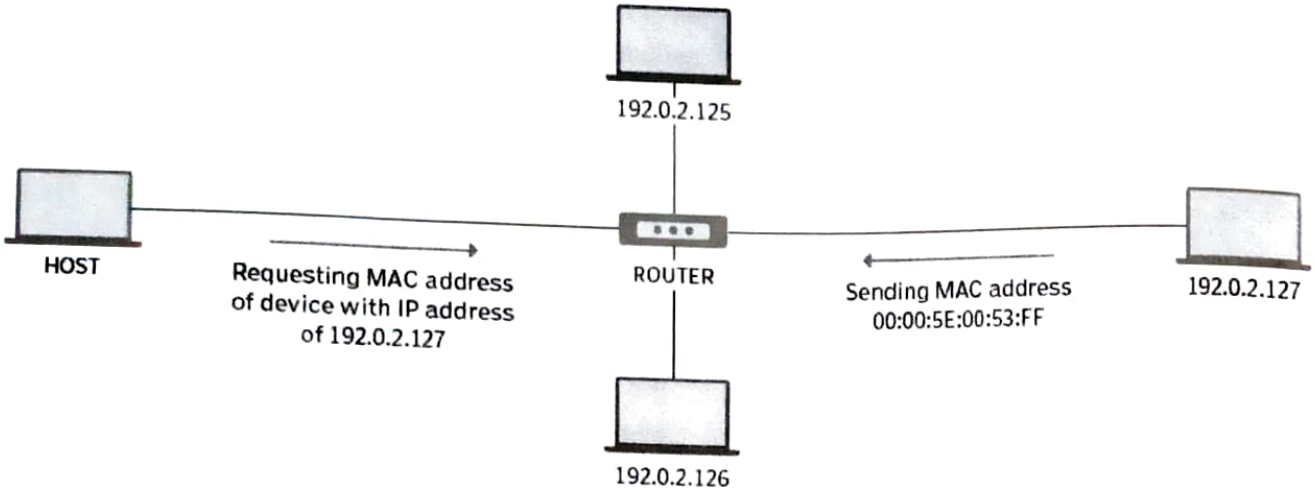
UNIT–II

PROTOCOL AND ADDRESSING SCHEME

2.1 INTRODUCTION TO ARP

ARP stands for Address Resolution Protocol. It is a vital part of how computers communicate with each other on a local network. When a computer wants to send data to another computer on the same network, it needs to know the recipient's physical address, called the MAC address. However, computers use IP addresses (like 192.168.1.100) to identify each other, not MAC addresses (like 12:34:56:78:90:ab). ARP helps the computer find the MAC address of another device based on its IP address, making communication possible.

How Address Resolution Protocol (ARP) works



WORKING OF ARP PROTOCOL

Example

Imagine you have two computers at home, Computer A and Computer B, connected to the same Wi-Fi network.

a. Computer A wants to send a message to Computer B, but it only knows Computer B's IP address (let's say 192.168.1.200). Computer A needs to find out Computer B's MAC address to deliver the message directly to it.

b. To do this, Computer A sends an ARP request to the network, essentially asking, "Hey everyone, I need to talk to the device with IP address 192.168.1.200. Can someone tell me their MAC address?"

c. All the devices on the local network, including Computer B, receive this ARP request.

d. Computer B recognizes its own IP address (192.168.1.200) and replies to Computer A with an ARP reply, saying, "I am the one with that IP address (192.168.1.200), and my MAC address is 12:34:56:78:90:ab."

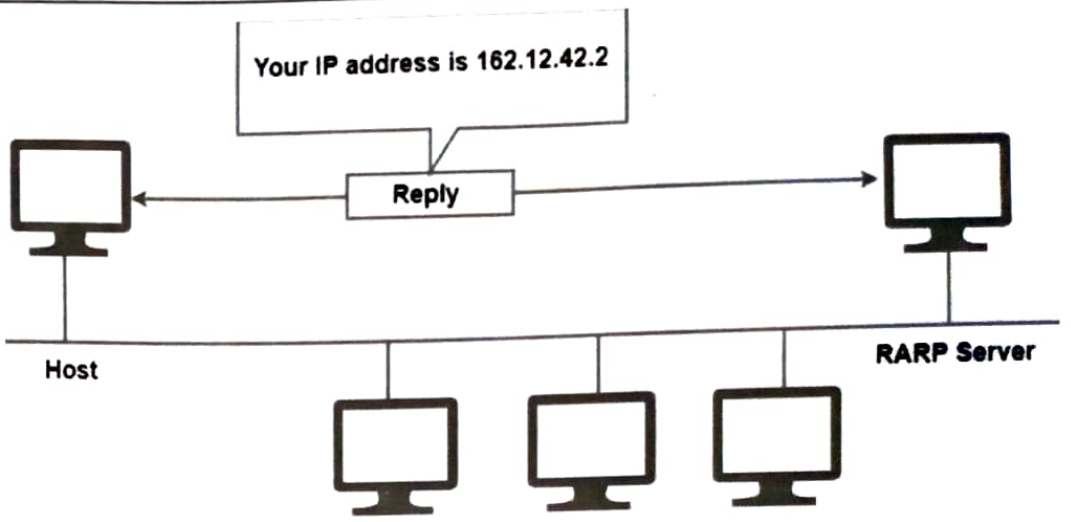
e. Computer A receives the ARP reply and now knows the MAC address of Computer B (12:34:56:78:90:ab).

f. Now, Computer A can package the message with Computer B's MAC address and directly send it over the network. The message reaches Computer B successfully.

ARP helps computers find each other's MAC addresses based on their IP addresses, enabling smooth communication on a local network. It's like a virtual postman that ensures the messages reach the correct recipients within your home network.

2.1.1 INTRODUCTION TO RARP

RARP, which stands for Reverse Address Resolution Protocol, is a network protocol used by computers to find their IP addresses when they only know their MAC addresses. It is the reverse process of ARP (Address Resolution Protocol). While ARP helps find the MAC address from an IP address, RARP helps find the IP address from a MAC address.



WORKING OF RARP PROTOCOL

Example

Imagine you have a computer that has just booted up and does not know its IP address. However, it knows its own MAC address, which is a unique hardware address assigned to its network adapter.

a. The computer broadcasts a RARP request on the network, essentially saying, "Hey, I have this MAC address (let's say 12:34:56:78:90:ab), and I need to know my IP address. Can anyone help?"

b. A RARP server on the network receives the request and checks its database to find a corresponding IP address for the given MAC address (12:34:56:78:90:ab).

c. If the RARP server finds a matching entry, it sends a RARP reply back to the requesting computer with the IP address (e.g., 192.168.1.100) associated with the MAC address (12:34:56:78:90:ab).

d. The computer receives the RARP reply and now knows its IP address (192.168.1.100).

e. With the IP address resolved, the computer can now communicate with other devices on the network using IP-based communication.

