Network Theory Exam Study Guide

Comprehensive Guide for Units III-VI

Undergraduate Level

Table of Contents

Unit III: Network Layer

1. Prerequisite Review

Before diving into the Network Layer, let's recall a few key concepts: * **Data Encapsulation**: Data moves down the layers, getting wrapped in headers. The Network Layer adds an **IP Header** to the Transport Layer segment, creating a **Packet**. * **Addressing**: Just like houses have addresses, devices on a network need unique identifiers. We've seen MAC addresses (Data Link Layer) which are physical and permanent. Now we introduce **IP Addresses** (Network Layer) which are logical and hierarchical. * **The Goal**: The Network Layer is responsible for **Host-to-Host delivery**. It's like the postal service ensuring a letter gets from your house to your friend's house, regardless of the path it takes.

---

2. IPv4 Addressing

Internet Protocol version 4 (IPv4) is the fundamental addressing system of the internet.

2.1 Structure and Notation

- **Analogy**: Think of an IP address like a phone number. It has an area code (Network ID) and a subscriber number (Host ID).
- **Format**: A 32-bit address, typically written in **Dotted Decimal Notation** (e.g., `192.168.1.1`).
- **Bits**: 32 bits total, divided into 4 octets (8 bits each).
    - Example: `11000000.10101000.00000001.00000001` = `192.168.1.1`

2.2 Classful Addressing (Legacy)

Originally, IP addresses were divided into "Classes" to define the network size.

Class	Leading Bits	Range (1st Octet)	Default Subnet Mask	Use Case
A	`0`	1 - 126	255.0.0.0 (/8)	Huge networks (ISPs, Gov)
B	`10`	128 - 191	255.255.0.0 (/16)	Medium networks (Universities)
C	`110`	192 - 223	255.255.255.0 (/24)	Small networks (Home/Office)
D	`1110`	224 - 239	N/A	Multicast (Streaming)
E	`1111`	240 - 255	N/A	Experimental/ Research

- **Problem**: Inefficient. A Class A network has 16 million addresses! If a company needed 5000, giving them a Class B (65,000) wasted 60,000 addresses.

2.3 Subnetting

Subnetting is the process of borrowing bits from the Host ID to create a Subnet ID. * **Analogy**: A large office building (Network) is divided into floors (Subnets). The main address gets you to the building, the floor number gets you to the department. * **Benefit**: Reduces network traffic (broadcast domains) and improves security. * **Subnet Mask**: A 32-bit number that tells the computer which part of

the IP is the Network and which is the Host. * `1`s represent the Network. * `0`s represent the Host.

2.4 CIDR (Classless Inter-Domain Routing)

CIDR replaced Classful addressing. It allows for flexible network sizes using "slash notation". * **Notation**: `IP_Address / Prefix_Length` * **Example**: `192.168.1.0/24` means the first 24 bits are the network. * **VLSM (Variable Length Subnet Mask)**: Allows a network administrator to use different subnet masks for subnets of the same network, optimizing address usage.

2.5 NAT (Network Address Translation)

NAT allows multiple devices on a private network to share a single public IP address. * **Analogy**: An apartment complex has one main mailing address. The concierge (NAT Router) sorts the mail to individual apartment numbers (Private IPs). * **Private IP Ranges** (Not routable on the internet): * `10.0.0.0` - `10.255.255.255` * `172.16.0.0` - `172.31.255.255` * `192.168.0.0` - `192.168.255.255`

---

3. IPv6 Addressing

We ran out of IPv4 addresses! **IPv6** is the successor.

3.1 Key Differences

- **Size**: 128-bit address (vs 32-bit). That's enough for every grain of sand on Earth to have an IP.
- **Format**: Hexadecimal, separated by colons.
    - Example: `2001:0db8:85a3:0000:0000:8a2e:0370:7334`
- **Header**: Simplified header for faster processing. No checksum (Transport layer handles it).

3.2 Address Types

1. **Unicast**: One-to-One. (Send to a specific device).
2. **Multicast**: One-to-Many. (Send to a group).
3. **Anycast**: One-to-Nearest. (Send to the closest server in a group, e.g., DNS).
    - Note: IPv6 has no Broadcast.

