Here is a comprehensive question bank for your internal test preparation, based on the syllabus (Modules 3 & 4), the PowerPoint slides (ch19-ch22), and the previous year's questions you provided.

Question Bank: Computer Networks (18CS53)

Module 3: The Network Layer

This section covers routing, congestion, IP addressing (IPv4/IPv6), and network layer protocols like ICMP, OSPF, and BGP.

Q1: What are the two main design issues for the Network Layer?

A: The two primary design issues are:

1. **Service provided to the Transport Layer:** The network layer can offer either a connection-oriented service (like a virtual circuit) or a connectionless service (like datagrams). The internet uses a connectionless service.
2. **Internal Network Structure:** This concerns how packets are routed within the subnet. It involves choosing routing algorithms, handling congestion, and managing quality of service.

Q2: Differentiate between Distance Vector and Link State routing algorithms.

A:

| Feature | Distance Vector (e.g., RIP) | Link State (e.g., OSPF) |

| :--- | :--- | :--- |

| Knowledge | Knows only its direct neighbors' routing tables. | Knows the entire network topology (all routers and links). |

| Information Shared | Shares its entire routing table. | Shares only the state of its own links (Link State Packet - LSP). |

| Sharing | Shares information only with its direct neighbors. | Floods its LSP to all other routers in the network. |

| Convergence | Slow. Prone to the "Count to Infinity" problem. | Fast. Each router computes its own routes. |

| Complexity | Simpler to implement. | More complex (requires Dijkstra's algorithm). |

Q3: What is congestion? Differentiate between the Leaky Bucket and Token Bucket algorithms.

A: Congestion is a state in a network where the load (number of packets) exceeds the network's capacity (link bandwidth and router processing speed), leading to high packet delay and loss.

- **Leaky Bucket:** This algorithm enforces a fixed output rate, regardless of the burstiness of the input. It's like a bucket with a small hole at the bottom. If the bucket is full, new packets are discarded. It smooths out traffic but doesn't allow for saving "credits" during idle times.
- **Token Bucket:** This algorithm allows for bursts. A "bucket" holds tokens, which are generated at a fixed rate. To send a packet, the router must consume a token. If the bucket is full of tokens, the router can send a large burst of packets at once. This allows the network to use idle capacity more effectively.

Q4: Draw and explain the IPv4 datagram header.

A: The IPv4 header is 20 bytes long (without options).

- **Version (4 bits):** Set to 4 for IPv4.
- **HLEN (4 bits):** Header Length, in 32-bit words (e.g., a value of 5 means $5 \times 4 = 20$ bytes).
- **Type of Service (8 bits):** Specifies Quality of Service (QoS) parameters like priority.
- **Total Length (16 bits):** Total length of the entire datagram (header + data) in bytes.
- **Identification (16 bits):** Used to identify fragments of a single original datagram.
- **Flags (3 bits):** Control fragmentation (e.g., "Don't Fragment", "More Fragments").
- **Fragment Offset (13 bits):** Specifies the position of the fragment in the original datagram.
- **Time to Live (TTL) (8 bits):** A hop limit. Decremented by each router. When it reaches 0, the packet is discarded (prevents infinite loops).
- **Protocol (8 bits):** Identifies the transport layer protocol (e.g., 6 for TCP, 17 for UDP).
- **Header Checksum (16 bits):** Error-checking for the header only.
- **Source IP Address (32 bits):** The sender's IP address.
- **Destination IP Address (32 bits):** The receiver's IP address.
- **Options (Variable):** Used for testing, debugging, or security (rarely used).

Q5: What is subnetting? An organization is given the block 172.16.0.0/16 and needs to create 8 subnets. Find the new subnet mask and the address range for the first two subnets.

A: Subnetting is the process of dividing a single large network block into multiple smaller, more manageable networks called subnets.

1. **Original Mask:** /16 (255.255.0.0). This leaves 16 bits for hosts.
2. **Required Subnets:** 8.
3. **Bits to Borrow:** To get 8 subnets, we need to borrow n bits from the host part such that $2^n \ge 8$. Here, $n=3$.
4. **New Subnet Mask:** We add these 3 bits to the network part. The new mask is $16 + 3 = 19$.

    - **In CIDR:** /19
    - **In dot-decimal:** 255.255.224.0
5. **Address Ranges:**
    - **Subnet 1:** 172.16.0.0/19 (Network address: 172.16.0.0, Broadcast: 172.16.31.255)
    - **Subnet 2:** 172.16.32.0/19 (Network address: 172.16.32.0, Broadcast: 172.16.63.255)

Q6: Explain the rules for IPv6 address compression and provide an example.

A: IPv6 addresses are 128 bits long, written as 8 blocks of 16 bits (hextets). Compression makes them easier to write.

- **Rule 1 (Omit Leading Zeros):** Within any 16-bit block, leading zeros can be omitted.
    - Example: `0040` becomes `40`, `0000` becomes `0`.
- **Rule 2 (Double Colon):** A single, contiguous sequence of all-zero blocks can be replaced with a double colon (`::`). This can only be done *once* per address.
    - Example: `FE80:0000:0000:0000:0202:B3FF:FE1E:8329`
    - Becomes: `FE80::0202:B3FF:FE1E:8329`
    - Example 2: `2001:0DB8:0000:0001:0000:0000:0000:ABCD`
    - Becomes: `2001:DB8:0:1::ABCD` (Leading zeros omitted, and the longest zero-block compressed).

Q7: What is the purpose of ICMP? List three common ICMP message types.

A: The Internet Control Message Protocol (ICMP) is a network layer protocol used by network devices to send error messages and operational information. It is essentially a feedback mechanism for IP.