RARP is useful in certain situations, such as when a diskless workstation or a computer without a local storage device needs to obtain its IP address during the boot-up process. The RARP server helps these devices discover their IP addresses dynamically based on their MAC addresses, making it possible for them to participate in the network activities.

2.2 ROUTING

Routing is like finding the best path for your data to reach its destination in a network. Imagine you want to send a letter to your friend who lives far away. You don't know the exact route to their house, but you have a map with some directions.

In computer networks, routers act like the postmen of the internet. They take your data (like emails or webpages) and look at the address (IP address) to figure out where it needs to go. Routers use a map (routing table) that tells them which direction to send the data. They pass it from one router to another, like passing your letter from one post office to the next until it finally reaches your friend's house.

The goal of routing is to efficiently deliver your data to its destination, avoiding traffic jams and taking the shortest or fastest path possible. This way, information can travel smoothly across the network, and you can access websites, send messages, and do many things online without getting lost along the way.

2.2.1 TYPES OF ROUTING

There are two main types of routing in computer networks:

1. **Static Routing:**
   a. Static routing is like following a predefined, fixed path on a map without considering current traffic conditions.
   b. In this method, network administrators manually configure the routes that data packets should take between devices. These routes don't change unless manually modified by the administrator.
   c. It is straightforward and suitable for small, simple networks with predictable traffic patterns.
2. **Dynamic Routing:**
   a. Dynamic routing is like using a GPS to find the best and fastest route on a map based on real-time traffic conditions.
   b. With dynamic routing, routers communicate with each other, sharing information about network changes and traffic conditions. They use this real-time data to automatically determine the best path for data packets.
   c. It is more flexible and adaptive, making it suitable for larger networks with changing traffic patterns and devices that may come online or go offline frequently.

2.2.2 ROUTING TABLE

A routing table is a critical data structure stored in routers and switches that contains information about the network's available routes. Each entry in the routing table includes the following key components:

a. **Destination Network:** The IP address or IP address range of the destination network.
b. **Next Hop:** The IP address of the next router or gateway to which the data should be forwarded to reach the destination network.
c. **Interface:** The physical network interface (e.g., Ethernet, Wi-Fi) through which the data should be sent to reach the next hop.
d. **Metric/Cost:** The value that represents the distance or cost to reach the destination network. In dynamic routing, this value is calculated based on factors like link bandwidth, delay, or administrative preference.

When a data packet arrives at a router, the router consults its routing table to determine the best route for forwarding the packet. It matches the destination IP address of the packet with the entries in the routing table and selects the appropriate next hop and outgoing interface to send the packet along the best path toward its destination.

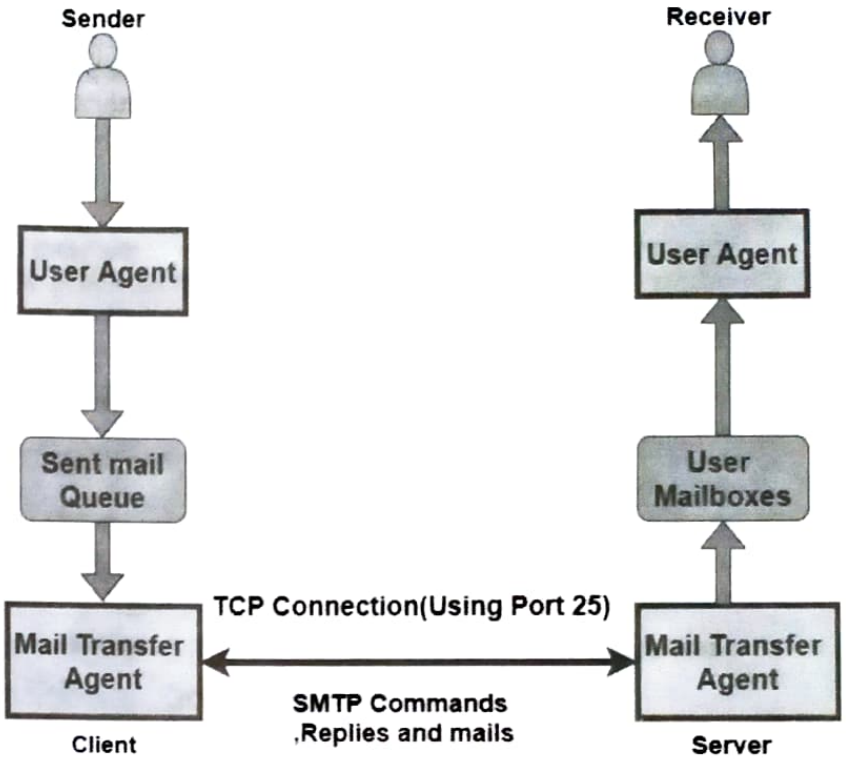
2.3 INTRODUCTION TO SMTP, POP, IMAP

2.3.1 INTRODUCTION TO SMTP

SMTP (Simple Mail Transfer Protocol) is like a postal service for emails.

Just like you send letters to friends through the postal service, SMTP helps your computer send emails to other computers across the internet.

When you write an email and click "send," your email client (like Gmail or Outlook) uses SMTP to package and deliver your email to the recipient's email server. It's like putting your letter in an envelope and handing it to the postal service.



WORKING OF SMTP PROTOCOL

Example

Let's say you want to send an email from your Gmail account (youremail@gmail.com) to your friend's email account (friend@example.com).

a. You compose the email in your Gmail account and click the "Send" button.
b. Gmail's email server processes your request and prepares the email to be sent.
c. Gmail's email server uses SMTP to connect to the recipient's email server (example.com) using its domain name (example.com) and the standard SMTP port (usually port 25).
d. The sending email server (Gmail) communicates with the receiving email server (example.com) and transfers the email message to it using SMTP.
e. The receiving email server (example.com) accepts the email and stores it in the recipient's mailbox.
f. Your friend can then log in to their email account, access the mailbox, and retrieve the email you sent.

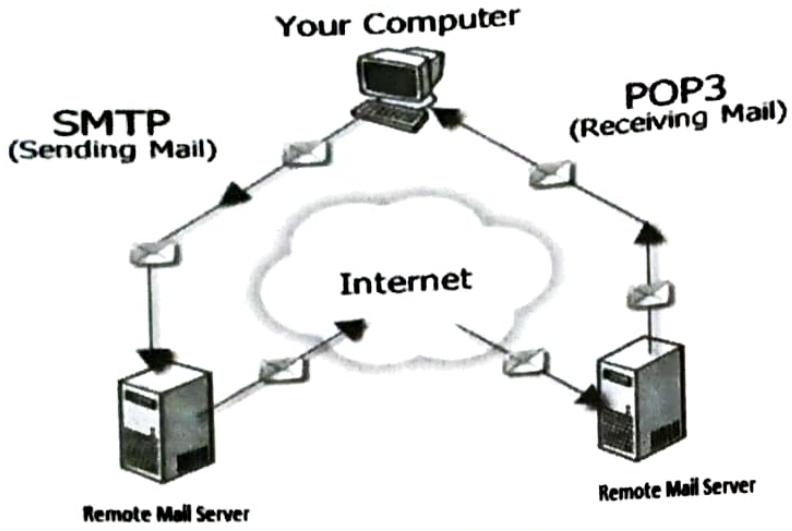
SMTP ensures that the email message is reliably delivered from one server to another, making email communication possible on a global scale. It works behind the scenes, handling the complexities of sending and receiving emails, so users can easily communicate with each other using their preferred email clients.

