**Networking Protocols [ NAT, DHCP , DNS, ARP, FTP, HTTP, SSH, SMB, SNMP, SMTP, POP3, IMAP, TCP, UDP ]**

1. **Network Address Translation (NAT) :**  Translates private IP addresses to public IP addresses for routing traffic on the internet.

* A device on a private network (e.g., 192.168.1.10) sends a packet to access the internet (e.g., to a website). The packet includes Source IP, Destination IP (the website's public IP), Source port number and Destination port number.
* The NAT device (the router) replaces the private source IP to its own public IP (203.0.113.1), the router also assigns a unique port number (e.g., 45000) to track the connection in its NAT table.
* The packet is modified with the router’s public IP, unique port number assigned by the router, destination IP and destination port number and sent to the website over the internet.
* The website sends a response packet to the router's public IP (203.0.113.1) and the assigned port (45000).
* The router checks its NAT table to find which private device and port correspond to the public IP (203.0.113.1) and port (45000). It replaces the public Destination IP (203.0.113.1) with the private IP (192.168.1.10) and forwards the packet to the original device.

1. **Dynamic Host Configuration Protocol (DHCP) :** Automatically assigns IP addresses to devices on a network.

* **DHCP Discover**: The client (device) sends a broadcast message to discover a DHCP server on the network (DHCPDISCOVER).
* **DHCP Offer**: The DHCP server responds with an offer, providing an available IP address, subnet mask, lease time, and other configuration details (DHCPOFFER).
* **DHCP Request**: The client selects an offer and sends a DHCPREQUEST message to the server, confirming that it accepts the offer.
* **DHCP Acknowledgment**: The DHCP server acknowledges the request with a DHCPACK message, finalizing the IP assignment.
* **Client Configures**: The client applies the assigned IP address and network configuration to communicate on the network.

1. **Domain Name System (DNS) :** Resolves domain names to IP addresses to facilitate website access.

* The user enters a URL (e.g.,www.apple.com) in the browser, it checks its browser cache if the IP for the domain name is available.
* If the IP is not cached in the browser, the browser sends a query to the local DNS resolver (usually provided by your ISP or a public DNS service like Google DNS).
* The DNS resolver checks its cache to see if it already has the IP address for the domain. If it does, it returns the IP address.
* If the IP address is not cached, the resolver sends a recursive query to the root DNS servers, which redirect the query to the TLD server and which again redirects it to the authoritative name server for the domain.
* The authoritative DNS server provides the IP address for the requested domain, and the resolver sends it back to the user's browser, which can now connect to the website.

1. **Address Resolution Protocol (ARP) :** Maps IP addresses to MAC addresses to enable communication within a local network.

* When a device (Device A) wants to communicate with another device (Device B) on the same local network, it needs to know Device B’s MAC address, but it only knows Device B's IP address.
* Device A sends an ARP request as a broadcast to the entire local network, asking, "Who has IP 192.168.1.2? Tell me your MAC address."
* All devices on the network receive the ARP request, but only Device B responds with an ARP reply containing its MAC address (e.g., 00:11:22:33:44:55).
* Device A receives Device B's MAC address and updates its ARP cache, storing the mapping of Device B's IP address to MAC address for future communication.
* Now, Device A can directly communicate with Device B using its MAC address to send data over the network.

1. **File Transfer Protocol (FTP) :** Transfers files between a client and a server over a network.

* FTP uses two channels – a control channel (port 21) for commands and responses, and a data channel for transferring files.
* The client sends login credentials (username and password) to authenticate with the server.
* When transferring files, a separate data channel is established for each file transfer, which can be in either active or passive mode.
* In Active Mode, the server opens a random port, and the client connects to it to transfer data.
* In Passive Mode, the client opens a random port and connects to the server to transfer data.
* Files are transferred over the data channel, and once the transfer is complete, the data connection is closed.
* The client sends a QUIT command to terminate the session and close the control channel.

