UNIT 5
NETWORK AND TRANSPORT LAYERS

•CONTENTS

- • Introduction; Transport and Network Layer Protocols (Transmission Control Protocol, Internet Protocol); Transport Layer Functions (Linking to the Application Layer, Segmenting, Session Management); Addressing (Assigning Addresses, Address Resolution); Routing (Types of Routing, Routing Protocols, Multicasting, The Anatomy of a Router); TCP/IP Example (Known Addresses, Unknown Addresses, TCP Connections, TCP/IP and Network Layers).

- **Introduction**
- **Transport Layer Functionality in Networking**
- **Linking Layers:**
  - The **Transport Layer** connects application software in the **Application Layer** (e.g., Web, Email) with the network.
- **End-to-End Delivery:**
- It ensures the **end-to-end delivery** of messages across the network.
- **Message Segmentation:**
- The Transport Layer **segments** outgoing messages from the Application Layer for transmission.
- **Role of Other Layers:**
- **Network Layer:** Routes messages by selecting the best path and adds an **IP packet**.
- **Data Link Layer:** Adds an **Ethernet frame** and coordinates with the **Physical Layer** to transmit the data.

- **The network and transport layers also accept incoming messages from the data link layer and organize them** into coherent messages that are passed to the application layer.
- **The transport layer at the sender would break the message into several smaller segments and give them to the network layer to route, which in turn gives them to the data link layer to transmit.** The network layer at the receiver would receive the individual packets from the data link layer, process them, and pass them to the transport layer, which would reassemble them into

- Network Layer is responsible for delivery of datagrams between two hosts. This is called **host-to-host delivery.**
- Transport Layer is responsible for delivery of entire message from one process running on source to another process running on destination. This is called **process-to process delivery.**

Introduction to Network Layer

- The network layer is the third layer of the Open Systems Interconnection Model (OSI Model) and the layer that provides data routing paths for network communication.
- Data is transferred in the form of packets via logical network paths in an ordered format controlled by the network layer.
- Logical connection setup, data forwarding, routing and delivery error reporting are the network layer’s primary responsibilities.
- The network layer involves each and every host and router in the network. The role of the network layer in a sending host is to begin the packet on its journey to the receiving host.

- Three important network-layer functions are:
- **Path determination.: The network layer must determine the route or path taken by** packets as they flow from a sender to a receiver. The algorithms that calculate these paths are referred to as routing algorithms.
- **Switching: When a packet arrives at the input to a router, the router must move it to the** appropriate output link.
- **Call setup: With TCP, a three-way handshake is required before data actually flow from** sender to receiver. This allowed the sender and receiver to set up the needed state information (for example, sequence number and initial flow control window size). Some network architectures require router call setup along path before data flows.

- Introduction to Transport Layer
- The transport layer is a 4th layer from the top.
- The main role of the transport layer is to provide the communication services directly to the application processes running on different hosts.
- **The transport layer provides a logical communication between application processes running on different hosts**. Although the application processes on different hosts are not physically connected, application processes use the logical communication provided by the transport layer to send the messages to each other.
- The transport layer **protocols are implemented in the end systems but not in the network routers.**
- A computer network provides more than one protocol to the network applications. For example, **TCP and UDP are two transport layer protocols that provide a different set of services to the network layer.**

- The functions of Transport layer are as follows:
- **Segmentation and Reassembly:**

Breaks down large data into smaller segments for transmission and reassembles them at the destination.

- **End-to-End Communication:**

Ensures reliable data transfer between source and destination, providing end-to-end communication

- **Flow Control:**

Manages data flow to prevent overwhelming the receiving device.

- **Error Detection and Correction:**

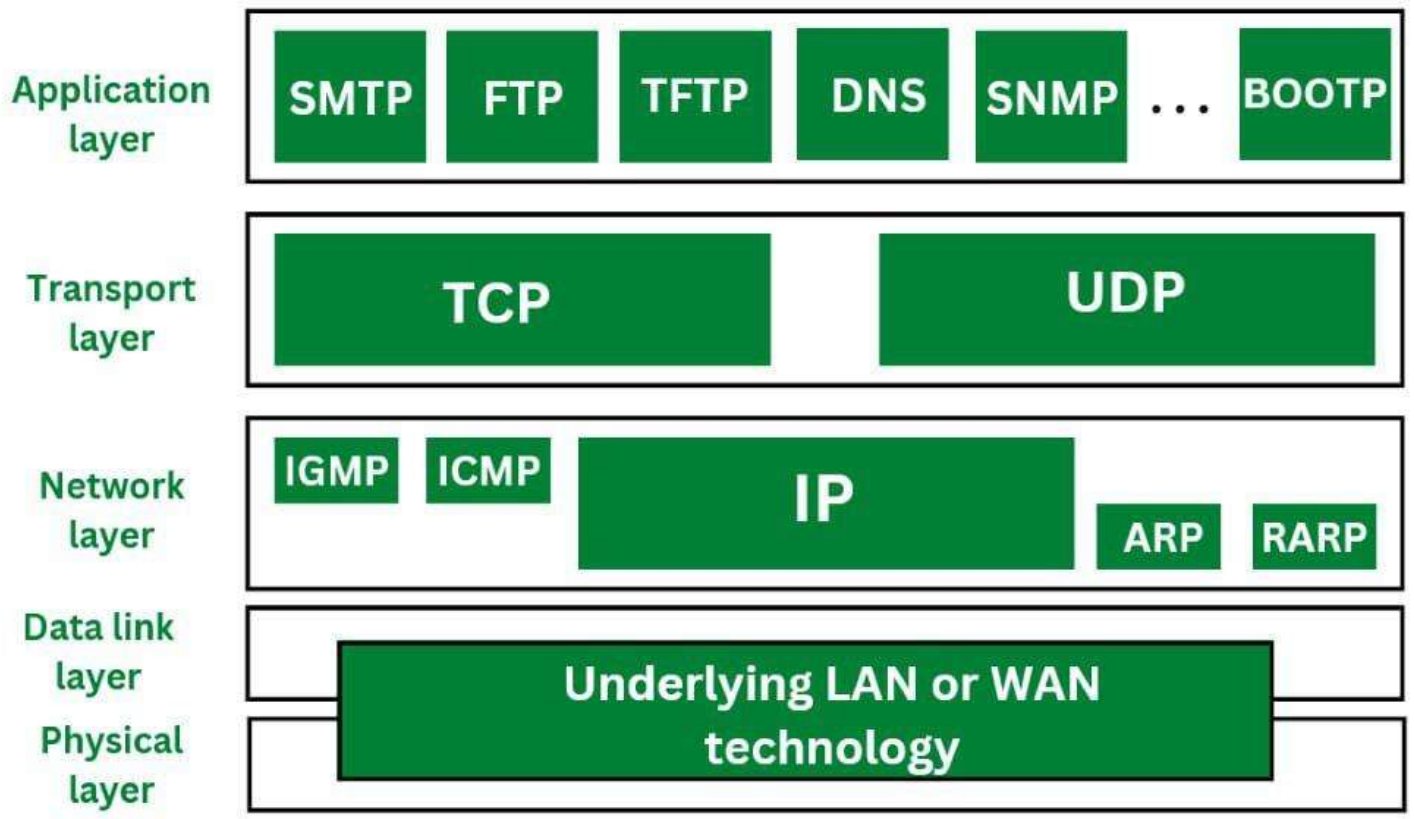
Identifies and corrects errors that may occur during transmission.

- **Connection Management:**

Establishes, maintains, and terminates connections between devices.

• Transport and Network Layer Protocols

- There are different transport/network layer protocols, but one family of protocols, **the Internet Protocol Suite, dominates**. Each transport and network layer protocol performs essentially the same functions, but each is incompatible with the others unless there is a special device to translate between them.



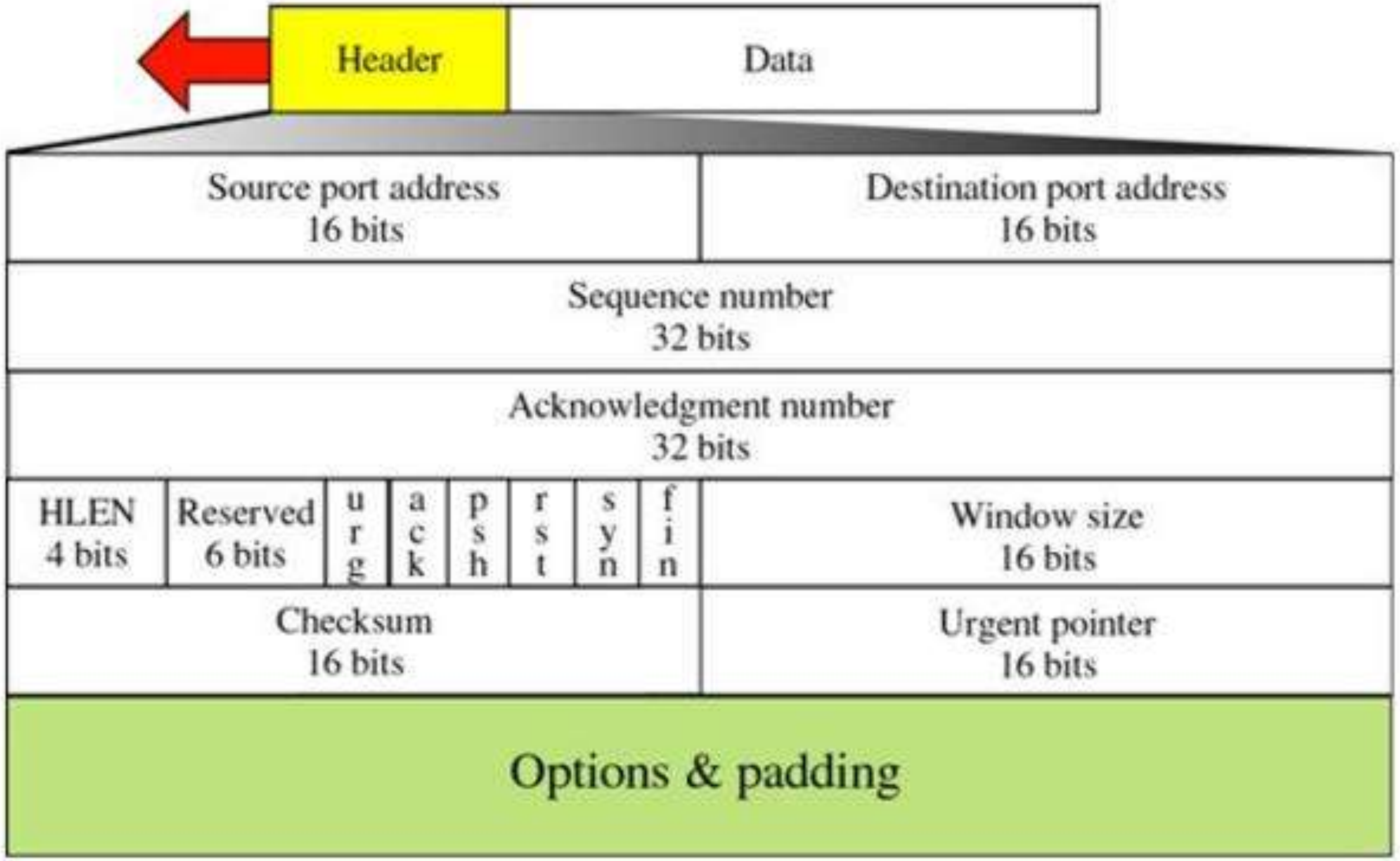
- **Transmission Control Protocol (TCP)**
- TCP is a standard that defines how to establish and maintain network conversations for data exchange between application programs.
- **Collaboration with IP:**
- Works with **Internet Protocol (IP)** to define how computers send and receive data packets, forming the foundation of the Internet.
- **Connection-Oriented Protocol:**
- **Connection Established:** TCP sets up and maintains a connection until data exchange is complete.
- **Key Functions:**
- **Data Segmentation:** Breaks application data into packets for network delivery.
- **Packet Management:** Sends and receives packets from the Network Layer.
- **Flow Control:** Manages data flow to prevent congestion.
- **Error Handling:** Ensures error-free transmission with packet retransmission and acknowledgment of received packets.

- **TCP in Action: Example & Operation**
- **Example:**
- **Web Server Sending HTML:**
  - Uses **HTTP** to request the **TCP** layer to set up a connection and send the file.
  - **TCP:** Divides the file into packets, numbers them, and forwards them to the **IP layer** for delivery.
  - **IP Layer:** Delivers packets, possibly along different routes, to the client.
  - **Client's TCP Layer:** Waits for all packets, acknowledges received ones, requests retransmission of missing packets, and assembles them into the complete file.
- **Operation of TCP:**

1. **Connection Establishment:**
   1. Sets up a connection before data transmission.
2. **Data Transmission:**
   1. Handles the reliable and sequential transmission of data packets.
3. **Connection Termination:**
   1. Properly terminates the connection after data exchange.

- Features of TCP
- Stream Data Transfer
- **Content:**Applications at the Application Layer transfer a continuous stream of bytes to lower layers.
- **TCP's Role:**
  - Packs the byte stream into packets called TCP segments.
  - Segments are passed to the IP layer for transmission.
- Reliability
- **Reliability:** The most important feature of TCP is reliable data delivery. In order to provide reliability, TCP must recover from **data that is damaged, lost, duplicated, or delivered out of order by the network layer**. **TCP assigns a sequence number to each byte transmitted, and expects a positive acknowledgment (ACK**) from the receiving TCP layer. If the ACK is not received within a timeout interval, the data is retransmitted. The receiving TCP uses the **sequence numbers to rearrange the TCP segments** when they arrive out of order, and to eliminate duplicate TCP segments.

- Flow Control
  - **Solution:** TCP uses a sliding window mechanism for flow control.
  - **Sequence Numbers:** Data segments are assigned sequence numbers at the byte level.
  - **Receiver’s Role:** The receiving device informs the sender about the number of bytes it can handle via the ACK.
- **Multiplexing**: Multitasking achieved through the use of port numbers.
- **Connections**
- Before data transmission, a connection is established between the sender and receiver.
- Connections are made using the port numbers of the devices.
- **Full duplex**: TCP provides for concurrent data streams in both directions


Header
Data
Source port address 16 bits
Destination port address 16 bits
Sequence number 32 bits
Acknowledgment number 32 bits
HLEN 4 bits
Reserved 6 bits
urg
ack
psh
rst
syn
fin
Window size 16 bits
Checksum 16 bits
Urgent pointer 16 bits
Options & padding

- **1.Source port address:**
- This is a 16-bit field that defines the port number of the application program in the host that is
- sending the segment.
- **2.Destination port address:**
- This is a 16-bit field that defines the port number of the application program in the host that is
- receiving the segment.
- **3. Sequence number (32-bit):**
- The sequence number of the first data octet in this segment (except when SYN is present). If SYN is
- present the sequence number is the initial sequence number (ISN) and the first data octet is ISN+1. It
- puts the data back in the correct order or retransmits missing or damages data, a process called
- sequencing.
- **4.Acknowledgement number (32 bit):**
- Defines which TCP octet expected next. If the ACK control bit is set this field contains the value of
- the next sequence number the sender of the segment is expecting to receive. Once a connection is
- established this is always sent.