3.3 Transition Strategies

How do we switch from IPv4 to IPv6? 1. **Dual Stack**: Devices run both IPv4 and IPv6 simultaneously. 2. **Tunneling**: Wrapping an IPv6 packet inside an IPv4 packet to cross an IPv4 network. 3. **Translation**: Converting headers between versions (NAT64).

---

4. Routing

Routing is finding the best path for a packet to travel from source to destination.

4.1 Static vs Dynamic

- **Static Routing**: Admin manually types in the routes.
    - Pros: Secure, no overhead.
    - Cons: Doesn't adapt to failures. Hard to manage in large networks.
- **Dynamic Routing**: Routers talk to each other to learn paths.
    - Pros: Automatic, scalable, adapts to broken links.
    - Cons: Uses bandwidth and CPU.

4.2 Distance Vector Routing (e.g., RIP)

- **Concept**: "Tell your neighbors about the world."
- **Metric**: Hop count (number of routers to cross).
- **Algorithm**: Bellman-Ford.

- **Problem**: **Count-to-Infinity**. If A tells B "I can reach C", and the link to C breaks, A might learn a path to C through B (which is actually A -> B -> A -> ...).
    - Fix: Split Horizon (don't tell a neighbor about a route you learned from them).

4.3 Link State Routing (e.g., OSPF)

- **Concept**: "Tell the world about your neighbors."
- **Mechanism**:
    1. Discover neighbors (Hello packets).
    2. Measure cost (delay, bandwidth).
    3. Flood **LSA (Link State Advertisements)** to everyone.
    4. Build a map of the entire network.
    5. Run **Dijkstra's Algorithm** to find the Shortest Path.

---

5. Protocols

5.1 IP (Internet Protocol)

- **Unreliable**: Best effort delivery. No guarantees.
- **Connectionless**: Each packet is independent.

5.2 ICMP (Internet Control Message Protocol)

- **Purpose**: Error reporting and diagnostics.
- **Tools**: `ping` (Echo Request/Reply) and `traceroute`.
- **Messages**: "Destination Unreachable", "Time Exceeded" (TTL expired).

5.3 ARP (Address Resolution Protocol)

- **Problem**: I know the IP address (`192.168.1.5`), but I need the MAC address to send the frame.
- **Solution**: Broadcast "Who has 192.168.1.5?"

- **Reply**: "I do! Here is my MAC."
- **RARP**: Reverse ARP (Diskless workstations asking "What is my IP?").

5.4 Fragmentation

- **MTU (Maximum Transmission Unit)**: The largest packet a network link can carry (usually 1500 bytes for Ethernet).
- **Process**: If a packet is too big, the router breaks it into fragments.
- **Reassembly**: Happens **only at the destination**, not intermediate routers.

Unit III Summary

- **IPv4** uses 32-bit addresses; **IPv6** uses 128-bit addresses.
- **Subnetting** divides networks; **CIDR** allows flexible sizing.
- **Routing** finds the path: **Distance Vector** (Hops, Neighbors) vs **Link State** (Map, Dijkstra).
- **ARP** maps IP to MAC. **ICMP** helps debug.

Unit IV: Transport Layer

1. Prerequisite Review

- **Host-to-Host vs. Process-to-Process**: The Network Layer gets the packet to the correct computer (Host). The Transport Layer gets it to the correct application (Process) running on that computer.
- **Reliability**: The Network Layer (IP) is unreliable. The Transport Layer is responsible for fixing errors, if needed.

---

2. Services & Fundamentals

2.1 Port Addressing

- **Analogy**: The IP address is the apartment building address. The **Port Number** is the specific apartment number where a person (application) lives.
- **Socket**: IP Address + Port Number = Socket (e.g., `192.168.1.1:80`). This uniquely identifies a connection.
- **Well-Known Ports**:
  - 80: HTTP (Web)
  - 443: HTTPS (Secure Web)
  - 25: SMTP (Email)
  - 53: DNS

2.2 Multiplexing & Demultiplexing

- **Multiplexing (Sender)**: Gathering data from different apps (Chrome, Spotify, Zoom), adding headers, and sending them out as a single stream of packets.
- **Demultiplexing (Receiver)**: Receiving the stream, checking the Port Number, and delivering the data to the correct app.

