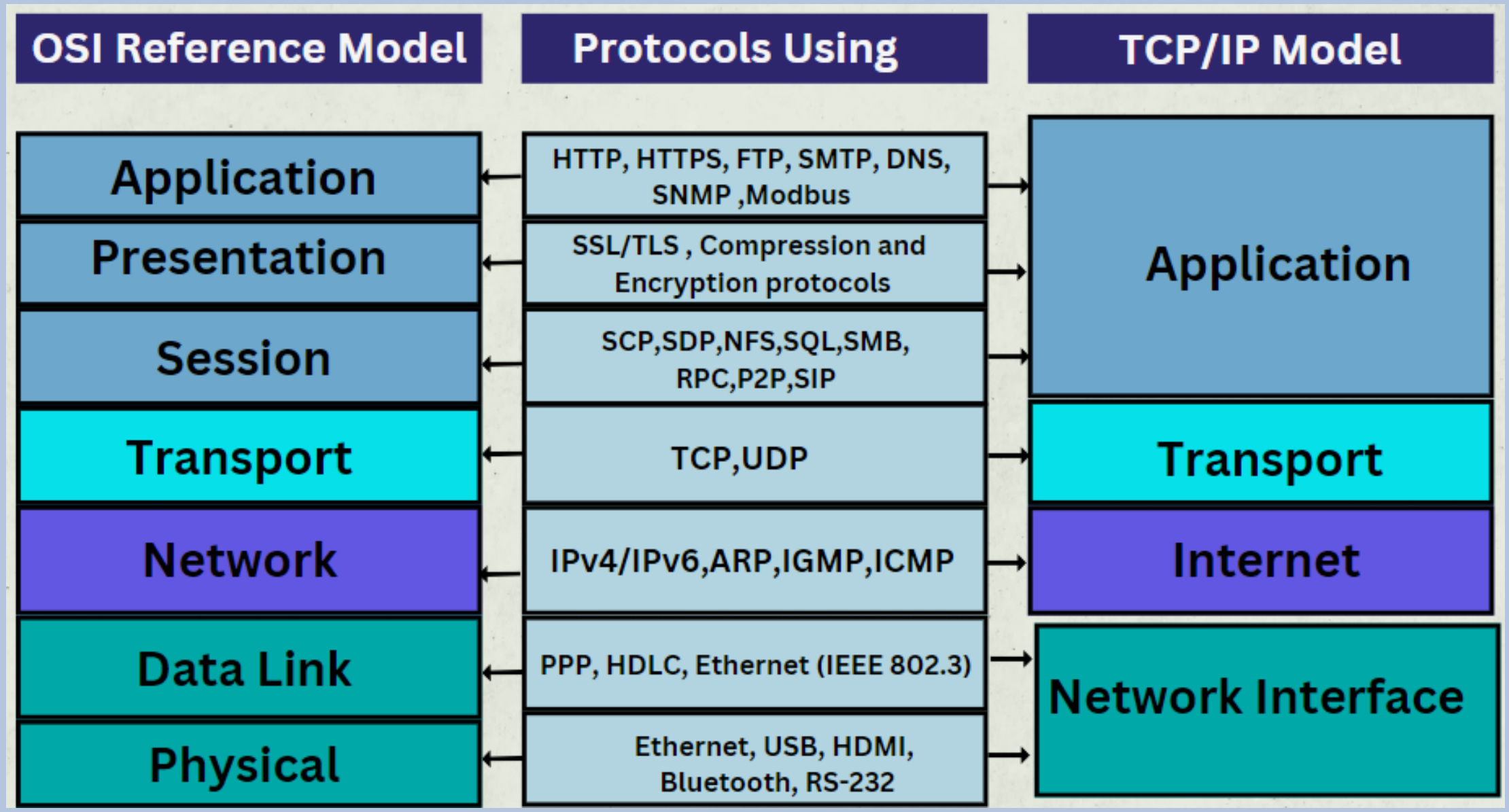
TOPICS

- Networking Models
- Network Protocols
- Ports and Protocols
- IP address , Subnet , CIDR and more
- Packet , Switch and Router
- Two-way and Three-way handshaking
- DNS
- DHCP
- Security Groups and firewalling
- HTTP
- SSL and TLS
- CDN
- VPN
- NFS
- Network Troubleshooting Tools

Networking Models



OSI Model Vs TCP/IP Model

1. **Application Layer:**
   - OSI Model: Serves as the interface between the user and the network, offering end-user services.
   - TCP/IP Model: Combines functionalities of OSI's Application and Presentation layers.
2. **Presentation Layer**:
   - OSI Model: Handles data translation, encryption, and compression.
   - TCP/IP Model: Functions integrated into the Application layer.
3. **Session Layer:**
   - OSI Model: Manages sessions or connections between applications.
   - TCP/IP Model: Functions incorporated into the Application layer.
4. **Transport Layer:**
   - OSI Model: Ensures end-to-end communication, managing flow control and error correction.
   - TCP/IP Model: Includes TCP for connection-oriented and reliable communication, and UDP for connectionless communication.
5. **Network Layer:**
   - OSI Model: Handles routing and forwarding of data between devices on different networks.
   - TCP/IP Model: Equivalent to the OSI Network Layer, managing logical addressing and routing. Uses the Internet Protocol (IP).
6. **Data Link Layer:**
   - OSI Model: Establishes a reliable link between directly connected nodes.
   - TCP/IP Model: Combines functionalities of OSI's Data Link and Physical layers. Concerned with link establishment and framing.
7. **Physical Layer:**
   - OSI Model: Manages the physical connection and transmission of raw binary data over a medium.
   - TCP/IP Model: Integrates functionalities of OSI's Data Link and Physical layers.

Similarities Between TCP/IP and OSI Model

- **Logical Networking Models:**
  - Both TCP/IP and OSI models offer structured and logical frameworks for networking.
- **Layered System:**
  - Both models use a layered system, dividing networking functionalities into distinct layers.
- **Specific Layer Functions:**
  - Each layer in both models has a designated and specific function, contributing to the overall communication process.
- **Ease of Issue Identification:**
  - The layered approach facilitates pinpointing issues during failures, as problems can be isolated to a specific layer.
- **Example Scenario:**
  - In both models, issues with data transmission to a hardware device can be traced by examining the data link layer (OSI) or the hardware layer (TCP/IP).

Differences Between the TCP/IP and OSI Model

- **Segmentation of Functions:**
  - OSI model segments functions into multiple layers, providing a more detailed breakdown.
  - TCP/IP model groups multiple functions into single layers, consolidating functionalities.
- **Complexity in TCP/IP:**
  - TCP/IP combines functions in the application and network access layers, potentially making troubleshooting and performance optimization more challenging.
- **Layered Approach in OSI:**
  - OSI model allows for focused troubleshooting by addressing specific layers, such as application, presentation, or session layers.
- **Confusion Potential in TCP/IP:**
  - In TCP/IP, functions of multiple OSI layers are combined within a single layer (e.g., application layer), potentially leading to ambiguity when diagnosing issues.
- **Example Scenario:**
  - In OSI, one can precisely address issues in the application, presentation, or session layers. In TCP/IP, the combined nature of these functions in the application layer may pose challenges in pinpointing specific problems.

Choosing Between TCP/IP and OSI Models

- **OSI Model Preference:**
  - Many users prefer the OSI model due to its segmentation of network functions into more layers.
  - This segmentation simplifies troubleshooting and enhances network performance management.
- **TCP/IP Preference:**
  - TCP/IP is favoured for its broader range of applications and widespread use in contemporary networking structures.
  - Common usage among teammates or administrators may make TCP/IP a preferable choice for practical reasons.
- **Preference Consideration:**
  - Administrators may choose between OSI and TCP/IP based on their familiarity with the model used by their team or peers.
  - The choice often depends on the specific needs and existing practices within a networking environment.

Network Protocols

A network protocol is a set of rules facilitating communication between devices over the internet. For successful communication, devices must support the same protocol, or a gateway can be used for translation.

Three main types of network protocols:

- **Network Management Protocols:** Define policies for monitoring and maintaining a network.
  Examples: SNMP, FTP, POP3, Telnet

- **Network Communication Protocols:** Establish rules and formatting for data exchange across a network.
  Examples: TCP, UDP, IP, HTTP, IRC, BGP, ARP.

- **Network Security Protocols:** Use security measures like cryptography and encryption to protect data.
  Examples: SFTP, SSL, HTTPS.

Network Protocol vs. Internet Protocol

Transmission Control Protocol (TCP) is a widely used network protocol known for breaking down data into packets for efficient transfer. In the TCP/IP model, TCP works in tandem with Internet Protocol (IP) to identify hosts and facilitate data transmission across the internet. IP assigns addresses, while TCP guides the data through the network to its destination, ensuring successful delivery.

Network Protocol vs. Communication Protocol

Communication protocols like TCP/IP and HTTP facilitate data exchange between devices, while network management protocols, such as SNMP, focus on managing and troubleshooting performance.
For instance, SNMP monitors and troubleshoots connections, providing administrators insights into infrastructure status. Communication protocols define formatting and syntax rules for data exchange, while network management protocols ensure efficient network performance.

Transmission Control Protocol (TCP) and Internet Protocol (IP)

TCP converts data into packets for secure and ordered transmission between a server and client. It ensures accurate and sequential delivery of content like files, text, images, and emails.

TCP establishes a connection through a three-way handshake:

1) The client or web browser transmits a Synchronize Sequence Number (SYN) to the destination server.