Common message types include:

1. **Echo Request / Echo Reply:** Used by the `ping` utility to test reachability.
2. **Destination Unreachable:** Sent by a router when it cannot deliver a packet (e.g., host unknown, port unreachable).
3. **Time Exceeded:** Sent when a packet's TTL reaches zero, or when a fragment reassembly timer expires.

Module 4: The Transport Layer

This section covers the transport service, UDP, and TCP (including its header, connection management, and congestion control).

Q1: What is the primary role of the Transport Layer? How does it achieve "process-to-process" communication?

A: The primary role of the Transport Layer is to provide logical communication between processes running on different hosts. While the Network Layer handles host-to-host delivery, the Transport Layer handles delivering data to the correct application on that host.

It achieves this using port numbers. Each application is assigned a unique port number (e.g., HTTP is port 80, FTP is port 21). The Transport Layer header includes source and destination port numbers, allowing the receiving host's operating system to demultiplex the incoming data and pass it to the correct application process.

Q2: Draw the UDP segment header and explain its fields. Why would an application use UDP?

A: The UDP header is very simple (8 bytes).

- **Source Port (16 bits):** Port number of the sending application.
- **Destination Port (16 bits):** Port number of the receiving application.
- **Length (16 bits):** The length of the entire UDP segment (header + data) in bytes.
- **Checksum (16 bits):** Optional error-checking for the header and data.

Applications use UDP when **speed is more important than reliability**. It is connectionless and has low overhead.

- **Use cases:** DNS, DHCP, streaming video/audio, online gaming.

Q3: Differentiate between TCP and UDP.

A:

| Feature | TCP (Transmission Control Protocol) | UDP (User Datagram Protocol) |

| :--- | :--- | :--- |

| Connection | Connection-oriented (uses 3-way handshake). | Connectionless. |

| Reliability | Reliable (uses acknowledgments, retransmissions). | Unreliable (best-effort delivery). |

| Ordering | Guarantees in-order delivery of data. | Does not guarantee order. |

| Flow Control | Yes (using sliding window). | No. |

| Congestion Control | Yes (e.g., Slow Start, Congestion Avoidance). | No. |

| Header Size | 20 bytes (plus options). | 8 bytes. |

| Speed | Slower, due to overhead. | Faster, lightweight. |

| Use Cases | HTTP, FTP, SMTP (email), SSH. | DNS, DHCP, VoIP, online gaming. |

Q4: Explain the TCP 3-way handshake for connection establishment.

A: TCP uses a 3-way handshake to establish a reliable connection.

1. **SYN:** The client (initiator) sends a TCP segment with the **SYN** (Synchronize) flag set to 1 and a random initial sequence number (e.g., `seq=x`).
2. **SYN-ACK:** The server receives the SYN, allocates resources, and sends back a segment with both the **SYN** flag set to 1 and the **ACK** (Acknowledgment) flag set to 1. It includes its own random initial sequence number (e.g., `seq=y`) and an acknowledgment number (e.g., `ack=x+1`) to confirm receipt of the client's SYN.
3. **ACK:** The client receives the SYN-ACK and sends a final segment with the **ACK** flag set to 1. The sequence number is $x+1$ and the acknowledgment number is $y+1$.

At this point, the connection is established, and data transfer can begin.

Q5: Draw and explain the key fields of the TCP segment header.

A: The TCP header is 20 bytes long (without options).

- **Source Port (16 bits):** Port number of the sending application.
- **Destination Port (16 bits):** Port number of the receiving application.
- **Sequence Number (32 bits):** The sequence number of the first data byte in this segment.
- **Acknowledgment Number (32 bits):** If the ACK flag is set, this field contains the value of the next sequence number the sender is expecting to receive.
- **Header Length (HLEN) (4 bits):** The size of the TCP header in 32-bit words.
- **Flags (6 bits):**
    - **URG:** Urgent pointer field is valid.
    - **ACK:** Acknowledgment field is valid.
    - **PSH:** Push function (receiver should pass data to app immediately).
    - **RST:** Reset the connection.
    - **SYN:** Synchronize sequence numbers (used to initiate a connection).
    - **FIN:** No more data from sender (used to terminate a connection).
- **Window Size (16 bits):** Used for flow control. Specifies the number of bytes the sender is willing to receive.
- **Checksum (16 bits):** Error-checking for the header, data, and a pseudo-header.
- **Urgent Pointer (16 bits):** Points to urgent data (if URG flag is set).

Questions from Previous Internal Test (PYQ)

These are the questions from the images you provided, with answers.

Q1. a. Explain Distance Vector Routing algorithm with an example. (6M)

A: See "Module 3, Q2" for the core concept.

Explanation: Distance Vector Routing is an iterative, asynchronous, and distributed routing algorithm.

- **Distributed:** Each router maintains its own routing table, which it computes based on information from its neighbors.
- **Iterative:** The process continues until no more information can be exchanged.
- **Asynchronous:** It does not require all routers to operate in lockstep.

Each router maintains a **distance vector** (routing table) containing the "distance" (cost, usually hops) to every other router in the network. It periodically shares this vector with its **direct neighbors only**.

When a router X receives a distance vector from its neighbor Y, X updates its own table using the Bellman-Ford equation:

$Cost(X, Z) = \min_{\text{all neighbors } Y} [Cost(X, Y) + Cost(Y, Z)]$

This means the cost for X to get to Z is the minimum of (cost to get to neighbor Y + Y's advertised cost to get to Z).