1. **HyperText Transfer Protocol (HTTP) :** Transfers hypertext documents (web pages) over the internet.

* The client (usually a web browser) sends an HTTP request to the server, asking for a specific resource (like a webpage or file).
* The server receives the request, processes it, and prepares the requested resource (e.g., an HTML page or image).
* The server sends an HTTP response back to the client, which contains the requested resource and a status code indicating the result of the request.
* The client receives the response and processes it, such as rendering the web page in the browser.
* HTTP is a stateless protocol, meaning once the response is sent, the connection is closed, and no session information is retained by the server.

1. **Secure Shell (SSH) :** Provides secure remote access to a computer over a network.

* The client initiates a connection to the server by sending an SSH request on port 22.
* The server responds by sending its public key to the client to establish a secure encrypted channel.
* The client generates a random session key (Shared Secret Key) and encrypts it using the server's public key, sending it back to the server.
* The server decrypts the session key using its private key, and both the client and server now share a secure session key for encrypted communication.
* The client then authenticates itself to the server by providing a password or an SSH key (public/private key pair) to prove its identity.
* Once authenticated, the client can send commands securely, and the server responds, all data being encrypted using the shared session key.
* The session remains open until the client or server closes the connection. The encrypted channel ensures confidentiality and integrity of the communication.

1. **Server Message Block (SMB) :** Shares files and resources like printers over a network.

* The client sends a request to the server using the SMB protocol, typically over port 445. This is used to access shared resources like files, printers, and devices on a network.
* The server and client establish a session using a negotiation phase to determine the supported SMB protocol version, encryption, and authentication methods.
* The client authenticates itself to the server by providing credentials such as a username and password. The server verifies these credentials before granting access to shared resources.
* The client sends requests for specific resources (e.g., files or printers) on the server. This is done using SMB commands that indicate what actions the client wants to perform, such as opening or reading a file.
* The server processes the requests and responds, either by granting access to the resource or denying access if the permissions are insufficient.
* Data is transferred between the client and the server over the SMB connection. The protocol allows the client to read, write, and manipulate files or use shared printers.
* Once the client is done accessing the resources, the session is terminated by sending a disconnect command to the server, closing the connection.

1. **Simple Network Management Protocol (SNMP) :** Monitors and manages network devices by exchanging data between managers and agents.

* The SNMP manager sends a request to the SNMP agent on a device to gather information, such as device status or performance metrics.
* The SNMP agent receives the request and accesses the Management Information Base (MIB), which contains data points about the device.
* The agent collects the requested data from the device, like CPU load or network traffic.
* The agent sends a response with the requested data back to the SNMP manager.
* If the agent detects significant changes or issues on the device, such as exceeding a threshold, it sends an SNMP trap to notify the manager.
* The SNMP manager interprets the received data and takes necessary actions, such as alerting the administrator or adjusting device settings.

1. **Simple Mail Transfer Protocol (SMTP) :** Sends email messages from a client to a mail server.

* The client sends an email message to the SMTP server, which includes the recipient's email address, subject, body, and any attachments.
* The client establishes a connection to the SMTP server on port 25 (or 587 for secure submission) and starts the session by sending a HELO or EHLO command.
* If required, the client authenticates with the server by providing a username and password.
* The SMTP server processes the email and checks the recipient's address. If the email is valid, the server prepares to forward it to the recipient's mail server.
* The SMTP server forwards the email to the recipient's mail server using DNS to find the correct destination server.
* Once the email reaches the recipient's mail server, it is stored and ready to be retrieved using protocols like POP3 or IMAP.
* After the email is sent, the client sends a QUIT command to the SMTP server to close the session.

1. **Post Office Protocol (POP3) :** Retrieving and downloading emails from a mail server to a local device and after downloading the messages are deleted from the mail server.

