**Networking Protocols**

1. **Network Address Translation (NAT)**

* 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)**

* **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)**

* 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)**

* 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)**

* 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)**

* 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)**

* 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.