NETWORK LAYER

INTERNET TECHNOLOGY

ISA- 4

Name: Olencia Fernandes
Sr No.: 29
SEM: 1

INDEX

The Network Layer

In the OSI (Open Systems Interconnection) and TCP/IP models, the Network Layer is essential because it facilitates data transport between devices over vast, interconnected networks like the Internet. In order to ensure that data goes from one network to another effectively and securely, the Network Layer—which sits between the Data Link Layer and the Transport Layer—is responsible for packet forwarding, logical addressing, and routing.

The Network Layer manages the organizational needs and technical difficulties of massive data flows through techniques including packet switching, logical IP addressing, and routing protocols. In order to make data routing and delivery scalable, safe, and effective, this layer combines IP protocols, subnetting, Network Address Translation (NAT), and routing algorithms. These functions serve as the foundation for business and service provider networks in addition to supporting regular Internet operations.

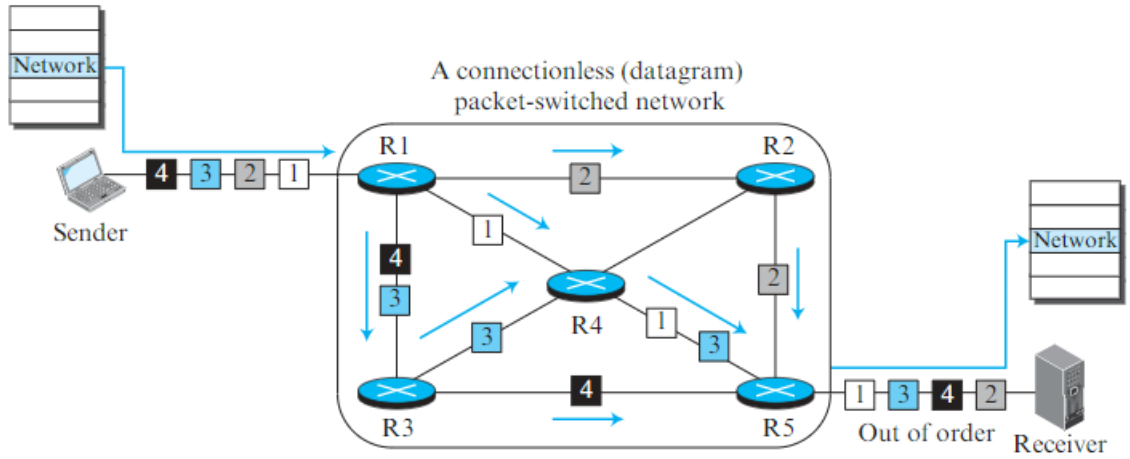
The main features of the network layer, such as packet switching, IP addressing, routing, and protocols like DHCP and ICMP, will be covered in this guide. These subjects show how data flows smoothly across various network contexts and offer fundamental knowledge for anybody working in digital networking, network design, or network management.

Packet Switching

Packet switching is a method used at the Network Layer to transfer data as packets, which are small chunks of a larger message. Each packet is routed independently, making packet switching essential for efficient, flexible data communication across complex networks. Unlike the circuit switching used in the Physical Layer, packet switching is the primary switching method used at the Network Layer. This section will explore two key approaches in packet-switched networks: the Datagram (connectionless) approach and the Virtual Circuit (connection-oriented) approach.

Datagram Approach: Connectionless Service

The Datagram approach offers a **connectionless service** where each packet is treated as an independent entity with no relationship to other packets. This approach is commonly used in the Internet due to its simplicity and flexibility.

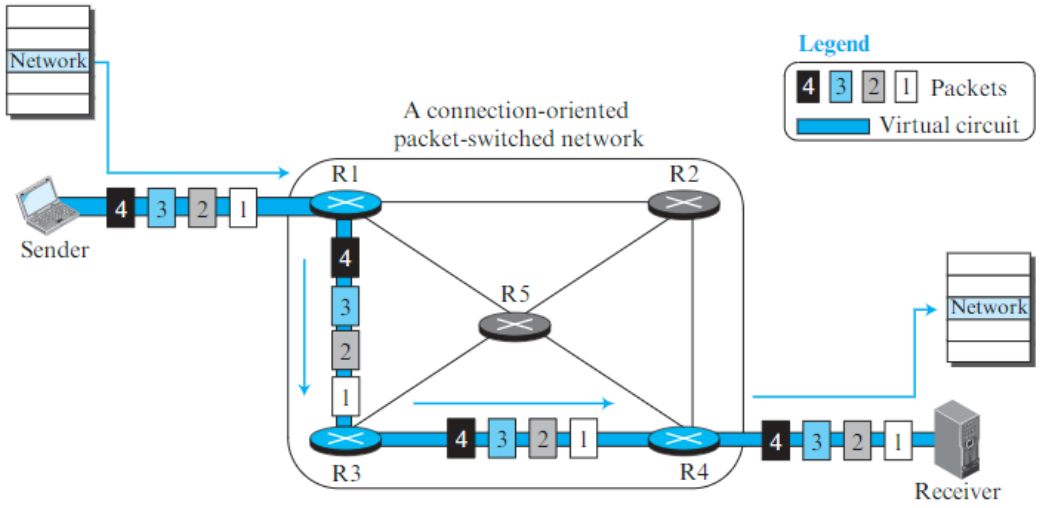


1. **Independent Packet Transmission**: In this method, packets from the same message do not need to travel the same path. Each packet may reach its destination independently based on the routing information available at each router, as shown in the above Figure.
2. **Routing Based on Destination Address**: Each packet contains a source and destination address, with the routing decision made based only on the destination address. This approach allows packets from different sources or even different messages to intermingle, giving each packet a unique path through the network.
3. **Out-of-Order Delivery**: Since packets can take different paths, they may arrive out of order. The destination device reassembles packets in the correct order before delivering the complete message to the upper layer.

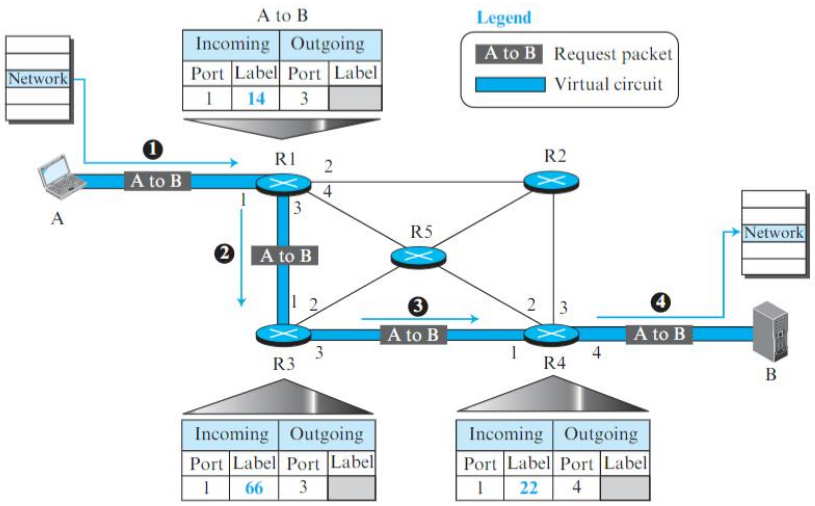
In a datagram-based network, **routers (switches)** forward packets based on their destination addresses. The forwarding table in each router dynamically determines the outgoing interface for each packet, making it well-suited for large, dynamic networks.