2) The destination server responds with an acknowledgment message, denoted as SYN-ACK.

3) The originating device receives the SYN-ACK message and responds with an ACK acknowledgment message, thereby completing the connection establishment.

User Datagram Protocol (UDP)

UDP, a communication protocol for network packet transmission, is favored by organizations seeking higher transfer speeds compared to TCP. Despite sacrificing some accuracy, UDP is well-suited for applications like video/audio streaming, online gaming, and VoIP calls, where tolerating some data loss is acceptable. Notably, UDP doesn't initiate a connection before transmitting packets and does not assure data delivery to the destination device.

File Transfer Protocol (FTP)

FTP, a network protocol for file transfer, operates over an unencrypted TCP/IP connection, enabling users to send up to 2GB of data using web browsers or FTP clients. Its efficiency in transferring large files quickly is valued by many organizations. However, the downside is its lack of security, as FTP transmits data in plain text. To address this, some organizations choose FTPS, a secure version of FTP that employs SSL encryption to protect transferred data, while maintaining the same functionality.

Hypertext Transfer Protocol (HTTP)

HTTP is a web communication protocol facilitating client-server interaction. A client sends a request to a server for webpage resources, and the server responds, allowing the client to load text, images, and videos.

HTTP request-response cycle is summarized as follows:

1. The client transmits an HTTP request message to the web server, seeking access to webpage content.
2. The web server handles the request message.
3. The web server dispatches a response message containing the requested content or webpage.
4. The client accepts the message and loads the content in the web browser for the end user to view.

Additionally, there exists a secure variant of HTTP known as HTTPS. It employs SSL/TLS encryption to secure requests and responses, preventing unauthorized access by third parties.

Simple Network Management Protocol (SNMP)

SNMP, an application layer protocol, gathers management information from devices like computers, routers, and printers. Network monitoring relies on SNMP to assess real-time performance and status across devices. The protocol involves an SNMP manager sending GET requests to SNMP-enabled devices, which have local SNMP agents collecting performance data and forwarding it to the manager. This approach provides administrators with a comprehensive overview of performance and status.

Internet Message Access Protocol (IMAP)

IMAP, a protocol for email retrieval, ensures emails remain on the remote server and are not downloaded locally when accessed. This allows users to check their emails from multiple devices. In contrast to POP3, which downloads emails locally, IMAP does not automatically delete emails from the server.

Internet Control Message Protocol (ICMP)

ICMP is a network protocol alerting devices to connectivity issues and errors, such as messages being too long or out of order, prompting resend requests. However, cybercriminals may exploit ICMP in ICMP flood attacks, overwhelming servers with bogus requests to divert computing resources from end users.

1. **Ping and Round-Trip Time (RTT):**
   - Ping is a troubleshooting tool that sends ICMP requests to a device.
   - It measures Round-Trip Time (RTT), the time for a device to respond.
   - RTT delay is indicative of connection quality.
2. **Traceroute and Network Routes:**
   - Traceroute employs ICMP to troubleshoot and measure network efficiency.
   - It reveals the time taken to traverse from one device to another.
   - Useful for identifying and analyzing the path of network packets.

Post Office Protocol (POP)

POP3 is a network protocol facilitating the retrieval of emails from a remote server to a local device. When the client connects to the server via TCP, it automatically downloads new messages, allowing offline access. Email platforms like Microsoft Outlook use POP3 to collect messages for offline availability. By default, downloaded emails are deleted from the server, but users can configure settings to retain them for a specific duration.

Simple Mail Transfer Protocol (SMTP)

SMTP, a mail delivery protocol, enables devices to send emails to remote endpoints via TCP. Providers like Microsoft Outlook, Gmail, and Yahoo Mail utilize SMTP for message delivery. Organizations establish an SMTP server for employees to connect to through a mail user agent (MUA) or email client, such as Gmail, to send emails. Unlike POP3, SMTP doesn't retrieve emails and doesn't automatically delete them.

How does the IP protocol manage error and control messages in a network?

Error and control messages within a network are managed by the Internet Protocol (IP) through the Internet Control Message Protocol (ICMP). ICMP, a supporting protocol, sends error messages and operational information indicating success or failure in communication with another IP address. IP doesn't directly handle error detection but relies on the Transmission Control Protocol (TCP) for this function. TCP ensures data integrity through checksums, acknowledges received packets, and retransmits lost ones. While IP doesn't inherently provide reliability, it serves as the foundation for dependable communication in higher-level protocols like TCP.

What is Telnet?

Telnet is the main Internet protocol for creating a connection to a remote server.

What is NNTP (Network News Transfer Protocol)?

NNTP is utilized for handling notes posted on Usenet newsgroups. NNTP servers globally manage Usenet newsgroup content. An NNTP client, also known as a newsreader and integrated into web browsers, interacts with these servers. The protocol operates on reserved port number 119.

Ports and Protocols

PORTS - A port serves as a digital identifier for a particular process or service actively running on a computer. These ports are assigned numerical values ranging from 0 to 65535, with certain numbers set aside for dedicated protocols or services.

PROTOCOLS - A protocol is set of rules and standards for data transmission among network devices. It specifies message format, encoding/decoding procedures, and addresses error handling.

Well Known Ports :

- Reserved ports in the range of **1 to 1,023**.
- Registered with IANA for specific services.

Registered Ports :

- Reserved ports in the range of **1,024 to 49,151.**
- Not as commonly utilized as Well Known Ports.

Dynamic and/or Private Ports :

- Reserved ports in the range of **49,152 to 65,535.**
- This port range for dynamic use, often for proprietary services or private use.

Port Number	Service Name	Protocol
20	FTP Data Transfer	TCP
21	FTP Control	TCP
22	SSH	TCP
23	Telnet	TCP
25	SMTP	TCP
53	DNS	TCP/UDP
67	DHCP servers to listen for client requests	UDP
68	DHCP clients to send requests to DHCP servers	UDP
80	HTTP	TCP
88	Kerberos Authentication System	TCP/UDP
110	POP3	TCP
119	NNTP	TCP
123	NTP	UDP
135	Microsoft RPC	TCP
137	NetBIOS Name Service	UDP
138	NetBIOS Datagram Service	UDP

Port Number	Service Name	Protocol
139	NetBIOS Session Service	TCP
143	IMAP	TCP
161	SNMP	UDP
389	LDAP	TCP/UDP
443	HTTPS	TCP
445	Microsoft SMB over TCP/IP	TCP
514	Syslog	UDP
636	LDAPS	TCP
993	IMAPS	TCP
995	POP3S	TCP
1433	Microsoft SQL Server	TCP
1521	Oracle SQL	TCP
3306	MySQL	TCP
3389	RDP	TCP
5432	PostgreSQL	TCP

What is the significance of port numbers?

Port numbers play a crucial role by identifying the particular application or service targeted by network traffic. This ensures accurate routing of network traffic to the designated destination, facilitating effective communication between applications and services across a network.

What are the common open ports?

FTP: 21; **SSH/SCP**: 22; **Telnet**: 23; **SMTP**: 25; **DNS**: 53; **POP3**: 110; **IMAP**: 145; **HTTP**: 80; **HTTPS**: 443; **MySQL**: 3306; **RDP**: 3389.

Is it possible to change port numbers?

Port numbers can be modified by adjusting the settings of the respective application or service, but it's crucial to check for potential conflicts with other applications or services using the chosen port.

IP Address

IP addresses are unique numerical identifiers assigned to devices connecting to the internet, facilitating communication between them. This 32-bit binary number is represented in decimal format for human readability.

An IP address is divided into four octets (e.g., 192.168.0.1), each representing a binary number ranging from 0 to 255. These octets compose two parts: the network portion and the host portion, determined by the subnet mask.

Difference between IPv4 and IPv6

IPv6 was invented primarily to address the limitations of IPv4 and to accommodate the growing number of devices connecting to the internet.

IPv4 offering about 4.3 billion unique addresses, shown in decimal format as four groups of numbers. IPv6 addresses utilize 128 bits, providing 340 undecillion unique addresses, presented in hexadecimal format as eight groups of four digits.

Classes of IP addresses:

Class	IP Range
A	0.0.0.0 – 127.255.255.255
B	128.0.0.0 – 191.255.255.255
C	192.0.0.0 – 223.255.255.255
D	224.0.0.0 – 239.255.255.255
E	240.0.0.0 – 255.255.255.255

The five classes of IP addresses, Class A through E, are categorized based on the range of addresses and their intended purposes. The first octet of an IP address determines its class:

- **Class A:**
  - Range: 0--- (0.0.0.0 to 127.255.255.255).
  - The first bit is "0".
  - Supports up to 126 networks, each with up to 16,777,214 hosts.
- **Class B:**
  - Range: 10-- (128.0.0.0 to 191.255.255.255).
  - The first bit is "1" and the second bit is "0".
  - Allows for up to 16,384 networks, each with up to 65,534 hosts.

- **Class C:**
  - Range: 110- (192.0.0.0 to 223.255.255.255).
  - The first two bits are "1" and the third bit is "0".
  - Supports up to 2,097,152 networks, each with up to 254 hosts.
- **Class D:**
  - Range: 1110 (224.0.0.0 to 239.255.255.255).
  - The first three bits are "1" and the fourth bit is "0".
  - Used for multicasting, not for network addressing.
- **Class E:**
  - Range: 1111 (240.0.0.0 to 255.255.255.255).
  - The first four bits are "1".
  - Reserved for experimental purposes, not for general networking.