2.3.2 INTRODUCTION TO POP

Post Office Protocol (POP) is a way for your email program (like Outlook or Thunderbird) to get the emails that are stored on an email server (where your emails are stored online). It's like a post office for your emails.

Imagine your emails are like letters, and the email server is like a post office where these letters are kept until you pick them up. POP allows your email program to go to the post office (email server)

and collect all the new letters (emails) waiting for you. Once your email program downloads the emails, they are removed from the server, just like when you pick up your letters from the post office, they are no longer there for others to see.



WORKING OF POP PROTOCOL

Example

Let's say you have an email account with a popular email service provider. Here's how POP works to retrieve your emails:

a. You open your email client (like Microsoft Outlook or Apple Mail) on your computer or mobile device.

b. In your email client settings, you configure the POP server settings for your email account. These settings include the incoming mail server (POP server) address, your email address, and your email account password.

c. When you click on the "Retrieve Emails" button or when your email client automatically checks for new messages, it connects to the email server using POP.

d. The email server verifies your credentials (email address and password) to ensure you are an authorized user.

e. Once authenticated, the email server allows your email client to download your new messages from the server to your device.

f. The downloaded messages are then stored in your email client's inbox or other specified folders.

g. You can now read, manage, and respond to your emails directly from your email client, even if you are offline.

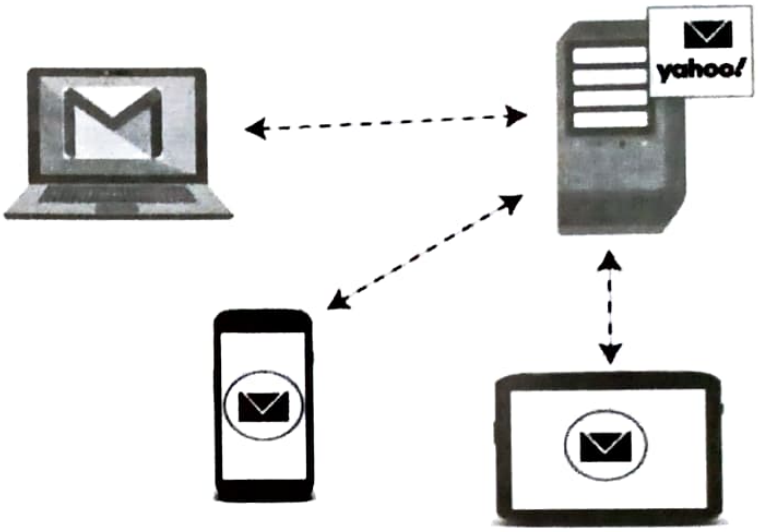
h. By default, when you retrieve your emails using POP, they are typically removed from the server. However, some email clients have an option to keep a copy of the messages on the server for a specific period.

POP is a widely used protocol for accessing emails and is suitable for users who want to keep a local copy of their messages and read them offline. However, it's important to note that since POP downloads messages to your local device, they are no longer stored on the email server, which means you won't be able to access them from other devices or webmail once they are downloaded.

2.3.3 INTRODUCTION TO IMAP

IMAP, which stands for Internet Message Access Protocol, is a protocol used to access and manage email messages stored on an email server. Unlike POP (Post Office Protocol), which downloads emails to the local device and removes them from the server, IMAP allows users to view and manage their emails directly on the server, from multiple devices. It keeps the emails synchronized between the server and the email client, making it easier to access messages from different devices.

Think of IMAP as a virtual window that shows you the contents of your mailbox on the email server. It lets you view, organize, and reply to your emails without actually downloading them permanently to your device.



WORKING OF IMAP PROTOCOL

Example

Let's use the same scenario as in the previous example, where you have an email account with a popular email service provider. Here's how IMAP works to access and manage your emails:

a. You open your email client (like Microsoft Outlook or Apple Mail) on your computer or mobile device.

b. In your email client settings, you configure the IMAP server settings for your email account. These settings include the incoming mail server (IMAP server) address, your email address, and your email account password.

c. When you click on the "Retrieve Emails" button or when your email client automatically checks for new messages, it connects to the email server using IMAP.

d. The email server verifies your credentials (email address and password) to ensure you are an authorized user.

e. Once authenticated, the email server allows your email client to access and display the list of emails stored on the server.

f. You can read, organize, delete, and reply to your emails directly from your email client. When you mark an email as read or move it to a specific folder, those changes are reflected on the server.

g. IMAP keeps your email client and the server in sync, so if you access your email from another device or webmail, you'll see the same changes you made from your email client.

h. IMAP allows you to access your full email history, including all folders and subfolders, as everything is stored on the server.

i. Since IMAP stores emails on the server, it also provides a way to access your emails from multiple devices seamlessly.

IMAP is a preferred choice for users who want to access and manage their emails across multiple devices while keeping them synchronized with the server. It provides a more flexible and convenient way to work with emails, especially for users who frequently switch between different devices.

2.4 INTRODUCTION TO WWW AND HTTP/S

2.4.1 INTRODUCTION TO WWW

The World Wide Web (WWW) is a vast and interconnected system of information found on the internet. It's like a gigantic library that holds all sorts of things like websites, videos, pictures, and much more.

The WWW operates on the internet's infrastructure and relies on specific protocols and technologies to enable the exchange of information. Some key components of the WWW in networking include:

a. **Web Pages:** Web pages are individual documents containing various types of content, such as text, images, videos, and interactive elements. These pages are written using a markup language called HTML (Hypertext Markup Language) and can be accessed through web browsers.

b. **Web Servers:** Web servers are computers or devices that store and serve web pages and other resources. They respond to requests from web clients (typically web browsers) by sending the requested content over the internet.

c. **Web Clients:** Web clients are software applications, such as web browsers, that users utilize to access web pages. These clients send requests to web servers, receive the response, and display the content to users.

d. **Web Browsers:** Web browsers are software applications that allow users to access and view web pages. Examples of popular web browsers include Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari. Browsers interpret HTML and display web content in a user-friendly format.

e. **Hyperlinks:** Hyperlinks are clickable text or images on a web page that allow users to navigate between different web pages. They connect related resources on the web, enabling users to access additional information with a simple click.

f. **Hypertext Transfer Protocol (HTTP):** HTTP is the fundamental protocol used for communication between web servers and web clients. It defines how requests and responses should be formatted and handled when accessing web resources.

g. **Uniform Resource Locators (URLs):** URLs are addresses that uniquely identify web resources on the internet. They specify the location of web pages, allowing users to access specific content.

h. **Hypertext Markup Language (HTML):** HTML is the standard language used to create the structure and content of web pages. Web servers serve HTML files to web clients, and web browsers interpret and display these files to users as web pages.

i. **Web Development:** Creating and maintaining web pages involves various technologies, including HTML for content structure, CSS (Cascading Style Sheets) for design and layout, and JavaScript for interactivity. Web developers use these technologies to build user-friendly and visually appealing websites.

