# Computer Networks - Exam Notes

## 1. Introduction to Networking

- IP Addressing: Provides unique, hierarchical addresses to devices, like a house address system.  
- Internet Protocol (IP): Routes packets from sender to receiver across the Internet.  
- Packets: Units of information exchanged across the Internet, each with layered headers.

## 2. Overview of TCP/IP

### Packets and Protocol Stack

- Data is split into packets, with each layer adding a header (encapsulation).  
- Headers contain control information: destination address, error checking, etc.  
- Intermediate devices (like routers) inspect only relevant headers.  
- TCP/IP ensures delivery of packets to the right application/process.

### Layers of TCP/IP Stack

|  |  |
| --- | --- |
| Layer | Description |
| Application Layer | User-facing applications that require network access (e.g., web browsers, email clients). |
| Transport Layer | Manages communication between processes; ensures reliability (TCP) or speed (UDP). |
| Internet Layer (IP) | Handles packet addressing and routing; best-effort delivery (no guarantees). |
| Network Interface Layer | Covers physical transmission technologies like Ethernet, PPP, ATM. |

### Transmission Control Protocol (TCP)

- Connection-oriented protocol:  
 - Establishes a connection before data transfer (3-way handshake).  
- Reliable:  
 - Guarantees data delivery (uses acknowledgments and retransmissions).  
- Ordered delivery:  
 - Packets are reassembled in the correct order.  
- Error Checking:  
 - Uses checksums to detect errors.  
- Flow Control:  
 - Uses sliding window to control the rate of data transmission.  
- Congestion Control:  
 - Adapts sending rate based on network congestion.

TCP Connection Lifecycle:  
1. Connection Establishment: 3-way handshake: SYN → SYN-ACK → ACK.  
2. Data Transfer: Reliable byte stream; sequencing numbers used.  
3. Connection Termination: 4-step process (FIN, ACK, FIN, ACK).

### Other Protocol Details

- UDP (User Datagram Protocol): Connectionless, no guarantees; faster, but unreliable.  
- IP (Internet Protocol): Provides addressing and routing.  
- Network Interface Layer: Physical medium technologies; easily integrates new technologies.

## 3. OSI Model vs TCP/IP Stack

| **OSI Model** | **TCP/IP Stack** |
| --- | --- |
| 7 layers | 4 layers |
| More theoretical | Practical, real-world use |
| Complex, late adoption | Simpler, early ARPANET use |
| Bureaucratic delays | Supported by DoD, industry |
| Less flexible | More flexible and open-source |

### Why TCP/IP Won Over OSI

- Early adoption: Used in ARPANET before OSI was developed.  
- Simplicity: Fewer layers, easier implementation.  
- Flexibility: Adaptable to various hardware and networks.  
- Industry Support: Backed by DoD, universities, and vendors.  
- Open Standard: Free and widely adopted.  
- Performance: Lower overhead.

## 4. Network Performance

### Key Metrics

- Bandwidth (Throughput): Amount of data transmitted per second (bps).  
- Latency (Delay): Time taken for a message to travel.  
 - Latency = Propagation + Transmission + Queuing.  
 - Round-Trip Time (RTT): Time for a message to go to destination and back.  
- Delay × Bandwidth Product: Latency = length of pipe, Bandwidth = width.

### Common Pitfalls

- b = bits, B = Bytes.  
- Bandwidth in bps, file sizes in Bytes.  
- 1 Mbps = 10^6 bps, 1 MB = 2^20 Bytes.

**Data Link Layer**

**Why Data Link Layer?**

- Provides separation between network and physical layers.  
- Simplifies IP stack design by hiding physical medium specifics.  
- Avoids frequent changes to IP for each hardware type.  
- Promotes 'IP over Everything' by abstracting the medium.