Each class has a specific range of addresses and serves different purposes, from addressing individual hosts to multicast groups and experimental use.

The traditional division of IPv4 address space into classes (A, B, and C) was initially designed to accommodate networks of varying sizes. However, this class-based approach has become outdated, primarily due to the rapid depletion of IPv4 addresses. As a result, more flexible allocation methods like CIDR have been adopted, rendering classful addressing a legacy concept.

Public IP address - assigned by IANA and allocated to ISPs and organizations, are globally unique and routable on the internet.

The public IP address range - five classes:

- **Class A:**
  - 1.0.0.0 to 9.255.255.255
  - 11.0.0.0 to 126.255.255.255
- **Class B:**
  - 128.0.0.0 to 172.15.255.255
  - 172.32.0.0 to 191.255.255.255
- **Class C:**
  - 192.0.0.0 to 192.167.255.255
  - 192.169.0.0 to 223.255.255.255

Private IP address - designated for internal networks and non-routable on the internet, are assigned by network administrators.

The private IP address range - three classes:

- **Class A:**
  - 10.0.0.0 to 10.255.255.255
- **Class B:**
  - 172.16.0.0 to 172.31.255.255
- **Class C:**
  - 192.168.0.0 to 192.168.255.255

Netmasks and Subnets

- **Subnetting:** Dividing a network into smaller sections to enhance efficiency and management.
- **Netmask**: Specifies the number of bits used for the network portion of an IP address.
- **Subnet mask**: Another netmask used to further divide a network into smaller subsections, allowing for finer segmentation and resource management.

CIDR Notation

CIDR (Classless Inter-Domain Routing) notation expresses the division between network and host portions of an IP address. It uses a forward slash followed by a number to denote the number of network bits. A higher number indicates fewer available host addresses.

Loopback Address

The class A network 127.0.0.0 (CIDR notation 127.0.0.0/8) is reserved for loopback. IP packets with source addresses in this network should never leave a host. Any packets received on a non-loopback interface with a loopback source or destination address should be discarded.

Gateway IP

A gateway IP is the address of the router connecting a local network to external networks like the internet, acting as the entry and exit point for network traffic. Typically, it's the first usable address in a subnet, obtained by setting the host portion to zeros in the network address.

Example: For the IP address 10.0.0.0/24, the network portion is "10.0.0," and the host portion is "0." Therefore, the gateway IP for this subnet is 10.0.0.1.

Broadcast IP

A broadcast IP address sends data to all devices in a subnet and is the highest address in that subnet. To find it, set all bits in the host portion to ones while keeping the network portion unchanged.

Example: In subnet 10.0.0.0/24, with network portion "10.0.0.," the broadcast IP is 10.0.0.255.

Host Calculation

To calculate the number of hosts that can be assigned to a network, use the following formula:

Number of Hosts=2^(Number of Host Bits)−2

Where:

- Number of Host Bits = Total bits in the host portion of the IP address.

Remember to subtract 2 from the result to account for the network address (all zeros) and the broadcast address (all ones), which cannot be assigned to hosts.

For example, in a network with a subnet mask of /24 or 255.255.255.0 (which means there are 24 bits allocated for the network portion), leaving 8 bits for the host portion:

Number of Hosts=2^8−2=256−2=254

So, 254 hosts can be assigned to this network.

Calculate Network Address and Broadcast Address

To calculate the network address:

1. Write the given IP address and subnet mask in binary format.
2. Perform a logical AND operation between corresponding octets of the IP address and subnet mask.
3. Convert the result back to decimal format; this is the network address.

To calculate the broadcast address:

1. Write the given IP address in binary format.
2. Write the inverse of the subnet mask in binary form.
3. Perform a logical OR operation between corresponding octets of the IP address and the inverse of the subnet mask.

Example - IP address 192.168.5.50 with a subnet mask of /28(255.255.255.240)

Network Address

IP Address in Decimal notation	192	168	5	50
Binary equivalent IP address(A)	11000000	10101000	00000101	00110010
Binary equivalent Subnet mask(B)	11111111	11111111	11111111	11110000
A AND B	11000000	10101000	00000101	00110000
Network Address	192	168	5	48

Broadcast Address

IP Address in Decimal notation	192	168	5	50
Binary equivalent IP address(A)	11000000	10101000	00000101	00110010
Inverse of Subnet mask(B)	00000000	00000000	00000000	00001111
A OR B	11000000	10101000	00000101	00111111
Broadcast Address	192	168	5	63

Subnet Mask Values

CIDR	SUBNET MASK	Number of IP ADDRESSES	Number of USABLE IP ADDRESSES
/32	255.255.255.255	1	1
/31	255.255.255.254	2	2*
/30	255.255.255.252	4	2
/29	255.255.255.248	8	6
/28	255.255.255.240	16	14
/27	255.255.255.224	32	30
/26	255.255.255.192	64	62
/25	255.255.255.128	128	126
/24	255.255.255.0	256	254
/23	255.255.254.0	512	510
/22	255.255.252.0	1,024	1,022
/21	255.255.248.0	2,048	2,046
/20	255.255.240.0	4,096	4,094
/19	255.255.224.0	8,192	8,190
/18	255.255.192.0	16,384	16,382
/17	255.255.128.0	32,768	32,766
/16	255.255.0.0	65,536	65,534
/15	255.254.0.0	131,072	131,070
/14	255.252.0.0	262,144	262,142
/13	255.248.0.0	524,288	524,286
/12	255.240.0.0	1,048,576	1,048,574
/11	255.224.0.0	2,097,152	2,097,150
/10	255.192.0.0	4,194,304	4,194,302
/9	255.128.0.0	8,388,608	8,388,606
/8	255.0.0.0	16,777,216	16,777,214
/7	254.0.0.0	33,554,432	33,554,430
/6	252.0.0.0	67,108,864	67,108,862
/5	248.0.0.0	134,217,728	134,217,726
/4	240.0.0.0	268,435,456	268,435,454
/3	224.0.0.0	536,870,912	536,870,910
/2	192.0.0.0	1,073,741,824	1,073,741,822
/1	128.0.0.0	2,147,483,648	2,147,483,646
/0	0.0.0.0	4,294,967,296	4,294,967,294

Packet:

Think of a packet as a parcel containing data. It's like a letter you send through the postal service. The data you want to send gets divided into packets before traveling across a network.

Switch:

Imagine a switch as a post office sorting facility. When packets arrive at a switch, it reads the address (MAC address) on each packet and decides where to send it next within the local network. It's like the switchboard operator connecting calls to the right extensions.

Router:

Now, picture a router as a postal worker delivering parcels to different cities. Routers connect different networks (like cities), directing packets between them based on their IP addresses. Just as a postal worker sorts mail to send it to the correct city, a router routes packets to the appropriate network.

Understanding Data Transmission in Networks

Let's explore how data moves across networks from the left host to the right host. We have two networks: one on the left with IP address 192.168.100.0 and another on the right with private IP address 10.10.10.0, connected via a router. Each host on these networks has specific IP and MAC addresses. The left host has IP 192.168.100.2 with MAC address A, and the right host has IP 10.10.10.2 with MAC address B. Routers, acting as default gateways, enable communication between networks. For instance, when the left computer needs to communicate with the right network, it reaches out to its default gateway, the router.

Switch

The switch examines the frame and identifies the destination MAC address. It forwards the frame to the port connected to that MAC address. The frame reaches the router, which analyzes the frame's MAC header. Upon discovering the IP packet with the destination IP address, the router creates a new frame with the appropriate MAC address for the destination host. The router sends the frame back to the switch, which forwards it to the port connected to the host. The host receives the frame, opens it, and processes the data contained within the IP packet. Finally, it discards the packet and frame, retaining only the data for further analysis or application processing.