When a user types a URL into a web browser's address bar or clicks on a hyperlink, the web browser sends an HTTP request to the corresponding web server. The web server processes the request and sends back an HTTP response, which includes the requested web page's HTML content and any associated resources (such as images or stylesheets). The web browser then interprets the HTML, displays the web page to the user, and may initiate additional requests for linked resources to fully render the page.

2.4.2 INTRODUCTION TO HTTP/S

HTTP (Hypertext Transfer Protocol) and HTTPS (Hypertext Transfer Protocol Secure) are fundamental protocols used in computer networking and the World Wide Web to facilitate the exchange of information between web servers and web clients (typically web browsers). They define the rules and conventions for how data is formatted and transmitted over the internet, enabling the retrieval and display of web pages and resources.

1. HTTP (Hypertext Transfer Protocol)

Hypertext Transfer Protocol (HTTP) is a fundamental protocol used for communication on the World Wide Web. It governs how data is exchanged between web browsers (clients) and web servers. HTTP enables the retrieval and display of web pages, images, videos, and other resources from servers to browsers, allowing users to access and interact with online content.

The HTTP request includes specific information, such as the method (e.g., GET, POST, PUT, DELETE), the URL, and additional headers. The web server processes the request and sends back an HTTP response, which contains the requested web page's content and relevant status codes (e.g., 200 OK for success, 404 Not Found for page not found).

HTTP is a stateless protocol, meaning each request-response cycle is independent of previous ones. Therefore, it does not maintain any memory of past interactions between the client and server.

Key features and aspects of HTTP include:

a. **Client-Server Model:** HTTP follows a client-server architecture. The client, typically a web browser, sends requests to the server for specific resources. The server processes these requests and sends back the requested resources along with appropriate status codes.

b. **Stateless Protocol:** HTTP is stateless, meaning each request from the client to the server is independent and doesn't retain any information about previous requests. This requires additional mechanisms (such as cookies and sessions) to maintain user state across multiple requests.

c. **Request and Response:** An HTTP request is sent by the client to the server, indicating the type of resource being requested and other relevant details. The server processes the request and sends back an HTTP response containing the requested resource along with a status code that indicates the outcome of the request.

d. **URLs and URIs:** HTTP uses Uniform Resource Locators (URLs) or Uniform Resource Identifiers (URIs) to identify the location of a resource on the web. URLs typically include the protocol (HTTP or HTTPS), domain name, path, and any query parameters.

e. **HTTP Methods:** HTTP defines various methods (also known as verbs) that specify the action to be performed on the server. Common methods include:

- GET: Retrieve a resource from the server.
- POST: Send data to the server to create or update a resource.

- PUT: Update a resource on the server.
- DELETE: Remove a resource from the server.
- HEAD: Retrieve metadata about a resource without the actual content.

f. **Status Codes:** HTTP responses include status codes that indicate the outcome of the request. Examples include:

- 200 OK: Request successful.
- 404 Not Found: Resource not found on the server.
- 500 Internal Server Error: Server error.

g. **Headers:** Both requests and responses include headers, which provide additional information about the request/response or modify its behavior. Headers can include details like content type, length, caching instructions, and more.

h. **Security Considerations:** HTTP transmits data in plain text, making it susceptible to interception and tampering. To enhance security, HTTPS (HTTP Secure) encrypts data using SSL/TLS protocols, ensuring that sensitive information remains confidential during transmission.

i. **Evolution:** Over time, HTTP versions have evolved. HTTP/1.1 was a widely used version for many years. More recently, HTTP/2 and HTTP/3 have introduced improvements in performance, multiplexing, and reduced latency.

2. HTTPS (Hypertext Transfer Protocol Secure)

HTTPS is an extension of HTTP that adds a layer of security to the communication between web browsers and web servers. It uses encryption and authentication mechanisms to protect the privacy and integrity of data transmitted over the internet.

With HTTPS, the data exchanged between the client and server is encrypted, ensuring that any potential eavesdroppers cannot easily intercept or understand the information being transmitted. This is particularly important when sensitive data, such as login credentials, credit card information, or personal details, are exchanged on websites.

HTTPS utilizes SSL (Secure Socket Layer) or TLS (Transport Layer Security) protocols to establish a secure connection between the client and server. When you visit a website using HTTPS, your web browser verifies the authenticity of the website's SSL/TLS certificate to ensure you are connecting to the legitimate server.

HTTP and HTTPS are essential protocols in networking that enable the smooth functioning of the World Wide Web. HTTP governs the standard data exchange between web clients and servers, while HTTPS adds an extra layer of security to protect sensitive information during communication. As internet security concerns continue to rise, the adoption of HTTPS has become increasingly prevalent, making secure web browsing a crucial aspect of modern networking practices.

Key features and aspects of HTTPS include:

a. **Encryption:** The primary purpose of HTTPS is to provide encryption for the data transmitted between a user's browser and the web server. This encryption is achieved using Secure Sockets Layer (SSL) or Transport Layer Security (TLS) protocols. These protocols create a secure, encrypted channel through which data can travel, preventing unauthorized access and eavesdropping.

b. **Data Integrity:** In addition to encryption, HTTPS also ensures data integrity. This means that the data exchanged between the user and the server remains unchanged during transit. Any tampering or alteration of data can be detected, as it would break the encryption and cause validation errors.

c. **Authentication:** HTTPS provides authentication, ensuring that users are connecting to the intended website. Websites using HTTPS obtain an SSL/TLS certificate from a trusted Certificate Authority (CA). This certificate contains information about the website's identity and is used to verify that the website is legitimate.

d. **Trust and Security Indicators:** Modern web browsers display visual indicators to let users know if a website is using HTTPS. These indicators include a padlock icon in the address bar, indicating a secure connection. Some browsers may also display the company name or website owner alongside the padlock.

e. **Mixed Content Blocking:** Browsers may block mixed content, which refers to insecure resources (such as images or scripts) being loaded on a secure HTTPS page. This helps maintain the security of the connection and ensures that all resources are encrypted.

f. **SEO Benefits:** Search engines often prioritize websites using HTTPS in search results, as it demonstrates a commitment to security and user privacy.

g. **Compatibility:** Modern web browsers strongly encourage the use of HTTPS and may display warnings for websites that don't use it, especially when transmitting sensitive data like passwords or credit card information.