Example:

- Consider routers A, B, C. A-B cost is 2, B-C cost is 1, A-C cost is 5.
- **B's Table:** {A: 2, B: 0, C: 1}
- **C's Table:** {A: 5, B: 1, C: 0}
- A's initial table is {A: 0, B: 2, C: 5}.
- A receives B's table. A recalculates its path to C:
    - Path via B: Cost(A,B) + Cost(B,C) = 2 + 1 = 3.
    - This is less than its current cost of 5.
- A updates its table: {A: 0, B: 2, C: 3} (via B).
  This process repeats until all tables stabilize. A major drawback is the Count to Infinity problem, where good news (a link coming up) travels fast, but bad news (a link failing) travels very slowly.

Q1. b. What is congestion? Explain leaky bucket algorithm. (4M)

A: See "Module 3, Q3" for the definitions.

Q2. a. Explain the concept of IP addressing and subnetting with an example. (6M)

A: See "Module 3, Q5" for a detailed explanation and example of IP addressing and subnetting.

Q2. b. Differentiate between IPv4 and IPv6. (4M)

A:

| Feature | IPv4 (Internet Protocol v4) | IPv6 (Internet Protocol v6) |

| :--- | :--- | :--- |

| Address Size | 32 bits | 128 bits |

| Address Notation | Dotted-decimal (e.g., 192.168.1.1) | Hexadecimal (e.g., 2001:db8::1) |

| Address Space | ~ 4.3 billion addresses (exhausted) | $\sim 3.4 \times 10^{38}$ addresses (vast) |

| Header | 20-byte base header with options | 40-byte fixed header (options are extension headers) |

| Configuration | Manual or via DHCP | Stateless Address Autoconfiguration (SLAAC) and DHCPv6 |

| Security | Optional (IPsec) | Required (IPsec is built-in) |

| Fragmentation | Done by sending host and routers | Done only by the sending host |

Q3. a. Explain the header format of IPv4. (7M)

A: See "Module 3, Q4" for a diagram and full explanation of the fields.

Q3. b. What is BGP? Explain the BGP protocol. (3M)

A: BGP (Border Gateway Protocol) is the standardized inter-domain routing protocol used on the internet. It manages how packets are routed between different Autonomous Systems (AS), which are large networks (e.g., an ISP, a university, a large corporation).

BGP is a **Path Vector** protocol. Unlike OSPF or RIP which focus on the lowest "cost", BGP makes routing decisions based on **policies**, AS-path length, and other attributes.

- When a BGP router advertises a route to a neighbor, it includes the entire **path of AS numbers** it took to reach that destination.
- This path information is used to prevent routing loops (a router discards any route that already contains its own AS number).
- Policies (which are manually configured) are the most important factor, allowing an AS to decide *how* it wants to route traffic (e.g., "don't route traffic for my competitor," or "prefer this cheaper link").

Q4. a. Explain the elements of transport protocol. (6M)

A: Key elements (or services) of a transport protocol include:

1. **Addressing (Port Numbers):** Provides process-to-process communication by using port numbers to identify specific applications on a host.
2. **Connection Establishment:** For connection-oriented services (like TCP), this involves a handshake (e.g., 3-way handshake) to establish a logical connection before data is sent.
3. **Connection Release:** A mechanism to gracefully terminate the connection, ensuring all data is delivered (e.g., TCP's 4-way handshake with FIN flags).
4. **Flow Control:** Prevents a fast sender from overwhelming a slow receiver. TCP uses a "sliding window" mechanism where the receiver advertises its available buffer space (the "window size").
5. **Error Control:** Ensures data is delivered reliably. This is done using checksums (to detect corruption), acknowledgments (to confirm receipt), and retransmissions (to resend lost or corrupt packets).
6. **Segmentation and Reassembly:** Breaks large messages from the application layer into smaller segments for transmission and reassembles them at the destination.
7. **Congestion Control:** Manages network congestion (as opposed to receiver flow control). TCP uses algorithms like Slow Start to probe for available bandwidth and reduce its sending rate when congestion is detected.

Q4. b. With a neat diagram, explain the TCP segment header. (4M)

A: See "Module 4, Q5" for a diagram and full explanation of the fields.

Q5. a. Explain the working of OSPF protocol. (5M)

A: OSPF (Open Shortest Path First) is an intra-domain Link State routing protocol. It is used within a single Autonomous System (AS).

The working of OSPF can be summarized in 5 steps:

1. **Discover Neighbors:** Each router sends "Hello" packets on all its links to discover its directly connected neighbors and establish adjacencies.
2. **Measure Link Cost:** Each router measures the cost (e.g., based on bandwidth) to its neighbors.
3. **Build Link State Packet (LSP):** Each router creates an LSP that describes its own links and their costs (e.g., "I am router A. I can reach B with cost 2 and C with cost 5").
4. **Flood LSPs:** Each router floods its LSP to *all other routers* within the same area. This ensures every router in the area builds an identical copy of the network topology.
5. **Compute Shortest Path:** Using the complete topology map, each router independently runs **Dijkstra's (Shortest Path First) algorithm** to compute the shortest path from itself to every other router. The results are stored in its routing table.

Q5. b. Explain the working of ICMP. (5M)

A: See "Module 3, Q7" for an explanation of ICMP's purpose and message types.

Working: ICMP is an auxiliary protocol to IP. It does not carry application data but rather control messages. When an IP datagram encounters an error (like an invalid destination, or its TTL expires), the router or host that discovers the error generates an ICMP message. This message is encapsulated within a new IP datagram and sent back to the original source of the problematic datagram.

- **Example (Ping):** A host sends an ICMP Echo Request to a destination. The destination host, upon receiving it, replies with an ICMP Echo Reply, confirming reachability.
- **Example (Error):** A host sends a UDP packet to a port that is not open. The destination host sends back an ICMP Destination Unreachable (Port Unreachable) message.