Virtual Circuit Approach: Connection-Oriented Service

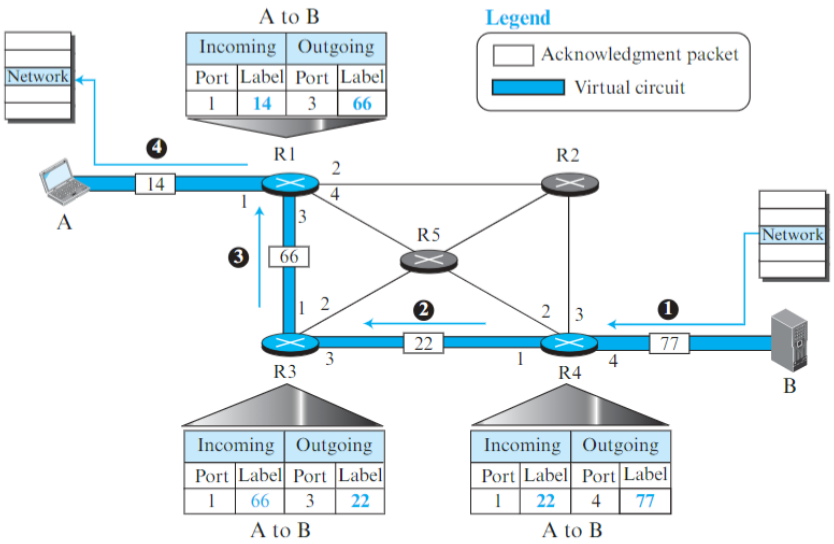
The Virtual Circuit approach offers a **connection-oriented service**, where a dedicated path, known as a virtual circuit, is set up between the source and the destination before data transfer begins.



1. **Three-Phase Process**:
    - **Setup Phase**: A virtual path is established using a unique identifier (label) to designate the route each packet will follow. The setup involves a request from the source to establish the connection and an acknowledgment from the destination.

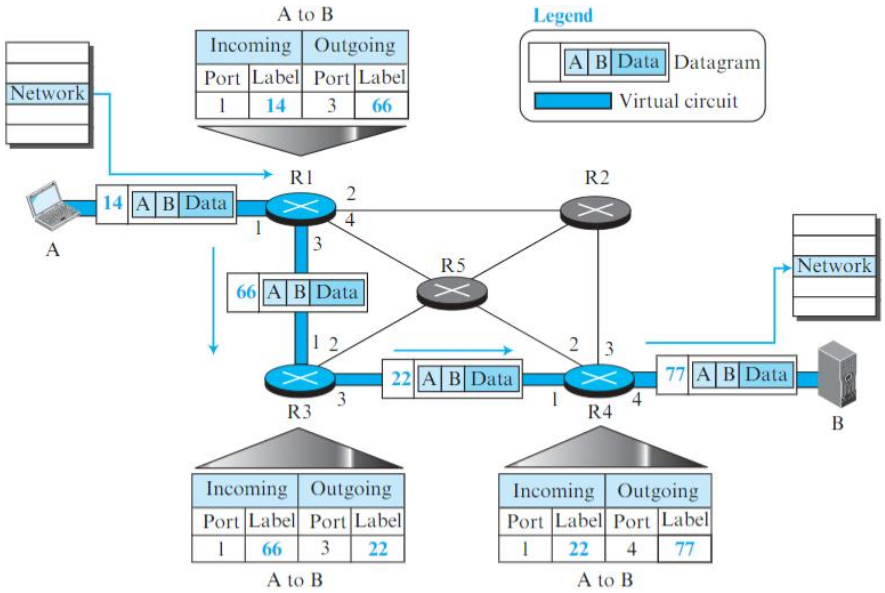


Sending request packet in a virtual-circuit network



Sending acknowledgments in a virtual-circuit network

- **Data Transfer Phase**: Once the path is set, all packets in the message follow the same sequence of routers, maintaining the order of packets, as shown in Figure below.



- **Teardown Phase**: After all packets have been sent, the source initiates a teardown message, which removes the virtual circuit from each router's table.

2. **Consistent Path with Flow Labelling**: Each packet in a virtual circuit contains a **label** (flow identifier) in addition to source and destination addresses. This label is used by routers to maintain the virtual path. The source and destination addresses are often kept as a fallback, especially in parts of the network that may switch to a connectionless service.
3. **In-Order Delivery**: Since packets travel along a predetermined route, they arrive in the same order as they were sent, simplifying the reassembly process at the destination.
4. **Advantages and Applications**: Connection-oriented packet switching provides predictable performance and Quality of Service (QoS), making it ideal for applications needing consistent delivery times. It is commonly used in networks where reliability and in-order packet delivery are essential, such as MPLS (Multiprotocol Label Switching) networks.

The virtual circuit approach ensures that routers use the label to forward each packet along the designated path, allowing for more reliable data transmission across the network.

Forwarding and Routing

Forwarding

Forwarding refers to the process by which a packet is transferred from one input link of a router to an appropriate output link. This is a router-local function, meaning it occurs within each router and does not involve the entire network.

- **Mechanism:** When a packet arrives at a router, the router examines a specific field in the packet's header, usually the destination IP address, and consults its forwarding table. This table contains mappings of network addresses to output links, directing packets based on their destination IP address.
- **Example:** Consider a packet with a header field value of 0111 arriving at a router. The router would use this header field value to look up the corresponding entry in its forwarding table, determining that this packet should be sent to output link interface 2. The router then transfers the packet to that output link for the next stage of its journey.
- **Local Process Analogy:** Forwarding can be compared to navigating a single interchange or intersection on a road trip. At each intersection, a driver chooses the appropriate exit to continue on the route, based on directions given along the way.