**Types of Layer 2 Networks**

|  |  |
| --- | --- |
| Type | Description |
| Point-to-Point | Direct device-to-device connection; no addressing needed. |
| Circuit-Based | Uses virtual circuits with identifiers; supports sharing of infrastructure. |
| Shared Networks | Multiple devices share the medium; uses addresses; collision management needed. |

**Ethernet Frame Structure**

|  |  |
| --- | --- |
| Field | Purpose |
| Preamble | Synchronization between sender and receiver. |
| SFD (Start Frame Delimiter) | Marks start of frame. |
| Destination MAC | Receiver address. |
| Source MAC | Sender address. |
| Length/Type | Indicates payload length or protocol type. |
| Data | Payload (e.g., IP packet). |
| FCS | Frame Check Sequence for error detection. |

**Ethernet Switching Domains**

- Hubs: Single collision domain, single broadcast domain.  
- Switches: Multiple collision domains (one per port), single broadcast domain.  
- Routers: Separate collision and broadcast domains.

**CSMA/CD Process**

1. Sense the medium.  
2. If idle, transmit.  
3. If collision detected, stop transmitting.  
4. Send 32-bit jam signal.  
5. Wait random backoff time, retry.

**Reliable Transmission Methods**

|  |  |
| --- | --- |
| Method | Description |
| Stop-and-Wait ARQ | Send one frame, wait for ACK, retransmit on timeout. |
| Sliding Window | Send multiple frames before requiring ACKs; keeps the pipe full. |
| Selective Acknowledgment (SACK) | ACK only received frames, improves recovery. |

**VLAN and Trunking**

- VLAN: Logical separation of networks, reduces broadcast traffic, improves security.  
- Trunking (802.1Q): Allows multiple VLANs over one physical link by tagging frames.  
- VLAN Stacking (Q-in-Q): Overcome VLAN ID limitations by nesting tags.

**Spanning Tree Protocol (STP)** - Prevents loops in Ethernet networks.  
- Blocks redundant paths while maintaining network connectivity.  
- Uses root/branch/leaf model to construct a loop-free topology.

**Network Layer**

**Why Layer 3 is Needed**

**- Ethernet lacks hierarchical addressing; scaling issues with large MAC tables.  
- Ethernet cannot easily communicate with non-Ethernet devices.  
- Need for protocol independent of underlying L2 technologies.  
- Network Layer (IP) provides logical addressing and routing.**

**Functions of Layer 3**

**- Logical, hierarchical addressing.  
- Separates broadcast domains; routers block broadcasts.  
- Routing based on Layer 3 forwarding tables.  
- Quality of Service (QoS) marking for priority handling.  
- Most common L3 protocol: Internet Protocol (IP).  
- IP is connectionless and unreliable (relies on upper layers like TCP for reliability).**

**IP Header Structure**

|  |  |
| --- | --- |
| **Field** | **Purpose** |
| **Version** | **IP version (4 for IPv4).** |
| **IHL** | **Header Length; typically 5 (20 bytes).** |
| **ToS/DSCP** | **Type of Service; prioritization (QoS).** |
| **Total Length** | **Total packet size (header + data).** |
| **Identification** | **Unique packet ID for fragmentation.** |
| **Flags** | **Fragment control: DF (Don’t Fragment), MF (More Fragments).** |
| **Fragment Offset** | **Position of fragment in original packet.** |
| **TTL** | **Time to Live; prevents loops.** |
| **Protocol** | **Upper layer protocol (e.g., TCP=6, UDP=17).** |
| **Header Checksum** | **Error check for the header.** |
| **Source Address** | **IP address of sender.** |
| **Destination Address** | **IP address of receiver.** |
| **Options** | **Rarely used; header extension.** |

**IP Fragmentation and Reassembly**

**- Fragmentation needed when packet > MTU (e.g., Ethernet MTU = 1500 bytes).  
- Fragmentation done by routers; reassembly at destination.  
- Fragments carry same Identification value.  
- IP does not recover missing fragments (no retransmission).**

**IP Addressing**

**- IPv4 address: 32 bits; shown in dotted-decimal format (e.g., 192.168.1.1).  
- Internally handled in binary.  
- Early addressing used classes (A, B, C, D, E):**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **Range (First Octet)** | **Network Bits** | **Purpose** |
| **A** | **1-126** | **8 bits** | **Large networks** |
| **B** | **128-191** | **16 bits** | **Medium networks** |
| **C** | **192-223** | **24 bits** | **Small networks** |
| **D** | **224-239** | **-** | **Multicast** |
| **E** | **240-255** | **-** | **Reserved** |