---

3. UDP (User Datagram Protocol)

UDP is the "fire and forget" protocol.

3.1 Characteristics

- **Connectionless**: No handshake. Just send.
- **Unreliable**: No guarantees. If a packet is lost, it's gone.
- **Fast**: Low overhead (small header).
- **Use Cases**: Streaming (YouTube, Netflix), VoIP (Skype), Gaming (where speed > perfection).

3.2 Header

- Very simple, only 8 bytes:
    1. Source Port
    2. Destination Port
    3. Length
    4. Checksum (for error detection)

---

4. TCP (Transmission Control Protocol)

TCP is the reliable workhorse of the internet.

4.1 Characteristics

- **Connection-Oriented**: Must establish a connection before sending data.
- **Reliable**: Guarantees delivery, in order, without errors.
- **Byte-Stream**: Data is sent as a continuous stream of bytes.

4.2 The 3-Way Handshake (Connection Setup)

Before talking, Alice and Bob must agree to talk. 1. **SYN** (Synchronize): Client sends "Let's connect! My sequence number is X." 2. **SYN-ACK**: Server says "Okay! I see X. My sequence number is Y." 3. **ACK**: Client says "Okay! I see Y. Connection established."

4.3 Flow Control

Flow Control prevents the sender from overwhelming the **receiver**. * **Stop-and-Wait**: The sender sends one packet and waits for an ACK before sending the next. * Analogy: Sending a letter, waiting for a reply, then sending the next. Very slow but simple. * **Pipelining**: The sender sends multiple packets without waiting for ACKs (filling the pipe). * Analogy: Sending 10 letters at once. Much faster. * **Sliding Window Protocol**: Used to implement pipelining. The receiver tells the sender "I have room for 5000 bytes" (Window Size). The sender

sends 5000 bytes and waits for an update. * Analogy: Eating a hot dog. You tell the person feeding you "Slow down!" if your mouth is full.

4.4 Congestion Control

Congestion Control prevents the sender from overwhelming the **network** (routers). * **Analogy**: Traffic jam. If everyone drives onto the highway at once, no one moves. * **Algorithms**: 1. **Slow Start**: Start sending slowly (1 packet). If successful, double it (2, 4, 8...). Exponential growth. 2. **Congestion Avoidance**: When a threshold is reached, grow linearly (add 1). 3. **Congestion Detection**: If a packet is lost (timeout), assume congestion. Drop the speed dramatically (back to 1 or half). 4. **AIMD (Additive Increase, Multiplicative Decrease)**: Slowly increase speed to find the limit, but cut speed by half immediately if a problem occurs.

---

5. Reliability Mechanisms

How does TCP ensure reliability?

1. **Sequence Numbers**: Every byte is numbered. The receiver can reorder packets if they arrive out of order.
2. **Acknowledgments (ACK)**: The receiver tells the sender "I received everything up to byte X."
3. **Retransmission**:
    - **Timeout**: If I don't get an ACK after a certain time, I assume the packet is lost and send it again.
    - **Fast Retransmit**: If I get 3 duplicate ACKs for the same packet, I know the next packet is missing, so I resend it immediately without waiting for the timer.

Unit IV Summary

- **UDP** is fast but unreliable (Streaming). **TCP** is reliable but slower (Web, Email).
- **Ports** identify applications.
- **3-Way Handshake** establishes a TCP connection.
- **Flow Control** protects the receiver (Sliding Window).
- **Congestion Control** protects the network (Slow Start, AIMD).

Unit V: Application Layer

1. Prerequisite Review

- **User Interface**: The Application Layer is where the human interacts with the network. It's the web browser, the email client, the file transfer tool.
- **Client-Server Model**: Most apps work this way. A **Client** (your phone) requests a service, and a **Server** (a powerful computer in a data center) provides it.