Routing

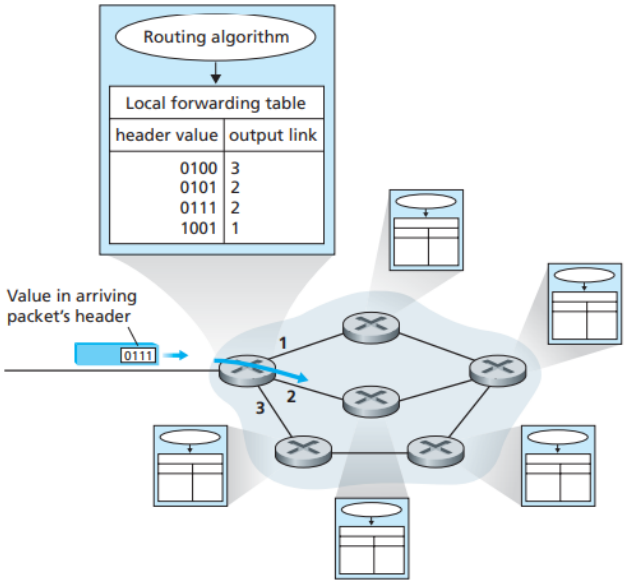
Routing is the network-wide process that determines the paths or routes packets take as they travel from the source to the destination host. Unlike forwarding, which operates at each individual router, routing involves planning the entire end-to-end path across multiple routers.

- **Routing Algorithms:** To select efficient routes, routers use routing algorithms, which can be either centralized or decentralised:
    - Centralized routing algorithms operate from a single location, calculating paths and disseminating routing information to all routers.
    - Decentralized routing algorithms distribute the routing calculation across multiple routers, each running part of the algorithm and sharing path information with neighbouring routers.
- **Dynamic Path Selection:** Routing algorithms dynamically determine the best paths based on factors like network topology, traffic load, and link quality. Routing decisions populate each router's forwarding table, dictating the path each packet will take.
- **Global Planning Analogy**: Routing is akin to planning a complete road trip from one location to another. Before departing, a driver selects an optimal route from

Pennsylvania to Florida, choosing a series of connected road segments and interchanges for the entire journey.

3. Forwarding Tables and Routing Interplay

- The forwarding table within each router directs packets based on information generated by routing algorithms. The routing algorithm, whether centralized or decentralized, fills each forwarding table with data on the optimal links to use for different destination addresses.
- In a hypothetical scenario where each forwarding table is manually configured, routing protocols would not be necessary. However, such a network would be inflexible and slow to respond to changes in topology, making automated routing protocols essential for modern networks.



Routing algorithms determine values in forwarding tables

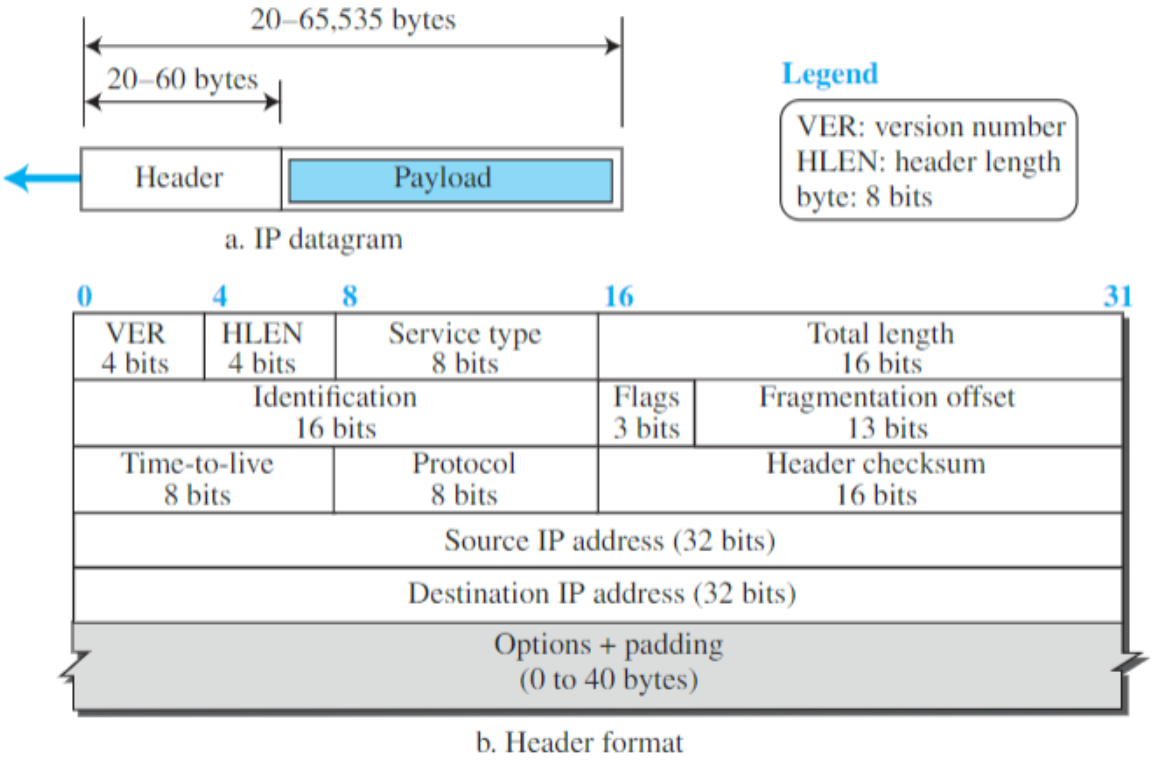
4. Connection Setup

- In some networks, an additional network-layer function, connection setup, is required before data packets begin flowing. This process is similar to TCP's three-way handshake, where routers along a chosen path exchange information to establish a connection state before any packet transmission. Connection setup ensures routers are prepared to handle packets for a specific connection in a controlled and coordinated manner.

The Internet Protocol (IP)

IPv4 Datagram Format

An **IP datagram** in the IPv4 protocol is a structured packet used for data transmission across network layers. Each datagram consists of a header and payload, containing important fields for addressing, routing, and handling the data. Here's an overview based on the detailed structure from your reference:



IP Datagram

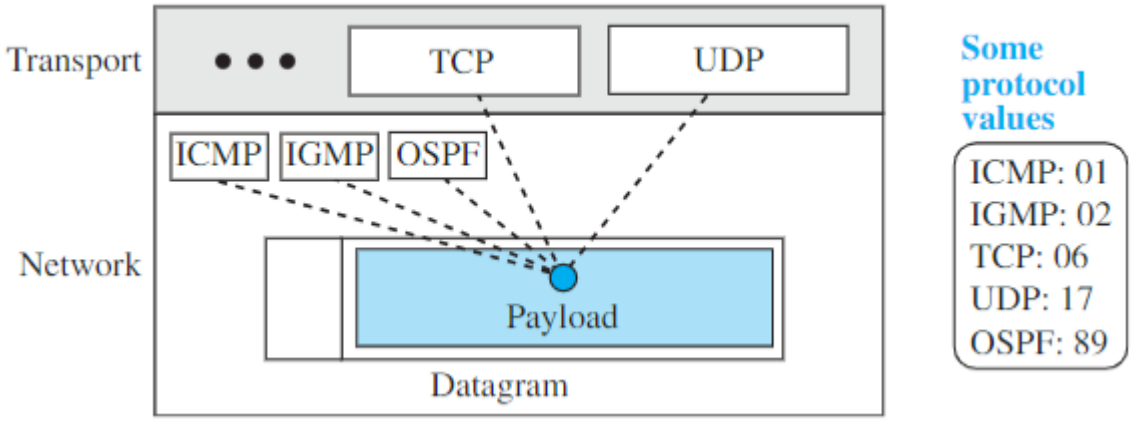
1. IP Datagram Structure

- **Header**: Between 20 and 60 bytes, consisting of fields necessary for network-layer processing.
- **Payload**: The data or content being transmitted, such as a TCP or UDP packet.