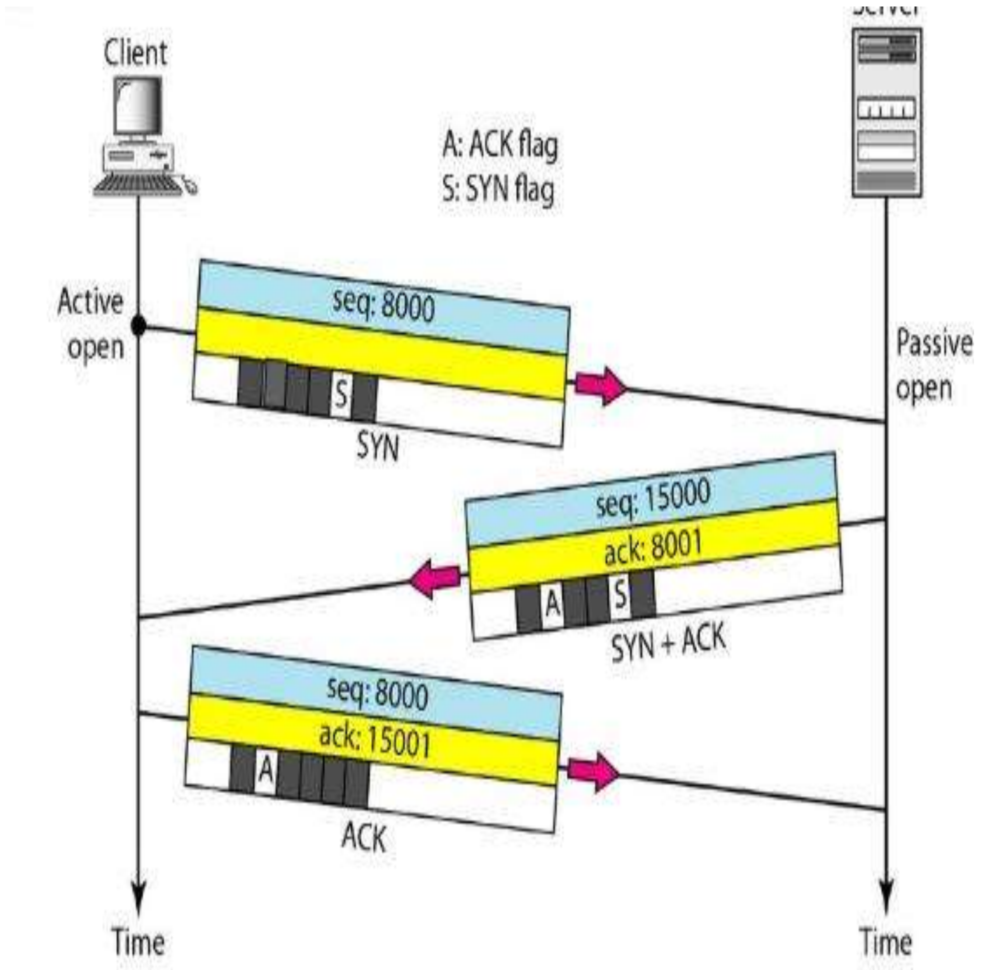
- **Header Length (4 bit):**
- Stands for header length, which defines the number of 32 bit words in the header. **This indicates where the data begins**. The TCP header (even one including options) is an integral number of 32 bits long.
- **6. Reserved (6 bits):**
- Reserved for future use, it always set to 0.
- **7. Control Bits (6 bits from left to right):**
- URG: Urgent Pointer field significant
- ACK: Acknowledgment field significant
- PSH: Push Function
- RST: Reset the connection
- SYN: Synchronize sequence numbers
- FIN: No more data from sender

- **Window Size (16 bits):**
- The number of data octets beginning with the one indicated in the acknowledgement field
- which the sender of this segment is willing to accept.
- **9.Checksum (16 bits):**
- This field contains a checksum of the header. It actually uses a modified form of the header
- which includes some of the information from the IP header to detect some unusual types of
- errors.
- **10. Urgent Pointer:**
- It is only valid if the urgent flag is set, i.e. **it is used when the segment contains urgent data.**
- There is provision in TCP for some urgent messages to be sent bypassing the normal
- sequence.
- **11.Options and Padding:**
- There can be upto 40 bytes of optional information in the TCP header according to the need.
- It can be padded with 0’s if the length is less

- **TCP Connection Establishment using Three-way Handshaking**

- TCP provides reliable communication with something called **Positive Acknowledgement with Re- transmission (PAR).** The Protocol Data Unit (PDU) of the transport layer is called segment. Now a device using PAR resend the data unit until it receives an acknowledgement. **If the data unit received at the receiver's end is damaged** (It checks the data with checksum functionality of the transport layer that is used for Error Detection), **then receiver discards the segment**. **So, the sender has to resend the data unit for which positive acknowledgement is not received**. You can realize from above mechanism that three segments are exchanged between sender (client) and receiver (server) for a reliable TCP connection to get established. Let's see how this mechanism works:

- **Step 1: SYN (Client Initiates Connection)**
- **Client Sends SYN:** The client wants to start a connection, so it sends a segment with a SYN (Synchronize Sequence Number).
- **Purpose:** This tells the server that the client is ready to communicate and indicates the starting sequence number.
- **Step 2: SYN + ACK (Server Responds)**
- **Server Sends SYN-ACK:** The server responds with a segment that has both SYN and ACK bits set.
- **Purpose:**
    - **ACK:** Acknowledges the client's SYN.
- **SYN:** Indicates the server's starting sequence number for the connection
- **Step 3: ACK (Client Confirms Connection)**
- **Client Sends ACK:** The client acknowledges the server's response.
- **Purpose:** Confirms the connection, allowing both the client and server to start data transfer.



- Data Transfer

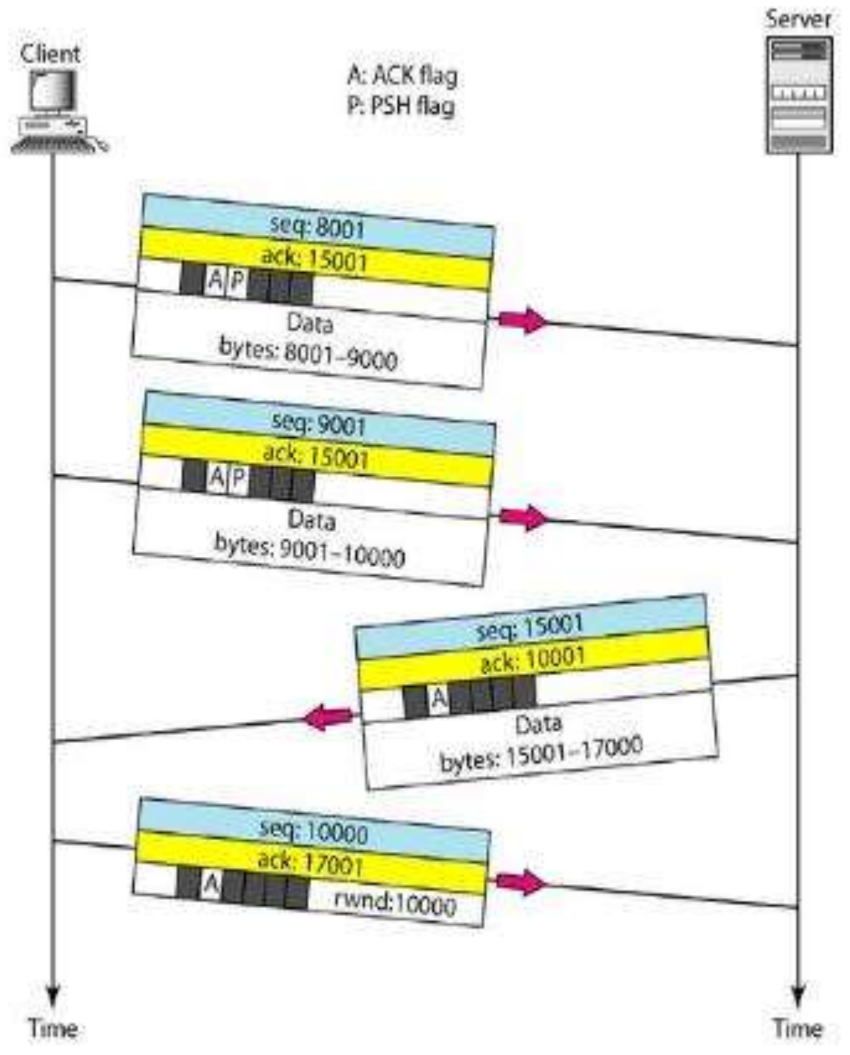
Data are buffered by the transport entity on both transmission and reception. TCP

normally exercises its own discretion as to when to construct a segment for transmission

and when to release received data to the user. The PUSH flag is used to force the data so

far accumulated to be sent by the transmitter and passed on by the receiver. This serves an end-of-block function. The user may specify a block of data as urgent. TCP will designate the end of that block with an urgent pointer and send it out in the ordinary data stream. The receiving user is alerted that urgent data are being received. If during data exchange, a segment arrives that is apparently not meant for the current connection, the (REST)RST flag is set on an outgoing segment. Examples of this situation are delayed duplicate SYNs and a acknowledgment of data not yet sent.

- **Connection Established**
  - **Client Sends Data:** 2,000 bytes in two segments.
  - **Server Responds:** 2,000 bytes in one segment.
  - **Final Client Segment:** Sends one more segment with only an acknowledgment (no data).
- **Sequence and Acknowledgment Numbers**
  - **Track Data:** Each segment has sequence and acknowledgment numbers to manage data flow.
- **Push (PSH) Flag**
  - **Client Segments:** PSH flag set to prompt immediate data delivery to the server process.
  - **Server Segment:** PSH flag not set; TCP implementations can choose to use this flag.



- **Connection Termination**

Initiating Closure

- **Either Party:** Both client and server can close the connection.
- **Common Practice:** Usually initiated by the client.

Termination Methods

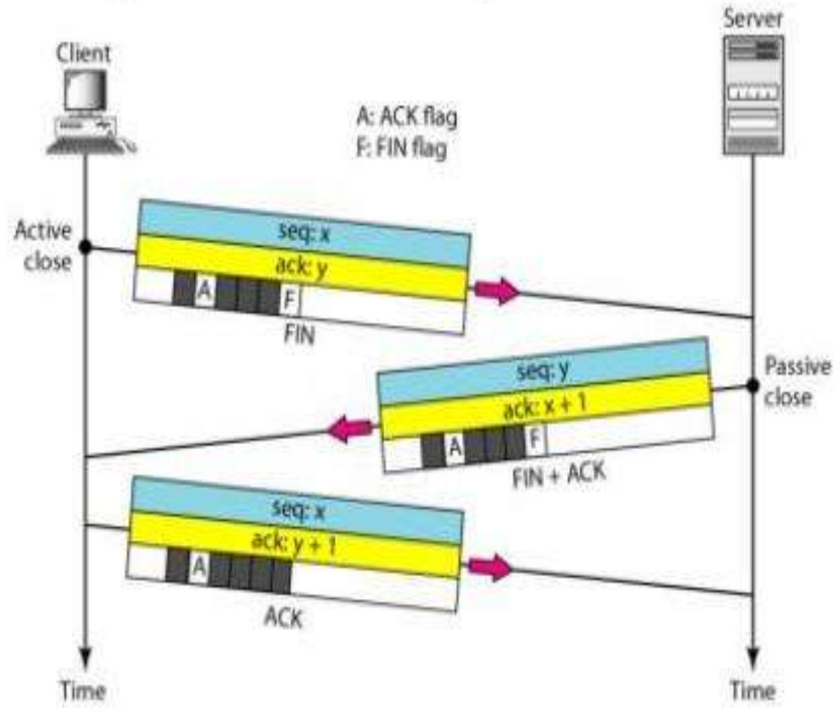
- **Three-Way Handshaking:** Commonly used for connection termination.
- **Four-Way Handshaking with Half-Close:** Alternative method, allowing partial closure.

- **Three-Way Handshaking Process**

1. **FIN:** One party sends a FIN (Finish) segment to close the connection.

2. **ACK:** The other party acknowledges with an ACK segment.

3. **FIN + ACK:** The second party sends a FIN segment , and the first party acknowledges.

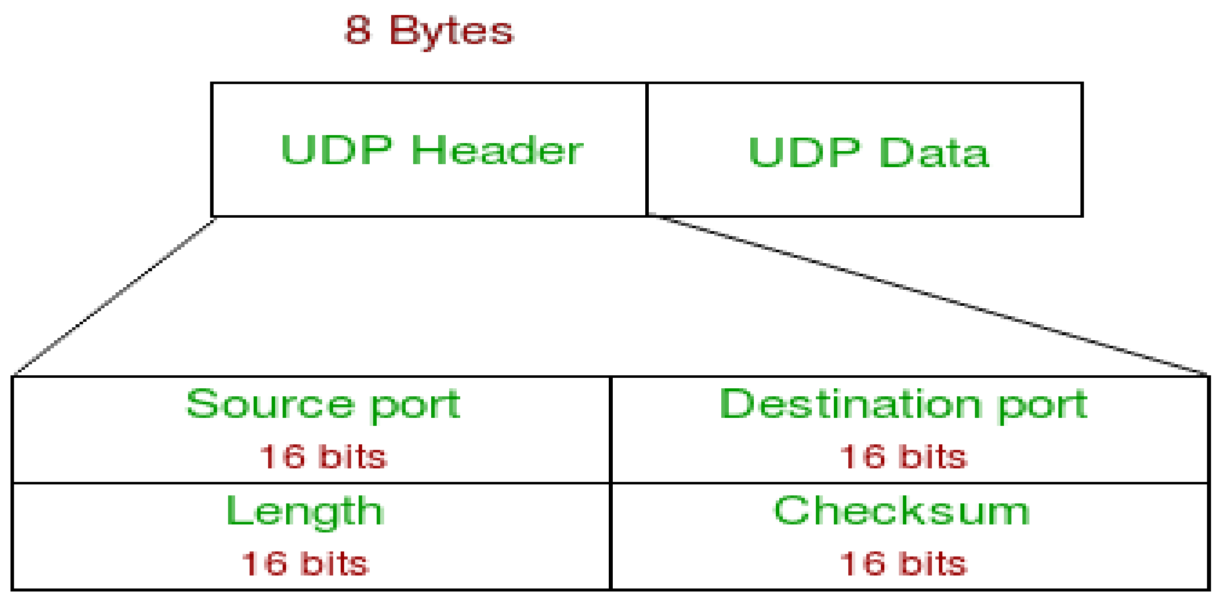


- **Termination Process**
- **1. Client Sends FIN**
    - **Action:** Client TCP sends a FIN (Finish) segment to indicate it wants to close the connection.
    - **Details:** Can include final data or just be a control segment (consumes one sequence number if no data).
- **2. Server Responds with FIN + ACK**
    - **Action:** Server TCP sends a FIN + ACK segment to:
        - **Acknowledge** the client's FIN segment.
        - **Initiate** the closure of the connection in the server-to-client direction.
    - **Details:** Can include final data or just be a control segment (consumes one sequence number if no data).
- **3. Client Sends ACK**
    - **Action:** Client TCP sends an ACK segment to acknowledge receipt of the server’s FIN segment.
    - **Details:** Contains an acknowledgment number (one more than the sequence number of the server's FIN) and does not carry data or consume sequence numbers.