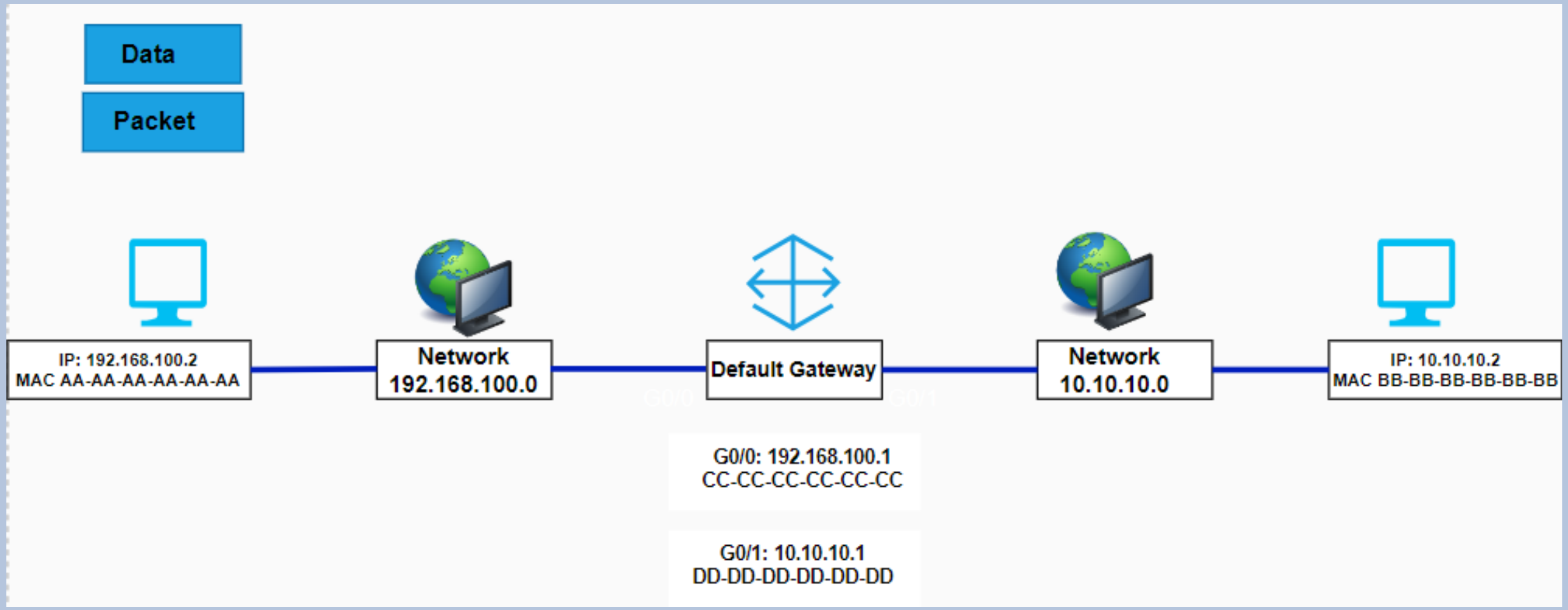
ROUTER

The router serves as a gateway to connect you to external networks such as the internet. In your IP configuration, the router is crucial. Each interface of the router is assigned additional IP addresses and MAC addresses. Its primary function is to route traffic between different networks, with each IP address having an associated MAC address for its respective interface. This relationship becomes significant as traffic traverses through the network.

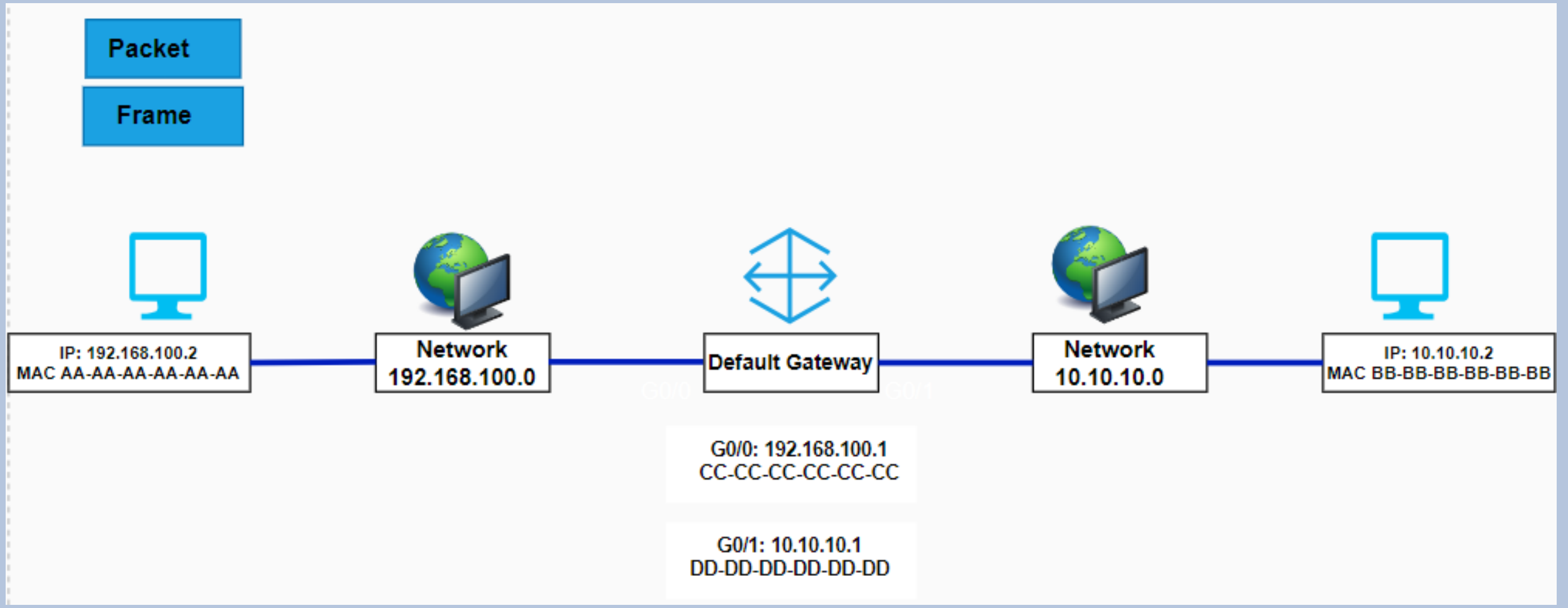
Host Creates a Packet

We've identified two separate local area networks. The host on the left intends to send data to the host on the right. Before transmitting, it must create an IP packet. This packet includes the destination address (10.10.10.2) and its own IP address as the source (192.168.100.0). With the packet prepared, the next step is to encapsulate it into a frame for transmission over the Ethernet LAN.

Host Creates a Packet

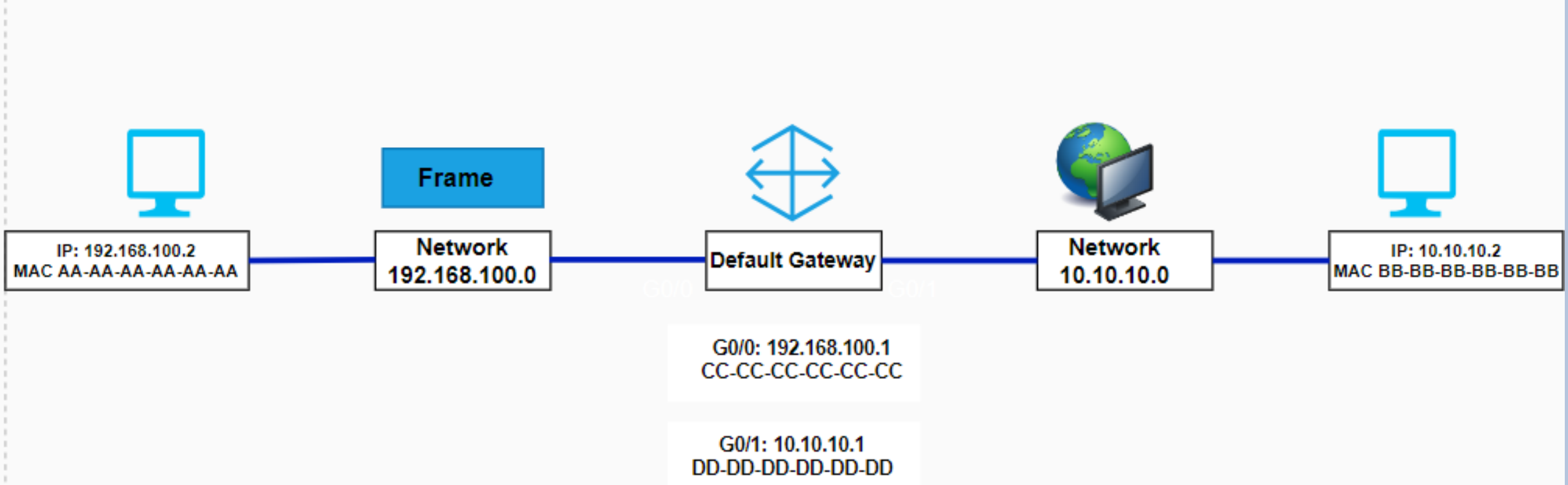


Packet Creates A Frame



Packet Routing

The packet forms a frame, inserting the destination MAC address. However, it lacks the MAC address of the computer 10.10.10.2 as it's on a different network. Instead, it uses the MAC address of its default gateway, known to be the G0/0 interface. The frame reaches the default gateway, guided by its MAC address. Then, the packet is encapsulated into the frame. In summary, the data is placed into the packet, then the packet into the frame. With a destination MAC address, the frame is sent to the LAN and forwarded to the switch by the computer.



Two-way handshaking:

This method entails a straightforward exchange of messages between two devices to acknowledge and verify the connection. For instance, in TCP/IP communication, when a client sends a SYN packet to the server, the server responds with a SYN-ACK packet, thereby confirming the connection.

Example: Imagine this scenario: when you send a friend a text message asking if they're available to chat, and they respond with a quick "Yes," confirming the connection.

Three-way handshaking:

This process is more elaborate, involving three sequential steps to establish a secure connection. In TCP/IP, it commences with a SYN packet from the client to the server, followed by a SYN-ACK packet from the server acknowledging the request. Finally, an ACK packet from the client confirms receipt, ensuring both parties are prepared to exchange data.

Example: Three-way handshaking: Imagine you're arranging a meeting with a colleague. You send them a message suggesting a time, they reply with their availability, and you confirm the finalized time.

DNS

The Domain Name System (DNS) functions as the internet's directory, converting user-friendly domain names into machine-readable IP addresses. This translation process facilitates smooth communication between individuals and online services, enabling activities such as website browsing and email transmission.