**- Classful addressing caused inefficiency and address depletion.  
- Address reuse via NAT, CIDR introduced.**

**IP Address Types**

**- Unicast: Single unique destination.  
- Broadcast: All devices in a network.  
- Anycast: Same IP assigned to multiple devices; routed to nearest one.  
- Multicast: Traffic sent to multiple interested receivers.**

**IP Subnetting**

**- Solves inefficiency of classful addressing.  
- Subnet mask divides host part into subnets.  
- Example: 192.168.2.0/25 splits Class C network into two subnets.**

**CIDR (Classless Inter-Domain Routing) and Route Aggregation**

**- Replaces classful addressing with prefix-based.  
- Allows route aggregation (supernetting).  
- More efficient IP space management.**

**IPv6 Overview**

**- 128-bit addresses solve IPv4 exhaustion.  
- Simplified header; no fragmentation by routers.  
- Mandatory IPsec for security.  
- Improved auto-configuration; critical for IoT.**

**IPv6 Header Fields**

|  |  |
| --- | --- |
| **Field** | **Purpose** |
| **Version** | **6 (0110).** |
| **Traffic Class** | **Differentiated Services (DS) and Explicit Congestion Notification (ECN).** |
| **Flow Label** | **Ensures in-order delivery.** |
| **Payload Length** | **Length of payload. (no header checksum)** |
| **Next Header** | **Type of next header (TCP, UDP, etc.).** |
| **Hop Limit** | **Like TTL in IPv4.** |
| **Source Address** | **128-bit source IP.** |
| **Destination Address** | **128-bit destination IP.** |

**IPv4 to IPv6 Transition - Tunneling**

**- No 'flag day' for upgrade; IPv4 and IPv6 must coexist.  
- Tunneling: IPv6 packet is encapsulated inside IPv4 packet.  
- Used extensively in modern networks (e.g., 4G/5G).**

**Network Layer - IPv4 Focus**

**IPv4 Overview**

**- IPv4 stands for Internet Protocol version 4, defined in RFC 791.  
- It is a connectionless protocol used for packet-switched networks.  
- Provides logical addressing to enable devices to locate each other across large networks, like the Internet.  
- IPv4 offers best-effort delivery: No guarantees about packet delivery, order, or duplicate protection.  
- Upper-layer protocols like TCP handle reliability, sequencing, and retransmissions.**

**IPv4 Addressing**

**- IPv4 addresses are 32 bits long, typically shown in dotted-decimal format (e.g., 192.168.1.1).  
- The address consists of a network portion and a host portion.  
- Originally, classful addressing divided IP addresses into Class A, B, C, D, and E.  
- Modern addressing uses CIDR (Classless Inter-Domain Routing) to provide more flexibility.  
- IP addresses must be unique in the network to avoid conflicts.**

**IPv4 Header Structure**

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Version** | **4 for IPv4.** |
| **IHL** | **Internet Header Length; header size in 32-bit words.** |
| **Type of Service (ToS)** | **Defines priority and QoS handling (now DSCP).** |
| **Total Length** | **Entire packet size including header and data.** |
| **Identification** | **Unique ID for fragmentation and reassembly.** |
| **Flags** | **Control fragmentation (DF, MF).** |
| **Fragment Offset** | **Position of the fragment in the original packet.** |
| **TTL** | **Time to Live; prevents looping by limiting hops.** |
| **Protocol** | **Indicates upper-layer protocol (e.g., TCP=6, UDP=17).** |
| **Header Checksum** | **Error checking for header integrity.** |
| **Source Address** | **IP address of sender.** |
| **Destination Address** | **IP address of receiver.** |
| **Options** | **Extended options; rarely used now.** |
| **Padding** | **Extra bytes to align header to 32-bit boundary.** |

