Resolving domain names to IP addresses
- **POP3 / IMAP** – Email retrieval

Q2-10: Minimum Transport Layer Header Size (TCP/IP Protocol Suite)

Since **a port number is 16 bits (2 bytes)**, and a **TCP/UDP header contains two port numbers (source + destination)**, the minimum transport-layer header size is:

2+2=4 bytes(for UDP)2 + 2 = 4 \text{ bytes} \quad (\text{for UDP})2+2=4 bytes(for UDP)

For TCP, the minimum header size is **20 bytes** due to additional fields like sequence numbers.

Q2-11: Types of Addresses Used in Each Layer

- **Physical Layer → Bit transmission** (No addresses, only raw signals)
- **Data Link Layer → MAC Address (48-bit address, unique to each device)**
- **Network Layer → IP Address (Logical address for routing across networks)**
- **Transport Layer → Port Number (Used to identify application services like HTTP, FTP, etc.)**

Q2-12: Transport Layer Multiplexing & Demultiplexing

Multiplexing → The transport layer allows multiple application messages (from different applications) to share a single connection (e.g., TCP sockets).
Demultiplexing → It ensures each received message is delivered to the correct application based on **port numbers**.

- **However, it does not combine multiple messages into one packet.**
- Each application message is **encapsulated individually** within a transport-layer segment.

Q2-13: Why No Multiplexing/Demultiplexing at the Application Layer?

- **Multiplexing/Demultiplexing happens at the Transport Layer** because different applications share the same network connection.
- **The Application Layer does not handle transport concerns**; it only focuses on the actual data being sent.

Q2-14: Do We Need a Link-Layer Switch to Connect Two Hosts?

No.

- Two hosts can be directly connected using a **crossover cable (for wired networks)** or a **direct Wi-Fi connection**.
- A **switch is needed** only if **more than two hosts** need to communicate.

Q2-15: Do We Need a Router if There is a Single Path Between Two Hosts?

No.

- A **router is only required** if the two hosts are in **different networks**.
- If they are in the **same network**, they can communicate directly via **switches or hubs** without a router.

Problems:

P2-1: Services Provided by Layer 1 (Physical Layer) to Layer 2 (Data Link Layer)

a. At Maria's Site

- **Layer 1 (Physical Layer)** transmits **bits** over the network medium.
- **Service to Layer 2**: It provides a way to send and receive **frames** as signals.

b. At Ann's Site

- **Layer 1 (Physical Layer)** receives **signals** and converts them back into **bits**.
- **Service to Layer 2**: It delivers received **frames** to the data link layer.

P2-2: Services Provided by Layer 2 (Data Link Layer) to Layer 3 (Network Layer)

a. At Maria’s Site

- **Layer 2 (Data Link Layer)** packages data into **frames** and adds **MAC addresses** for delivery.
- **Service to Layer 3**: It ensures that packets (from the network layer) are correctly placed into frames for transmission.

b. At Ann’s Site

- **Layer 2 (Data Link Layer)** extracts **packets** from the received frames.
- **Service to Layer 3**: It passes these packets to the **network layer**.

P2-3: Number of Hosts in 2020 Given 20% Growth Per Year

N=500 million×(1.2)10N = 500 \text{ million} \times (1.2)^{10}N=500 million×(1.2)10 N=500×6.191736N = 500 \times 6.191736N=500×6.191736 N≈3.1 billion hostsN \approx 3.1 \text{ billion hosts}N≈3.1 billion hosts

---

P2-4: Efficiency of a 5-Layer System with 10-byte Headers

- **Application data** = 100 bytes
- **Total header size** = $10 \times 5 = 50$ bytes
- **Total transmitted data** = $100 + 50 = 150$ bytes
- **Efficiency** = (Application Data / Total Data)

\frac{100}{150} = 0.6667 \text{ (or 66.67%)}

P2-5: Advantages and Disadvantages of Sending Large Packets in TCP/IP

Advantages:

- Fewer packets = **Lower overhead** (less protocol header overhead).
- **Better throughput** (fewer acknowledgments required).

Disadvantages:

- **More retransmission delay** if errors occur.
- **Fragmentation** may be needed if the packet exceeds network MTU.

P2-6: Matching Terms to TCP/IP Layers

- **a. Route determination → Network Layer** (Layer 3)
- **b. Connection to transmission media → Physical Layer** (Layer 1)
- **c. Providing services for the end user → Application Layer** (Layer 5)

P2-7: Matching More Terms to TCP/IP Layers

- **a. Creating user datagrams → Transport Layer** (Layer 4)
- **b. Handling frames between adjacent nodes → Data Link Layer** (Layer 2)
- **c. Transforming bits to electromagnetic signals → Physical Layer** (Layer 1)

P2-8: How Does IP Know Whether to Deliver to TCP or UDP?

- The **IP header contains a "Protocol" field** that indicates the transport layer protocol (e.g., **6 for TCP, 17 for UDP**).
- The IP layer uses this field to direct packets to the correct transport-layer protocol.

P2-9: Multiplexing & Demultiplexing at Data-Link Layer

- If a private network has three different **Data-Link Layer** protocols (L1, L2, L3):
  - **Source Node:** Demultiplexes packets based on destination **data-link layer protocol**.
  - **Destination Node:** Multiplexes incoming packets based on **source link layer protocol**.

Diagram Adjustment:
In Figure 2.10, **three separate Data Link Layers (L1, L2, L3)** must be shown at both sender and receiver ends.

P2-10: Adding an Encryption Layer to TCP/IP

- If encryption is required at the **Application Layer**, a new **Security Layer** should be added **above Transport Layer** but **below Application Layer**.

New Layered Model:

1. **Application Layer**
2. **Security Layer (Encryption/Decryption)**
3. **Transport Layer**
4. **Network Layer**
5. **Data Link Layer**
6. **Physical Layer**

P2-11: Protocol Layering in Air Travel

1. **Baggage Checking / Claiming (Physical Layer)**
2. **Boarding / Unboarding (Data Link Layer)**
3. **Takeoff / Landing (Network Layer)**

4. **Flight Route Determination (Transport Layer)**
5. **Passenger Services & Tickets (Application Layer)**

P2-12: Where Should a New Presentation Layer Be in TCP/IP?

- A **Presentation Layer** (for encoding, compression, encryption) should be added **between the Application Layer and Transport Layer**.
- **New TCP/IP Model:**
    - Application Layer
    - **Presentation Layer** *(New! Encoding, compression, encryption)*
    - Transport Layer
    - Network Layer
    - Data Link Layer
    - Physical Layer

P2-13: Changing LAN Technology in TCP/IP

- **Only the Data Link Layer & Physical Layer change.**
- **Network, Transport, and Application Layers remain the same.**

P2-14: Can an Application-Layer Protocol Written for UDP Work with TCP?

No, unless modified.

- **UDP is connectionless**, while **TCP is connection-oriented**.
- If an app protocol was designed for UDP, it would **not expect TCP's features** (like acknowledgments, congestion control).
- **Modification would be required** to handle **TCP's flow control & connection setup**.

P2-15: Data Flow in TCP/IP from West Coast to East Coast

1. **Application Layer** (Message is created).
2. **Transport Layer** (Message is **divided into segments**).
3. **Network Layer** (Segments **converted into packets with IP addresses**).
4. **Data Link Layer** (Packets **framed and sent via physical network**).
5. **Physical Layer** (Bits **transmitted over cable/wireless**).
6. **Router forwards packets from the West Coast to East Coast** via the Internet.
7. **Destination Host reverses the process** to extract the message.