Q6. a. Explain the three-way handshake mechanism in TCP. (5M)

A: See "Module 4, Q4" for a diagram and full explanation.

Q6. b. Explain the UDP segment header. (5M)

A: See "Module 4, Q2" for a diagram and explanation.

Q7. a. What is quality of service (QoS)? Explain two techniques for achieving good QoS. (5M)

A: Quality of Service (QoS) refers to the ability of a network to provide better or special service to selected network traffic. It is a set of techniques used to manage network resources (bandwidth, delay, jitter, packet loss) to meet the needs of specific applications. For example, a video conference needs low delay and jitter, while a file transfer can tolerate more delay.

Two techniques for achieving good QoS are:

1. **Traffic Shaping (e.g., Leaky Bucket):** This technique smooths out traffic bursts by buffering packets and sending them out at a fixed, average rate. This prevents a bursty application from causing congestion.
2. **Integrated Services (IntServ):** A resource reservation model. Applications use a protocol like **RSVP (Resource Reservation Protocol)** to request a specific amount of bandwidth and a bound on delay for a particular data flow. Routers along the path then reserve these resources, creating a "virtual circuit" for that flow.

Q7. b. Write a short note on flooding as a routing algorithm. (5M)

A: Flooding is a simple routing algorithm where every incoming packet is sent out on every outgoing link except the one it arrived on.

- **Advantages:**

    1. **Robust:** It is guaranteed to find the shortest path (and every other path) because it explores all possibilities.
    2. **Simple:** No complex tables or calculations are needed.
- **Disadvantages:**
    1. **Generates vast traffic:** Creates an exponential number of duplicate packets, which can congest the network.
    2. **Inefficient:** Wastes bandwidth.
- **Controls:** To be practical, flooding must be controlled. This is done using:
    1. **Hop Count (TTL):** A packet is discarded after a certain number of hops.
    2. **Sequence Numbers:** Routers keep track of (source, sequence_number) pairs and discard any duplicate packets they have already seen.
- **Use:** Pure flooding is rarely used for data. However, a controlled version of flooding is the core mechanism used by **Link State protocols (like OSPF)** to distribute Link State Packets (LSPs) to all routers.

Q8. a. Explain link state routing algorithm. (6M)

A: See "Q5. a. Explain the working of OSPF protocol." OSPF is a prime example of a Link State routing algorithm. The 5 steps listed there (Discover, Measure, Build, Flood, Compute) are the core explanation of Link State routing.

Q8. b. Differentiate between TCP and UDP. (4M)

A: See "Module 4, Q3" for a complete table differentiation.

Here is a comprehensive question bank for your internal test preparation, based on the syllabus (Modules 3 & 4), the PowerPoint slides (ch19-ch22), and the previous year's questions you provided.

Question Bank: Computer Networks (18CS53)

Module 3: The Network Layer

This section covers routing, congestion, IP addressing (IPv4/IPv6), and network layer protocols like ICMP, OSPF, and BGP.

Q1: What are the two main design issues for the Network Layer?

A: The two primary design issues are:

1. **Service provided to the Transport Layer:** The network layer can offer either a connection-oriented service (like a virtual circuit) or a connectionless service (like datagrams). The internet uses a connectionless service.

2. **Internal Network Structure:** This concerns how packets are routed within the subnet. It involves choosing routing algorithms, handling congestion, and managing quality of service.

Q2: Differentiate between Distance Vector and Link State routing algorithms.

A:

| Feature | Distance Vector (e.g., RIP) | Link State (e.g., OSPF) |

| :--- | :--- | :--- |

| Knowledge | Knows only its direct neighbors' routing tables. | Knows the entire network topology (all routers and links). |

| Information Shared | Shares its entire routing table. | Shares only the state of its own links (Link State Packet - LSP). |

| Sharing | Shares information only with its direct neighbors. | Floods its LSP to all other routers in the network. |

| Convergence | Slow. Prone to the "Count to Infinity" problem. | Fast. Each router computes its own routes. |

| Complexity | Simpler to implement. | More complex (requires Dijkstra's algorithm). |

Q3: What is congestion? Differentiate between the Leaky Bucket and Token Bucket algorithms.

A: Congestion is a state in a network where the load (number of packets) exceeds the network's capacity (link bandwidth and router processing speed), leading to high packet delay and loss.

- **Leaky Bucket:** This algorithm enforces a fixed output rate, regardless of the burstiness of the input. It's like a bucket with a small hole at the bottom. If the bucket is full, new packets are discarded. It smooths out traffic but doesn't allow for saving "credits" during idle times.
- **Token Bucket:** This algorithm allows for bursts. A "bucket" holds tokens, which are generated at a fixed rate. To send a packet, the router must consume a token. If the bucket is full of tokens, the router can send a large burst of packets at once. This allows the network to use idle capacity more effectively.

Q4: Draw and explain the IPv4 datagram header.

A: The IPv4 header is 20 bytes long (without options).

- **Version (4 bits):** Set to 4 for IPv4.