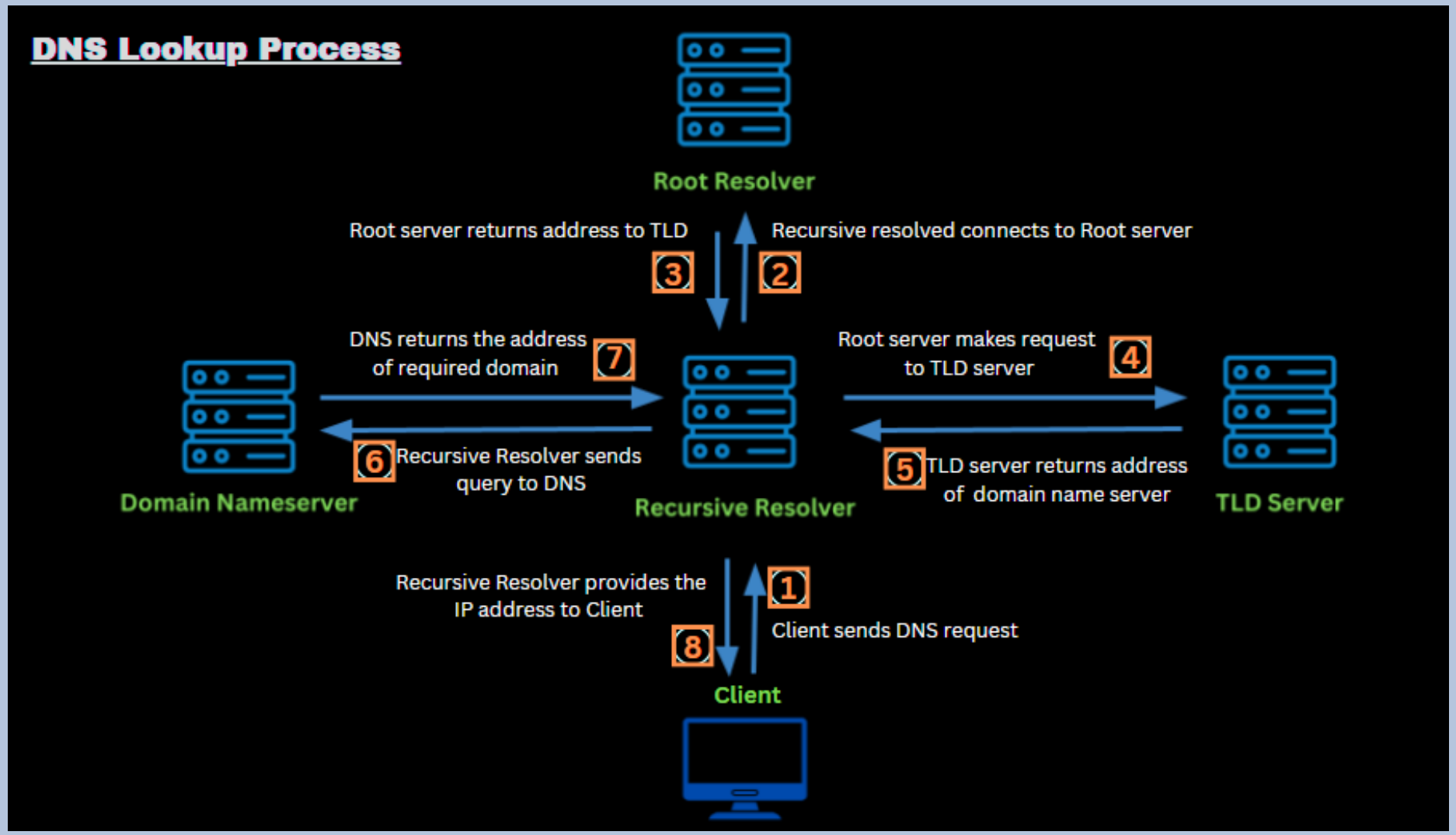
DNS employs a hierarchical structure, comprising Recursive Resolvers, Root Servers, Top-Level Domain (TLD) Servers, and authoritative Nameservers, to efficiently resolve domain names to their respective IP addresses.

DNS Lookup Process

1- The client sends a query to a Recursive Resolver.
2- The Recursive Resolver establishes a connection with a Root Server.
3- The Root Nameserver then provides the resolver with the address of a Top-Level Domain (TLD) Server (such as .com or .net).
4- The Root Resolver forwards a request to the TLD Server.
5- The TLD Server returns the IP address of the Domain Nameserver, which stores the information about the requested domain.
6- The Recursive Resolver sends a query to the Domain Nameserver.

7- The Domain Nameserver returns the IP address for the requested domain to the Recursive Resolver.

8- Finally, the Recursive Resolver provides the client with the IP address of the requested domain.



DHCP

DHCP (Dynamic Host Configuration Protocol) is a network management protocol used to automatically assign IP addresses and other network configuration parameters to devices on a network.

Understanding DHCP Client-Server Communication:

DHCP Discover Message –This message is produced by the client host to discover if there are any DHCP servers present in a network or not.

DHCP Offer Message – The DHCP server responds to the host, providing the assigned IP address and TCP configuration; if multiple DHCP servers exist, the client accepts the first DHCP OFFER message received.

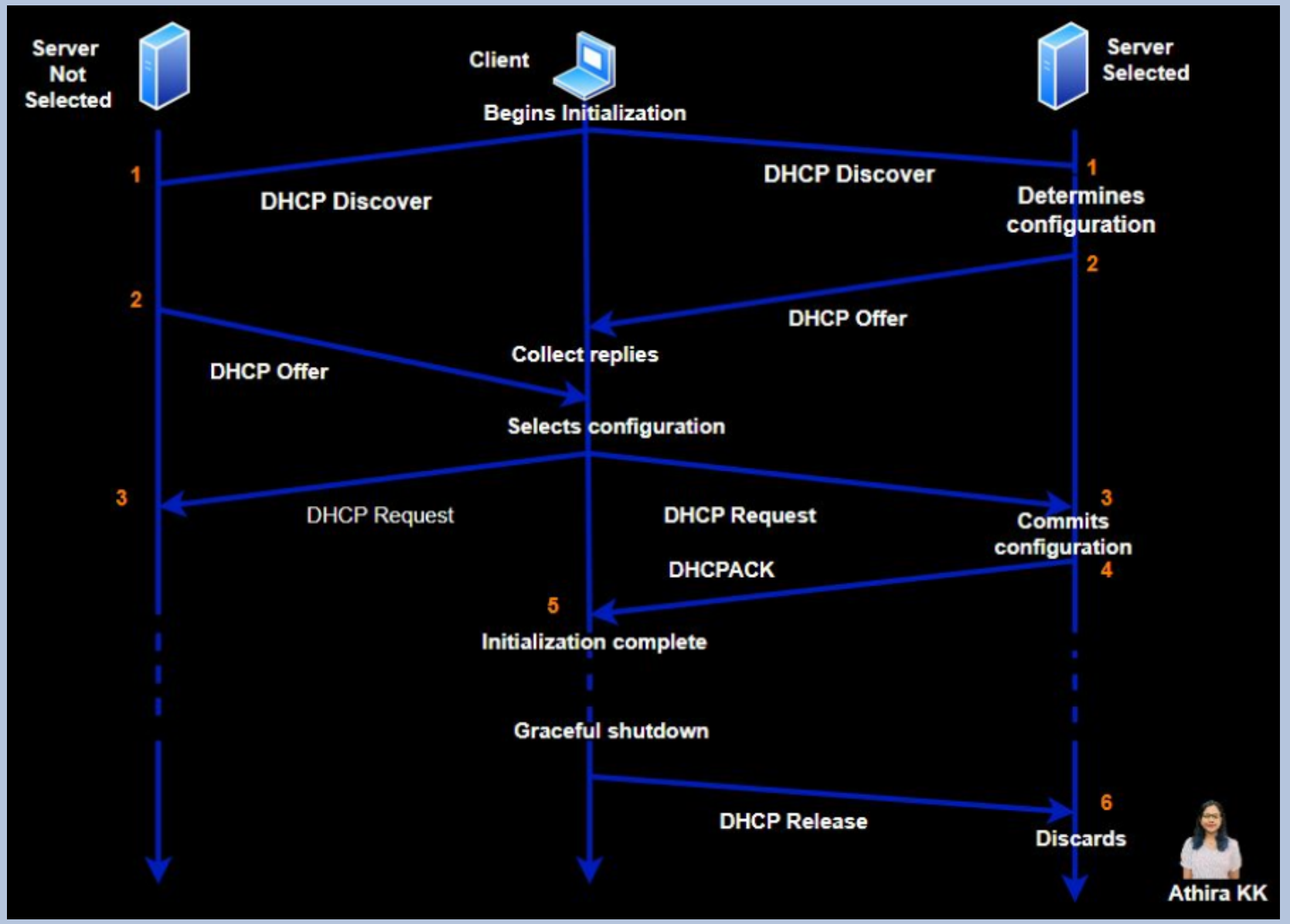
DHCP Request Message – The client receives and compares DHCP offer messages, selects a server, sends a DHCP Request message indicating its choice, and broadcasts this selection to all DHCP servers on the network.

DHCP Acknowledgment Message – When a server receives a DHCP Request message, it leases the address, returns unselected addresses to the pool, and sends the client an acknowledgment with configuration details, allowing the client to use the IP address until lease expiration or until releasing it with a DHCP Release message.

DHCP Request, DHCP ACK Message – The client initiates lease renewal halfway through its duration by sending a DHCP Request to the server, which, if accepted, triggers a DHCP ACK; if unanswered, the client may continue using the IP until lease expiration, bypassing the need for DHCP Discover and Request until renewal or lease expiry.