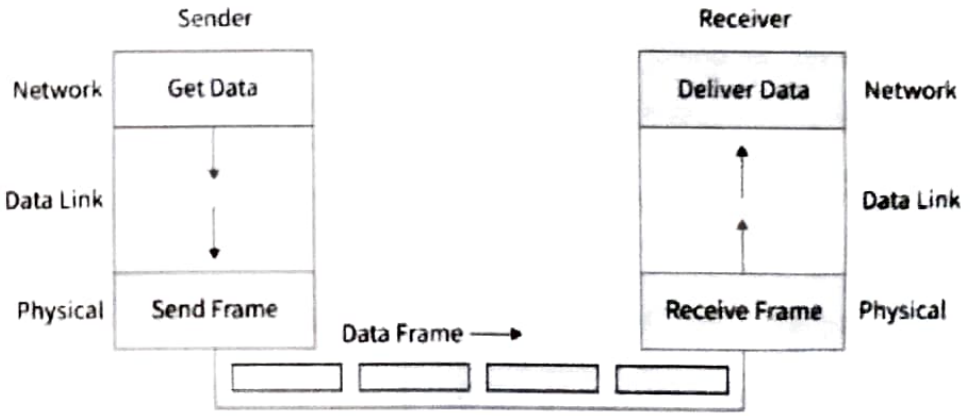
h. **Evolution:** As technology evolves, HTTPS standards are updated to maintain security. Older SSL versions, which have vulnerabilities, are being phased out in favor of more secure TLS versions.

2.5 DATA LINK LAYER PROTOCOLS

Data link layer protocols are sets of rules and procedures that govern how data is transmitted and received between devices in a computer network. They ensure reliable communication over a physical connection, such as Ethernet or Wi-Fi. Let's explore three specific data link layer protocols:

1. Simplest Data Link Layer Protocol

The Simplest Protocol is the most basic data link layer protocol. In this protocol, the sender simply sends a data frame to the receiver, assuming it will be successfully delivered without any error detection or acknowledgment. There is no flow control or error recovery mechanism. If the frame is lost or corrupted during transmission, there is no way to detect or recover from it.



SIMPLEST PROTOCOL

Example

Consider two computers, A (sender) and B (receiver), connected over a simple point-to-point link. A wants to send a data frame to B using the Simplest Protocol.

Steps:

a. A sends the data frame to B.

b. B receives the frame.

If the frame is successfully received, the communication is considered successful. However, if the frame is lost or corrupted during transmission, there is no mechanism to handle such errors, and the data will be lost without any indication.

2. Stop and Wait Protocol

The Stop-and-Wait Protocol is an improved version of the Simplest Protocol. In this protocol, the sender sends one data frame and then waits for an acknowledgment (ACK) from the receiver before sending the next frame. If the sender doesn't receive an ACK within a certain timeout period, it assumes the frame was lost or corrupted and retransmits the frame.

Example

Let's consider the same setup with computers A (sender) and B (receiver) connected over a network link. A wants to send a data frame to B using the Stop-and-Wait Protocol.

Steps:

a. A sends the data frame to B.

b. B receives the frame and sends an ACK back to A to acknowledge successful reception.

c. A receives the ACK and proceeds to send the next data frame if there is more data to send.

If B does not receive the frame correctly or the ACK is lost, A will not receive the ACK within the timeout period and will retransmit the same frame.

3. Stop and Wait ARQ (Automatic Repeat Request) Protocol

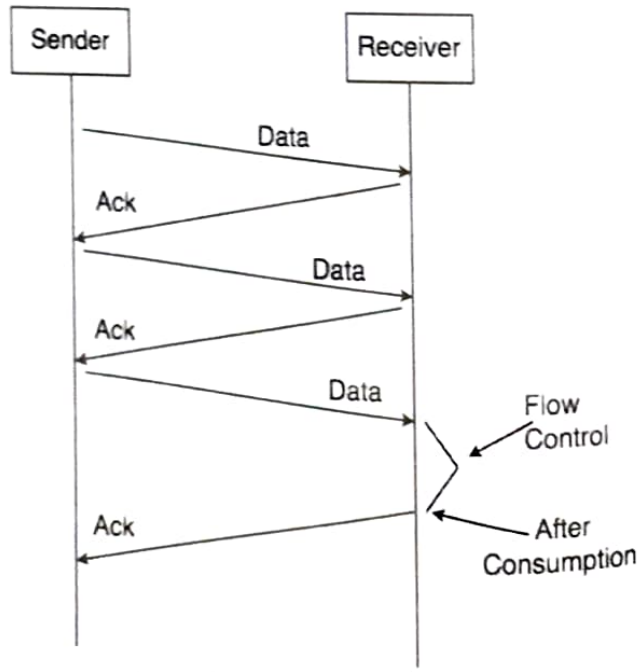
The Stop-and-Wait ARQ Protocol is an enhanced version of the Stop-and-Wait Protocol with error recovery mechanisms. If the receiver detects errors in the received frame, it sends a negative acknowledgment (NAK) to the sender, requesting retransmission of the frame. Additionally, the sender implements a sequence number for the frames it sends, allowing the receiver to detect duplicate frames.

Example

Consider the same setup with computers A (sender) and B (receiver) using the Stop-and-Wait ARQ Protocol.

Steps:

a. A sends the data frame to B with a unique sequence number.



STOP AND WAIT ARQ PROTOCOL

b. B receives the frame and checks for errors. If the frame is error-free, B sends an ACK back to A with the same sequence number, indicating successful reception.

c. If B detects errors in the frame, it sends a NAK to A, requesting retransmission of the frame with the specific sequence number.

d. Upon receiving the NAK, A retransmits the frame with the same sequence number.

e. B, upon receiving the retransmitted frame, either sends an ACK or a NAK as appropriate.

This process continues until all frames are successfully received and acknowledged by the receiver.

The Stop-and-Wait ARQ Protocol provides a more reliable and efficient data transfer by ensuring that lost or corrupted frames are retransmitted and acknowledged correctly. It reduces the chance of data loss and ensures data integrity during transmission. However, it can introduce additional delays due to the time taken for acknowledgment and retransmission. More sophisticated ARQ protocols, like Go-Back-N and Selective Repeat, are used in practice to further improve the efficiency of data link layer communication.

2.6 IPV4 ADDRESSING SCHEME

IPv4 (Internet Protocol version 4) addressing scheme is a way to give unique identities to devices connected to the internet, like your computer, smartphone, or any other device. It's like having a phone number for each device so they can talk to each other and exchange information.

2.6.1 CLASSFUL AND CLASSLESS NOTATIONS

In IPv4 addressing, there are two ways to represent IP addresses: Classful notation and Classless notation.