**IPv4 Fragmentation and Reassembly**

**- Routers fragment packets when they exceed the Maximum Transmission Unit (MTU) of the outgoing link.  
- Each fragment has the same Identification number.  
- Fragment Offset indicates the fragment’s position.  
- The MF (More Fragments) flag is set on all fragments except the last.  
- Reassembly occurs only at the destination.  
- If any fragment is lost, the entire packet is discarded.  
- IPv4 has no mechanism to recover lost fragments.**

**Types of IPv4 Addresses**

**- \*\*Unicast\*\*: Sent to a single, specific device.  
- \*\*Broadcast\*\*: Sent to all devices in the network (255.255.255.255).  
- \*\*Multicast\*\*: Sent to multiple devices in a group (224.0.0.0 to 239.255.255.255).  
- \*\*Anycast\*\*: Sent to the nearest device in a group (implemented at routing level).**

**IPv4 Subnetting**

**- Subnetting divides a network into smaller sub-networks.  
- A subnet mask determines how many bits are used for network vs. host.  
- Allows better utilization of address space and improves network management.  
- Example: 192.168.1.0/24 vs. 192.168.1.0/25 creates two subnets.**

**CIDR (Classless Inter-Domain Routing)**

**- Replaces classful addressing.  
- Uses network prefixes (e.g., 192.168.0.0/16).  
- Enables route aggregation; reduces size of routing tables.  
- Increases flexibility in address allocation.**

**IPv4 Limitations**

**- Limited address space (~4.3 billion addresses).  
- Address exhaustion due to Internet growth.  
- No inherent security (IPsec is optional).  
- Complex NAT (Network Address Translation) needed to extend address space.  
- IPv6 designed to overcome these limitations.**

**IPv4 vs IPv6**

|  |  |
| --- | --- |
| **IPv4** | **IPv6** |
| **32-bit address** | **128-bit address** |
| **Dotted-decimal notation** | **Hexadecimal colon notation** |
| **Header checksum** | **No header checksum** |
| **Fragmentation allowed** | **No fragmentation by routers** |
| **Optional IPsec** | **Mandatory IPsec** |
| **Broadcast supported** | **No broadcast; multicast and anycast only** |

**Transport Layer**

**Overview of the Transport Layer**

**- Provides end-to-end communication services for applications.  
- Responsible for reliable or unreliable data transfer.  
- Key protocols: TCP (reliable) and UDP (unreliable).  
- TCP/UDP provide multiplexing via port numbers.  
- Some protocols (e.g., OSPF) bypass the transport layer.**

**Transmission Control Protocol (TCP)**

**- TCP (RFC 793) ensures reliable, ordered, error-checked delivery of data.  
- Provides:  
 • Stream data transfer.  
 • Reliability (acknowledgments, retransmissions).  
 • Flow control (window size adjustment).  
 • Congestion control.  
 • Multiplexing via ports.  
 • Logical connections (stateful).  
 • Full-duplex communication.**

**TCP Header Structure**

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Source/Destination Ports** | **Identify application services.** |
| **Sequence Number** | **First byte’s number in this segment.** |
| **Acknowledgment Number** | **Next expected sequence number.** |
| **Header Length (HLEN)** | **Length of TCP header.** |
| **Flags (URG, ACK, PSH, RST, SYN, FIN)** | **Control flags for the connection.** |
| **Window Size** | **Receiver buffer space (flow control).** |
| **Checksum** | **Ensures integrity of header and data.** |
| **Urgent Pointer** | **Pointer to urgent data (rarely used).** |
| **Options** | **Extra settings like MSS (Maximum Segment Size).** |

**TCP Connection Establishment - Three-Way Handshake**

**1. SYN: Client sends SYN with initial sequence number.  
2. SYN-ACK: Server responds with SYN + ACK.  
3. ACK: Client acknowledges server’s SYN.  
- Connection is established after these 3 steps.**

**TCP Reliable Data Transfer**

**- Uses sequence numbers and acknowledgements.  
- Positive acknowledgment with retransmission (PAR).  
- Lost or out-of-order segments are detected via duplicate ACKs.  
- Retransmission timer ensures lost data is resent.**