•**User Datagram Protocol (UDP)**

- User Datagram Protocol (UDP) is a Transport Layer protocol. UDP is a part of Internet Protocol suite, referred as UDP/IP suite. Unlike TCP, it is **unreliable and connectionless protocol. So, there is no need to establish connection prior to data trans**fer.
- For the real-time services **like computer gaming, voice or video communication, live conferences**; we need UDP. Since high performance is needed, UDP permits packets to be dropped instead of processing delayed packets. There is no error checking in UDP, so it also save bandwidth.
- User Datagram Protocol (UDP) is more efficient in terms of both latency and bandwidth.

- **UDP Header:**
- UDP header is 8-bytes fixed and simple header. First 8 Bytes contains all necessary header information and remaining part consist of data. UDP port number fields are each 16 bits long, therefore range for port numbers defined from 0 to 65535; port number 0 is reserved. Port numbers help to distinguish different user requests or process.



1.Source Port : Source Port is 2 byte long field used to identify port number of source.

2.Destination Port : It is 2 byte long field, used to identify the port number of destination.

3.Length : It is 2 bytes long field which is the length of UDP including header and the data.

4.Checksum : Checksum is 2 bytes long field for error control.

Notes – Unlike TCP, Checksum calculation is not mandatory in UDP. No Error control overflow control is provided by UDP. Hence UDP depends on IP and ICMP for error reporting.

- **UDP Operations:**

The basic steps for transmission using UDP are:

1. Higher-Layer Data Transfer: An application sends a message to the UDP software.
2. UDP Message Encapsulation: The higher-layer message is encapsulated into the Data field of a UDP message. The headers of the UDP message are filled in, including the Source Port of the application that sent the data to UDP, and the Destination Port of the intended recipient. The checksum value may also be calculated.
3. Transfer Message To IP: The UDP message is passed to IP for transmission.

On reception at the destination device this short procedure is reversed.

- **Applications of UDP:**

- Used for simple request response communication when size of data is less and
- hence there is lesser concern about flow and error control.
- It is suitable protocol for multicasting as UDP supports packet switching.
- UDP is used for some routing update protocols like RIP(Routing Information Protocol).
- Normally used for real time applications which can not tolerate uneven delays

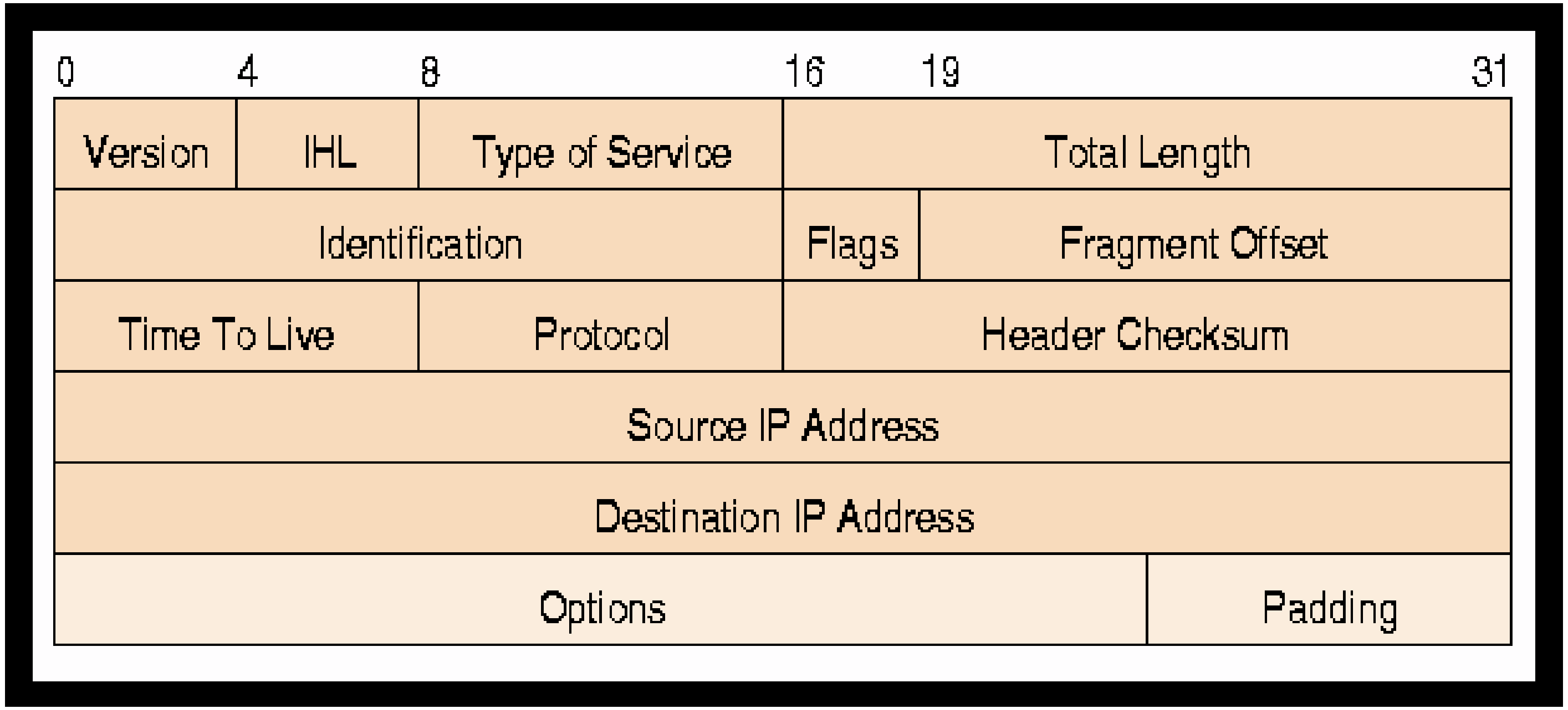
between sections of a received message.

- UDP takes datagram from Network Layer, attach its header and send it to the user. So, it works fast
- Following implementations uses UDP as a transport layer protocol:
- –DNS (Domain Name Service)
- –DHCP

- Internet Protocol
- Purpose: Provides connectionless data transfer over diverse networks.Data Packet: IP datagram is a data packet encapsulated with an IP header for routing.
- **Versions of IPIPv4:**
  - **Introduced:** Early 1980s.
  - **Address Size:** 32 bits.
  - **Header Size:** 20 bytes.
  - **Common Usage:** Still widely used.
- **IPv6:**
  - **Introduced:** 1998.
  - **Address Size:** 128 bits (larger address space).
  - **Header Size:** 40 bytes.
  - **Benefits:** Simpler packet structure, improved routing, and enhanced addressing.

- IPv4
- Definition: A 32-bit address uniquely identifying a device on the Internet. Format: Example - 192.168.1.1
- Total Addresses: 2^32 (4,294,967,296)
- Scope: Unique and universal for devices.

- Global Management IANA: Manages global IP address allocation.
- RIRs: Regional Internet Registries manage allocation in specific regions and to local ISPs.


0
4
8
16
19
31
Version
IHL
Type of Service
Total Length
Identification
Flags
Fragment Offset
Time To Live
Protocol
Header Checksum
Source IP Address
Destination IP Address
Options
Padding

- The header format is 22-60 bytes in length. It contains the necessary information for routing and delivery of data. Data field carries data for next layer.
- 1. Version (ver): It is 4 bits & keeps track of which version of protocol the datagram belongs. The current version is IPv4 and latest is IPv6.
- 2. Header length (HLEN): It is a 4 bit field that represents the size of a header. It rangers from 5 –15 in value.
- 3. Service(8 bit): determines datagram handling
- DS (Differentiated Service): 6 bits for precedence, delay, throughput, and reliability.ECN (Explicit Congestion Notification): 2 bits for congestion signaling.
- Total length (16 bit): It defines the total length of an IP datagram including IP header and IP payload. It is usually written in octets.

5. Identification: It allows the destination host to determine which datagram a newly arrived fragment belongs to. It is primarily used for uniquely identifying fragments of an original IP datagram. If IP packet is fragmented during the transmission, all the fragments contain same identification number to identify original IP packet they belong to.

6. Flag (3 bit): As required by the network resources if IP packet is too large to handle these flag bits tell if they can be fragmented or not. **It is used in fragmentation and reassembly.**

7. Fragmentation offsets (13 bits): **It is used to define where in the current datagram this fragment belongs.** This offset tells the exact position of the fragment in the original IP Packet.

8. Time to live (TTL) (8 bits): It specifies how long in seconds a data gram is allowed to remain in the internet. The maximum life is 255 seconds.

9.Protocols (8 bits): It defines the next higher level protocols that is to receive the data frame at the destination. Example: TCP, UDP.

10. Header checksum (16 bits): It consists an **error detecting code for header** only because some header fields may alter during transmission.

11. Source and destination address (32 bits each): It denotes the network address of source and destination.

12. Options: It is used for future. The current defined options are security, source routing, route recording, string identification, time stamp.

13. Padding: It is used to ensure the datagram header is a multiple of 32 bits in length.

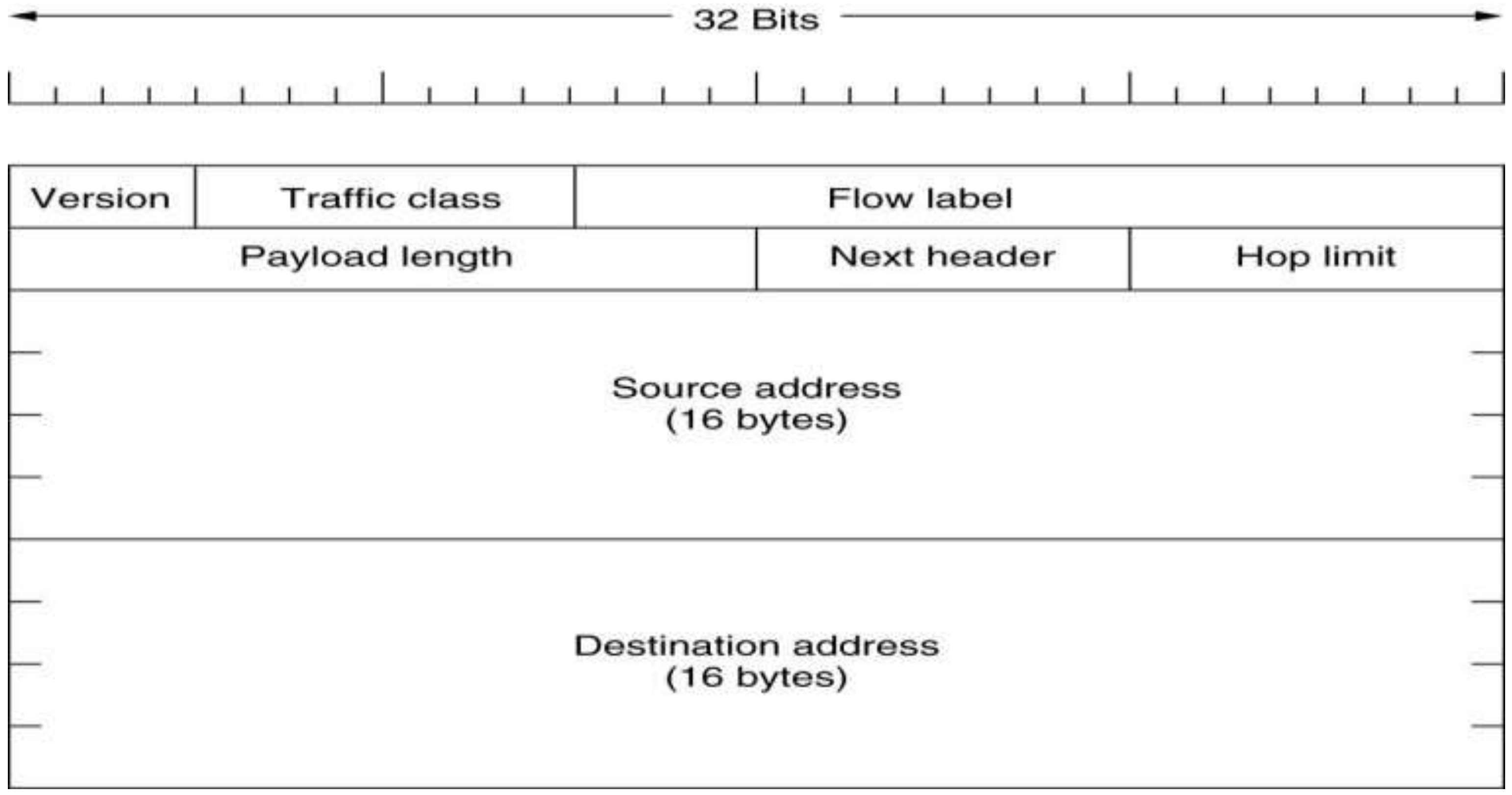
- Notation of IP Address:
- There are two common notations to show IP address
- Binary notation
- Dotted decimal notation
- Binary Notation:
- In binary notation, The IP address is displayed as 32-bits and this 32-bits is represented in 4-Octet
- (8bits) address or 4-bytes address. Example: 01110101. 10010101. 00011101. 11101010

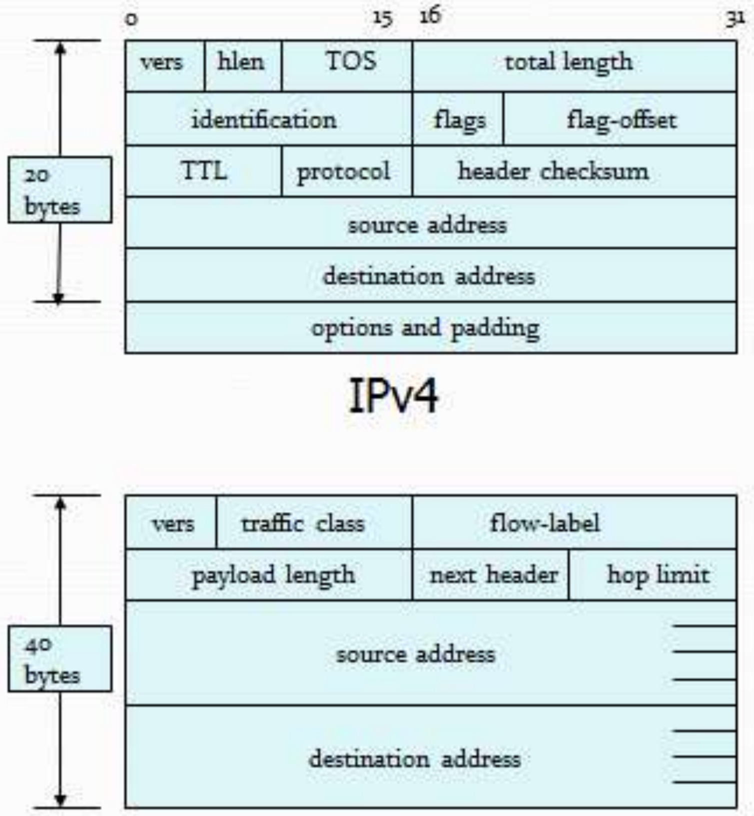
- • Dotted-Decimal Notation:
- • To make the IP address more compact and easier to read, IP addresses are usually written in
- decimal form with separating each byte by a dot (.). Each number in dotted-decimal notation is between 0 and 255.
- • For Example: 192.168.0.100

- **IP address: It can be assigned in two ways:**
- Static
- Dynamic
- Static address: It is manually inserted while joining the network. It is quite manageable as the user defines it, him/herself. It isn’t feasible in large organization.
- • Dynamic address: Dynamic address are automatically assigned to a device. DHCP (Dynamic Host Configuration Protocol) assigns IP addresses to any new device in the network automatically. It’s quite feasible in large organization as a large amount of address can be assigned without manual intervention.