1. **Classful Notation:**

   a. **Class A:** These addresses start with a number from 1 to 126 in the first octet (the first group of numbers in the IP address). Class A addresses are used for large networks and have 8 bits reserved for the network portion and 24 bits for the host portion.

   b. **Class B:** These addresses start with a number from 128 to 191 in the first octet. Class B addresses are used for medium-sized networks and have 16 bits reserved for the network portion and 16 bits for the host portion.

   c. **Class C:** These addresses start with a number from 192 to 223 in the first octet. Class C addresses are used for small networks and have 24 bits reserved for the network portion and 8 bits for the host portion.

   d. **Class D:** These addresses start with a number from 224 to 239 in the first octet. Class D addresses are used for multicast groups and not for regular network devices.

   e. **Class E:** These addresses start with a number from 240 to 255 in the first octet. Class E addresses are reserved for experimental purposes and not used for general networking.

2. **Classless Notation:**

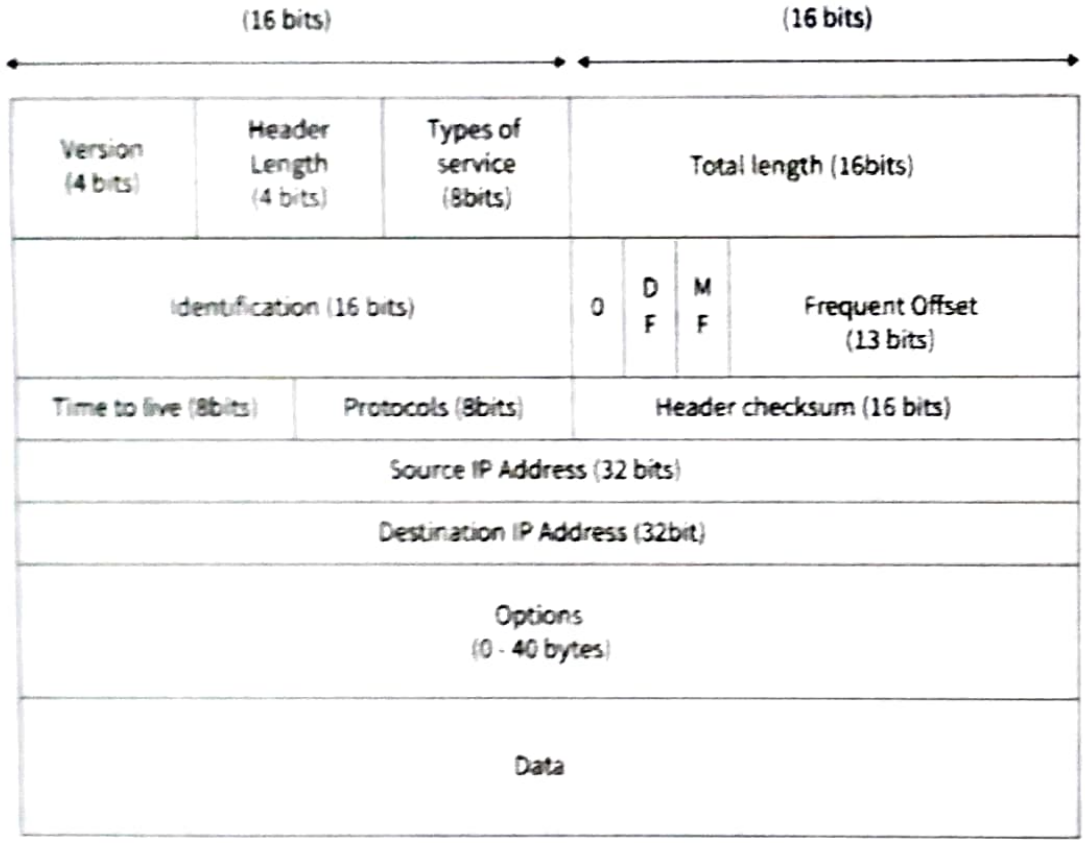
Classless notation, also known as CIDR (Classless Inter-Domain Routing), is a more flexible way of representing IP addresses. Instead of fixed classes like in classful notation, classless notation allows for variable-length subnet masks. This means you can divide an IP address into smaller or larger blocks as needed.

In classless notation, an IP address is followed by a forward slash and a number (e.g., 192.168.1.0/24). The number after the slash indicates the number of bits used for the network portion. So, /24 means the first 24 bits represent the network, and the remaining bits (8 in this case) represent the hosts.

Classless notation provides more efficient address allocation and allows for better use of available IP addresses, especially as the internet grew and the number of devices increased.

In simple words, classful notation divides IP addresses into predefined classes (A, B, C, etc.), while classless notation allows for more flexible division of IP addresses by using a variable-length subnet mask. Classless notation became more popular as it allowed better management of IP addresses and avoided wasting them, especially with the growth of the internet and the increasing number of connected devices.

2.6.2 IPV4 DATAGRAM HEADER



IPV4 DATAGRAM HEADER

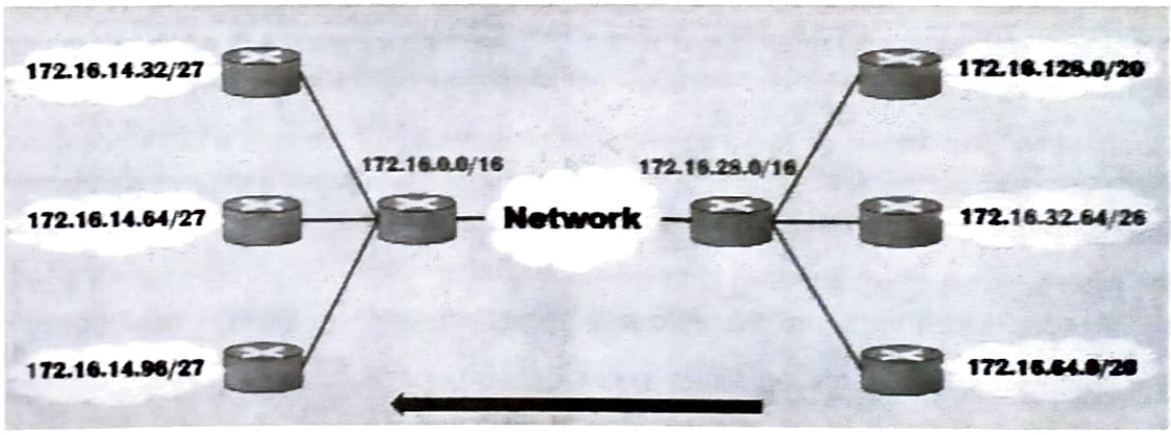
The IPv4 datagram header is a crucial part of the IPv4 (Internet Protocol version 4) packet structure. It contains essential information that helps in the delivery of data across a network.

1. **Version (4 bits)**: Specifies the version of IP being used, and for IPv4, it is always set to "0100."
2. **Header Length (4 bits)**: Indicates the length of the header in 32-bit words. Since the header can vary in size, this field allows the receiver to know where the actual data (payload) begins.
3. **Type of Service (TOS) (8 bits)**: Originally meant to define the type of service for the packet (e.g., high priority or low delay), it's now mostly unused and often replaced by the Differentiated Services Code Point (DSCP).