2. Fields in the IP Header

- **Version (4 bits)**: Specifies the IP version (IPv4 or IPv6).
- **Header Length (4 bits)**: Length of the IP header in 4-byte words. Multiplying this value by 4 gives the header's total byte length.
- **Service Type (8 bits)**: Initially the Type of Service (ToS), now redefined as Differentiated Services (DiffServ) for QoS.

- **Total Length (16 bits)**: Total length of the datagram (header + data) in bytes, up to a maximum of 65,535 bytes.
- **Identification, Flags, and Fragmentation Offset**: Used for fragmentation and reassembly:
    - **Identification (16 bits)**: Unique ID for the datagram to facilitate reassembly.
    - **Flags (3 bits)**: Controls fragmentation; D bit for "do not fragment" and M bit for "more fragments."
    - **Fragmentation Offset (13 bits)**: Offset of the fragment's data relative to the original datagram, measured in 8-byte units.
- **Time-to-Live (TTL, 8 bits)**: Limits the datagram's lifetime in the network. Each router decrements TTL by one, and if TTL reaches zero, the datagram is discarded.
- **Protocol (8 bits)**: Indicates the higher-layer protocol (e.g., TCP = 6, UDP = 17) to direct the payload at the destination.

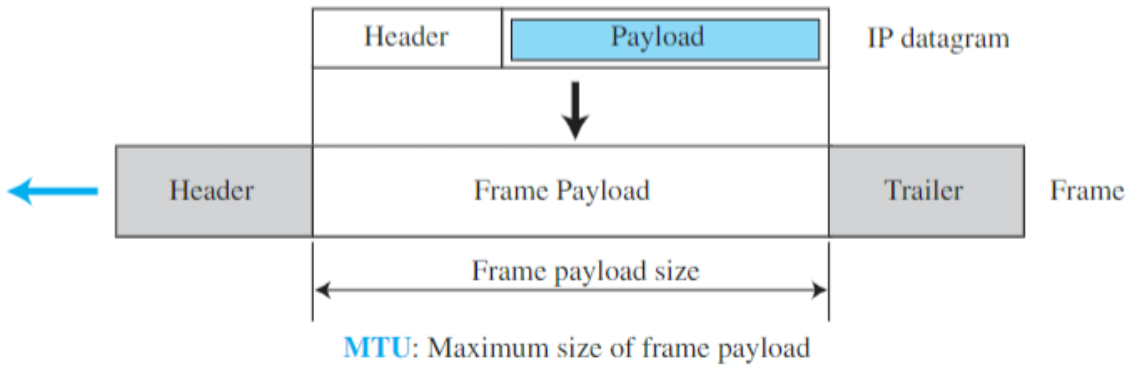


Multiplexing and demultiplexing using the value of the protocol field

- **Header Checksum (16 bits)**: Used for error-checking the header; recalculated at each router due to header modifications.
- **Source and Destination IP Addresses (32 bits each)**: Specifies the sender and receiver's IP addresses, remaining unchanged throughout transit.
- **Options (up to 40 bytes)**: Optional and used for network testing or diagnostics, increasing processing overhead.

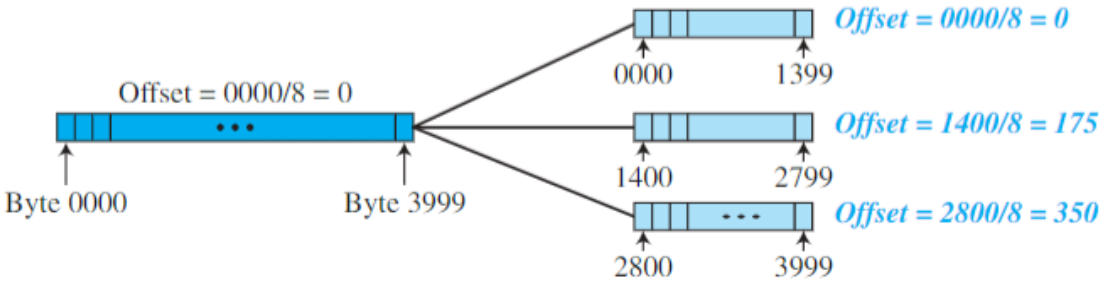
3. Fragmentation

- When a datagram exceeds the Maximum Transfer Unit (MTU) of a network segment, it is split into fragments, each with its own header. Fragmentation occurs at the sender or intermediate routers, and reassembly happens only at the destination