DHCPRELEASE Message - The client terminates the lease with a DHCP Release message to the server, prompting the return of its IP address to the available pool and cancellation of any remaining lease time.

The below diagram shows the interaction between clients and servers involves the exchange of DHCP messages, forming the foundation for establishing network connections.


Server Not Selected
Client
Server Selected
Begins Initialization
1
DHCP Discover
DHCP Discover
1
Determines configuration
2
2
DHCP Offer
DHCP Offer
Collect replies
Selects configuration
3
DHCP Request
DHCP Request
3
Commits configuration
DHCPACK
4
5
Initialization complete
Graceful shutdown
DHCP Release
6
Discards
Athira KK

Security Groups and Firewalling

Firewall: Imagine a firewall as a digital barrier watching over the traffic coming in and going out of our network. It works like a gatekeeper, checking the traffic against set rules to stop unauthorized access and protect against cyber threats. Firewalls are flexible and can be set up in different ways, like using special devices or software. They're super important for keeping sensitive data safe and making sure everyone follows security rules.

Security Groups: Security groups, which we often find in cloud services like AWS, Azure and GCP, are like virtual versions of firewalls. They control the traffic going to and from our cloud resources.
Security groups give us the power to decide exactly who or what can talk to their cloud stuff, like which IP addresses, protocols, and ports are allowed. By setting up security groups, we can make sure their cloud apps and services stay safe from cyberattacks by limiting who can access them and how.

Security Groups and Firewall Rules for Instance Setup:
When we set up an instance, we can assign it one or more security groups. These groups have firewall rules that are more important than the default ones, which are:

- All outgoing traffic is allowed.
- No incoming traffic is permitted.

If we use the default security group without making any changes, our new instance won't be reachable from outside.
To be able to ping or connect to our instance using SSH, we need to create incoming rules for it.

HTTP

HTTP, or Hypertext Transfer Protocol, is essential for internet data exchange. It allows clients to communicate with servers, retrieving resources like HTML documents, images, and APIs. Using a request-response model, clients send requests to servers, which respond with the requested data. HTTP employs methods like GET, POST, PUT, and DELETE for resource manipulation. Response codes indicate request status, such as success, redirection, client error, or server error. Overall, HTTP is fundamental to web communication and shapes modern internet interactions.

HTTP Request

An HTTP request is a request message from a client to a server asking for access to a resource.
Every HTTP request sent over the Internet carries encoded data containing various types of information.
A standard HTTP request includes:

- The HTTP version type
- A URL
- An HTTP method
- HTTP request headers
- An optional HTTP body.

HTTP Methods

HTTP methods, specify the action an HTTP request expects from the server. For instance, 'GET' retrieves information (like a webpage), while 'POST' sends data to the server (e.g., form submissions).

Method Name	Description
GET	Initiates a request for object data, with the response body containing the requested information.
HEAD	Resembles GET, lacking a response body, primarily used for metadata or testing purposes.
POST	Transmits changes to an object.
PUT	Replace an existing object with a new one.
PATCH	Updates an object's content.
OPTIONS	Defines the communication possibilities for an object.
DELETE	Deletes an object.

HTTP Request Headers

HTTP headers, structured as key-value pairs, convey essential details in every HTTP request and response. They transmit vital data like the client's browser type and the requested information.

HTTP Request Body

The HTTP request body carries the payload of data sent to the web server, including form submissions like usernames, passwords, or any other input data.

HTTP Response

An HTTP response is the information that web clients, often browsers, receive from an Internet server in response to an HTTP request. This response conveys valuable data corresponding to the request made.

A standard HTTP response includes:

- an HTTP status code
- HTTP response headers
- an optional HTTP body

HTTP Status Codes

An HTTP status code is a three-digit number a server generates in response to a browser's request.

Five distinct categories of HTTP status codes

(The "xx" refers to different numbers between 00 and 99.)

Status Code	Description	Context
1XX	Informational codes	The server acknowledges and is processing the request.
2XX	Success codes	The server successfully received, understood, and processed the request.
3XX	Redirection codes	The server received the request, but there’s a redirect to somewhere else (or, in rare cases, some additional action other than a redirect must be completed).
4XX	Client error codes	The server couldn’t find (or reach) the page or website. This is an error on the site’s side.
5XX	Server error codes	The client made a valid request, but the server failed to complete the request.

HTTP Response Headers

An HTTP response includes headers that communicate crucial details like the language and data format transmitted within the response body.

HTTP Response Body

Typically, when 'GET' requests are successful in HTTP, the response body carries the requested data, commonly in the form of HTML content interpreted by web browsers to display as webpages.

⭐**Complete List of HTTP Status Codes**

- **1XX — Informational codes**

Status Code	Description
100	Continue
101	Switching Protocols
102	Processing
103	Early Hints

- **2XX — Success Codes**

Status Code	Description
200	OK
201	Created
202	Accepted
203	Non-Authoritative Information
204	No Content
205	Reset Content
206	Partial Content
207	Multi-Status
208	Already Reported
226	IM Used

- **3XX — Redirection Codes**

Status Code	Description
300	Multiple Choices
301	Moved Permanently
302	Found
303	See Other
304	Not Modified
307	Temporary Redirect
308	Permanent Redirect

- **4XX — Client Error Codes (In next page)**

Status Code	Description
400	Bad Request
401	Unauthorized
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Timeout
409	Conflict
410	Gone
411	Length Required
412	Precondition Failed
413	Content Too Large
414	URI Too Long
415	Unsupported Media Type

Status Code	Description
416	Range Not Satisfiable
417	Expectation Failed
421	Misdirected Request
422	Unprocessable Content
423	Locked
424	Failed Dependency
425	Too Early
426	Upgrade Required
428	Precondition Required
429	Too Many Requests
431	Request Header Fields Too Large
451	Unavailable for Legal Reasons

- **5XX — Server Error Codes**

Status Code	Description
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout
505	HTTP Version Not Supported
506	Variant Also Negotiates
507	Insufficient Storage
508	Loop Detected
511	Network Authentication Required

Is it possible to initiate DDoS attacks via HTTP?

- DDoS attacks can be launched over HTTP.
- These attacks flood a web server with a large number of HTTP requests, causing overload.
- This renders the server unable to respond to legitimate user requests.
- Attackers use techniques like botnets or amplification to generate large volumes of HTTP traffic.
- HTTP is a "stateless" protocol where each command runs independently.
- In HTTP 1.1 and above, persistent connections allow multiple requests over a single TCP connection.
- HTTP requests in large quantities can be part of application layer attacks or layer 7 attacks.

SSL - Secure Socket Layer

- Supports the Fortezza algorithm.
- The Message digest is used to create a master secret.
- The Message Authentication Code protocol is used.
- More complex than TLS (Transport Layer Security).
- Less secured as compared to TLS.
- Less reliable and slower.
- SSL has been deprecated.
- SSL uses port to set up explicit connection.
- SSL is obsolete (all versions), no more recommended for use.

TLS - Transport Layer Security

- Does not support the Fortezza algorithm.
- A Pseudo-random function is used to create a master secret.
- Hashed Message Authentication Code protocol is used.
- TLS (Transport Layer Security) is simple.
- Provides high security.
- Highly reliable and upgraded. It provides less latency.
- TLS is still widely used.
- TLS uses protocol to set up implicit connection.
- TLS 1.3 is the latest version.

SSL

SSL, or Secure Sockets Layer, is a protocol ensuring the confidentiality and integrity of data transmitted over the Internet.

Here's how SSL works:

1. A user connects to an HTTPS-enabled website.
2. The browser requests the server's public key and exchanges it for its own.
3. Both parties use their private and public keys to encrypt and decrypt messages, ensuring only the intended recipient can read them.
4. Messages sent from the user's browser are encrypted using the server's private key, and vice versa.
5. This encryption process, combined with unique keys generated for each session, prevents third parties from intercepting data during transmission.

TLS

TLS, or Transport Layer Security, comprises protocols for securing web pages and the transport layer. It utilizes symmetric and asymmetric cryptography for data transmission security and efficiency.

Symmetric encryption employs secret keys shared between parties, typically 128 bits long, ensuring message encryption and decryption. However, securely sharing these keys poses challenges.