---

2. DNS (Domain Name System)

DNS is the phonebook of the internet.

2.1 Hierarchy

- **Root Servers**: The top of the tree (represented by a dot `.`).
- **TLD (Top-Level Domain)**: `.com`, `.org`, `.edu`, `.in`.
- **Authoritative Servers**: The servers that actually know the IP address for a specific domain (e.g., `google.com`).

2.2 Resolution Process

How does your computer find `www.example.com`? 1. **Browser Cache**: "Have I been there recently?" 2. **OS Cache**: "Does my computer know?" 3. **Resolver (ISP)**: "Hey ISP, do you know?" 4. **Root Server**: "I don't know, but here is the address for `.com`." 5. **TLD Server**: "I don't know, but here is the address for `example.com`." 6. **Authoritative Server**: "Yes! The IP is `93.184.216.34`."

2.3 Record Types

- **A**: IPv4 Address.
- **AAAA**: IPv6 Address.
- **MX**: Mail Exchange (Email server).
- **CNAME**: Canonical Name (Alias, e.g., `www` -> `server1`).
- **NS**: Name Server (Who is authoritative?).
- **PTR**: Pointer Record (Reverse DNS). Maps IP to Domain Name.

---

3. HTTP (HyperText Transfer Protocol)

HTTP is the protocol of the World Wide Web.

3.1 Versions

- **HTTP/1.0**: Old. Created a new TCP connection for every file (very slow).
- **HTTP/1.1**: Persistent connections (keep-alive). Reuse the connection for multiple files.
- **HTTP/2.0**: Multiplexing. Send multiple requests at once over a single connection. Binary (not text).

3.2 Request/Response

- **Request**: Client sends "GET /index.html HTTP/1.1".

- **Response**: Server sends "HTTP/1.1 200 OK" followed by the webpage.

3.3 Methods (Verbs)

- **GET**: Retrieve data (Loading a page).
- **POST**: Submit data (Filling a form).
- **PUT**: Update data (Uploading a file).
- **DELETE**: Remove data.

3.3 Status Codes

- **2xx (Success)**: 200 OK.
- **3xx (Redirection)**: 301 Moved Permanently.
- **4xx (Client Error)**: 404 Not Found (You typed the URL wrong).
- **5xx (Server Error)**: 500 Internal Server Error (The server crashed).

3.4 HTTPS & TLS/SSL

- **HTTP** is plain text. Anyone can spy on it.
- **HTTPS** is HTTP + **SSL/TLS** (Encryption). It ensures:
    1. **Privacy**: No one can read the data.
    2. **Integrity**: No one changed the data.
    3. **Authentication**: You are talking to the real bank, not a fake one.

---

4. Email Protocols

Sending an email involves multiple protocols.

4.1 SMTP (Simple Mail Transfer Protocol)

- **Role**: **Pushing** mail.

- Used by the sender to send email to the server, and by servers to send email to other servers.

4.2 POP3 (Post Office Protocol v3)

- **Role**: **Pulling** mail.
- **Behavior**: Downloads email to your device and deletes it from the server. Good for one device, bad for multiple (phone + laptop).

4.3 IMAP (Internet Message Access Protocol)

- **Role**: **Syncing** mail.
- **Behavior**: Reads email on the server. Keeps everything in sync across multiple devices.

4.4 MIME (Multipurpose Internet Mail Extensions)

- SMTP only supports text. **MIME** allows sending images, audio, video, and non-English characters in email.

---

5. Other Protocols

5.1 FTP (File Transfer Protocol)

- Used for transferring large files.
- **Two Connections**:
    1. **Control Connection (Port 21)**: Sends commands (Login, CD, LS).
    2. **Data Connection (Port 20)**: Sends the actual file.
- **Modes**:
    - **Active Mode**: Client opens a random port and server connects back to it (Problematic for firewalls).
    - **Passive Mode**: Server opens a random port and client connects to it (Firewall friendly).