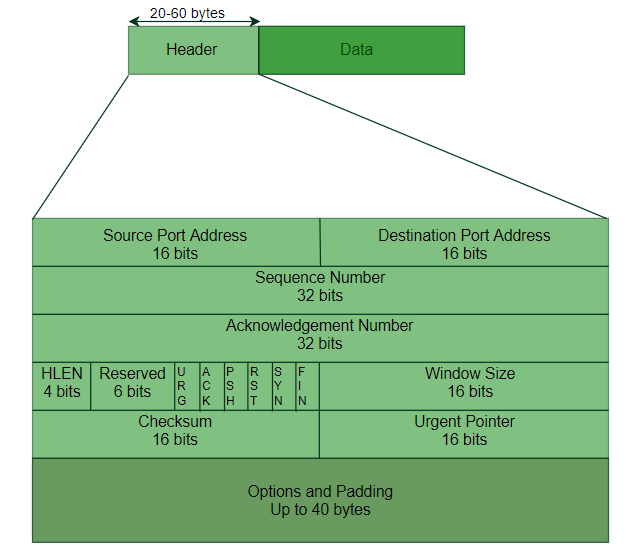
* The client connects to the mail server on port 110 (or 995 for secure connections) using the POP3 protocol.
* The client sends its login credentials (username and password) to the server for authentication.
* The mail server verifies the credentials and, if valid, retrieves the list of emails stored in the mailbox.
* The client requests to download the emails, and the server sends the email messages to the client.
* The emails are downloaded to the client’s local device, typically removing them from the server (unless the "leave messages on server" option is enabled).
* The client can then read, store, and manage the downloaded emails offline.
* Once done, the client disconnects from the server, closing the session.

1. **Internet Message Access Protocol (IMAP) :** To view and manage your emails directly on the mail server without downloading them.

* The client connects to the mail server on port 143 (or 993 for secure connections) using the IMAP protocol.
* The client authenticates by providing a username and password to the mail server.
* The mail server allows the client to access and view email messages stored on the server, without downloading them locally.
* The client retrieves email headers and content as needed, leaving the messages on the server.
* Any actions performed by the client (e.g., marking an email as read, moving it to a folder, or deleting it) are synchronized with the server, ensuring consistency across all devices.
* The client disconnects from the server, but emails remain stored on the server, accessible from other devices.

1. **Transmission Control Protocol (TCP) :** Establishes reliable, ordered communication between network devices.

* The client initiates a connection request by sending a **SYN** (synchronize) packet to the server, indicating the desire to start a connection.
* The server responds with a **SYN-ACK** (synchronize-acknowledge) packet, confirming it received the client's request and is ready to establish the connection.
* The client sends an **ACK** (acknowledge) packet to the server, confirming the server's response, and the connection is established.
* Once the connection is established, both the client and server can send data packets back and forth. Each packet includes sequence numbers to ensure data is transmitted in the correct order.
* If a packet is lost or corrupted, the receiver can request a retransmission of the missing data using **ACK** messages, ensuring reliable delivery.
* After the data transfer is complete, the client or server sends a **FIN** (finish) packet to initiate the termination of the connection.
* The other side responds with an **ACK** to confirm the termination, and the connection is closed.



1. **User Datagram Protocol (UDP) :** Provides faster, connectionless communication for applications like streaming.

* The client sends a data packet to the server without establishing a connection beforehand. The packet is transmitted using the UDP protocol on a specified port.
* The server listens for incoming UDP packets on the designated port. Upon receiving a packet, it processes the data without any handshake or acknowledgment.
* There is no acknowledgement from the server back to the client, meaning the client doesn't know if the packet was successfully received.
* If the client needs to send more data, it sends additional packets independently of the previous ones, with no connection management involved.
* UDP does not guarantee that the data packets will arrive in order, nor does it check for missing or corrupted packets. If a packet is lost or corrupted, the application using UDP must handle it.
* After data transmission, the session terminates without any formal closing procedure, as UDP is connectionless.