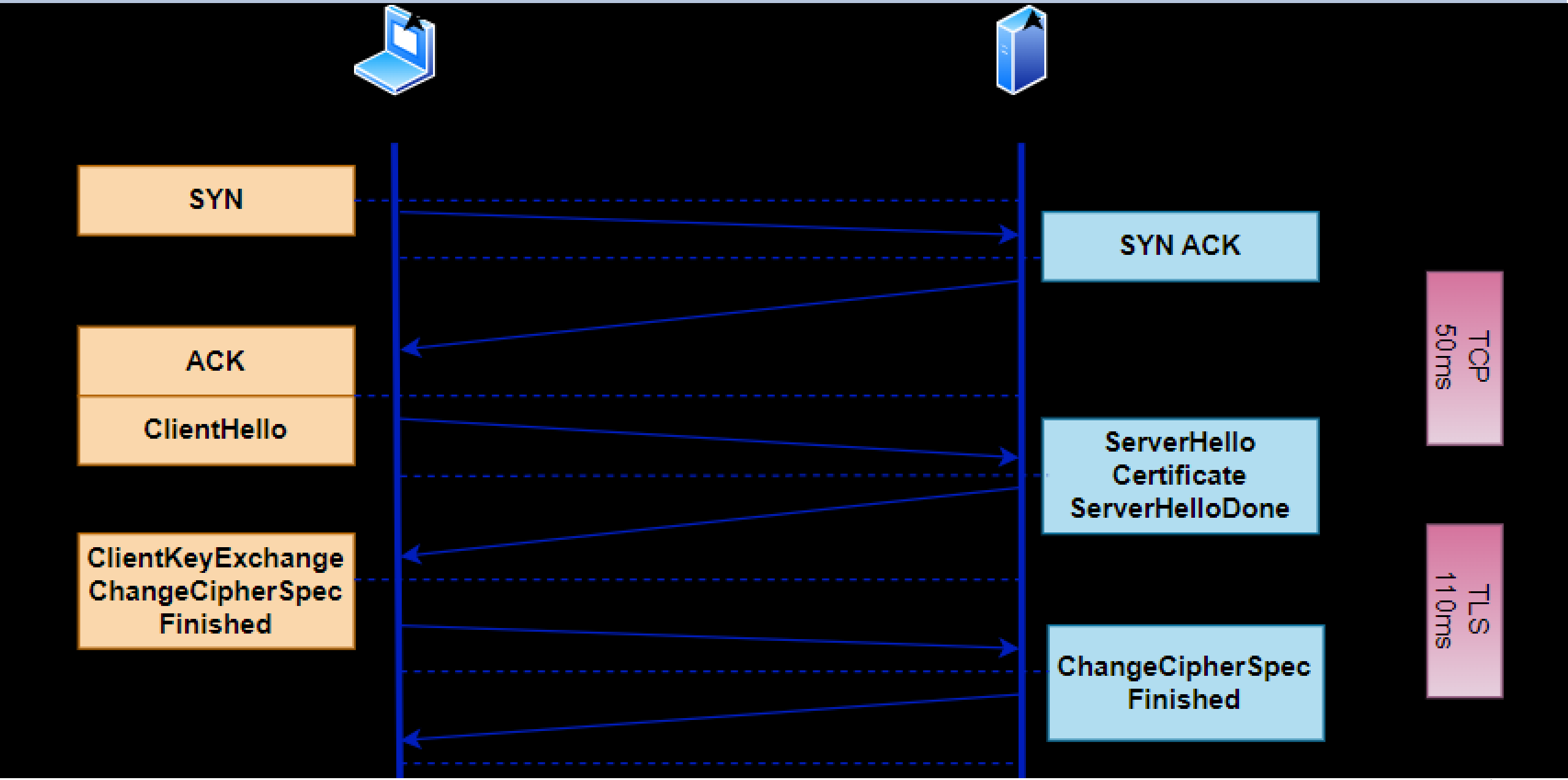
Asymmetric encryption doesn't require shared secure channels; parties can exchange public keys via email, for instance. But it requires larger keys for enhanced security, making it computationally intensive.

TLS ensures identity verification and detects tampering by having clients validate server public keys. Connections rely on X.509 digital certificates issued by Certificate Authorities for authenticity.

Important Terms of TLS

1. **Asymmetric Encryption:** Asymmetric or "public key" cryptography employs two interrelated keys: a public key and a private key. Information encrypted with a public key can only be decrypted with its corresponding private key. Similarly, a digital signature produced with a private key can be authenticated using the corresponding public key.
2. **Symmetric Encryption:** Symmetric encryption utilizes a single key for both encryption and decryption operations. This approach is preferred for its efficiency compared to asymmetric cryptography. During the TLS handshake, a shared symmetric encryption key is established to facilitate secure communication.
3. **Cipher Suites:** Cipher suites are configurations of cryptographic algorithms implemented within the TLS protocol. These include algorithms for asymmetric encryption during the handshake, symmetric encryption for data transmission, digital signature algorithms, and hash functions to ensure data integrity during transit.
4. **Digital Certificate:** A digital certificate serves as proof of ownership for a public key. Servers present digital certificates during the TLS handshake to verify their identity to the client, ensuring secure and authenticated communication.

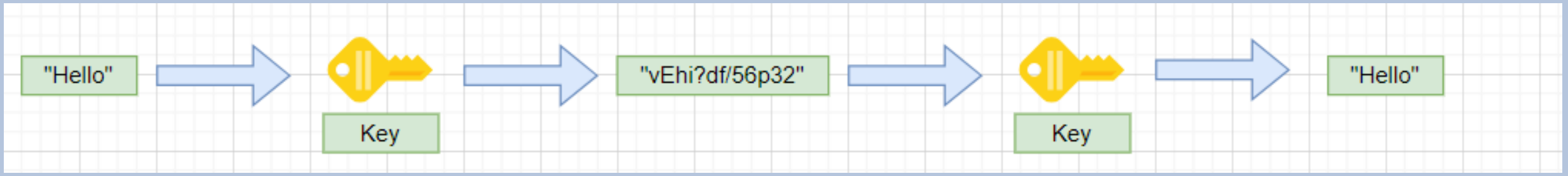
TLS handshake depend on Asymmetric encryption



TLS handshake depend on Asymmetric encryption

1. **Client Hello:** The client initiates the handshake by sending configuration data to the server, including supported TLS versions, cipher suites, and a random value called "client random."
2. **Server Hello:** The server responds with its chosen TLS version, cipher suite, digital certificate, and its own random value, "server random."
3. **Premaster Secret:** After verifying the server's digital certificate, the client generates a random value known as the premaster secret. This value is encrypted with the server's public key and sent to the server.
4. **Session Keys:** The server decrypts the premaster secret using its private key. Both client and server use the client and server random values, along with the premaster secret, to calculate a shared session key. As the premaster secret was encrypted during transmission, the session key remains secret.
5. **Client Finished:** To finalize the handshake, the client sends an encrypted message to the server using the session key. The server decrypts and verifies the message's integrity, confirming the client's ability to calculate the session key correctly.
6. **Server Finished:** Similarly, the server sends an encrypted message to the client using the shared session key. Decrypting and verifying this message assures the client of the server's ability to calculate the session key accurately.

Symmetric encryption with session keys



In symmetric encryption, both parties in a conversation utilize the same key. Following the TLS handshake, both sides employ identical session keys for encryption purposes. Once session keys are activated, the public and private keys become inactive. Session keys, being temporary, are discarded once the session concludes and are not reused thereafter. Subsequent sessions necessitate the creation of a fresh set of random session keys.

Data Encryption:

SSL and TLS protocols enable secure communication between server and client endpoints, ensuring that data exchanged remains confidential and unaltered by third parties. This encryption safeguards sensitive information like passwords and credit card numbers from exposure during transmission.

Identification:

Encryption technology, underpinned by SSL/TLS protocols based on public-key cryptography, secures information exchanges between the client (browser) and the server. This established secure connection, fortified by years of key exchange testing, resists brute force attacks and hacking attempts. The TLS connection's true security hinges on its configuration, including parameters such as secure version usage and certificate authentication. The server's SSL/TLS certificate, validated by a Certificate Authority, confirms its identity with certainty, establishing a fully secure connection.

CDN (Content Delivery Network)

A CDN is a global network of servers strategically located to speed up web content delivery by bringing it closer to users. Using caching techniques, CDNs store duplicate files worldwide, enabling fast access to internet content from nearby servers. This ensures swift page loading times and enhances the web browsing experience, facilitating activities like streaming, downloading, banking, social media, and online shopping without delays.

Example: Imagine you're in Tokyo and you're trying to access a popular restaurant's website in Paris, which is hosted on a server in France. If the request had to travel all the way from Tokyo to Paris, you'd likely experience slow loading times. However, a CDN would store a cached version of the restaurant's website in various locations worldwide, called Points of Presence (PoPs). These PoPs have their own caching servers and ensure that you can access the content quickly from a location close to you in Tokyo.

Amazon CloudFront, a CDN service from Amazon Web Services, swiftly delivers web pages, videos, images, and more to users with minimal delay and fast speeds. It stores copies of content near users through a global network of edge locations, cutting retrieval time and enhancing web app performance. CloudFront seamlessly works with other AWS services, providing dynamic content delivery, instant analytics, and robust security for content in transit

Mechanism of a CDN

A content delivery network relies on three types of servers:

- **Origin servers:** This hold the original content and are where updates are made. They can be owned by the content provider or hosted by third-party cloud providers like Amazon's AWS S3 or Google Cloud Storage.

- **Edge servers:** Spread across various locations worldwide, also known as "points of presence" (PoPs), these servers cache content from origin servers. They deliver this content to users nearby. When users request content, they are directed to a nearby cached copy on an edge server. If the cached content is outdated, the edge server requests updated content from the origin server. CDN edge servers are managed by the CDN hosting provider.

- **DNS servers:** These servers manage and provide IP addresses for both origin and edge servers. When a client requests content from an origin server, DNS servers respond with the name of an edge server that can serve the content faster.

To guarantee a smooth experience while browsing websites , **CDNs perform two primary functions:**

- CDN make things faster by reducing the annoying wait time, called **latency**. This happens when the stuff you want to see takes too long to reach your device. CDNs fix this by putting the content closer to you, so it arrives quicker and more reliably.

- CDN spread out internet traffic evenly, called load balancing, like how roads manage traffic. If one route gets too busy, they send some cars down different roads to keep things moving smoothly. This balancing act helps websites handle lots of visitors without crashing or slowing down.