4. **Total Length (16 bits):** Specifies the total size of the IPv4 packet, including the header and the data (payload). It ensures that the receiving device knows the entire size of the packet.
5. **Identification (16 bits):** Helps in identifying fragments of a larger packet. When a packet is too big for a network link, it can be broken into smaller fragments, and this field assists in reassembling them at the destination.
6. **Flags (3 bits):** Contains control flags related to packet fragmentation. For example, it indicates whether the packet can be fragmented or if it is the last fragment of a packet.
7. **Fragment Offset (13 bits):** Specifies the position of the current fragment in the original packet, ensuring the correct order of reassembly.
8. **Time to Live (TTL) (8 bits):** Represents the maximum number of router hops (network devices) the packet can pass through before being discarded. It prevents packets from endlessly circulating in the network.
9. **Protocol (8 bits):** Identifies the higher-layer protocol to which the data should be passed (e.g., TCP or UDP) after reaching the destination.
10. **Header Checksum (16 bits):** Helps in error detection for the header. The checksum is calculated at the source and rechecked at each router to ensure the header's integrity.
11. **Source IP Address (32 bits):** Represents the IP address of the sender (the source device).
12. **Destination IP Address (32 bits):** Represents the IP address of the intended receiver (the destination device).
13. **Options (variable length):** An optional field that is used to include additional information or specify certain actions related to the packet. It is rarely used in practice.

2.6.3 SUBNETTING AND SUPERNETTING

Subnetting and Supernetting are techniques used in IPv4 addressing to divide or combine IP address blocks.



SUBNETTING AND SUPERNETTING

1. Subnetting

Subnetting is a technique used in computer networks to divide a larger network into smaller, more manageable sub-networks called subnets. It helps in efficient allocation of IP addresses and in controlling network traffic.

In a computer network, each device (such as computers, printers, or smartphones) is assigned a unique IP address to communicate with other devices on the network. IP addresses consist of two parts: the network portion, which identifies the network the device belongs to, and the host portion, which identifies the specific device within that network.

When a network is subnetted, the original network is split into multiple smaller subnets, and each subnet gets its own range of IP addresses. Subnetting is typically done to achieve several goals:

a. **Efficient Use of IP Addresses:** Subnetting allows network administrators to allocate IP addresses more efficiently. Instead of using a single large network for all devices, smaller subnets can be created for different departments, floors, or buildings, reducing IP address wastage.

b. **Improved Network Performance:** Smaller subnets can help manage network traffic more effectively. Devices within the same subnet can communicate directly without causing unnecessary broadcast traffic across the entire network, which can lead to improved performance.

c. **Enhanced Security:** Subnetting can be used to create security boundaries within a network. By dividing the network into subnets, access controls and security policies can be applied more granularly, restricting access between certain subnets for added security.

The process of subnetting involves borrowing bits from the host portion of the IP address to create the subnet portion. The subnet mask is used to determine the boundary between the network portion and the host portion.

For example, a subnet mask of 255.255.255.0 (or /24 in CIDR notation) means that the first 24 bits are used for the network portion, leaving 8 bits for the host portion. This allows for a maximum of 2^8 = 256 unique IP addresses in the subnet.

By customizing the subnet mask, different numbers of IP addresses can be assigned to each subnet, depending on the requirements of the network.

For example, if you have an IP address block with a range from 192.168.1.0 to 192.168.1.255, you can use subnetting to create smaller subnetworks like 192.168.1.0/24, 192.168.1.128/25, and 192.168.1.192/26. Each of these subnets can be assigned to different departments or locations within an organization, making network management more efficient.

2. Supernetting

Supernetting, also known as route aggregation or prefix aggregation, is a technique used in computer networking to combine multiple smaller IP address ranges or subnets into a single, larger range. It is the opposite process of subnetting.

The main purpose of supernetting is to simplify the routing process and reduce the size of the routing tables in routers, which improves the efficiency of data forwarding within a network.

In a computer network, routing tables contain information about how to reach different IP address ranges or subnets. When a router receives a data packet, it consults its routing table to determine the next hop for forwarding the packet to its destination. If there are numerous small subnets listed in the routing table, it can become large and unwieldy, leading to increased processing overhead and memory usage in the router.

Supernetting helps address this issue by aggregating multiple smaller subnets with contiguous address ranges into a single larger network. The process involves borrowing bits from the host portion of the IP addresses in the subnets to create a supernet, also known as a supernetwork.

For supernetting to work, the subnets being aggregated must have certain characteristics:

a. They must be adjacent to each other in terms of IP address range.

b. They must have contiguous address ranges, meaning there are no gaps between the address ranges.

c. They must have the same prefix length or subnet mask.

Supernetting is represented using a shorter subnet mask than the original subnets. For example, if you have two subnets, 192.168.1.0/24 and 192.168.2.0/24, with contiguous address ranges, they can be supernetted into a single network with a shorter mask, such as 192.168.1.0/23.

The /23 subnet mask means that the first 23 bits are used for the network portion, and the last 9 bits (32 - 23) are used for the host portion. This results in a larger address range that includes both the original subnets.

By using supernetting, the routing tables in routers become smaller and more efficient, which helps in faster routing decisions and overall improved network performance.

2.6.5 NETWORK ADDRESS TRANSLATION

Network Address Translation (NAT) is a technique used in computer networks to help multiple devices within a private network share a single public IP address when connecting to the internet. It's like having a receptionist at the entrance of a building who takes phone calls from the outside and forwards them to the right person inside.

1. **Private Network:** Imagine you have a small office with multiple employees (computers, smartphones, etc.), and you want them all to access the internet using a single main office phone number (public IP address).

2. **NAT Device:** You install a special device called a NAT router, which acts as the receptionist. It's connected to both your office's private network and the internet. The router has two interfaces, one facing the private network and the other facing the internet.

3. **Outbound Communication:** When any employee wants to browse the internet or send data outside the office, the data goes to the NAT router first. The router keeps track of which device made the request and temporarily replaces the private IP address of that device with the single public IP address of the router.

4. **Internet Communication:** Now, when the data goes out to the internet, it carries the router's public IP address as the sender's address. The destination servers on the internet respond to the router's public IP address.

5. **Inbound Communication:** When the responses come back to the router, it looks at the data's destination and remembers which internal device requested it. It then replaces the router's public IP address with the original private IP address of the corresponding device and sends the data to the correct employee (device) within the office.

2.6.6 ADVANTAGES AND DISADVANTAGES

Advantages of IPv4 Addressing Scheme:

1. **Familiarity:** IPv4 has been widely used for decades, so network administrators and users are already familiar with its setup and configuration.

2. **Simple Implementation:** IPv4 addresses are represented in four groups of numbers (e.g., 192.168.1.1), making them relatively straightforward to understand and work with.
3. **Device Compatibility:** Many devices and network equipment are designed specifically to work with IPv4, ensuring broad compatibility across different devices.
4. **Wide Adoption**: IPv4 has been used for a long time and is still prevalent, making it widely supported by internet service providers and devices around the world.