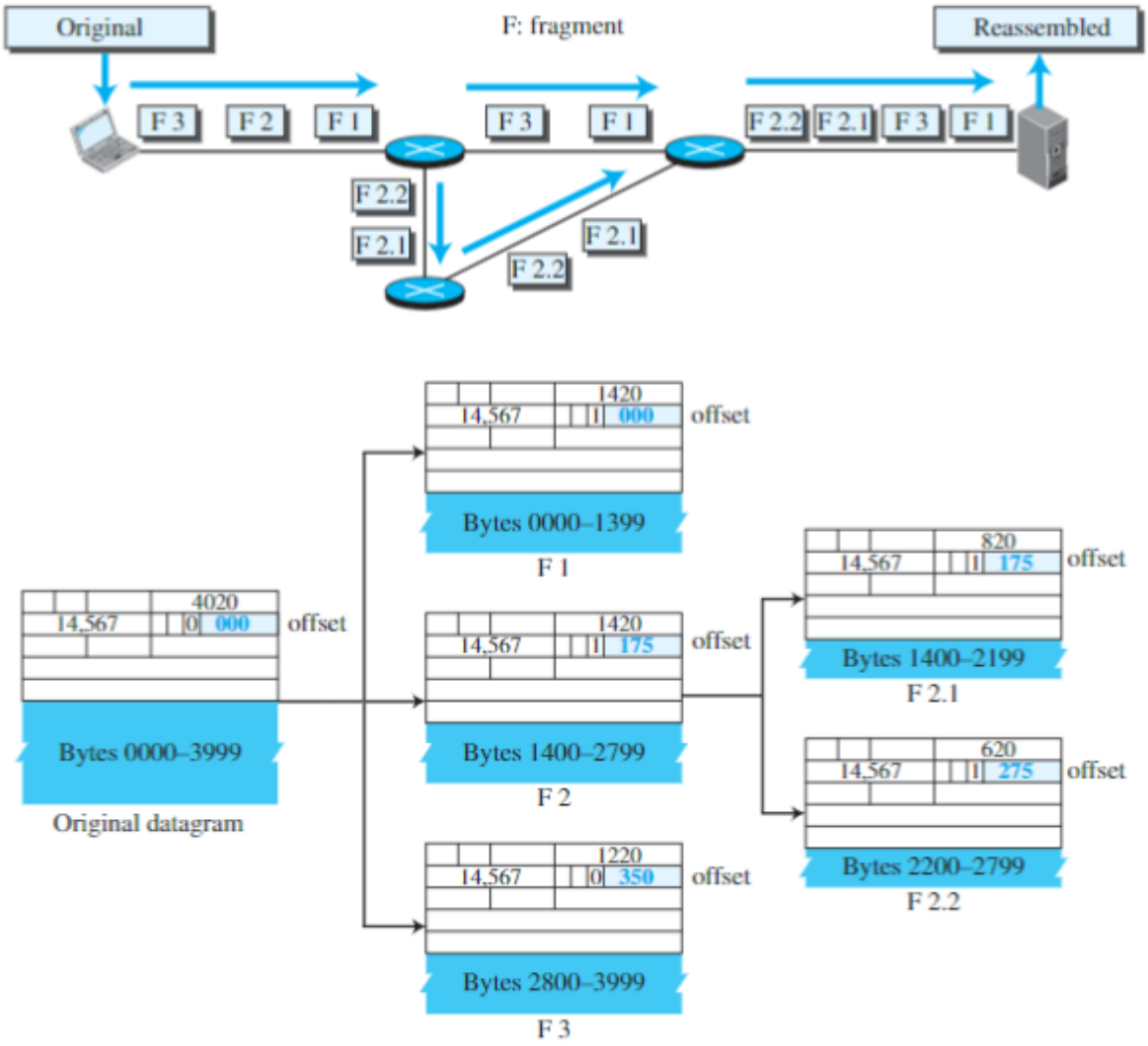


Maximum transfer unit (MTU)

- Fields like **Identification**, **Flags**, and **Fragmentation Offset** guide reassembly:
    - Fragments share the same **Identification** value.
    - The **M bit** in Flags marks whether a fragment is the last one (0) or if more follow (1).
    - **Fragmentation Offset** gives the position of each fragment in the original datagram.



Fragmentation example



Detailed Example

4. Security Issues and IPSec

- **Security Challenges**: IP is inherently insecure with vulnerabilities to packet sniffing, modification, and spoofing.
- **IPSec**: Addresses security needs by adding encryption, integrity checks, and authentication for secure packet transmission.

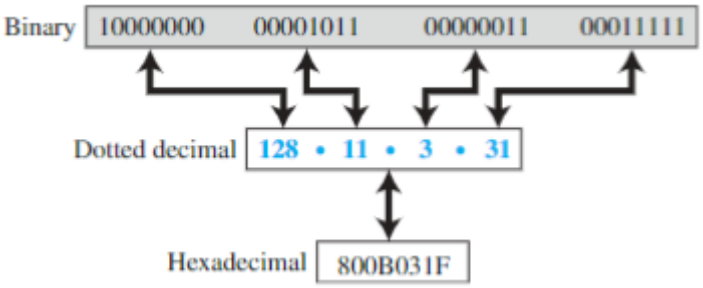
IP datagrams form the backbone of internet communications by ensuring data packets are correctly routed and reassembled across various networks, even with MTU restrictions, making them robust yet flexible for large-scale network operations.

IPv4 Addressing

IPv4 addressing forms the basis of internet communication by assigning unique, hierarchical addresses to each device or network connection. Here's a comprehensive look at IPv4 addresses and how they are organized, referenced, and managed.

IPv4 Address Basics

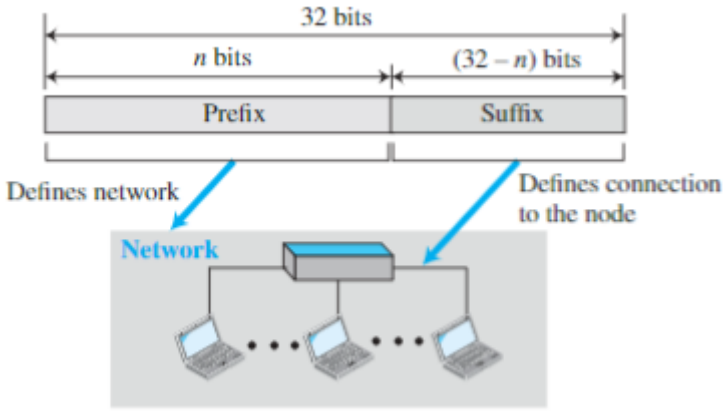
1. **Definition and Universality**:
    - An IPv4 address is a 32-bit number uniquely identifying a device's connection to the Internet. Since addresses are associated with the connection, rather than the device itself, moving a device to a different network typically requires a new IP address.
    - Each IPv4 address is unique across the Internet, ensuring every device with an internet connection can be reached precisely.
2. **Address Space**:
    - IPv4 uses a 32-bit address space, resulting in a theoretical total of 2322^{32}232 (4,294,967,296) possible addresses.
    - This space can support over 4 billion unique addresses, though practical limitations reduce the usable address pool due to reserved addresses.
3. **Notations**:



Three different notations in IPv4 addressing

- **Binary Notation**: An IPv4 address is represented as four groups of 8 bits (octets), often separated by spaces for readability.
- **Dotted-Decimal Notation**: To improve readability, the address is divided into four octets separated by dots (e.g., 192.168.1.1). Each octet is represented as a decimal number from 0 to 255.
- **Hexadecimal Notation**: In some programming contexts, IPv4 addresses are represented in hexadecimal notation, which uses 8 hexadecimal digits (e.g., C0A80101 for 192.168.1.1).