- **IPv6 Address Format**
    - **Structure:** Eight groups of four hexadecimal digits separated by colons.
    - **Example:** 2001:0db8:85a3:0000:0000:8a2e:0370:7334
- **Address Length**
    - **Size:** 128 bits (16 bytes)
    - **Benefit:** Provides a vastly larger address space compared to IPv4.
- **Shortened Notation**
    - **Leading Zeros:** Can be omitted within each group.
        - **Example:** 2001:db8:85a3:0:0:8a2e:370:7334
    - **Consecutive Zeros:** Use :: to replace one or more consecutive groups of zeros.
        - **Example:** 2001:db8:85a3::8a2e:370:7334
- **Key Features**
    - **Larger Address Space:** Supports 2^128 addresses.
    - **Simplified Header:** Improved efficiency for routing and processing.
    - **Enhanced Security:** Built-in support for IPsec.
- **IPv6 Transition**
    - **Coexistence with IPv4:** IPv6 is designed to coexist with IPv4 during the transition period.

- IPv6 Header format
- **Header Fields**
- **1. Version (4 bits)**
  - **Function:** Indicates the IP version (IPv6).
- **2. Traffic Class (8 bits)**
  - **Function:** Similar to the "Service" field in IPv4; it is used for quality of service (QoS) and traffic management.
- **3. Flow Label (20 bits)**
  - **Function:** Identifies packets that belong to the same flow, providing a way to handle traffic with special requirements.
- **4. Payload Length (16 bits)**
  - **Function:** Specifies the length of the payload (data) following the header.
- **5. Next Header (8 bits)**
  - **Function:** Indicates the type of header immediately following the IPv6 header (e.g., TCP, UDP).
- **6. Hop Limit (8 bits)**
  - **Function:** Specifies the maximum number of hops (routers) the packet can pass through; similar to the TTL in IPv4.
- **7. Source Address (128 bits)**
  - **Function:** Specifies the address of the sender.
- **8. Destination Address (128 bits)**
  - **Function:** Specifies the address of the recipient


32 Bits
Version
Traffic class
Flow label
Payload length
Next header
Hop limit
Source address
(16 bytes)
Destination address
(16 bytes)



IPv4

IPv6

Removed (6)

- ID, flags, flag offset
- TOS, hlen
- header checksum

Changed (3)

- total length => payload
- protocol => next header
- TTL => hop limit

Added (2)

- traffic class
- flow label

Expanded

- address 32 to 128 bits

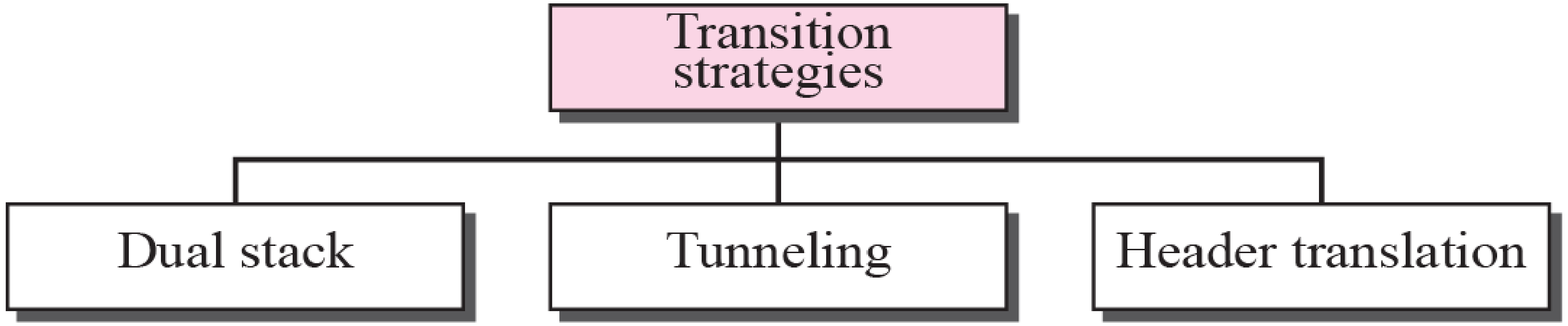
- **IPv4 to IPv6 Transition**

Because of the huge number of systems on the Internet, the transition from IPv4 to IPv6 cannot happen suddenly.

It will take a considerable amount of time before every system in the internet can move from IPv4 to IPv6

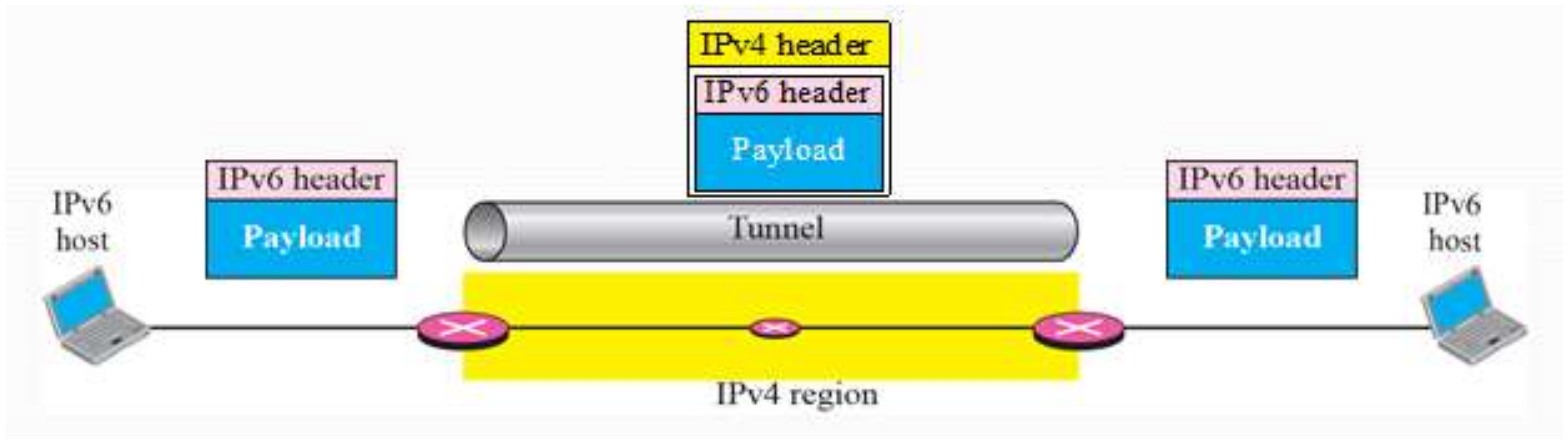
The transition must be smooth to prevent any problems between IPv4 and IPv6 systems.

Three strategies to help the transition :

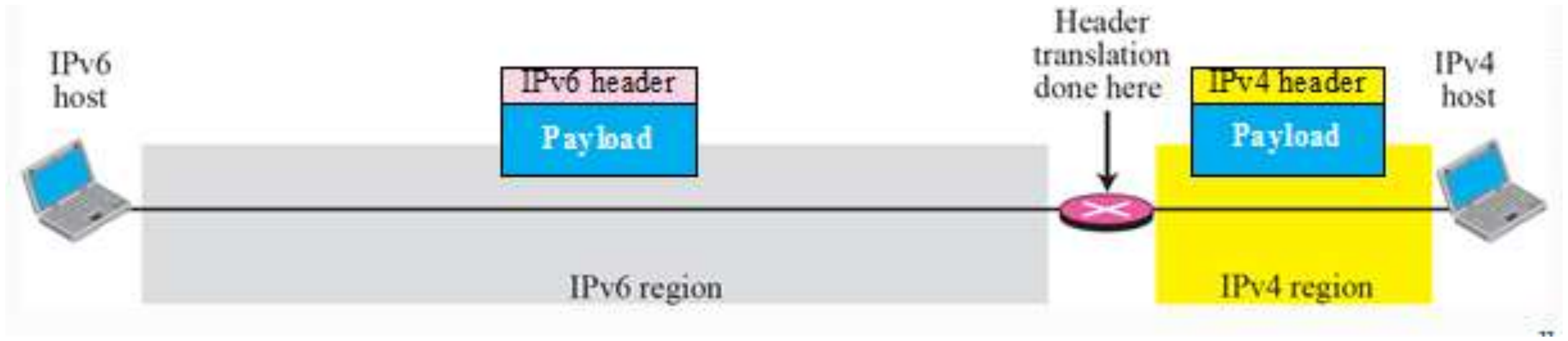


- **Dual Stack Overview**
- **Definition:** Devices run both IPv4 and IPv6 simultaneously.
- **Packet Handling**
- **Received Packets:** The version field in the header determines whether to use IPv4 or IPv6.
- **Sending Packets:** The source host queries DNS to determine the appropriate protocol.
- **DNS Query Process**
- **IPv4 Address:** If DNS returns an IPv4 address, the packet is sent using IPv4.
- **IPv6 Address:** If DNS returns an IPv6 address, the packet is sent using IPv6.
- **Transition Period**
- **Parallel Operation:** Allows gradual transition to IPv6 while maintaining IPv4 connectivity

- **Tunnelling**
- Tunneling is used to deal with networks where v4 router(s) sit between two v6 routers
- Simply encapsulate v6 packets and all of their information in v4 packets and leaves its capsule when it exist the region.



- **Header Translation**
- Header Translation is necessary when the majority of the internet has moved to IPv6 but some still use IPv4.
- Converting IPv6 Header to IPv4 and vice versa
- Used to link IPv6 and IPv4 Networks
- Some header fields may lost during translation
- Eg. When converting IPv6 Packet to IPv4, the header information like *Flow-label gets lost*



- Transport Layer Functions
- **Linking Application to Network**
- Role: Connects application software in the application layer with the network.
- **Segmenting Large Messages**
- Function: Breaks down large messages into smaller segments for transmission.
- Purpose: Ensures efficient data transfer and reassembly at the destination.
- **Session Management**
  - **Function:** Manages the end-to-end delivery of messages.
  - **Purpose:** Maintains the connection and ensures reliable communication.
- **Address Resolution**
  - **Process:** Finding the numeric network address of the destination.
  - **Note:** Can involve transport, network, data link, or application layer functions depending on the protocol.

- Linking to the Application Layer
- **Multiple Applications**
- **Scenario:** Computers often run multiple application programs simultaneously (e.g., Web browsers, email clients).
- **Message Delivery**
- **Transport Layer Role:** Decides which application program should receive the incoming message.
- **Purpose:** Ensures messages are directed to the correct application (e.g., Web page requests to Web server softwar
- **Port Addresses**
- **Unique Identifiers:** Each application has a unique port address.
- **Format:** Port addresses are 16-bit (2-byte) numbers.

- **Scenario**
- **User's Applications:** Internet Explorer, Outlook, RealPlayer running simultaneously.
- **Temporary Port Numbers**
- **Assigned Ports:**
    - **Internet Explorer:** Port 1027
    - **Outlook (Email Client):** Port 1028
    - **RealPlayer:** Port 1029
- **Simultaneous Communication**
- **Internet Explorer:** Sends data to Web server on **port 80** at **xyz.com**.
- **Outlook:** Receives data from mail server on **port 25**.
- **RealPlayer:** Requests data from music server on **port 554** at **123.com**.
- **Port Function**
- **Purpose:** Ensures data from different applications is correctly routed to and from the appropriate services

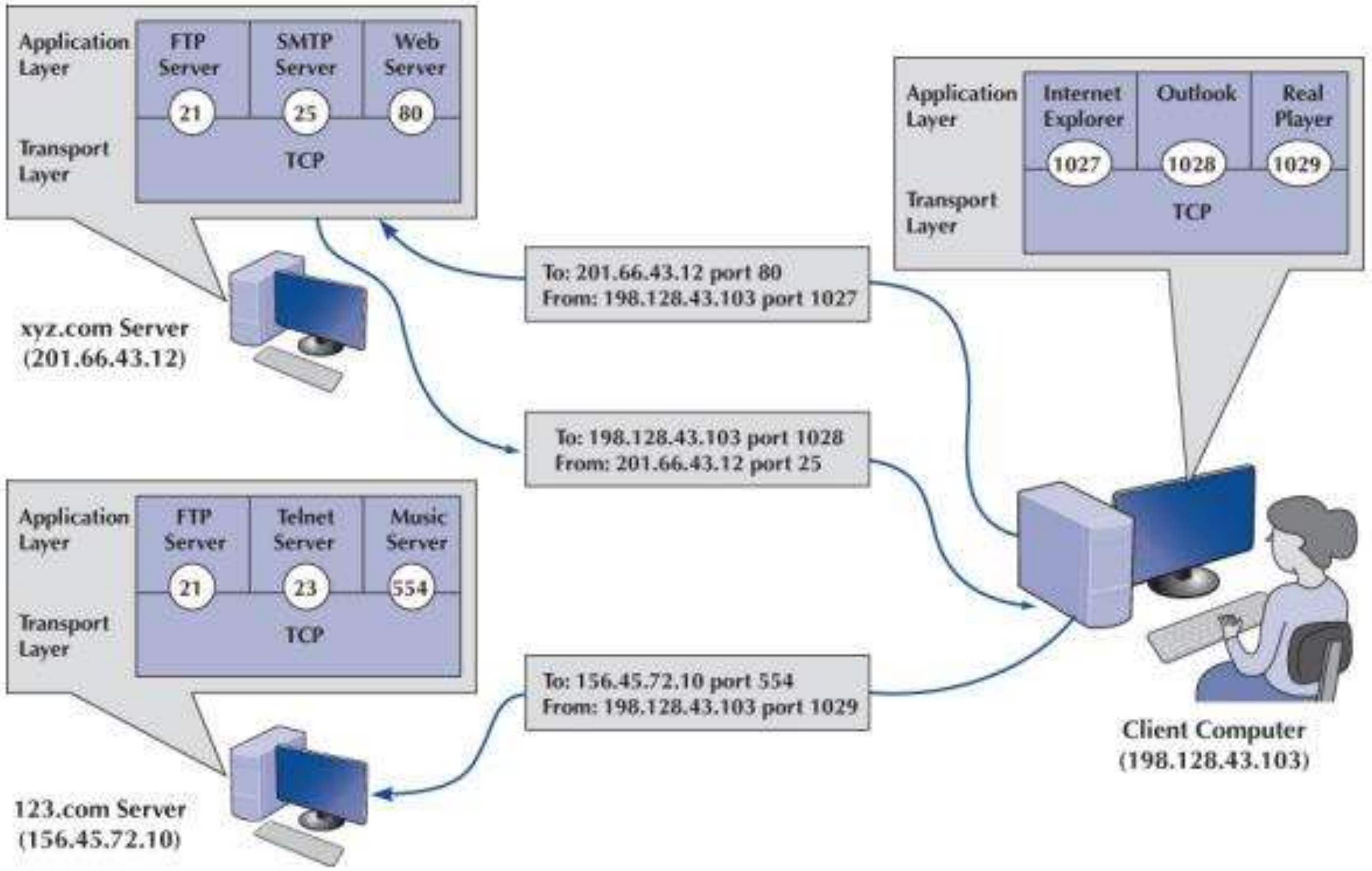


FIGURE 5-5 Linking to application layer services

- **Segmenting**
- **What is Segmenting?**
- **Definition:**
  - Breaking large messages into smaller segments for transmission.
  - Reassembling segments into complete messages on receipt.
- **Why Segmenting is Needed**
- **Purpose:**
  - Large data may exceed the frame size at the data link layer.
  - Transport layer splits data to match the frame size.
- **Segmenting in Action**
- **Examples:**
  - **Web Browsers:** Gradual page loading as packets arrive.
  - **Email:** Full message displayed after all packets are received.