Benefits of CDN

- Enhance performance
- Enhance security
- Ensure availability
- Improve customer experience
- Offload traffic
- Reduce bandwidth costs

VPN (Virtual Private Network)

VPN create a secure connection when you're using public networks. It encrypts your internet traffic and hides your online identity, making it harder for others to spy on you or steal your information. This encryption happens instantly as you browse the web.

When you activate a VPN, it establishes a secure connection (often referred to as a "tunnel") between your device and a distant server managed by the VPN provider. Your internet activity is directed through this tunnel to the server, which forwards it to the public internet as usual. Any data returning to your device follows the same path: from the internet, to the VPN server, through the encrypted connection, and finally back to your device.

Example: Imagine you're sending a letter to a friend. Instead of just sealing the envelope and dropping it in the mailbox, you put it in a secure locker at the post office. This locker acts like a VPN server. Your letter is encrypted and sent to the locker, where it's forwarded to your friend's address. When your friend responds, their letter goes through the same process: from their address to the locker, then decrypted and delivered to you. This added security layer, much like a VPN, guarantees that your correspondence remains private during its journey.

Main types of VPNs

SSL VPN:

1. SSL VPNs enable remote users to securely connect to a company's network resources.
2. Users can access the VPN through a web browser, simplifying the connection process without requiring additional software.
3. SSL VPNs offer encryption, authentication, and access controls to protect data and ensure secure access to network resources from various devices and locations.

Site-to-site VPN:

1. Site-to-Site VPNs establish encrypted tunnels between geographically separated networks or sites using dedicated VPN routers or firewalls.
2. These VPNs connect entire networks together, enabling seamless communication between different locations as if they were part of the same local network.
3. Site-to-Site VPNs are commonly used by organizations with multiple branch offices or remote sites, offering a scalable and cost-effective solution for securely connecting distributed networks and facilitating collaboration and resource sharing.

Client-to-Server VPN:

- Client-to-Server VPNs enable individual users or devices to securely connect to a central network, such as a corporate network, from remote locations.
- Users install VPN client software on their devices, establishing encrypted connections to a VPN server within the central network, granting access to resources as if physically present.
- Client-to-Server VPNs offer strong encryption and authentication to protect data transmission, while allowing remote workers to access company resources securely from anywhere with an internet connection, improving productivity and collaboration

Benefits of VPN

- Keeps your online activities secure and private.
- Allows secure access to company networks from anywhere.
- Protects against cyber threats like hacking.
- Encryption of data transmitted over public Wi-Fi networks.
- Enables remote work securely.

How AWS Client VPN works

The Client VPN endpoint is utilized by **two categories of users: Administrators and Clients.**

The **administrator** is responsible for setting up and configuring the service. They make the Client VPN endpoint, link it with the target network, configuring the authorization rules, and setting up additional routes (if required).

- **Example:** Administrator is like the event organizer who sets up a secure venue for a party. They arrange the venue, decide who can enter, and provide access cards. Once everything's ready, they share the venue details and access cards with invited guests.

The **client** is the end user. They're the ones who actually log in and start the VPN session.

- **Example:** Client is like the party guest who arrives at the venue and uses their access card to enter. They can enjoy the party safely inside the venue. If the organizer has allowed access to other areas, like a VIP lounge or outdoor terrace, the guest can also explore those areas.

NFS (Network File System)

Network File System (NFS) finds extensive application in scenarios necessitating centralized file storage and sharing across multiple systems or users.

Imagine a big company with many different groups of people working together. They all need to share files and work on them together to do their jobs well. NFS helps with this by keeping all the files in one place, so everyone can access them easily, no matter where they are.

Example: let's say there's a team of software developers working on a project. They need to use code, documents, and other files stored in one central spot. With NFS, they can easily connect their computers to this central storage and work on the files as if they were saved right on their own computers. This makes it easy for everyone to work together on the same files, helping the team collaborate better and keep track of changes they make.

Network Troubleshooting Tools

✓**ping** - Checks the reachability of a host on an Internet Protocol (IP) network.

✓**traceroute** - Traces the route that packets take from the source to the destination network.
✓**telnet** - Establishes a connection to a remote host using the Telnet protocol.
✓**curl** - Transfers data to or from a server, supporting various network protocols like HTTP, HTTPS, FTP, etc.
✓**dig** - Performs DNS lookups and displays the results from the queried DNS server.
✓**netstat** - Displays network connections, routing tables, interface statistics, masquerade connections, and multicast memberships.
✓**nmap** - Scans networks for open ports, services, and vulnerabilities.
✓**ssh** - Securely connects to a remote device or server over a network, providing encrypted communication.
✓**scp** - Securely copies files between hosts on a network using Secure Copy Protocol (SCP).

ping

Ping is a versatile command-line tool used to check network connectivity between devices. It sends ICMP echo requests to a specified destination and waits for an ICMP echo reply.

Here's how to use it:

- Basic Usage: Open our command prompt or terminal and type ping [destination], where [destination] is the hostname or IP address we want to ping.
  For example, ping www.google.com.
- Interpreting Results: After executing the command, we'll see statistics about the ping, including the round-trip time (RTT) in milliseconds and any packet loss. This information helps diagnose network issues.
- Example: Suppose we want to check our connection to Google's servers. we enter ping www.google.com in our command prompt. The output shows the time it takes for packets to travel to Google's servers and back, along with any packet loss.

traceroute

The traceroute command is used to trace the path that packets take from our device to a destination host.

Example: traceroute google.com
It displays the IP addresses of the routers along the path and the time it takes for packets to reach each router, helping diagnose network routing issues.

How does it work?
Traceroute launches UDP probes with increasing TTL values until it receives a "time exceeded" message from ICMP. This iterative process uncovers each hop along the route, incrementing TTL until reaching the destination or TTL limit.
The report we receive details Time to Live, IP addresses of route stops, and round-trip times. Plus, asterisks mark unresponsive routers, aiding troubleshooting.

telnet

The telnet command is used to establish a TCP connection to a remote host.

Example: telnet google.com 443
It allows us to check if a specific port on a remote host is open and responsive, aiding in troubleshooting network connectivity.

- While ping and traceroute can indicate server responsiveness, it doesn't guarantee operational status. For instance, a server may respond to ping but lack essential services like Apache, rendering it inaccessible. Conversely, servers not responding to ping may still be reachable via other protocols due to firewall configurations.
- To test network connectivity and protocol allowance effectively, Telnet proves invaluable. Although outdated for remote access due to lack of encryption, Telnet remains useful for protocol testing.
- For example, testing a connection to google.com at port 443 with Telnet can confirm successful network connections.

curl

The curl command is a versatile tool for transferring data using various protocols.

Example: curl https://example.com
It can retrieve and display content from web servers, making it useful for testing APIs and fetching web pages.

- curl supports various protocols, primarily used for sending HTTP requests.
- To perform a basic HTTP GET request,
  curl http://example.com
- For checking response codes and viewing headers only,
  curl -I http://example.com
- To utilize different request methods, such as POST, utilize the -X flag:
  curl -X POST http://example.com
- Moreover, curl excels at file downloads or storing responses with the -o flag:
  curl http://example.com/file -o output.file

dig

The dig command is a DNS lookup utility used to query DNS servers for information about domain names. It retrieves various DNS records, including A, AAAA, CNAME, MX, and TXT records.

Example : dig google.com - will query the DNS server for the IP address of "google.com".

- For instance, querying the DNS record for google.com with **dig google.com** returns vital information like the queried server, TTL, query class, query type, and the associated IP address, such as 172.217.0.46.

- While dig defaults to querying servers specified in /etc/resolv.conf, we can specify a particular DNS server using the @ flag. Additionally, we can explore different record types like MX, NS, or ALL using the -t option.

netstat

The netstat command displays network connections, routing tables, interface statistics, masquerade connections, and multicast memberships. It provides information about network connections, routing information, and interface statistics.

Example: netstat -lp - shows the protocol, local address, and process ID (PID) of the listening program.

Netstat offers various flag commands like:

- -a: all active ports
- -n: only numerical IP addresses and ports
- -f: whenever possible, provide all names of foreign connections
- -o: show process ID
- -r: routing table

nmap

The nmap command is a network scanning tool used to discover hosts and services on a computer network. It sends packets to target hosts and analyses the responses to identify open ports, running services, and potential vulnerabilities.

Example: nmap -sS target_ip scans the target IP address using TCP SYN scan.

In troubleshooting, nmap used to identify network assets and vulnerabilities.

For example, by conducting a ping scan to discover hosts and then probing specific hosts to reveal active services like SSH and HTTP, facilitating further diagnostics or actions like SSH connections or HTTP requests using tools like curl.

ssh

SSH, or Secure Shell, is a protocol used to securely access and control remote computers over a network. It encrypts all data transferred between the local and remote machines, making it safe to use even on untrusted networks. This level of security is essential for protecting sensitive data and troubleshooting networks with unknown security levels.

To launch SSH, use the basic syntax:
[ssh] [user_name@hostname] or [ssh] [user_name@ipaddress].
For example, ssh user@example.com.

After executing the command, we'll be prompted to enter the password associated with the remote user account for authentication. While passwords are a common method, it's advisable to set up password less authentication for added security whenever feasible.

scp

SCP, or Secure Copy Protocol, is a secure method for transferring files between a local and remote host. Unlike SSH, which allows us to execute commands on a remote server, SCP focuses solely on file transfer.

If we want to copy a file named "example.txt" from our local machine to a remote server.
Example of SCP command like this:
scp example.txt user@example.com:/path/to/destination