Hierarchical Structure in IPv4 Addressing



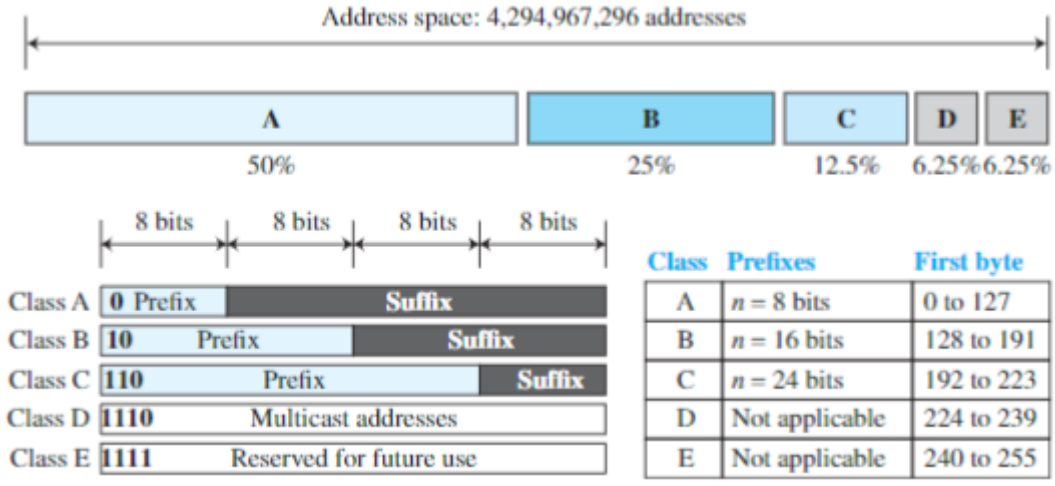
Hierarchy in addressing

IPv4 addresses have a hierarchical structure, divided into a **prefix** (network portion) and a **suffix** (host portion):

- The **prefix** specifies the network, while the **suffix** uniquely identifies a specific connection within that network.

Classful Addressing (Legacy)

Originally, IPv4 addresses were divided into fixed-length classes (Class A, B, C, D, and E) based on the prefix length, to suit varying network sizes:



Class	Prefixes	First byte
A	$n = 8$ bits	0 to 127
B	$n = 16$ bits	128 to 191
C	$n = 24$ bits	192 to 223
D	Not applicable	224 to 239
E	Not applicable	240 to 255

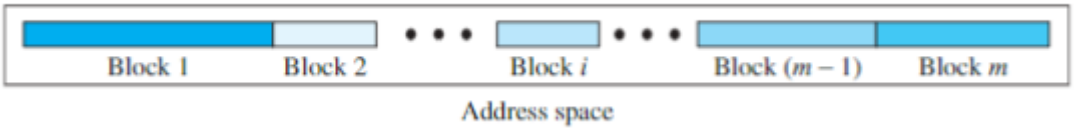
Occupation of the address space in classful addressing

- **Class A**: 8-bit prefix, designed for large networks (up to ~16 million hosts).
- **Class B**: 16-bit prefix, for medium-sized networks.
- **Class C**: 24-bit prefix, suitable for smaller networks.
- **Class D**: For multicast addressing, with no fixed prefix/suffix split.
- **Class E**: Reserved for experimental purposes.

This fixed-length prefix scheme simplified routing and address interpretation, but it led to inefficient address use, especially in Classes A and B, where many addresses remained unused in assigned blocks, resulting in significant **address depletion**.

Classless Addressing and CIDR

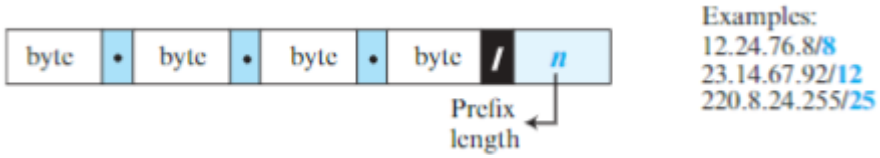
Classless Inter-Domain Routing (CIDR) introduced **variable-length prefixes** to overcome limitations of classful addressing, allowing for flexible allocation based on network size requirements.



Variable-length blocks in classless addressing

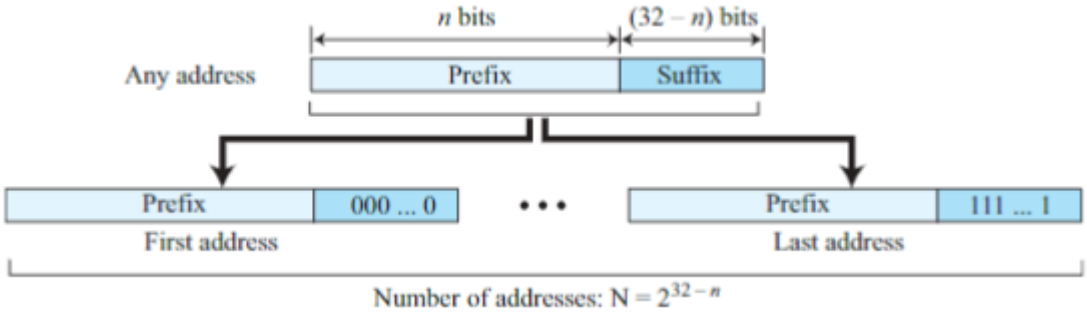
Here's how CIDR works:

1. **Prefix Length in Slash Notation**:
    - An IPv4 address in CIDR includes a suffix, or prefix length, specified after a slash. For example, 192.168.1.0/24 indicates the first 24 bits are the network prefix, leaving the remaining 8 bits for host addresses.



Slash notation (CIDR)

2. **Address Block Sizes**:
    - Blocks can vary in size (e.g., 2, 4, 8, 16 addresses), allowing efficient allocation for networks of different sizes. The number of addresses in a block is calculated as 2^(32−n), where n is the prefix length.



Information extraction in classless addressing

3. **Calculating First and Last Addresses in a Block**:
    - **First Address**: Retain the prefix bits and set remaining bits to 0.
    - **Last Address**: Retain the prefix bits and set remaining bits to 1.

Route Aggregation

Route Aggregation, also known as **prefix aggregation** or **supernetting**, is a method used in IP networking to reduce the number of routing table entries by consolidating multiple IP address prefixes into a single, summarized route. This technique improves network efficiency, reduces router memory and CPU usage, and minimizes the amount of information exchanged between routers. It is particularly useful in large networks or backbone networks like the internet, where it helps manage scalability by reducing the size and complexity of routing tables.

Key Concepts in Route Aggregation

1. **Summarization of IP Prefixes**:
    - Route aggregation allows multiple IP subnets with contiguous address ranges to be represented by a single summary address. For example, instead of advertising four separate /24 subnets (like 192.168.1.0/24 to 192.168.4.0/24), they could be aggregated into a single /22 route (192.168.0.0/22).
    - This consolidated route covers all subnets in the range, thus reducing the number of individual routes in the routing table.
2. **CIDR (Classless Inter-Domain Routing)**:
    - CIDR notation (e.g., 192.168.0.0/22) is widely used in route aggregation to express aggregated addresses without being constrained by classful IP address boundaries. CIDR allows routers to combine addresses based on the common prefix, enabling more flexible and efficient aggregation.
    - For example, in CIDR, the IP range 192.168.0.0 to 192.168.3.255 can be represented as 192.168.0.0/22, covering 1024 addresses within a single route entry.
3. **Hierarchical Network Design**:
    - Route aggregation works best in hierarchical network designs, where IP addresses are assigned logically in a way that reflects the network topology. For instance, an organization might allocate IP ranges based on geographic location or departmental divisions, which can be aggregated at a regional or departmental router.
    - Internet Service Providers (ISPs) also use hierarchical design to assign IP addresses to customers in contiguous blocks, allowing ISPs to aggregate and advertise fewer routes to other ISPs.
4. **Reduces Routing Table Size and Updates**:

- By consolidating routes, routers store fewer entries, which saves memory and reduces processing load.
- Fewer routes need to be shared or updated across networks, which reduces the bandwidth used by routing protocols (like BGP, OSPF, or EIGRP) and enhances convergence times in large networks.