- **TCP’s Role in Segmenting**
- **Reliability:**
  - Ensures all segments are received using Continuous (Automatic repeat request)ARQ.
  - Negotiates optimal segment size with the receiver.

- Session Mgmt
- **What is a Session?**
- **Definition:** A session is like a conversation between two computers.
- **Purpose:** Manages the sequence and flow of data between sender and receiver.
- **Connection-Oriented Messaging**
- **Process:**
  - Establishes a TCP connection (session) between sender and receiver.
  - Sender initiates with a SYN segment; session closed with a FIN segment.
  - Ensures all segments arrive using continuous ARQ (sliding window).
- **Connectionless Messaging**
- **Process:**
  - No session setup; each packet is sent independently.
  - Packets may take different routes and arrive out of order.
  - Network layer adds sequence numbers to ensure correct reassembly.

- **TCP vs. UDP**
- **TCP:** Used for connection-oriented messaging; reliable, ordered delivery.
- **UDP:** Used for connectionless messaging; smaller packets, faster, but no guarantee of order or delivery.

- Addressing
- **Four Levels: Physical Address**
- **Logical Address**
- **Port Address**
- **Specific Address**
- **Physical Addresses**
- **Definition:** Also known as the link address.
- **Characteristics:** Defined by LAN or WAN; varies by network.
- **Example:** Ethernet uses a 48-bit (6-byte) physical address.
- **Logical Addresses**
- **Purpose:** Used by networking software for independent packet delivery across different network types.
- **Characteristics:** A 32-bit address in IPv4.
- **Example:** Internet address like 132.24.75.9.

- **What Are Port Addresses?**
- **Port Number**: A port number is a logical identifier for a specific process or service on a computer within a network. It works alongside an IP address to help differentiate between multiple services or applications running on the same device.
  - **IP Address + Port Number**: Think of an IP address as the street address of a building (your computer), while a port number is like an apartment number within that building (specific application). Together, they ensure that data is delivered to the correct application.
- **Application-Specific Addresses**
- In addition to port numbers, some applications use more human-readable addresses:
- Port 80: HTTP (Web traffic) Port 443: HTTPS (Secure web traffic)
- Port 25: SMTP (Email sending) Port 21: FTP (File Transfer Protocol)

• IP address

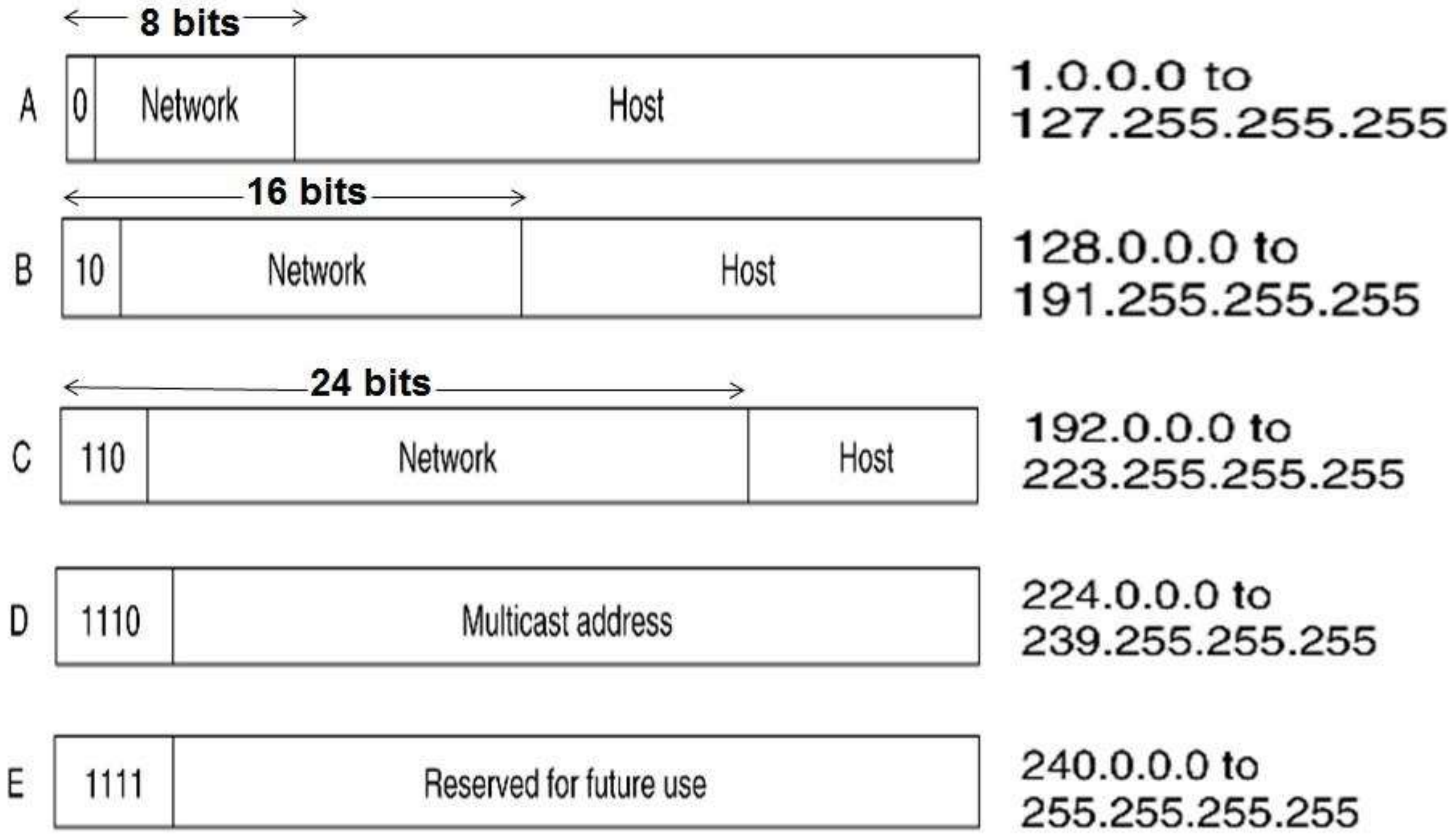
- An **IP Address** (Internet Protocol Address) is a unique identifier assigned to each device connected to a network that uses the Internet Protocol for communication. Think of it as the "address" of your device on the internet or a local network, much like how your home has a unique postal address.

- There are two versions of IP addresses:

1. **IPv4**: This is the most commonly used version. It consists of four numbers separated by dots (e.g., 192.168.1.1). Each number can range from 0 to 255.
2. **IPv6**: This is the newer version designed to replace IPv4 because of the limited number of IPv4 addresses. It looks different (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334).

- **IPv4**
- An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a device (for example, a computer or a router) to the Internet
- The IPv4 addresses are unique and universal.
- The address space of IPv4 is 232 or 4,294,967,296
- The IP address space is managed globally by the Internet Assigned Numbers Authority (IANA), and by five regional Internet registries (RIRs) responsible in their designated territories for assignment to end users and local Internet registries, such as Internet service providers

- Classful Addressing (IP Address Classes):
- Class A: The range of Class A IP address is 1.0.0.0 to 127.255.255.255
- Class B: The range of Class B IP address is 128.0.0.0 to 191.255.255.255
- Class C: The range of Class C IP address is 192.0.0.0 to 223.255.255.255
- Class D: The range of Class D IP address is 224.0.0.0 to 239.255.255.255
- Class E: The range of Class E IP address is 240.0.0.0 to 255.255.255.255


8 bits
A
0
Network
Host
1.0.0.0 to
127.255.255.255
16 bits
B
10
Network
Host
128.0.0.0 to
191.255.255.255
24 bits
C
110
Network
Host
192.0.0.0 to
223.255.255.255
D
1110
Multicast address
224.0.0.0 to
239.255.255.255
E
1111
Reserved for future use
240.0.0.0 to
255.255.255.255

	First byte	Second byte	Third byte	Fourth byte
Class A	🟨 0			
Class B	🟨 10	🟨		
Class C	🟨 110	🟨	🟨	
Class D	1110			
Class E	1111			

a. Binary notation

	First byte	Second byte	Third byte	Fourth byte
Class A	🟨 0–127			
Class B	🟨 128–191	🟨		
Class C	🟨 192–223	🟨	🟨	
Class D	224–239			
Class E	240–255			

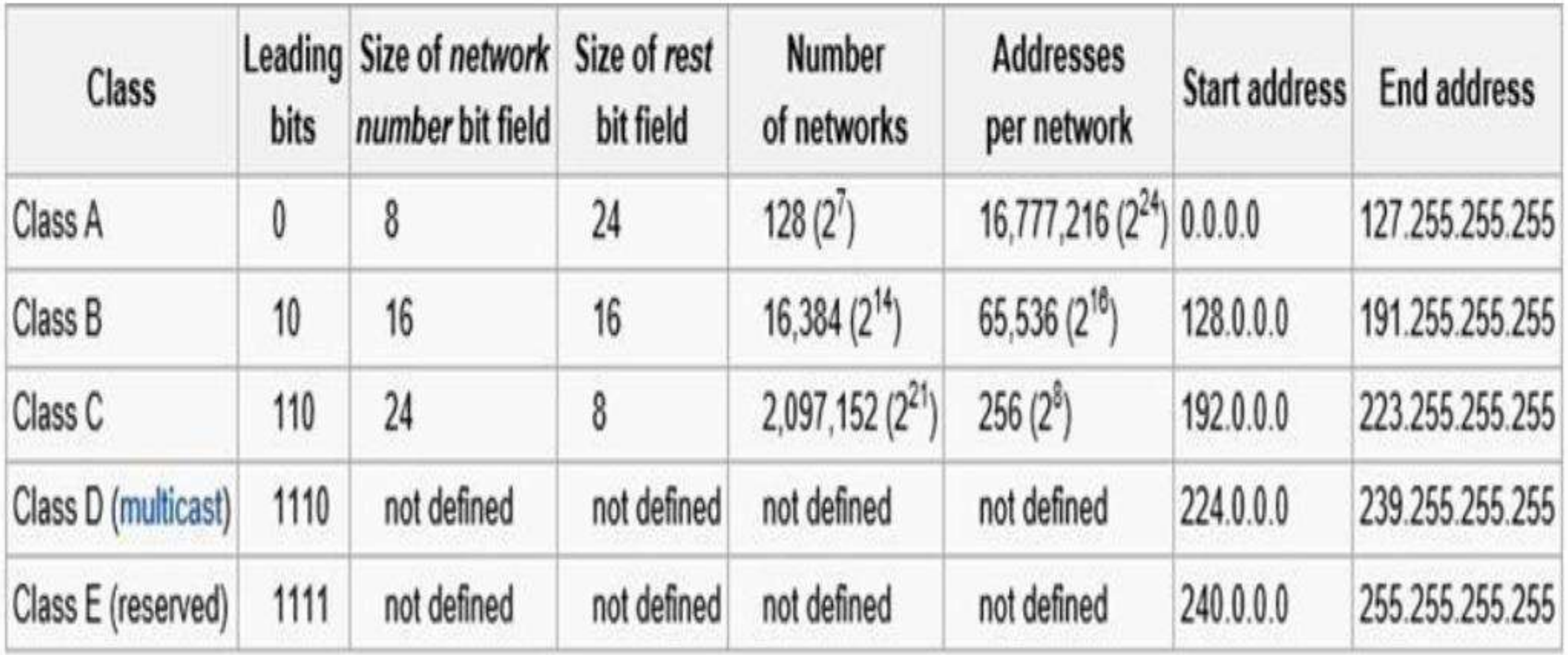
b. Dotted-decimal notation

- **Class Address**
- •The first bit of the first octet is always set to zero. So that the first octet ranges from 1 –127.
- •The class A address only include IP starting from 1.x.x.x to 126.x.x.x. The IP range 127.x.x.x is reserved for
- loop back IP addresses.
- •The default subnet mask for class A IP address is 255.0.0.0. **This means it can have 126 networks (2^7-2) and**
- **16777214 hosts (2^24-2).** (Minus 2 because 2 addresses are reserved for network and broadcast address)
- Class A IP address format is thus: **0NNNNNNN.HHHHHHHH.HHHHHHHH.HHHHHHHH**

- **Class B Address**
- •Here the first two bits in the first two bits is set to zero.
- •Class B IP Addresses range from 128.0.x.x to 191.255.x.x.
- •The default subnet mask for Class B is 255.255.x.x. **Class B has 16384 (2^14) network address and 65534(2^16-2)**
- Host addresses. Class B IP address format is: **10NNNNNN.NNNNNNNN.HHHHHHHH.HHHHHHHH**

- **Class C Address**
- •The first octet of this class has its first 3 bits set to 110. Class C IP addresses range from 192.0.0.x to
- 223.255.255.x.
- •The default subnet mask for Class C is 255.255.255.x. **Class C gives 2097152 (2^21) Network address and 254 (2^8-2)** Host addresses.
- Class C IP address format is: **110NNNNN.NNNNNNNN.NNNNNNNN.HHHHHHHH**
- **Class D Address**
- The first four bits of the first octet in class D IP address are set to 1110.
- Class D has IP address rage from 224.0.0.0 to 239.255.255.255.
- Class D is reserved for Multicasting. In multicasting data is not intended for a particular host, but multiple ones.
- That is why there is no need to extract host address from the class D IP addresses.
- The Class D does not have any subnet mask.
- **Class E Address**
- •The class E IP addresses are reserved for experimental purpose only for R&D or study.
- •IP addresses in the class E ranges from 240.0.0.0 to 255.255.255.254
- This class too is not equipped with any subnet

Summary



Class	Leading bits	Size of *network number* bit field	Size of *rest* bit field	Number of networks	Addresses per network	Start address	End address
Class A	0	8	24	128 (2^7)	16,777,216 (2^{24})	0.0.0.0	127.255.255.255
Class B	10	16	16	16,384 (2^{14})	65,536 (2^{16})	128.0.0.0	191.255.255.255
Class C	110	24	8	2,097,152 (2^{21})	256 (2^8)	192.0.0.0	223.255.255.255
Class D (multicast)	1110	not defined	not defined	not defined	not defined	224.0.0.0	239.255.255.255
Class E (reserved)	1111	not defined	not defined	not defined	not defined	240.0.0.0	255.255.255.255

- **Netid (or network part) and Hostid (or host part)**
- In classful addressing an IP address in classes A, B, and C is divided into netid and host id. These parts are varying length depending on the class of the address.
- **Netid (network part): The portion of the IP address that identifies the network is called the netid.**
- The network part of IP address identifies the network to which the host is attached. All the hosts attached to the same network part (netid) in their IP address.
- **Hostid: The portion of the IP address that identifies the host or router on the network is called the** hostid. The host part identifies each host uniquely on that particular network.