- **HLEN (4 bits):** Header Length, in 32-bit words (e.g., a value of 5 means $5 \times 4 = 20$ bytes).
- **Type of Service (8 bits):** Specifies Quality of Service (QoS) parameters like priority.
- **Total Length (16 bits):** Total length of the entire datagram (header + data) in bytes.
- **Identification (16 bits):** Used to identify fragments of a single original datagram.
- **Flags (3 bits):** Control fragmentation (e.g., "Don't Fragment", "More Fragments").
- **Fragment Offset (13 bits):** Specifies the position of the fragment in the original datagram.
- **Time to Live (TTL) (8 bits):** A hop limit. Decremented by each router. When it reaches 0, the packet is discarded (prevents infinite loops).
- **Protocol (8 bits):** Identifies the transport layer protocol (e.g., 6 for TCP, 17 for UDP).
- **Header Checksum (16 bits):** Error-checking for the header only.
- **Source IP Address (32 bits):** The sender's IP address.
- **Destination IP Address (32 bits):** The receiver's IP address.
- **Options (Variable):** Used for testing, debugging, or security (rarely used).

Q5: What is subnetting? An organization is given the block 172.16.0.0/16 and needs to create 8 subnets. Find the new subnet mask and the address range for the first two subnets.

A: Subnetting is the process of dividing a single large network block into multiple smaller, more manageable networks called subnets.

1. **Original Mask:** /16 (255.255.0.0). This leaves 16 bits for hosts.
2. **Required Subnets:** 8.
3. **Bits to Borrow:** To get 8 subnets, we need to borrow n bits from the host part such that $2^n \ge 8$. Here, $n=3$.
4. **New Subnet Mask:** We add these 3 bits to the network part. The new mask is $16 + 3 = 19$.
    - **In CIDR:** /19
    - **In dot-decimal:** 255.255.224.0
5. **Address Ranges:**
    - **Subnet 1:** 172.16.0.0/19 (Network address: 172.16.0.0, Broadcast: 172.16.31.255)
    - **Subnet 2:** 172.16.32.0/19 (Network address: 172.16.32.0, Broadcast: 172.16.63.255)

Q6: Explain the rules for IPv6 address compression and provide an example.

A: IPv6 addresses are 128 bits long, written as 8 blocks of 16 bits (hextets). Compression makes them easier to write.

- **Rule 1 (Omit Leading Zeros):** Within any 16-bit block, leading zeros can be omitted.
    - Example: `0040` becomes `40`, `0000` becomes `0`.

- **Rule 2 (Double Colon):** A single, contiguous sequence of all-zero blocks can be replaced with a double colon (::). This can only be done *once* per address.
    - Example: FE80:0000:0000:0000:0202:B3FF:FE1E:8329
    - Becomes: FE80::0202:B3FF:FE1E:8329
    - Example 2: 2001:0DB8:0000:0001:0000:0000:0000:ABCD
    - Becomes: 2001:DB8:0:1::ABCD (Leading zeros omitted, and the longest zero-block compressed).

Q7: What is the purpose of ICMP? List three common ICMP message types.

A: The Internet Control Message Protocol (ICMP) is a network layer protocol used by network devices to send error messages and operational information. It is essentially a feedback mechanism for IP.

Common message types include:

1. **Echo Request / Echo Reply:** Used by the ping utility to test reachability.
2. **Destination Unreachable:** Sent by a router when it cannot deliver a packet (e.g., host unknown, port unreachable).
3. **Time Exceeded:** Sent when a packet's TTL reaches zero, or when a fragment reassembly timer expires.

Module 4: The Transport Layer

This section covers the transport service, UDP, and TCP (including its header, connection management, and congestion control).

Q1: What is the primary role of the Transport Layer? How does it achieve "process-to-process" communication?

A: The primary role of the Transport Layer is to provide logical communication between processes running on different hosts. While the Network Layer handles host-to-host delivery, the Transport Layer handles delivering data to the correct application on that host.

It achieves this using port numbers. Each application is assigned a unique port number (e.g., HTTP is port 80, FTP is port 21). The Transport Layer header includes source and destination port numbers, allowing the receiving host's operating system to demultiplex the incoming data and pass it to the correct application process.

Q2: Draw the UDP segment header and explain its fields. Why would an application use UDP?

A: The UDP header is very simple (8 bytes).

- **Source Port (16 bits):** Port number of the sending application.
- **Destination Port (16 bits):** Port number of the receiving application.
- **Length (16 bits):** The length of the entire UDP segment (header + data) in bytes.
- **Checksum (16 bits):** Optional error-checking for the header and data.

Applications use UDP when **speed is more important than reliability**. It is connectionless and has low overhead.

- **Use cases:** DNS, DHCP, streaming video/audio, online gaming.

Q3: Differentiate between TCP and UDP.

A:

| Feature | TCP (Transmission Control Protocol) | UDP (User Datagram Protocol) |

| :--- | :--- | :--- |

| Connection | Connection-oriented (uses 3-way handshake). | Connectionless. |

| Reliability | Reliable (uses acknowledgments, retransmissions). | Unreliable (best-effort delivery). |

| Ordering | Guarantees in-order delivery of data. | Does not guarantee order. |

| Flow Control | Yes (using sliding window). | No. |

| Congestion Control | Yes (e.g., Slow Start, Congestion Avoidance). | No. |

| Header Size | 20 bytes (plus options). | 8 bytes. |

| Speed | Slower, due to overhead. | Faster, lightweight. |

| Use Cases | HTTP, FTP, SMTP (email), SSH. | DNS, DHCP, VoIP, online gaming. |

Q4: Explain the TCP 3-way handshake for connection establishment.

A: TCP uses a 3-way handshake to establish a reliable connection.

1. **SYN:** The client (initiator) sends a TCP segment with the **SYN** (Synchronize) flag set to 1 and a random initial sequence number (e.g., `seq=x`).
2. **SYN-ACK:** The server receives the SYN, allocates resources, and sends back a segment with both the **SYN** flag set to 1 and the **ACK** (Acknowledgment) flag set to 1. It includes its own random initial sequence number (e.g., `seq=y`) and an acknowledgment number (e.g., `ack=x+1`) to confirm receipt of the client's SYN.
3. **ACK:** The client receives the SYN-ACK and sends a final segment with the **ACK** flag set to 1. The sequence number is $x+1$ and the acknowledgment number is $y+1$.