**TCP Flow Control**

**- Prevents sender from overwhelming receiver.  
- Uses window size in TCP header.  
- Receiver advertises buffer space (window).  
- Zero window: sender stops sending.  
- Window size updated as buffer frees.**

**User Datagram Protocol (UDP)**

**- Simple, connectionless, unreliable protocol.  
- Lower overhead compared to TCP.  
- No flow control, no retransmissions.  
- Suitable for real-time applications (VoIP, DNS, streaming).**

**UDP Header Structure**

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Source Port** | **Port of sending application.** |
| **Destination Port** | **Port of receiving application.** |
| **Length** | **Length of UDP header and data.** |
| **Checksum** | **Integrity check (optional in IPv4).** |

**Ports and Sockets**

**- Port Numbers:  
 • Well-Known Ports (1-1023): Reserved for standard services.  
 • Ephemeral Ports (1024-65535): Temporary ports for clients.  
- Socket = IP Address + Port Number + Protocol (TCP/UDP).  
- Unique identifier for each network connection.**

**Other Transport Protocols**

|  |  |
| --- | --- |
| **Protocol** | **Year** |
| **RTP (Real-time Transport Protocol)** | **1996** |
| **SCTP (Stream Control Transmission Protocol)** | **2000** |
| **DCCP (Datagram Congestion Control Protocol)** | **2006** |
| **UDT (UDP-based Data Transfer Protocol)** | **2005** |
| **FASP (Fast And Secure Protocol)** | **2006** |
| **MPTCP (Multipath TCP)** | **2013** |
| **QUIC** | **2021** |

**MPTCP (Multipath TCP) Features**

**- Allows a single connection to use multiple paths.  
- Backward-compatible with standard TCP.  
- Splits into subflows.  
- Aggregates bandwidth and improves throughput.  
- Resilient to path failures.  
- Transparent to applications.  
- Provides security via HMACs.  
- Defined in RFC 8684.**

**QUIC Protocol Features**

**- Runs on top of UDP.  
- 0-RTT and 1-RTT handshakes for fast connection setup.  
- Integrated TLS 1.3 encryption.  
- Multiplexed streams with no Head-of-Line blocking.  
- Stable Connection IDs support migration across networks.  
- Custom congestion control; per-stream retransmission.  
- Mobility support and enhanced security.  
- Primary transport for HTTP/3.**

**Local and Remote Port Forwarding (SSH)**

**\*\*Local Port Forwarding:\*\*  
- Redirects local port to remote server port via SSH.  
- Syntax: ssh -L [local\_port]:[destination\_host]:[destination\_port] user@ssh\_server  
- Example: ssh -L 8080:web-app.corp.local:80 user@gateway.company.com**

**\*\*Remote Port Forwarding:\*\*  
- Redirects remote server port to local machine port.  
- Syntax: ssh -R [remote\_port]:[destination\_host]:[destination\_port] user@ssh\_server  
- Example: ssh -R 8080:localhost:3000 user@public-server.com**

**Port Forwarding on Wireless Routers**

**- NAT (Network Address Translation) allows multiple devices to share one public IP.  
- NAT blocks unsolicited incoming traffic.  
- Port forwarding maps external ports to internal devices.  
- Example: Forward external port 25565 to internal 192.168.1.100:25565 (e.g., Minecraft server).**

**Wireless Networks**

**Overview of Wireless Networks**

**- Wireless networks allow communication without fixed infrastructure wiring.  
- Includes mobile phones, Wi-Fi, Bluetooth, and ad hoc networks.  
- Wireless links connect devices via radio waves, infrared, or other non-wired media.  
- Key components: wireless hosts, base stations, wireless links.**

**Elements of a Wireless Network**

**- \*\*Wireless Hosts\*\*: Devices like smartphones, laptops, IoT devices; may be stationary or mobile.  
- \*\*Base Station\*\*: Connects wireless hosts to wired network; e.g., Wi-Fi Access Points (APs), cell towers.  
- \*\*Wireless Links\*\*: Used for host-base station communication or backbone link; multiple access protocols manage access.**