Disadvantages of IPv4 Addressing Scheme:

1. **Address Exhaustion:** The primary disadvantage of IPv4 is its limited address space, resulting in IPv4 address exhaustion. With only about 4.3 billion addresses available, there aren't enough unique addresses to accommodate the increasing number of devices connecting to the internet.
2. **NAT Overload:** To cope with the address shortage, Network Address Translation (NAT) is widely used, but it can introduce complexities, such as difficulty in hosting services and potential performance bottlenecks.
3. **Complex Subnetting:** Subnetting IPv4 addresses can be challenging, especially when dealing with different address classes and ranges, requiring careful planning and management.
4. **Security Concerns:** The widespread use of NAT and the shortage of public IP addresses can make it harder to trace the source of network attacks or malicious activities.

2.7 IPV6 ADDRESSING

Imagine the internet is like a neighborhood, and every device connected to it, like your computer or smartphone, is a house. Each house needs a unique address to receive mail and packages (data) from the outside world.

Now, in the past, we had a limited number of addresses (IPv4 addresses) available for all the houses in the neighborhood. As more and more devices joined the internet, we ran out of unique addresses, like running out of house numbers.

IPv6 addressing scheme comes to the rescue! It's like expanding the neighborhood and giving every house a much longer address. With IPv6, we have an enormous amount of unique addresses, enough for every house in the world and beyond! This means all our devices can get their unique addresses, and the internet can keep growing without any worries about running out of addresses.

In simple words, the need for IPv6 addressing scheme is like making sure every house in the neighborhood has its own unique address, so we have enough room for all the devices to join the internet and keep it growing for the future.

2.7.1 NEED OF IPV6 MIGRATION

The need for IPv6 migration is driven by the limitations of the current IPv4 addressing scheme and the growing demands of an increasingly connected world. Let's understand why we need to transition to IPv6 in simple words:

1. **Address Space Exhaustion:** The main reason for IPv6 migration is the depletion of available IPv4 addresses. With the explosive growth of the internet and the ever-increasing number of connected devices, we are running out of unique IPv4 addresses. IPv6 provides a vast address space that ensures we have enough unique addresses for all the devices now and in the future.

2. **Global Connectivity:** As the internet connects more people, places, and things, IPv6 enables seamless global communication. The large address space allows for direct peer-to-peer connections without the need for complex workarounds like Network Address Translation (NAT), enhancing efficiency and improving end-to-end connectivity.
3. **Internet of Things (IoT):** The rise of the Internet of Things (IoT) means we have an ever-growing number of smart devices that need unique IP addresses. IPv6's huge address space is essential for assigning unique addresses to each IoT device, facilitating their direct communication and management.
4. **Streamlined Routing:** IPv6 has a simplified and hierarchical addressing structure, which streamlines the routing process for routers and enhances overall network performance. This leads to more efficient data transmission and reduces the burden on internet infrastructure.
5. **Security and Mobility:** IPv6 includes built-in security features like IPsec, which helps protect data during transmission. Additionally, IPv6 has improved support for mobile devices, allowing seamless connectivity as devices move between networks.
6. **Future-Proofing:** By migrating to IPv6, we are ensuring the longevity and scalability of the internet. IPv6 is designed to meet the growing demands of emerging technologies, applications, and devices, safeguarding the internet's ability to accommodate future innovations.
7. **Coexistence with IPv4:** During the migration to IPv6, both IPv4 and IPv6 will coexist. This allows for a gradual transition and ensures that devices using IPv4 can still communicate with devices using IPv6 through transition mechanisms.

2.7.2 IPV6 ADVANTAGES

IPv6 offers several advantages over the older IPv4 addressing scheme in simple words:

1. **Vast Address Space:** IPv6 provides an incredibly large number of unique addresses, so every device in the world can have its own address. It's like having enough phone numbers for every person on the planet and even more!
2. **Global Connectivity:** IPv6 enables direct communication between devices across the internet, making it easier for them to talk to each other without needing complicated workarounds like NAT. It's like having a direct line to reach anyone worldwide without going through an operator.
3. **Efficient Routing:** IPv6 uses a smarter and simpler addressing structure, which helps routers send data more efficiently. It's like having better road signs and clear directions, allowing data packets to reach their destination faster.
4. **Simplified Configuration:** IPv6 can automatically assign addresses to devices, making it easier to set up new devices on a network. It's like your computer automatically knowing its phone number when you connect it to the internet.
5. **Better Security:** IPv6 includes built-in security features, making it more resistant to cyberattacks and ensuring that data is protected during transmission. It's like having a secure lock on your door to keep your information safe.
6. **Support for IoT:** With the rise of smart devices like smart homes, cars, and wearables, IPv6 provides enough addresses for each device to connect to the internet directly. It's like giving every smart device its unique nameplate to be easily found.

7. **Future-Proofing:** IPv6 is designed to meet the ever-growing demands of the internet and emerging technologies, making it a future-proof solution. It's like building a house that can accommodate all your family members and future generations.

QUESTION BANK

■ Multiple Choice Questions (MCQs):

1. _____ helps the computer find the MAC address of another device based on its IP address, making communication possible.

   (a) ARP (b) RARP (c) POP (d) SMTP

2. _____ is like finding the best path for your data to reach its destination in a network.

   (a) Classful addressing (b) Surfing

   (c) Routing (d) Searching

3. Dynamic routing is like using a GPS to find the best and fastest route on a map based on real-time traffic conditions.

   (a) True (b) False

4. _____ protocol provides mail service.

   (a) HTTP (b) HTTPS (c) SMTP (d) POP

5. What is the size of ip address?

   (a) 8 bits (b) 16 bits. (c) 32 bits (d) 64 bits

6. What is the size of MAC address?

   (a) 8 bits (b) 16 bits (c) 32 bits (d) 48 bits

7. _____ also known as route aggregation or prefix aggregation, is a technique used in computer networking to combine multiple smaller IP address ranges or subnets into a single, larger range.

   (a) Suppernetting (b) Subnetting (c) Masking (d) Rotuing

ANSWERS

1. (a) ARP
2. (c) Routing
3. (a) True
4. (d) POP
5. (c) 32 bits
6. (d) 48 bits
7. (a) Suppernetting

■ Short Questions:

1. What is the role of ARP and RARP?
2. Enlist types of routing.
3. Explain the functionalities of SMTP,POP,IMAP.
4. What is the need of routing table.
5. Enlist data link layer protocols.
6. What is the need of ipv6.
7. What is supernetting and subnetting.

Long Questions:

1. Explain ARP and RARP with its working.
2. Explain types of routing.
3. Briefly explain SMTP protocol.
4. Briefly explain POP protocol.
5. Briefly explain IMAP protocol.
6. Explain datalink layer protocols.
7. What is ipv4 addressing scheme? Explain with a neat and clean diagram with its working.
8. What is ipv4 addressing scheme? How it proves to be beneficial over ipv4?

■ ÷ ■