Subnetting

- **Subnetting**: Divides a larger IP block (e.g., a Class A or B address) into smaller sub-networks. This is useful for managing IPs within an organization.

Subnetting is a process used in IP networking to divide a larger IP network into smaller, more manageable subnetworks, or "subnets." This is achieved by modifying the IP address prefix, which creates distinct network segments within the original address space. Subnetting is essential for efficient IP address management, improving network organization, and enhancing security and performance by controlling traffic flows within different segments.

Key Concepts in Subnetting

1. **IP Address and Subnet Mask:**
    - An IP address (IPv4, for example) consists of 32 bits, typically written in four octets separated by dots (e.g., 192.168.1.1).
    - A subnet mask is used to determine which portion of the IP address refers to the network and which part refers to the host. For instance, in the IP 192.168.1.1 with subnet mask 255.255.255.0, the first three octets (192.168.1) identify the network, while the last octet (1) identifies the host within that network.
    - CIDR notation (e.g., 192.168.1.0/24) is often used, where “/24” indicates that the first 24 bits are used for the network portion, leaving the last 8 bits for host addresses.
2. **Subnetting a Network:**
    - To create subnets, we "borrow" bits from the host portion of the IP address to create additional network segments.
    - For example, starting with a network 192.168.1.0/24, you can subnet it into two /25 networks by using one additional bit in the subnet mask, resulting in two subnets: 192.168.1.0/25 and 192.168.1.128/25.

3. **Subnet Size and Host Calculation:**
    - The number of subnets and hosts per subnet depends on the number of bits borrowed for subnetting and the number of bits remaining for hosts.
    - For instance, with a /26 subnet (255.255.255.192), 6 bits are available for the host portion, allowing up to 26−2=622^6 - 2 = 6226−2=62 hosts per subnet (the subtraction of 2 accounts for the network and broadcast addresses).
4. **Benefits of Subnetting:**
    - Efficient IP Address Management: Subnetting allows better allocation of IP addresses by matching network sizes to actual requirements, avoiding wasted IPs.
    - Improved Security and Isolation: Different departments or segments can have separate subnets, limiting access and isolating network traffic.
    - Optimized Network Performance: Subnets reduce broadcast traffic, as broadcast domains are limited to the subnet scope, improving overall network performance.
    - Easier Troubleshooting: Smaller, logically segmented networks are easier to manage, monitor, and troubleshoot.

Special Addressing and Address Conservation

1. **Private IP Addresses**: Defined by RFC 1918, these addresses (e.g., 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16) are used within private networks and are not routable on the public internet.
2. **Network Address Translation (NAT)**: Allows multiple devices in a private network to share a single public IP address, helping to conserve IPv4 addresses.

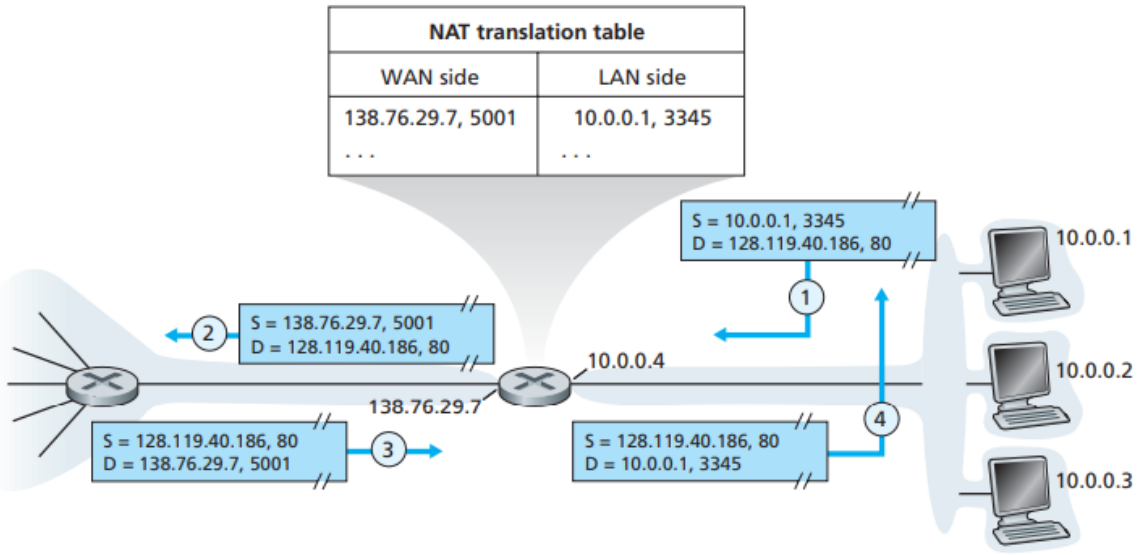
   Network Address Translation (NAT) provides a practical solution for connecting multiple devices in private networks to the global Internet using a single public IP address. This method addresses the shortage of IPv4 addresses and simplifies IP management in small office or home office (SOHO) networks. Here's a breakdown of how NAT works, its benefits, challenges, and NAT traversal solutions:

   ### How NAT Works

   In a NAT setup, a router manages the connection between a private network and the Internet:

   1. **Private IP Addressing**: Devices within a local network use private IP addresses (such as 10.0.0.0/8, per RFC 1918). These private addresses only have meaning within the local network and cannot be routed directly over the Internet.
   2. **Single Public IP**: The NAT-enabled router has a single public IP address (e.g., 138.76.29.7), which it uses for all outgoing traffic to the Internet.
   3. **Address Translation**: When a device in the private network initiates a connection to an external server, the NAT router:
      - Replaces the device's private IP address with the router's public IP address in the packet header.
      - Assigns a unique source port number for the session, creating an entry in the **NAT translation table**. This entry maps the private IP and port of the device to the router's public IP and assigned port.
   4. **Returning Traffic**: When the external server responds, it sends packets to the router's public IP and port number. The NAT router uses the translation table to match the destination port with the internal device, then rewrites the destination address and forwards the packet.