- **Network Address**
- • The network address is an address that defines the network itself; it cannot be assigned to a host.
- Network address plays a very important role in classful addressing. A network address has several properties:
- All hostid bytes are 0s
- The network address defines the network to the rest of internet. Latter we learn that router can route a packet on the network address
- Network address is first address in the block
- Given the network address we can find the class of the address
- **Network address is different from Netid . A network address has both netid and hostid with 0s for hostid.**

- **Reserved IP Address**
- Certain IP addresses are reserved and cannot be assigned to device on a network.
- The addresses in class E are reserved for research purpose.
- The network address is reserved: all 0s in hostid of network (network address is used to identify the network itself)
- Broadcast address is reserved: all 1 in hostid of network (broadcast address is used for broadcasting packets to all the devices on a network)
- In class A IP address 127.xx.yy.zz is reserved for internal loop-back
- **Private IP address**
- Some addresses are allocated to be used for private networks- called private IP address
- ○ 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

- **Classless Addressing**
- To reduce the wastage of IP addresses in a block, we use sub-netting. What we do is that we use host id bits as net id bits of a classful IP address.
- We give the IP address and define the number of bits for mask along with it (usually followed by a ‘/’ symbol), like, 192.168.1.1/28.
- Here, subnet mask is found by putting the given number of bits out of 32 as 1, like, in the given address, we need to put 28 out of 32 bits as 1 and the rest as 0, and so, the subnet mask would be 255.255.255.240.

- Classless Inter-Domain Routing (CIDR)
- **CIDR Overview**:
- A method for efficient IP address allocation, replacing the old Class A, B, C system.
- Allows for variable-length subnet masking (VLSM) to create subnets of varying sizes.
- **Benefits of CIDR**:
- **Efficient IP Address Use**: Prevents wastage by allocating only the needed IP addresses.
- **Route Summarization**: Combines multiple IP networks into a single route, reducing routing table size.

- CIDR Notation:
- Format: IP Address/Prefix Length
- Example: 192.168.1.0/24
- IP Range: 192.168.1.0 - 192.168.1.255
- Subnet Mask: 255.255.255.0

- How CIDR Works
- Assigns a numerical prefix to each IP address (e.g., 177.67.5.44/13).
- Prefix Length: The prefix (/13) indicates that the first 13 bits represent the network.
- Remaining 19 bits identify the host.
- **Routing**:
  - **Longest Prefix Match**: Routers use the longest prefix in routing tables to determine packet forwarding.
  - **Efficiency**: CIDR allows packets to be routed to groups of networks, simplifying routing tables on backbone routers.

- **Dynamic Addressing**
- **Static addressing** requires manually configuring network layer addresses (IP addresses) on each computer.
- This creates a **management challenge** when devices move or network changes occur, as each system needs manual updates.
- **Dynamic addressing** solves this by allowing a server (like a **DHCP server**) to assign an IP address to a device automatically whenever it connects.
- This method is commonly used for **client computers**, making network management easier and more efficient.
- Servers usually maintain **static addresses** for consistent identification

- **DHCP - Dynamic Host Configuration Protocol**
- **Application layer protocol** used by hosts to automatically obtain network setup information.
- **Automates IP address assignment**, removing the need for manual configuration by network administrators.
- **DHCP server** maintains a pool of IP addresses and leases one to hosts as they come online.
- **Simplifies network configuration**, allowing devices to receive their entire setup in one message.
- **Standard protocol** for dynamically configuring hosts in an IP network, extending from the earlier **BOOTP** protocol.

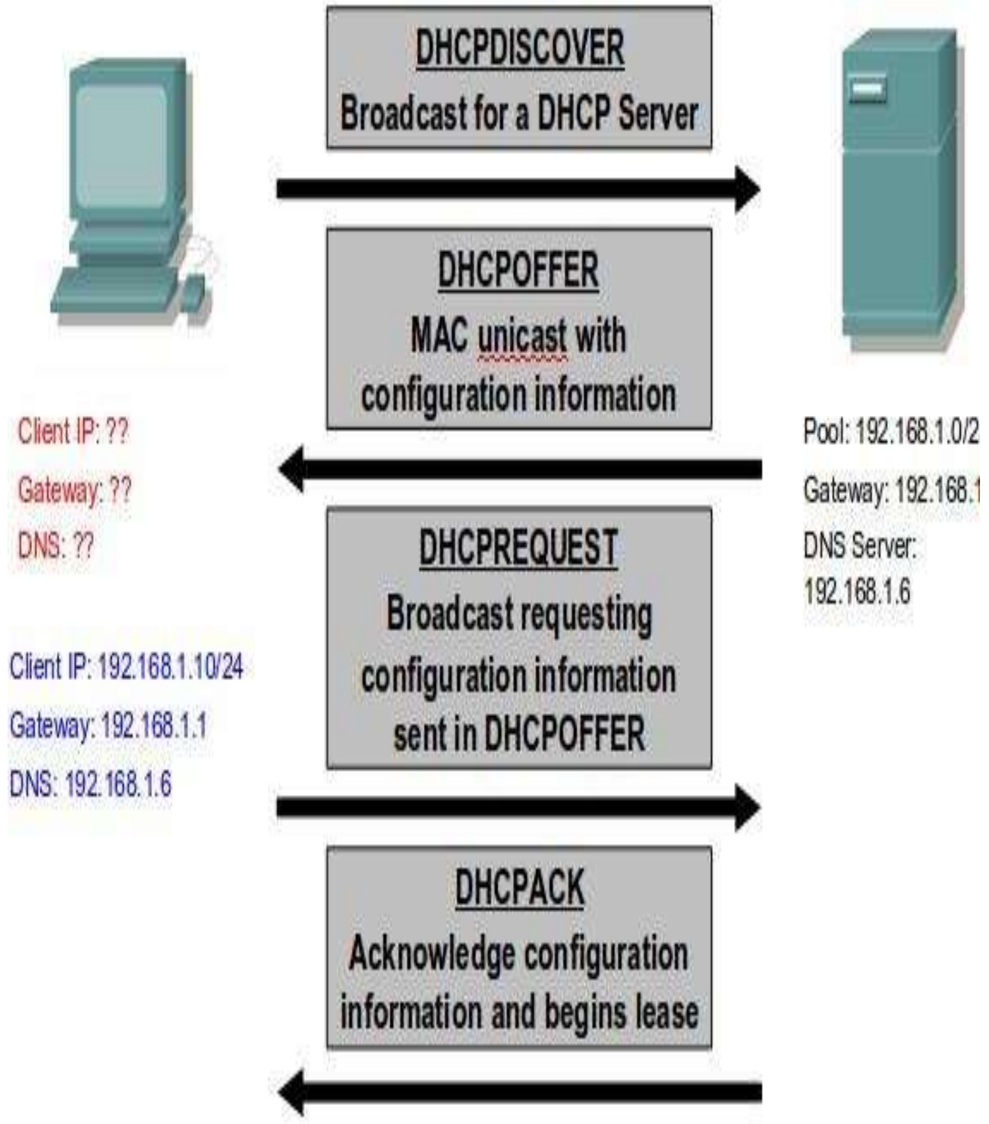
- How it Works?

- DHCP is a client-server protocol that uses DHCP servers and DHCP clients. A DHCP server is a machine that runs a service that can lease out IP addresses and other TCP/IP information to any client that requests them. For example, on Linux System example Ubuntu you can install the DHCP Server service to perform this function. The DHCP server typically has a pool of IP addresses that it is allowed to distribute to clients, and these clients lease an IP address from the pool for a specific period of time, usually several days. Once the lease is ready to expire, the client contacts the server to arrange for renewal.

- DHCP clients are client machines that run special DHCP client software enabling them to communicate with DHCP servers. All versions of Linux and Windows include DHCP client software, which is installed when the TCP/IP protocol stack is installed on the machine.

- **DHCP Lease Process (4 Steps)**

1. **DHCPDISCOVER**:
   The client broadcasts a request looking for a DHCP server.
2. **DHCPOFFER**:
   DHCP servers on the network offer an IP address to the client.
3. **DHCPREQUEST**:
   The client selects one DHCP server's offer and requests to lease the IP address.
4. **DHCPACK**:
   The selected DHCP server acknowledges the request, assigns the IP address, and sends any configured options. The client finalizes its configuration and can start network communication.


DHCPDISCOVER
Broadcast for a DHCP Server
DHCPOFFER
MAC unicast with configuration information
DHCPREQUEST
Broadcast requesting configuration information sent in DHCPOFFER
DHCPACK
Acknowledge configuration information and begins lease
Client IP: ??
Gateway: ??
DNS: ??
Client IP: 192.168.1.10/24
Gateway: 192.168.1.1
DNS: 192.168.1.6
Pool: 192.168.1.0/2
Gateway: 192.168.1
DNS Server: 192.168.1.6

- **Address Resolution**
- To send data, the sender must translate the application layer address (e.g., a domain name) into a network layer address (IP address) and then into a data link layer address (MAC address).
- **TCP/IP** uses two methods for address resolution:
  - **Server Name Resolution**: Translating a domain name like www.facebook.com into an IP address (e.g., 157.240.15.35) using **DNS**.
  - **IP to Data Link Layer Resolution**: Resolving an IP address into a MAC address.

- **Server Name Resolution with DNS**
- **Domain Name System (DNS)** is used to map human-readable domain names to numeric IP addresses.
- Process:
  - User enters a domain name (e.g., example.com) in their browser.
  - The browser sends a **DNS query** to a DNS server.
  - The DNS server responds with the corresponding IP address.
  - The browser connects to the web server using the IP address.
- In cases like file transfers, this process ensures that the file transfer client first resolves the server's name to an IP address, and then connects to the file server for the transfer.

- **DNS Query Types**

1. **Recursive Query**:
    1. The DNS client requests the DNS server to resolve the domain name completely. The server must respond with either the requested IP address or an error (e.g., "domain not found").
2. **Iterative Query**:
    1. The DNS client allows the DNS server to return the best answer it can. If the server doesn't know the IP address, it returns a referral to other DNS servers that might have the answer.
3. **Non-Recursive Query:**

- In a **non-recursive query**, the DNS client asks a DNS server for the IP address, and the server either knows the answer (because it has it cached) or returns the best possible result without contacting other DNS servers.
- The server **does not need to perform additional queries** to other servers. It simply provides the answer if it has it.

- Data Link Layer Address Resolution
- **Address Resolution Protocol (ARP):**
  ARP is used to map a dynamic Internet Protocol (IP) address to a fixed **Media Access Control (MAC) address** within a Local Area Network (LAN). This mapping is essential because IP and MAC addresses differ in format and function.
- **IP Address (Network Layer):**
  Identifies devices on a network and is responsible for routing data between different networks through routers.
- **MAC Address (Data Link Layer):**
  A unique physical address assigned to each device, allowing communication over a local network. It is fixed and does not change, unlike IP addresses

- **How ARP Works:**
  - When a device joins a LAN, it is assigned an IP address for communication.
  - When a data packet arrives at the gateway for a specific device, the gateway uses ARP to find the corresponding MAC address for that IP.
- There are three basic ARP terms:
- Reverse ARP: Reverse Address Resolution Protocol is used in local area networks LAN) by client machines for requesting IP Address from Router's ARP Table.
- Proxy ARP: Proxy Address Resolution Protocol work to enable devices that are separated into network segments connected through the router in the same IP to resolve IP Address to MAC Address.
- Inverse ARP: Inverse Address Resolution Protocol uses MAC Address to find the IP Address.

- **Routing in Networks**
- **What is Routing?**
  Routing is the process of determining the **path** or route a message takes from the sending computer to the receiving computer across a network. This ensures data reaches its correct destination.
- **Types of Networks:**
  - **Multiple Routes:** In large networks like the **Internet**, multiple possible routes exist between computers.
  - **Single Route:** In smaller or internal company networks, there may only be one logical route.
- **Routers:**
  - **Role:** Routers are specialized devices responsible for **routing messages** through the network.
  - **Placement:** Typically found at the **edge of subnets**, routers connect subnets and facilitate communication across different parts of a network.

- **Types of Routing**
- **1. Centralized Routing:**
  - All routing decisions are made by a **central computer or router**.
  - Commonly used in **host-based networks** where routing is straightforward.
  - Messages are sent to the central computer, which forwards them to the correct destination.
- **2. Static Routing:**
  - Routes are **manually added** to the routing table by the network administrator.
  - No automatic changes occur unless modified **manually**.
  - Best suited for networks with **predictable traffic**.
  - Simple to design and implement, without the need for complex routing protocols.

- Advantages of Static Routing
- No routing overhead for router CPU which means a cheaper router can be used to do routing.
- It adds security because only administrator can allow routing to particular networks only.
- No bandwidth usage between routers.
- Disadvantage of Static Routing
- For a large network, it is a hectic task for administrator to manually add each route for the network in the routing table on each router.
- The administrator should have good knowledge of the topology. If a new administrator comes, then he has to manually add each route so he should have very good knowledge of the routes of the topology.

- **Dynamic Routing**
- **What is Dynamic Routing?**
  - Dynamic routing automatically **adjusts routes** based on the current state of the network.
  - It uses protocols like **RIP (Routing Information Protocol)** and **OSPF (Open Shortest Path First)** to discover network destinations and the best paths to reach them.
  - If a route goes down, the router **automatically finds an alternative path** to the destination.
- **Key Features of Dynamic Routing:**
  - All routers must run the **same dynamic protocol** to exchange routing information.
  - When a router detects a change in network topology, it **broadcasts the change** to other routers, ensuring all routers are updated.

- Advantages of Dynamic Routing
- Easy to configure.
- More effective at selecting the best route to a destination remote network and also for discovering remote network.
- Disadvantage of Dynamic Routing

Consumes more bandwidth for communicating with other neighbors.

- Less secure than static routing

- **Distance Vector Routing**
- **What is Distance Vector Routing?**
  - A dynamic routing algorithm where each router calculates the **distance** to all possible destinations based on its **immediate neighbors**.
  - Routers exchange information about the network with their neighbors and **update routing tables** accordingly.
  - Information sharing occurs at **regular intervals**.
- **Algorithm & Protocols:**
  - Utilizes the **Bellman-Ford Algorithm** to create routing tables.
  - Common protocols: **RIP (Routing Information Protocol)** and **IGRP (Interior Gateway Routing Protocol)**, which use hop counts or other metrics for routing decisions.
- **Problems in Distance Vector Routing:**
  - **Count-to-Infinity Problem:** Can result in incorrect routing updates, solved using techniques like **Split Horizon**.
  - **Persistent Looping Problem:** Routing loops can last indefinitely, causing inefficiency.

- The Bellman Ford algorithm operates by having each router maintain a table (vector). The routing table contains two parts. They are:

a) The preferred outgoing line to use for that destination.

b) Estimate of the time or distance to that destination.

- **Procedure of Distance Vector Routing**

1. Initial Routing Table Creation:

   1. Each router creates a list of networks it can reach, noting the **number of hops** to each.
   2. Initially, this includes only directly connected networks, with a **hop count of 1**.

2. Periodic Sharing:

   1. Every 30 seconds (typically), routers share their **routing tables** with neighboring routers using a specific protocol.
   2. Neighbors add 1 to each hop count received, as it costs **1 hop to reach** the sender.