**Wireless Link Characteristics**

**- \*\*Decreased Signal Strength\*\*: Attenuation as signal propagates.  
- \*\*Interference\*\*: Other devices operating on similar frequencies cause interference.  
- \*\*Multipath Propagation\*\*: Signal reflects off surfaces; causes multiple delayed copies.  
- \*\*Hidden Terminal Problem\*\*: Devices not directly reachable interfere at the receiver.  
- \*\*Signal Attenuation\*\*: Signal strength weakens with distance and obstructions.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Range** | **Speed** | **Technology** | **Frequency** |
| **10-30m** | **2 Mbps** | **Bluetooth** | **2.4 GHz** |
| **30m** | **11 Mbps** | **802.11b** | **2.4 GHz** |
| **30m** | **54 Mbps** | **802.11g** | **2.4 GHz** |
| **70m** | **600 Mbps** | **802.11n (WiFi 4)** | **2.4/5 GHz** |
| **70m** | **3.5 Gbps** | **802.11ac (WiFi 5)** | **5 GHz** |
| **70m** | **14 Gbps** | **802.11ax (WiFi 6)** | **2.4/5 GHz** |
| **1 Km** | **35-560 Mbps** | **802.11af** | **TV Bands (54–790 MHz)** |
| **1 Km** | **347 Mbps** | **802.11ah** | **900 MHz** |
| **4G LTE** | **Varies** | **Mobile Networks** | **Various** |
| **5G** | **10 Gbps** | **Mobile Networks** | **Various** |

**Infrastructure vs Ad-Hoc Mode**

**- \*\*Infrastructure Mode\*\*: Base station connects devices to wired network; supports handoff between stations.  
- \*\*Ad-Hoc Mode\*\*: No base stations; devices communicate directly and route among themselves.**

**Wireless Network Taxonomy**

**- \*\*Single Hop, Infrastructure\*\*: Device connects to AP, AP connects to Internet.  
- \*\*Single Hop, No Infrastructure\*\*: Direct host-to-host communication (e.g., Bluetooth).  
- \*\*Multiple Hops, Infrastructure\*\*: Host relays through nodes to AP (mesh network).  
- \*\*Multiple Hops, No Infrastructure\*\*: Ad hoc networks like MANETs, VANETs.**

**IEEE 802.11 Wireless LAN Standards**

|  |  |  |  |
| --- | --- | --- | --- |
| **Standard** | **Max Speed** | **Range** | **Frequency** |
| **802.11b** | **11 Mbps** | **30 m** | **2.4 GHz** |
| **802.11g** | **54 Mbps** | **30 m** | **2.4 GHz** |
| **802.11n (WiFi 4)** | **600 Mbps** | **70 m** | **2.4/5 GHz** |
| **802.11ac (WiFi 5)** | **3.5 Gbps** | **70 m** | **5 GHz** |
| **802.11ax (WiFi 6)** | **14 Gbps** | **70 m** | **2.4/5 GHz** |
| **802.11af** | **35–560 Mbps** | **1 Km** | **TV Bands** |
| **802.11ah** | **347 Mbps** | **1 Km** | **900 MHz** |

**IEEE 802.11 MAC Protocol**

**- \*\*CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance)\*\*:  
 • Sense medium before transmitting.  
 • No collision detection (difficult due to fading).  
- \*\*Collision Avoidance\*\*:  
 • Request-to-Send (RTS) and Clear-to-Send (CTS) frames.  
 • Reduces hidden terminal problem.**

**802.11 Frame Structure**

**- \*\*Address 1\*\*: Destination address.  
- \*\*Address 2\*\*: Source address.  
- \*\*Address 3\*\*: Router MAC address.  
- \*\*Address 4\*\*: Used in ad-hoc mode.  
- \*\*Frame Control\*\*: Frame type (RTS, CTS, data, ACK).  
- \*\*Sequence Control\*\*: Frame sequencing.  
- \*\*Payload\*\*: Data carried.  
- \*\*CRC\*\*: Error detection.**

**Mobility in 802.11 Networks**

**- Mobile device can roam between access points.  
- Stays within same IP subnet.  
- Switch updates which port/device is connected based on frame source.**