5.2 DHCP (Dynamic Host Configuration Protocol)

- **Role**: Automatically assigns IP addresses to devices when they join a network.
- **Process (DORA)**:
  1. **D**iscover: "Is there a DHCP server?"
  2. **O**ffer: "Yes, you can use `192.168.1.50`."
  3. **R**equest: "Great, I'll take it."
  4. **A**cknowledge: "It's yours for 24 hours."

5.3 SSH (Secure Shell) vs Telnet

- **Telnet**: Remote login, but sends passwords in plain text (Unsafe!).
- **SSH**: Encrypted remote login (Safe!). Used by admins to manage servers.

5.4 SNMP (Simple Network Management Protocol)

- **Purpose**: Monitoring network devices (routers, switches, printers).
- **Components**:
  - **Manager**: The central computer monitoring everything.
  - **Agent**: The software running on the router/switch.
  - **MIB (Management Information Base)**: The database of variables (CPU usage, bandwidth, errors).

Unit V Summary

- **DNS** translates Names to IPs.
- **HTTP** transfers web pages; **HTTPS** secures them.
- **SMTP** sends email; **IMAP** syncs it.
- **DHCP** gives you an IP address automatically.

Unit VI: Network Security

1. Prerequisite Review

- **The Internet is Public**: By default, the internet is like a postcard. Anyone handling it can read it. Security is about putting that postcard in a sealed envelope (Encryption).
- **Trust**: How do you know the website you are visiting is actually your bank and not a hacker? (Authentication).

---

2. Fundamentals

2.1 Security Services (CIA Triad + 2)

1. **Confidentiality**: Only authorized people can read the data. (Privacy).
2. **Integrity**: The data hasn't been changed in transit.
3. **Availability**: The system is up and running when needed.
4. **Authentication**: Verifying the identity of the user/sender.
5. **Non-Repudiation**: The sender cannot deny sending the message later (like a digital signature).

2.2 Threats & Attacks

- **Passive Attack**: The hacker just listens (Eavesdropping). Hard to detect, easy to prevent (Encryption).
- **Active Attack**: The hacker changes data or disrupts service. Easy to detect, hard to prevent.
    - **Masquerading**: Pretending to be someone else (Spoofing).
    - **Replay**: Recording a valid message (like a bank transfer) and sending it again later.

    - **DoS (Denial of Service)**: Flooding a server with traffic so it crashes.

---

3. Cryptography

Cryptography is the science of secret writing.

3.1 Symmetric Key Cryptography

- **Concept**: Use the **SAME key** to lock (encrypt) and unlock (decrypt) the box.
- **Analogy**: A house key. You give a copy to your friend. Both of you can lock and unlock the door.
- **Algorithms**: DES (Old, broken), 3DES (Better), **AES** (Current standard, very strong).
- **Problem**: How do you share the key securely in the first place?

3.2 Asymmetric Key Cryptography (Public Key)

- **Concept**: Use **TWO keys**. A **Public Key** (to lock) and a **Private Key** (to unlock).
- **Analogy**: A mailbox. Anyone can put a letter in (Public Key), but only the postman with the key can take it out (Private Key).
- **Algorithms**: **RSA**, Diffie-Hellman (for key exchange).
- **Benefit**: Solves the key sharing problem.

3.3 Hash Functions

- **Concept**: A one-way fingerprint of data. You cannot get the data back from the hash.
- **Use**: Checking Integrity. If even one bit of the file changes, the hash changes completely.
- **Algorithms**: MD5 (Old), SHA-1, **SHA-256**.

3.4 Digital Signatures

- **Goal**: Prove authenticity and integrity.
- **Process**:
    1. Hash the message.
    2. Encrypt the hash with your **Private Key**.
    3. Receiver decrypts with your **Public Key** and compares the hash.

---

4. Protocols

4.1 IPSec (Internet Protocol Security)

- Secures IP packets at the Network Layer. Used in VPNs.
- **Modes**:
    - **Transport Mode**: Encrypts only the payload (data).
    - **Tunnel Mode**: Encrypts the entire packet (header + data).
- **Components**: AH (Authentication Header) and ESP (Encapsulating Security Payload).