At this point, the connection is established, and data transfer can begin.

Q5: Draw and explain the key fields of the TCP segment header.

A: The TCP header is 20 bytes long (without options).

- **Source Port (16 bits):** Port number of the sending application.
- **Destination Port (16 bits):** Port number of the receiving application.
- **Sequence Number (32 bits):** The sequence number of the first data byte in this segment.
- **Acknowledgment Number (32 bits):** If the ACK flag is set, this field contains the value of the next sequence number the sender is expecting to receive.
- **Header Length (HLEN) (4 bits):** The size of the TCP header in 32-bit words.
- **Flags (6 bits):**
    - **URG:** Urgent pointer field is valid.
    - **ACK:** Acknowledgment field is valid.
    - **PSH:** Push function (receiver should pass data to app immediately).
    - **RST:** Reset the connection.
    - **SYN:** Synchronize sequence numbers (used to initiate a connection).
    - **FIN:** No more data from sender (used to terminate a connection).
- **Window Size (16 bits):** Used for flow control. Specifies the number of bytes the sender is willing to receive.
- **Checksum (16 bits):** Error-checking for the header, data, and a pseudo-header.
- **Urgent Pointer (16 bits):** Points to urgent data (if URG flag is set).

Questions from Previous Internal Test (PYQ)

These are the questions from the images you provided, with answers.

Q1. a. Explain Distance Vector Routing algorithm with an example. (6M)

A: Distance Vector Routing is an iterative, asynchronous, and distributed routing algorithm.

- **Distributed:** Each router maintains its own routing table, which it computes based on information from its neighbors.
- **Iterative:** The process continues until no more information can be exchanged.
- **Asynchronous:** It does not require all routers to operate in lockstep.

Each router maintains a **distance vector** (routing table) containing the "distance" (cost, usually hops) to every other router in the network. It periodically shares this vector with its **direct neighbors only**.

When a router X receives a distance vector from its neighbor Y, X updates its own table using the Bellman-Ford equation:

$Cost(X, Z) = \min_{\text{all neighbors } Y} [Cost(X, Y) + Cost(Y, Z)]$

This means the cost for X to get to Z is the minimum of (cost to get to neighbor Y + Y's advertised cost to get to Z).

Example:

- Consider routers A, B, C. A-B cost is 2, B-C cost is 1, A-C cost is 5.
- **B's Table:** {A: 2, B: 0, C: 1}
- **C's Table:** {A: 5, B: 1, C: 0}
- A's initial table is {A: 0, B: 2, C: 5}.
- A receives B's table. A recalculates its path to C:
    - Path via B: Cost(A,B) + Cost(B,C) = 2 + 1 = 3.
    - This is less than its current cost of 5.
- A updates its table: {A: 0, B: 2, C: 3} (via B).
  This process repeats until all tables stabilize. A major drawback is the Count to Infinity problem, where good news (a link coming up) travels fast, but bad news (a link failing) travels very slowly.

Q1. b. What is congestion? Explain leaky bucket algorithm. (4M)

A: Congestion is a state in a network where the load (number of packets) exceeds the network's capacity (link bandwidth and router processing speed), leading to high packet delay and loss.

Leaky Bucket Algorithm: This algorithm is used for traffic shaping to control congestion.

1. It is implemented as a finite queue (a "bucket").
2. When a packet arrives, if the bucket is not full, the packet is added. If the bucket is full, the packet is discarded.
3. Packets are sent from the bucket at a **fixed, constant rate** (like a "leak"), regardless of how many packets are in the bucket (as long as it's not empty).
4. This smooths out bursty traffic into a steady stream, but it does not allow for saving "credits" during idle periods.

Q2. a. Explain the concept of IP addressing and subnetting with an example. (6M)

A: IP Addressing: An IP address is a unique 32-bit (for IPv4) numerical label assigned to each device on a network, used for identification and location. It's written in dot-decimal notation (e.g., 172.16.10.5). An IP address is divided into two parts: a Network ID (identifies the network) and a Host ID (identifies the specific device).

Subnetting: This is the process of dividing a single large network into multiple smaller, more manageable networks called subnets. It is done by "borrowing" bits from the Host ID portion of the address to create a **Subnet ID**.

- This is controlled by a **Subnet Mask**, a 32-bit number that "masks" the network and subnet parts from the host part.

Example:

An organization is given the block 172.16.0.0/16.

- **Original IP range:** 172.16.0.0 to 172.16.255.255
- **Original Subnet Mask:** /16 or 255.255.0.0

They need to create 8 subnets.

1. **Bits to Borrow:** To get 8 subnets, we need n bits where $2^n \ge 8$. So, $n=3$ bits.
2. **New Subnet Mask:** We borrow 3 bits from the host part, making the new mask $16 + 3 = 19$.
    - The new mask is **/19** or **255.255.224.0**.
3. **Address Ranges:** This creates 8 subnets, each with $2^{(32-19)} - 2 = 2^{13} - 2 = 8190$ usable hosts.
    - **Subnet 1:** 172.16.0.0/19 (Range: 172.16.0.1 to 172.16.31.254)
    - **Subnet 2:** 172.16.32.0/19 (Range: 172.16.32.1 to 172.16.63.254)
    - **Subnet 3:** 172.16.64.0/19
    - ...and so on, up to Subnet 8.