NAT Translation Table Example



Network address translation

Using the above figure, here's how a NAT translation occurs:

- A device with IP 10.0.0.1 and port 3345 sends a request to an external web server (IP 128.119.40.186, port 80).
- The NAT router changes the source to its public IP 138.76.29.7 and assigns a new port, say 5001.
- The translation table entry is:
    - WAN Side: (138.76.29.7, 5001)
    - LAN Side: (10.0.0.1, 3345)
- When a response arrives at 138.76.29.7:5001, the NAT router uses the table to redirect it to 10.0.0.1:3345.

Benefits of NAT

- **Conserves Public IPs**: By using a single public IP, NAT conserves the limited IPv4 address space.
- **Security Through Obfuscation**: NAT conceals the structure of the internal network, providing a basic layer of security.
- **Ease of Network Expansion**: New devices can be added to the private network without requiring additional public IP addresses.

Criticisms and Challenges

1. **Port Numbers as Host Identifiers**: Critics argue that port numbers are meant for identifying processes, not for distinguishing hosts. This can create problems, particularly with servers behind NAT that rely on specific well-known ports.
2. **Layer Violation**: Routers are expected to handle layer 3 (network layer) processing only. NAT, however, involves modifying layer 4 (transport layer) information, such as port numbers, which can interfere with packet integrity.
3. **End-to-End Principle Violation**: NAT disrupts the direct host-to-host communication model, modifying IP addresses and ports along the route.
4. **IPv6 Alternative**: IPv6, with its vast address space, offers a longer-term solution for address scarcity, while NAT is viewed as a temporary workaround.

NAT Traversal and UPnP

One significant drawback of NAT is its interference with peer-to-peer (P2P) applications, which require direct connections between peers (e.g., for file sharing or VoIP). Solutions for NAT traversal include:

- **Connection Reversal**: If one peer is not behind NAT, it can connect to an intermediate peer and request a direct connection, known as **connection reversal**.
- **Universal Plug and Play (UPnP)**: This protocol allows internal devices to request the NAT router to create a public mapping (public IP and port) to their private IP and port. For instance:
    - A device with IP 10.0.0.1 on port 3345 can request a mapping to 138.76.29.7:5001 for a BitTorrent session.
    - The application then advertises 138.76.29.7:5001, enabling external peers to connect directly.

UPnP enables TCP or UDP connections to NATed devices, addressing P2P compatibility issues by dynamically configuring NAT entries.

DHCP

The Dynamic Host Configuration Protocol (DHCP) is an essential networking protocol that automates IP address allocation and management for devices connecting to a network. Instead of manual configuration, DHCP dynamically assigns IP addresses, which is highly practical for environments where devices frequently join and leave networks, such as in offices, homes, and public places with Wi-Fi.

Here's an overview of DHCP's functionality:

- **Automated IP Assignment:** DHCP allows a device (client) to automatically request an IP address when connecting to a network. The DHCP server assigns an available IP from its pool to the client, making networking setup effortless and reducing the need for users to configure IPs manually.
- **Leasing IP Addresses:** Instead of providing a permanent IP, DHCP leases it for a set period, meaning the IP may be recycled and assigned to other devices as needed. This process optimizes the IP usage, especially in situations where not all networked devices are active simultaneously. For example, an ISP might have 2,000 customers but only 400 active at once. DHCP allows the ISP to manage a smaller block of IP addresses and allocate them on demand.
- **Plug-and-Play Capability:** DHCP's plug-and-play functionality makes it especially valuable in scenarios where devices change networks regularly, like laptops or mobile devices in schools, libraries, or offices. With DHCP, a device connecting to a network automatically receives an IP address, the subnet mask, the default gateway, and DNS server information, among other configurations.

Four-Step DHCP Process:

- **Discovery:** When a new device connects, it sends a broadcast message on the network to discover any available DHCP server.
- **Offer:** A DHCP server responds with an IP address offer, including configuration details like the subnet mask, lease time, and gateway address.
- **Request:** The client selects one of the offers (if multiple servers responded) and sends a request to the chosen DHCP server to finalize the IP assignment.
- **Acknowledgment (ACK):** The DHCP server acknowledges the request and assigns the IP to the client for the specified lease duration. Once the client has an IP, it can begin network communication.
- **Renewal Process:** To keep an IP address beyond the initial lease time, the device can request a renewal from the DHCP server. If granted, the device can retain the same IP address without interruption.

Limitations of DHCP

- One significant limitation of DHCP is that IP addresses are local to each network, meaning the IP address changes as the device moves between networks. This prevents the maintenance of persistent connections (e.g., TCP connections) across subnets, a drawback addressed in part by Mobile IP, which we'll explore further.
- In sum, DHCP's ability to allocate IPs dynamically, renew leases, and provide additional configuration details makes it indispensable for modern networks. The protocol minimizes administrative overhead, simplifies network management, and supports flexibility in device connectivity.

ICMP

Internet Control Message Protocol (ICMP): Used for error reporting and operational diagnostics, with tools like ping and traceroute relying on ICMP to test network connectivity.

The **Internet Control Message Protocol (ICMP)** is a core component of the Internet's network layer, primarily used for error reporting and network diagnostics, as specified in **RFC 792**. Operating above IP, ICMP helps hosts and routers communicate issues like unreachable destinations and timeouts. Each ICMP message includes a **Type** and **Code** field that specify the error or request type, and it attaches a header with the original message, allowing senders to identify and address the issue.

Key ICMP Functions:

1. **Error and Status Messages**:
    - **Destination Unreachable** (Type 3) indicates network, host, or port unavailability.
    - **Time Exceeded** (Type 11) signals that a packet's **Time-to-Live (TTL)** expired, useful in route diagnostics.
2. **Ping**:
    - Uses **Echo Request (Type 8)** and **Echo Reply (Type 0)** to test connectivity and round-trip time (RTT), verifying if a host is reachable.
3. **Traceroute**:
    - Sends packets with incrementally higher TTLs, receiving **Time Exceeded** messages from each router along the route. When the final destination is reached, a **Port Unreachable** message signals completion of the route.

ICMP Type	Code	Description
0	0	echo reply (to ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	6	destination network unknown
3	7	destination host unknown
4	0	source quench (congestion control)
8	0	echo request
9	0	router advertisement
10	0	router discovery
11	0	TTL expired
12	0	IP header bad

ICMP message types

ICMP Limitations:

Although critical for diagnostics, ICMP's congestion control (e.g., **Source Quench**) has mostly been replaced by more advanced mechanisms in TCP. Additionally, due to potential misuse in **DoS attacks**, ICMP traffic is often regulated on networks.

ICMP's capabilities in troubleshooting tools like Ping and Traceroute make it invaluable for network management and connectivity checks, despite its security considerations

References:

https://broman.dev/download/Computer%20Networking:%20A%20Top-Down%20Approach%206th%20Edition.pdf
https://studylib.net/doc/25925860/computer-networks-a-top-down-approach