4.2 SSL/TLS (Secure Sockets Layer / Transport Layer Security)

- Secures the Transport Layer (Web, Email). TLS is the modern version of SSL.
- **Handshake**:
    1. Client says "Hello, I support these algorithms."
    2. Server says "Hello, let's use this one. Here is my Certificate."
    3. Client verifies Certificate with a CA (Certificate Authority).
    4. They exchange keys and start encrypted communication.

4.3 S/MIME (Secure/Multipurpose Internet Mail Extensions)

- **Purpose**: Securing Email (Authentication, Integrity, Privacy).
- **Mechanism**:
  - **Signing**: Encrypt hash with sender's Private Key (Proof of Origin).
  - **Encryption**: Encrypt message with receiver's Public Key (Confidentiality).

---

5. Infrastructure

5.1 PKI (Public Key Infrastructure)

- The system that manages Digital Certificates.
- **CA (Certificate Authority)**: A trusted company (like Verisign or Let's Encrypt) that issues certificates. They vouch that "This Public Key belongs to Google.com".

5.2 Firewalls

- **Gatekeeper**: Sits between your network and the internet.
- **Packet Filtering**: "Block all traffic on Port 80 except from this IP."
- **Stateful**: "Only allow incoming traffic if it's a reply to an outgoing request."

5.3 IDS/IPS (Intrusion Detection/Prevention System)

- **IDS (Detection)**: The burglar alarm. Watches network traffic for suspicious patterns (signatures) and alerts the admin. Does not stop the attack.
- **IPS (Prevention)**: The security guard. Watches traffic and blocks it if it looks like an attack.

5.4 VPN (Virtual Private Network)

- **Tunnel**: Creates a secure, encrypted tunnel over the public internet.
- **Use**: Employees working from home can access the office network securely.

Unit VI Summary

- **CIA Triad**: Confidentiality, Integrity, Availability.
- **Symmetric**: One key (Fast). **Asymmetric**: Two keys (Secure key exchange).
- **HTTPS** uses **TLS** to secure the web.
- **Firewalls** filter traffic; **VPNs** create secure tunnels.

Glossary of Key Terms

- **ACK (Acknowledgment)**: A signal passed between communicating processes to signify acknowledgment of receipt of response.
- **Address Resolution Protocol (ARP)**: A protocol used to map an IP address to a physical machine address (MAC address).
- **Bandwidth**: The maximum amount of data that can travel through a channel.
- **CIDR (Classless Inter-Domain Routing)**: A method for allocating IP addresses and IP routing that replaces the previous classful IP addressing system.
- **DNS (Domain Name System)**: The system that translates human-readable domain names (like google.com) into IP addresses.
- **Encryption**: The process of converting information or data into a code, especially to prevent unauthorized access.

- **Firewall**: A network security device that monitors and filters incoming and outgoing network traffic based on an organization's previously established security policies.
- **Gateway**: A node (router) in a computer network, a key stopping point for data on its way to or from other networks.
- **HTTP (Hypertext Transfer Protocol)**: The foundation of data communication for the World Wide Web.
- **IP (Internet Protocol)**: The principal communications protocol in the Internet protocol suite for relaying datagrams across network boundaries.
- **MAC Address (Media Access Control Address)**: A unique identifier assigned to a network interface controller (NIC) for use as a network address in communications within a network segment.
- **NAT (Network Address Translation)**: A method of remapping one IP address space into another by modifying network address information in the IP header of packets while they are in transit.
- **Packet**: A formatted unit of data carried by a packet-switched network.
- **Router**: A device that forwards data packets between computer networks.
- **Subnet Mask**: A 32-bit number that masks an IP address, and divides the IP address into network address and host address.
- **TCP (Transmission Control Protocol)**: A standard that defines how to establish and maintain a network conversation through which application programs can exchange data.
- **UDP (User Datagram Protocol)**: A communications protocol that is primarily used for establishing low-latency and loss-tolerating connections between applications on the internet.
- **VPN (Virtual Private Network)**: A service that creates a safe, encrypted online connection.