Q2. b. Differentiate between IPv4 and IPv6. (4M)

A:

| Feature | IPv4 (Internet Protocol v4) | IPv6 (Internet Protocol v6) |

| :--- | :--- | :--- |

| Address Size | 32 bits | 128 bits |

| Address Notation | Dotted-decimal (e.g., 192.168.1.1) | Hexadecimal (e.g., 2001:db8::1) |

| Address Space | ~ 4.3 billion addresses (exhausted) | $\sim 3.4 \times 10^{38}$ addresses (vast) |

| Header | 20-byte base header with options | 40-byte fixed header (options are in extension headers) |

| Configuration | Manual or via DHCP | Stateless Address Autoconfiguration (SLAAC) and DHCPv6 |

| Security | Optional (IPsec) | Required (IPsec is built-in) |

| Fragmentation | Done by sending host and routers | Done only by the sending host |

Q3. a. Explain the header format of IPv4. (7M)

A: The IPv4 header is 20 bytes long (without options).

- **Version (4 bits):** Set to 4 for IPv4.
- **HLEN (4 bits):** Header Length, in 32-bit words (e.g., a value of 5 means $5 \times 4 = 20$ bytes).
- **Type of Service (8 bits):** Specifies Quality of Service (QoS) parameters like priority.
- **Total Length (16 bits):** Total length of the entire datagram (header + data) in bytes.
- **Identification (16 bits):** Used to identify fragments of a single original datagram.
- **Flags (3 bits):** Control fragmentation (e.g., "Don't Fragment", "More Fragments").
- **Fragment Offset (13 bits):** Specifies the position of the fragment in the original datagram.
- **Time to Live (TTL) (8 bits):** A hop limit. Decremented by each router. When it reaches 0, the packet is discarded (prevents infinite loops).
- **Protocol (8 bits):** Identifies the transport layer protocol (e.g., 6 for TCP, 17 for UDP).
- **Header Checksum (16 bits):** Error-checking for the header only.
- **Source IP Address (32 bits):** The sender's IP address.
- **Destination IP Address (32 bits):** The receiver's IP address.
- **Options (Variable):** Used for testing, debugging, or security (rarely used).

Q3. b. What is BGP? Explain the BGP protocol. (3M)

A: BGP (Border Gateway Protocol) is the standardized inter-domain routing protocol used on the internet. It manages how packets are routed between different Autonomous Systems (AS), which are large networks (e.g., an ISP, a university, a large corporation).

BGP is a **Path Vector** protocol. Unlike OSPF or RIP which focus on the lowest "cost", BGP makes routing decisions based on **policies**, AS-path length, and other attributes.

- When a BGP router advertises a route to a neighbor, it includes the entire **path of AS numbers** it took to reach that destination.
- This path information is used to prevent routing loops (a router discards any route that already contains its own AS number).

Q4. a. Explain the elements of transport protocol. (6M)

A: Key elements (or services) of a transport protocol include:

1. **Addressing (Port Numbers):** Provides process-to-process communication by using port numbers to identify specific applications on a host.
2. **Connection Establishment:** For connection-oriented services (like TCP), this involves a handshake (e.g., 3-way handshake) to establish a logical connection before data is sent.
3. **Connection Release:** A mechanism to gracefully terminate the connection, ensuring all data is delivered (e.g., TCP's 4-way handshake with FIN flags).
4. **Flow Control:** Prevents a fast sender from overwhelming a slow receiver. TCP uses a "sliding window" mechanism where the receiver advertises its available buffer space (the "window size").
5. **Error Control:** Ensures data is delivered reliably. This is done using checksums (to detect corruption), acknowledgments (to confirm receipt), and retransmissions (to resend lost or corrupt packets).
6. **Segmentation and Reassembly:** Breaks large messages from the application layer into smaller segments for transmission and reassembles them at the destination.
7. **Congestion Control:** Manages network congestion (as opposed to receiver flow control). TCP uses algorithms like Slow Start to probe for available bandwidth and reduce its sending rate when congestion is detected.

Q4. b. With a neat diagram, explain the TCP segment header. (4M)

A: The TCP header is 20 bytes long (without options).

- **Source Port (16 bits):** Port number of the sending application.
- **Destination Port (16 bits):** Port number of the receiving application.
- **Sequence Number (32 bits):** The sequence number of the first data byte in this segment.
- **Acknowledgment Number (32 bits):** If the ACK flag is set, this field contains the value of the next sequence number the sender is expecting to receive.
- **Header Length (HLEN) (4 bits):** The size of the TCP header in 32-bit words.
- **Flags (6 bits):** (e.g., SYN, ACK, FIN, RST, PSH, URG).
- **Window Size (16 bits):** Used for flow control.
- **Checksum (16 bits):** Error-checking for the header, data, and a pseudo-header.

Q5. a. Explain the working of OSPF protocol. (5M)

A: OSPF (Open Shortest Path First) is an intra-domain Link State routing protocol. It is used within a single Autonomous System (AS).

The working of OSPF can be summarized in 5 steps:

1. **Discover Neighbors:** Each router sends "Hello" packets on all its links to discover its directly connected neighbors and establish adjacencies.
2. **Measure Link Cost:** Each router measures the cost (e.g., based on bandwidth) to its neighbors.

3. **Build Link State Packet (LSP):** Each router creates an LSP that describes its own links and their costs (e.g., "I am router A. I can reach B with cost 2 and C with cost 5").
4. **Flood LSPs:** Each router floods its LSP to *all other routers* within the same area. This ensures every router in the area builds an identical copy of the network topology.
5. **Compute Shortest Path:** Using the complete topology map, each router independently runs **Dijkstra's (Shortest Path First) algorithm** to compute the shortest path from itself to every other router. The results are stored in its routing table.