3. Updating the Routing Table:

- Routers construct a new routing table using the **directly connected networks** and the new information from neighbors.
- Each path has an associated hop count and the **next router** to use for that path.

- **Removing Bad Routes:**
- Routes with **higher hop counts** or bad paths are removed from the table.
- If multiple paths to the same network exist, only the one with the **smallest hop count** is kept.
- **Propagating the Updated Table:**
- The updated table is communicated to neighboring routers.
- Over time, all routers will know the **optimal routing paths** to each network.

- **Link State Routing Algorithm**

1.Discover Neighbors:

1. Upon booting, the router sends a **hello packet** on each link. Neighbors respond, sharing their network addresses.

2.Measure Delay/Cost to Neighbors:

1. The router sends an **Echo packet**, and the round-trip time (RTT) is measured and divided by 2 to estimate the delay.
2. This can be repeated for better accuracy.

3.Build Link State Packets (LSPs):

1. The LSP contains:
    1. **Sender identity**
    2. **Sequence number**
    3. **Age**
    4. A list of neighbors and their associated costs.

4.Distribute Packets:

1. LSPs are distributed via **flooding**, using a **32-bit sequence number** to avoid duplication.
2. An **age field** ensures that packets don’t live indefinitely, being discarded when the age hits zero.

5.Compute Shortest Path:

1. Routers use **Dijkstra’s algorithm** to compute the shortest path to every other node.
2. After receiving all LSPs, routers construct a **topological map** of the network and determine the best route to each destination.

- **Dijkstra’s Algorithm**
- **Purpose:**
- Finds the **shortest paths** from a given **source node** sss to all other nodes in a network.
- **Steps of Dijkstra’s Algorithm:**

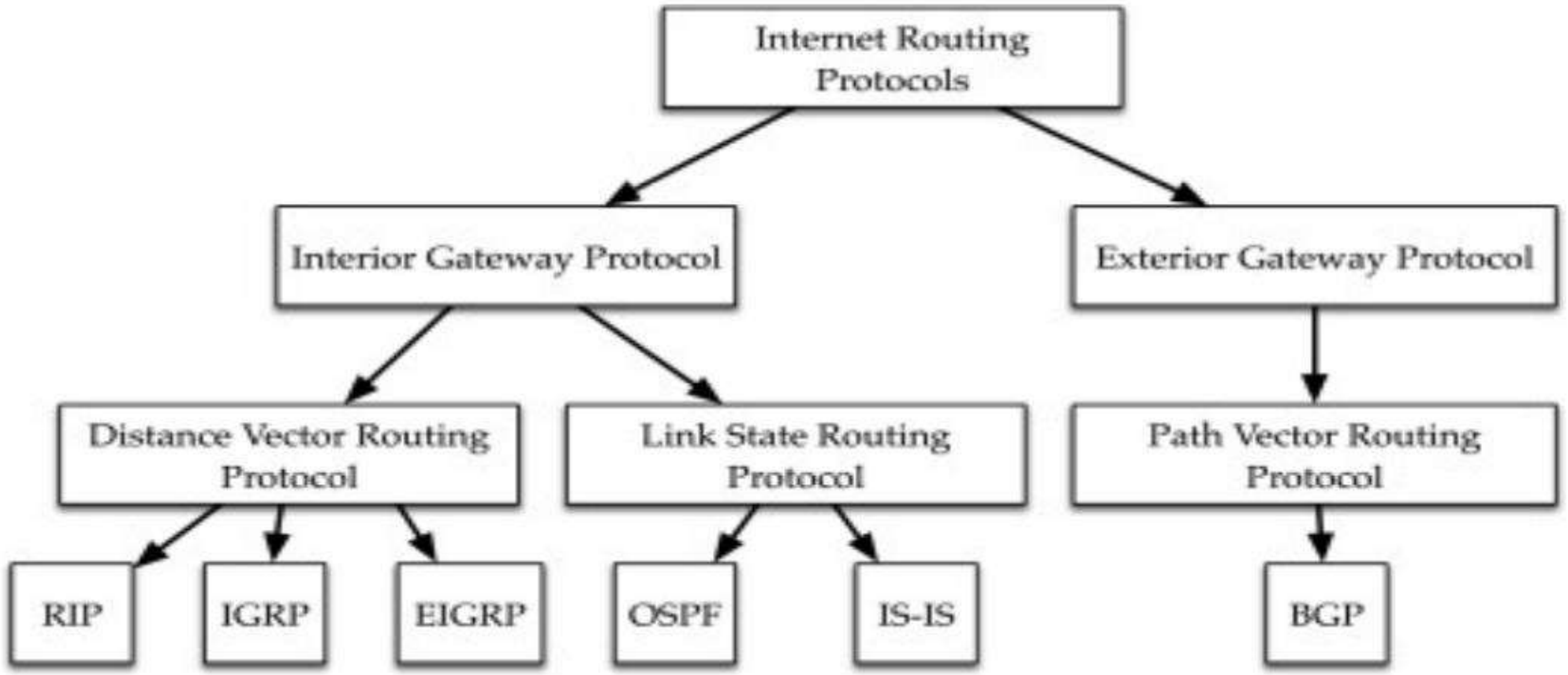
1. **Initialization:**
    1. **N** = Set of all nodes in the network.
    2. **s** = Source node.
    3. **T** = {s} (Set of nodes incorporated so far, initially containing only the source node).
    4. **w(i, j)** = Link cost between nodes iii and jjj.
        1. **w(i, i)** = 0 (Cost to itself).
        2. **w(i, j)** = ∞ if nodes iii and jjj are not directly connected.
        3. **w(i, j)** ≥ 0 if nodes iii and jjj are directly connected.
    5. **L(n)** = Cost of the least-cost path from source node sss to node nnn currently known. At the end, this will be the cost of the shortest path.
        1. **L(n) = w(s, n)** for nodes nnn directly connected to sss.

- **Step 2: Get Next Node**
- **Find the Neighboring Node:**
  - Identify the node not yet in **T** (the set of incorporated nodes) that has the **least-cost path** from the source node sss.
- **Incorporate the Node:**
  - Add this node to **T**.
  - Also include the edge connecting this node to a node in **T** that contributes to the least-cost path.
- **Step 3: Update Least-Cost Paths**
- **Repeat Steps 2 and 3:**
  - Continue finding and incorporating the next node with the least-cost path and updating the least-cost paths until **T = N** (all nodes are incorporated into **T**).
- **Termination:**
  - The algorithm terminates when all nodes have been added to **T**.
  - The shortest path from the source node to each node is finalized and **remains unchanged** thereafter.

Comparison

Feature	Distance Vector (DV)	Link State (LS)
Information Sharing	With direct neighbors only	With all routers in the network
Frequency of Updates	Periodic (e.g., every 30 seconds)	Triggered by topology changes
Algorithm Used	Bellman-Ford	Dijkstra's
Convergence Speed	Slower, may experience routing loops	Faster, minimal chance of routing loops
Network View	Routers have limited knowledge (neighbors only)	Routers have full network topology knowledge
Network Traffic	Higher due to regular updates	Lower due to event-driven updates

Different types of routing Protocols


Internet Routing Protocols
Interior Gateway Protocol
Exterior Gateway Protocol
Distance Vector Routing Protocol
Link State Routing Protocol
Path Vector Routing Protocol
RIP
IGRP
EIGRP
OSPF
IS-IS
BGP

- Five are commonly used on the Internet: Border Gateway Protocol (BGP), Internet Control Message Protocol (ICMP), Routing Information Protocol (RIP), Intermediate System to Intermediate System (IS-IS) Open Shortest Path First (OSPF), and Enhanced Interior Gateway Routing Protocol (EIGRP).

- **Interior Gateway Protocols (IGPs)**
- **Overview**:
  Interior Gateway Protocols are commonly used in smaller, cooperative networks like those found on university campuses or within a single organization.
- **Routing Information Protocol (RIP)**:
  One of the earliest IGPs, RIP uses a simple distance-vector algorithm to route data between networks. While easy to configure, it is not well-suited for larger, more complex networks.
- **Newer Protocols**:
  - **Interior Gateway Routing Protocol (IGRP)**:
    Developed by Cisco, IGRP offers better scalability and performance compared to RIP.
  - **Open Shortest Path First (OSPF)**:
    OSPF is a widely used link-state routing protocol that efficiently manages routing within larger networks by calculating the shortest path first.
- **Cisco’s Enhanced IGRP (EIGRP)**:
  A proprietary protocol by Cisco, EIGRP builds on IGRP with enhanced features like rapid convergence and better support for large networks.
- **Limitations of IGPs**:
  While these protocols are simple to set up, they typically do not scale well to large or highly dynamic networks.

- **Exterior Gateway Protocols (EGPs)**
- **Function**:
  EGPs are used to exchange routing information between different autonomous systems (AS), such as different Internet Service Providers (ISPs) or large corporate networks.
- **Use on the Internet**:
  EGPs, like the **Border Gateway Protocol (BGP)**, are commonly used on the Internet to manage routing between separate networks, ensuring data is directed through the most efficient path.
- **Routing Table**:
  These protocols maintain routing tables that list known routers, reachable network addresses, and associated cost metrics, helping routers select the best route for traffic.

- **Border Gateway Protocol (BGP)**
- **Overview**:
  BGP is an Internet Engineering Task Force (IETF) standard and is widely regarded as the most scalable of all routing protocols.
- **Functionality**:
  BGP manages how packets are routed across the internet by exchanging routing and reachability information between edge routers of different autonomous systems (AS). Each AS is a network managed by a single enterprise or service provider.
- **Key Role in the Internet**:
  BGP is critical to the global Internet, directing traffic between different autonomous systems, making it the backbone protocol for routing on the Internet. It is also commonly used in large Service Provider private networks.
- **Expanding Capabilities**:
  While BGP's original purpose was to carry Internet reachability information, it now supports additional functions like:
  - **Multicast routing**
  - **IPv6 support**
  - **Virtual Private Networks (VPNs)**
- **Types of BGP**:
  - **Internal BGP (iBGP)**:
    Routes traffic within a single autonomous system.
  - **External BGP (eBGP)**:
    Connects and routes traffic between different autonomous systems.

Feature	RIP	IGRP	EIGRP	OSPF	IS-IS
Type	Distance Vector	Distance Vector	Hybrid (Advanced Distance Vector)	Link State	Link State
Metric	Hop Count	Composite (Bandwidth, Delay)	Composite (Bandwidth, Delay, Load, Reliability)	Cost (Bandwidth)	Cost (Metric based on interface cost)
Max Hop Count	15 hops (limited network size)	255 hops	224 hops	Unlimited	Unlimited
Algorithm	Bellman-Ford	Bellman-Ford	Diffusing Update Algorithm (DUAL)	Shortest Path First (SPF) using Dijkstra	Shortest Path First (SPF) using Dijkstra
Convergence Time	Slow	Medium	Fast	Fast	Fast
Classful/Classless	RIP v1 is classful, RIP v2 is classless	Classful	Classless	Classless	Classless
Updates	Periodic (every 30	Periodic	Triggered and periodic	Triggered	Triggered

- **Internet Control Message Protocol (ICMP)**
- **Purpose**:
  The Internet Protocol (IP) lacks mechanisms for error reporting, error correction, or host management queries. ICMP was developed to address these limitations and serves as a companion protocol to IP.
- **Functionality**:
  ICMP facilitates communication between routers and hosts by transferring important messages, providing feedback about network issues, and ensuring error reporting.
- **Error Reporting and Diagnosis**:
  ICMP's primary responsibility is to report errors back to the original source. In addition to error reporting, ICMP helps diagnose network issues through specialized query messages.
- **Types of Messages**:
  - **Error-Reporting Messages**:
    These messages notify the source of errors encountered while delivering packets.
  - **Query Messages**:
    Used to diagnose network problems and gather information about the network’s status.

ICMP Message Types

Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo request	Ask a machine if it is alive
Echo reply	Yes, I am alive
Timestamp request	Same as Echo request, but with timestamp
Timestamp reply	Same as Echo reply, but with timestamp

Note: Choke packets are used for congestion and flow control over a network

A **timestamp** is a sequence of characters or encoded information identifying when a certain event occurred, usually giving date and time of day, sometimes accurate to a small fraction of a second.

- **Open Shortest Path First (OSPF)**
- **Overview**:
  OSPF is a **link-state** and **hierarchical** Interior Gateway Protocol (IGP) routing algorithm used for efficient network management.
- **Enhanced Features over RIP**:
  OSPF improves upon RIP with advanced features such as:
  - **Multipath routing**: Allows multiple paths to the same destination for load balancing.
  - **Least-cost routing**: Uses cost as the primary metric to determine the most efficient path.
- **Type of Service Routing**:
  OSPF supports routing based on different types of service, allowing multiple routes to be prioritized based on the service requirements.
- **Load Balancing**:
  It offers load balancing by distributing traffic across multiple routes, enhancing overall network efficiency.
- **Scalability and Management**:
  OSPF enables the division of networks into smaller subsets and areas, improving scalability and ease of network management.
- **Efficiency**:
  Unlike RIP, OSPF doesn’t rely on broadcasting messages. Instead, it sends status updates selectively to relevant routers or devices, reducing network overhead.
- **Preferred IGP for TCP/IP Networks**:
  Due to its efficiency, OSPF is widely regarded as the preferred interior routing protocol for managing TCP/IP networks.

- **Routing Information Protocol (RIP)**
- **Overview**:
  RIP is a **dynamic distance vector** interior routing protocol, widely used in smaller networks typically operated by a single organization.
- **Routing Process**:
  RIP works by counting the number of routers (hops) between the source and the destination. It selects the route with the fewest hops, simplifying the routing process.
- **Routing Table Updates**:
  Computers using RIP periodically send broadcast messages (at intervals set by the network manager) to announce their routing status, allowing the routing tables to update dynamically.
- **Protocol Support**:
  RIP is used by both the **TCP/IP** and **IPX/SPX(**Internetwork Packet Exchange/Sequenced Packet Exchange**)** protocol suites.

- **Intermediate System to Intermediate System (IS-IS)**
- **Overview**:
  IS-IS is a **link-state** interior routing protocol, primarily used in large networks and enterprise-level infrastructures.
- **Protocol Origin**:
  IS-IS is an ISO-developed protocol that has been adapted to work in **TCP/IP** networks, offering enhanced scalability and efficiency in managing large-scale routing.

- **Enhanced Interior Gateway Routing Protocol (EIGRP)**
- **Overview**:
  EIGRP is a **dynamic hybrid** interior routing protocol developed by Cisco, widely used within organizations. It combines features of both distance-vector and link-state protocols, providing a robust routing solution.
- **Hybrid Nature**:
  EIGRP incorporates characteristics of:
  - **Distance Vector Protocols**: Uses metrics like hop count.
  - **Link-State Protocols**: Maintains more detailed network topology information.
- **Improvement over IGRP**:
  EIGRP is an enhanced version of the **Interior Gateway Routing Protocol (IGRP)**, offering several advancements.
- **Metrics and Performance**:
  EIGRP records and uses multiple metrics to determine the best route:
  - **Transmission Capacity**
  - **Delay**
  - **Reliability**
  - **Load**
- **Routing Tables**:
  Routers running EIGRP maintain:
  - **Their own routing table**
  - **Routing tables for all neighboring routers**
    This dual storage allows EIGRP to have a more accurate and comprehensive understanding of the network topology.

- **Multicasting**
- **Overview**:
  Multicasting is a method for sending a message to multiple computers simultaneously, as opposed to unicasting (one-to-one) or broadcasting (one-to-all).
- **Unicast vs. Broadcast**:
  - **Unicast**: A message is sent from one computer to another specific computer (e.g., a client requesting a web page).
  - **Broadcast**: A message is sent to all computers on a specific LAN or subnet.
- **Multicast Messaging**:
  - **Purpose**: Efficiently sends the same message to multiple computers without overwhelming the network.
  - **Process**:
    - Computers wishing to receive multicast messages join a multicast group using the **Internet Group Management Protocol (IGMP)**.
    - Each multicast group is assigned a unique IP address to identify it.
    - Routers use this IP address to route multicast messages to the appropriate subnet.

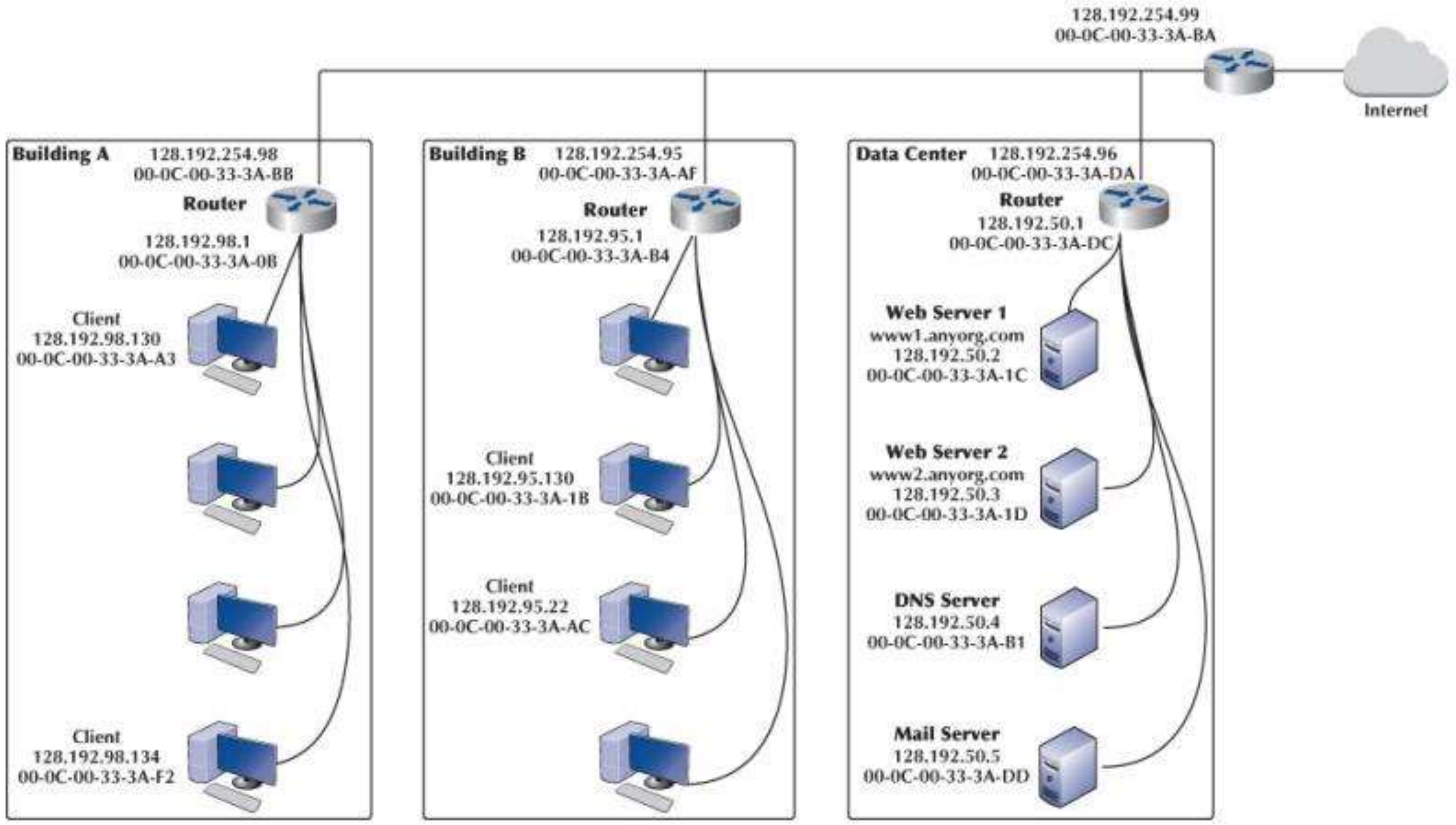
- **Data Link Layer Addressing**:
- Routers set the data link layer address to match the multicast data link layer address.
- Computers inform their data link layer software to process messages with this multicast address.
- **Example - Videoconferencing**:
- **Unicast**: Each participant sends multiple identical messages to every other participant, consuming significant network capacity.
- **Broadcast**: Sends a single message to all devices on the network, which can be inefficient and disruptive.
- **Multicast**: Sends a single message to a specific group of participants, reducing network load and avoiding unnecessary disruption.

- **Session Management**:
  - At the end of the multicast session, such as a videoconference, participating computers send an IGMP message to leave the multicast group, ensuring efficient resource use.

- **Anatomy of a Router**
- **1. Main Functions of a Router:**
- **Path Determination:** Determines the route a packet will take to its destination.
- **Packet Transmission:** Sends the packet across the chosen path.
- **Protocol Support:** Facilitates communication between various devices and protocols.
- **2. Connecting to and Configuring a Router:**
- **Console Port:**
  - Used for initial configuration when the router has no IP address.
  - Requires a blue rollover cable and terminal emulation software.

- **Network Interface Port:**
  - Allows access and configuration once the router has an IP address.
  - Supports TCP/IP, Telnet, and SSH.
- **Auxiliary Port:**
  - Offers a direct, non-network connection (e.g., with modems).
  - Rarely used in modern configurations.
- **3. Router Components:**
- **CPU:** Central Processing Unit.
- **Memory:** Includes volatile (RAM) and non-volatile (e.g., Flash) memory.
- **Ports/Interfaces:** Connects to networks and other devices.
- **4. Operating System:**
- **Cisco Internetwork Operating System (IOS):**
  - Used in approximately 90% of routers.
  - Command Line Interface (CLI) rather than Graphical User Interface (GUI).
  - Configurations are managed via command-line commands and stored in a config file.
- **5. Important Files:**
- **Config File:** Defines router operation, including routing protocols, interface statuses, and encryption methods.
- **Access Control List (ACL):** Manages network security by specifying which packets are allowed or denied.

- **CP/IP Example Overview**
- **1. **Initial Configuration for a Computer on a TCP/IP Network:**
- **IP Address:** Identifies the computer on the network.
- **Subnet Mask:** Determines the range of addresses within the local subnet.
- **DNS Server IP Address:** Translates domain names to IP addresses.
- **IP Gateway (Router) IP Address:** Routes traffic to destinations outside the local subnet.


128.192.254.99
00-0C-00-33-3A-BA
Internet
Building A
128.192.254.98
00-0C-00-33-3A-BB
Router
128.192.98.1
00-0C-00-33-3A-0B
Client
128.192.98.130
00-0C-00-33-3A-A3
Client
128.192.98.134
00-0C-00-33-3A-F2
Building B
128.192.254.95
00-0C-00-33-3A-AF
Router
128.192.95.1
00-0C-00-33-3A-B4
Client
128.192.95.130
00-0C-00-33-3A-1B
Client
128.192.95.22
00-0C-00-33-3A-AC
Data Center
128.192.254.96
00-0C-00-33-3A-DA
Router
128.192.50.1
00-0C-00-33-3A-DC
Web Server 1
www1.anyorg.com
128.192.50.2
00-0C-00-33-3A-1C
Web Server 2
www2.anyorg.com
128.192.50.3
00-0C-00-33-3A-1D
DNS Server
128.192.50.4
00-0C-00-33-3A-B1
Mail Server
128.192.50.5
00-0C-00-33-3A-DD

- 2. **Network Configuration Scenario:**
- **LANs and Subnets:**
    - **Building A:** Subnet 128.192.98.x
    - **Building B:** Subnet 128.192.95.x
    - **Data Center:** Subnet 128.192.50.x
    - **BN (Backbone Network):** Subnet 128.192.254.x
- **Connectivity:**
    - Each building connects to the BN via a router.
    - Routers have two IP addresses and two data link layer addresses: one for the building connection and one for the BN connection.
- **Servers:**
    - Web servers, DNS server, and Mail server are located in the data center.
- **Data Link Layer:**
    - Ethernet is used for the data link layer.
- **Application Layer Focus:**
    - Only Web requests are processed at the application layer.

- 3. **TCP/IP Communication Process:**
- **Creation of TCP Segments and IP Packets:**
  - Created by the sending computer and remain unchanged until reaching the final destination.
- **IP Packet:**
  - Contains the original source and ultimate destination IP addresses.
- **Data Link Layer Frame:**
  - Created by the sending computer for each message.
  - Contains the source and destination data link layer addresses (e.g., Ethernet addresses).
- **Frame Replacement:**
  - At each computer or router along the route, the data link layer frame is removed and replaced with a new frame.
  - Data link layer addresses change at each step, but IP addresses remain constant.

- **Key Takeaways:**
- The **IP addresses** are constant for the entire journey of the packet, while **data link layer addresses** change at each hop.
- **Configuration** involves setting up IP addresses, subnet masks, DNS, and gateway information.
- **Routing** is handled by routers which manage traffic between subnets and the Internet.

- **Handling Unknown Addresses in Network Communication**
- **1. **Client Requests Web Page:**
- **Client Computer (Building A):** IP Address 128.192.98.130
- **Desired Web Server:** www1.anyorg.com (IP address unknown)
- **2. **DNS Request:**
- **DNS Resolution:** The client's web browser needs to resolve the domain name to an IP address.
- **DNS Request Issued:** The web browser sends a DNS request to the DNS server (128.192.50.4) to get the IP address of www1.anyorg.com.
- **Transport Layer (TCP/UDP):** The DNS request is encapsulated in a UDP datagram and handed to the network layer (IP).

- 3. **Routing the DNS Request:**
- **Network Layer (IP):**
  - **Subnet Check:** The DNS server (128.192.50.4) is outside the client’s subnet (128.192.98.x).
  - **Packet Preparation:** The network layer creates an IP packet with the destination address of the DNS server.
  - **Data Link Layer Address:** Sets the data link layer address to the router’s address (128.192.98.1).
- **Transmission to Router (Building A):**
  - **Router Processing:** The router determines the packet needs to go to the data center router (128.192.254.96) using its MAC address (00-0-00-33-3A-DA).
  - **Data Center Router:** Receives the packet and forwards it to the DNS server using its MAC address (00-0C-00-CC-3A-B1).

- 4. **DNS Server Response:**
- **Processing DNS Request:** The DNS server resolves the domain name and sends the IP address (e.g., 128.192.50.2) back to the client.
- **Response Journey:** The response travels through the network, passing through the data center router, the router in Building A, and back to the client.
- **5. **Client Sends HTTP Request:**
- **HTTP Request:** With the IP address resolved, the client's web browser sends an HTTP request to the Web server (128.192.50.2).
- **Network Layer Routing:** The network layer recognizes that 128.192.50.2 is outside its subnet and sends the packet to the default gateway (128.192.98.1).
- **Data Link Layer Address:** The packet is encapsulated in an Ethernet frame with the router's MAC address (00-0C-00-33-3A-0B) and transmitted

- *6. **ARP Process:**
- **ARP Request:**
  - **Router’s Ethernet Address Unknown:** If the client does not know the router’s Ethernet address, it broadcasts an ARP request on the local subnet.
  - **ARP Request Content:** Requests the computer with IP address 128.192.98.1 (router) to reply with its MAC address.
- **ARP Response:**
  - **Router Response:** The router responds with its Ethernet address (e.g., 00-0C-00-33-3A-1B).
  - **ARP Cache Update:** The client updates its ARP cache with this address and sends the HTTP request.
- **7. **ARP at Intermediate Routers:**
- **Data Center Router:**
  - **ARP Request for DNS Server:** If the data center router does not know the DNS server’s MAC address, it issues an ARP request for 128.192.50.4.
  - **ARP Response:** The DNS server responds with its MAC address (00-0C-00-CC-3A-B1).
  - **Packet Forwarding:** The router uses this address to forward packets to the DNS server.

- TCP/IP and Network Layers
- **Step-by-Step Message Flow:**
- **1. Client's Application Layer (HTTP Packet Creation)**
- The process starts at the **application layer** on the sending computer (the client, 128.192.98.130, in Building A). Here, the **Web browser** generates an **HTTP packet** containing the request for the desired webpage.
- **2. Transport Layer (TCP Segment Creation)**
- The **transport layer** (TCP) wraps the **HTTP packet** within a **TCP segment**. This segment ensures reliable communication by handling sequencing and acknowledgment.
- **3. Network Layer (IP Packet Creation)**
- The **network layer** adds an **IP packet** around the TCP segment. The IP packet contains the IP address of the final destination (**128.192.50.2**, the Web server).
- **4. Data Link Layer (Ethernet Frame Creation)**
- The **data link layer** adds an **Ethernet frame** around the IP packet. This frame contains the **MAC address** of the next device (in this case, the router in Building A, **00-0C-00-33-3A-0B**).
- **5. Physical Layer (Electrical Signals Transmission)**
- Finally, the **physical layer** converts the Ethernet frame into **electrical impulses** and transmits the message over the cable to the **router** in Building A, which serves as the network gateway.

- **At the First Router (Building A Gateway):**
- **Physical Layer:** The router receives the electrical signals and converts them into digital data.
- **Data Link Layer:** The router verifies that the **Ethernet frame** is addressed to itself, performs **error detection**, strips off the Ethernet frame, and passes the **IP packet** to the network layer.
- **Network Layer:** The router checks the **destination IP address** (128.192.50.2), determines the next stop, and creates a new Ethernet frame with the MAC address of the **next device** (the router in the Data Center, **00-0C-00-33-3A-DA**).
- **Data Link & Physical Layers:** The new frame is passed down through the data link and physical layers, where it is transmitted as electrical signals through the network to the Data Center's router.
- **At the Data Center Router:**
- **Physical & Data Link Layers:** Similar to the Building A router, the Data Center router processes the incoming message by stripping off the Ethernet frame, performing error detection, and passing the IP packet to the network layer.
- **Network Layer:** The router recognizes that the destination IP address (128.192.50.2) is within its network and prepares the packet for the final leg of the journey.
- **Data Link Layer:** The router creates a new Ethernet frame with the **MAC address** of the **Web server** (**00-0C-00-33-3A-DC**) and sends the packet

- **At the Web Server (Final Destination):**
- **Physical & Data Link Layers:** The server receives the message, strips off the Ethernet frame, and passes the **IP packet** to the network layer.
- **Network Layer:** The server recognizes its own **IP address** (128.192.50.2) as the final destination and passes the packet to the **transport layer**.
- **Transport Layer:** The **TCP segment** is removed, and the **HTTP packet** is passed to the **application layer**, where the Web server processes the request.