Q5. b. Explain the working of ICMP. (5M)

A: The Internet Control Message Protocol (ICMP) is a network layer protocol used by network devices to send error messages and operational information. It is a feedback mechanism for IP.

Working: ICMP is an auxiliary protocol to IP. It does not carry application data but rather control messages. When an IP datagram encounters an error (like an invalid destination, or its TTL expires), the router or host that discovers the error generates an ICMP message. This message is encapsulated within a new IP datagram and sent back to the original source of the problematic datagram.

- **Example (Ping):** A host sends an ICMP Echo Request to a destination. The destination host, upon receiving it, replies with an ICMP Echo Reply, confirming reachability.
- **Example (Error):** A host sends a UDP packet to a port that is not open. The destination host sends back an ICMP Destination Unreachable (Port Unreachable) message.

Q6. a. Explain the three-way handshake mechanism in TCP. (5M)

A: TCP uses a 3-way handshake to establish a reliable connection.

1. **SYN:** The client (initiator) sends a TCP segment with the **SYN** (Synchronize) flag set to 1 and a random initial sequence number (e.g., seq=x).
2. **SYN-ACK:** The server receives the SYN, allocates resources, and sends back a segment with both the **SYN** flag set to 1 and the **ACK** (Acknowledgment) flag set to 1. It includes its own random initial sequence number (e.g., seq=y) and an acknowledgment number (e.g., ack=x+1) to confirm receipt of the client's SYN.
3. ACK: The client receives the SYN-ACK and sends a final segment with the ACK flag set to 1. The sequence number is $x+1$ and the acknowledgment number is $y+1$.
   At this point, the connection is established, and data transfer can begin.

Q6. b. Explain the UDP segment header. (5M)

A: The UDP header is a simple 8-byte header.

- **Source Port (16 bits):** Port number of the sending application.
- **Destination Port (16 bits):** Port number of the receiving application.
- **Length (16 bits):** The length of the entire UDP segment (header + data) in bytes.

- Checksum (16 bits): Optional error-checking for the header and data.
  UDP is used by applications like DNS, DHCP, and VoIP where speed is preferred over the reliability and overhead of TCP.

Q7. a. What is quality of service (QoS)? Explain two techniques for achieving good QoS. (5M)

A: Quality of Service (QoS) refers to the ability of a network to provide better or special service to selected network traffic. It is a set of techniques used to manage network resources (bandwidth, delay, jitter, packet loss) to meet the needs of specific applications (e.g., video conferencing needs low delay, while file transfer can tolerate it).

Two techniques for achieving good QoS are:

1. **Traffic Shaping (e.g., Leaky Bucket):** This technique smooths out traffic bursts by buffering packets and sending them out at a fixed, average rate. This prevents a bursty application from causing congestion.
2. **Integrated Services (IntServ):** A resource reservation model. Applications use a protocol like **RSVP (Resource Reservation Protocol)** to request a specific amount of bandwidth and a bound on delay for a particular data flow. Routers along the path then reserve these resources, creating a "virtual circuit" for that flow.

Q7. b. Write a short note on flooding as a routing algorithm. (5M)

A: Flooding is a simple routing algorithm where every incoming packet is sent out on every outgoing link except the one it arrived on.

- **Advantages:**
  1. **Robust:** It is guaranteed to find the shortest path (and every other path) because it explores all possibilities.
  2. **Simple:** No complex tables or calculations are needed.
- **Disadvantages:**
  1. **Generates vast traffic:** Creates an exponential number of duplicate packets, which can congest the network.
  2. **Inefficient:** Wastes bandwidth.
- **Controls:** To be practical, flooding must be controlled. This is done using:
  1. **Hop Count (TTL):** A packet is discarded after a certain number of hops.
  2. **Sequence Numbers:** Routers keep track of (source, sequence_number) pairs and discard any duplicate packets they have already seen.
- **Use:** Pure flooding is rarely used for data. However, a controlled version of flooding is the core mechanism used by **Link State protocols (like OSPF)** to distribute Link State Packets (LSPs) to all routers.

Q8. a. Explain link state routing algorithm. (6M)

A: Link State Routing is an intra-domain routing algorithm (like OSPF). Its operation is based on each router having a complete map of the network topology.

The process involves 5 steps:

1. **Discover Neighbors:** Each router sends "Hello" packets to find its directly connected neighbors.
2. **Measure Link Cost:** Each router measures the cost (e.g., delay or bandwidth) to its neighbors.
3. **Build Link State Packet (LSP):** Each router creates an LSP containing its identity, its list of neighbors, and the cost to each.
4. **Flood LSPs:** Each router broadcasts its LSP to *all other routers* in the network using controlled flooding. This ensures every router builds an identical copy of the network's topology map.
5. **Compute Shortest Path:** Using the complete topology map, each router independently runs **Dijkstra's (Shortest Path First) algorithm** to compute the shortest path from itself to every other router. This result is then used to build its local routing table.

Q8. b. Differentiate between TCP and UDP. (4M)

A:

| Feature | TCP (Transmission Control Protocol) | UDP (User Datagram Protocol) |

| :--- | :--- | :--- |

| Connection | Connection-oriented (uses 3-way handshake). | Connectionless. |

| Reliability | Reliable (uses ACKs and retransmissions). | Unreliable (best-effort). |

| Ordering | Guarantees in-order delivery. | Does not guarantee order. |

| Flow/Congestion Control | Yes (sliding window, slow start, etc.). | No. |

| Header Size | 20 bytes (plus options). | 8 bytes. |

| Use Cases | HTTP, FTP, SMTP (email). | DNS, VoIP, online gaming. |