

Domain Name System (DNS)

Last Updated : 15 Feb, 2025



The Domain Name System (DNS) translates human-readable domain names (e.g., `www.google.com`) into machine-readable IP addresses (e.g., `142.250.190.14`), enabling internet communication

- It enables computers to locate and communicate with each other on the internet.
- Functions as a **hierarchical, distributed database**.
- Queries pass through multiple levels:
 - **Root server**
 - **Top-Level Domain (TLD) server**
 - **Authoritative server** (stores the specific IP address).
- Ensures seamless website access using easy-to-remember names instead of numerical IP addresses.

How Does DNS Work?

- When we type a website like <https://www.geeksforgeeks.org> in our browser, our computer tries to find the IP address.
- First, it checks the local cache (our browser, operating system, or router) to see if it already knows the IP address.
- If the local cache doesn't have the IP, the query is sent to a DNS resolver to find it.
- DNS resolver may check host files (used for specific manual mappings), but usually, it moves on.
- Resolver sends the query to a Root DNS server, which doesn't know the exact IP address but points to the TLD server (e.g., .org server for this example).
- TLD server then directs the resolver to the authoritative nameserver for geeksforgeeks.org.
- Authoritative nameserver knows the exact IP address for geeksforgeeks.org and sends it back to the resolver.
- Resolver passes the IP address to our computer.

Types of Domain

There are various kinds of domains:

- **Generic Domains:** .com(commercial), .edu(educational), .mil(military), .org(nonprofit organization), .net(similar to commercial) all these are generic domains.
- **Country Domain:** .in (India) .us .uk
- **Inverse Domain:** if we want to know what is the domain name of the website. IP to domain name mapping. So DNS can provide both the mapping for example to find the IP addresses of [geeksforgeeks.org](https://www.geeksforgeeks.org) then we have to type

Domain Name Server

The client machine sends a request to the local name server, which, if the root does not find the address in its database, sends a request to the root name server, which in turn, will route the query to a top-level domain (TLD) or authoritative name server. The root name server can also contain some **hostName** to IP address mappings. The Top-level domain (TLD) server always knows who the authoritative name server is. So finally the IP address is returned to the local name server which in turn returns the IP address to the host.

DNS Lookup

[DNS Lookup](#), also called DNS Resolution, is the process of translating a human-readable domain name (like `www.example.com`) into its corresponding IP address (like `192.0.2.1`), which computers use to locate and communicate with each other on the internet. It allows users to access websites easily using names instead of remembering numeric IP addresses.

- DNS Lookup starts when a user types a domain name into their browser.
- The query goes through a series of servers: the DNS resolver, Root server, TLD server, and authoritative server.
- Each server plays a role in finding the correct IP address for the domain.
- Once the IP address is found, the browser connects to the website's server and loads the page.

DNS Resolver

DNS Resolver is simply called a DNS Client and has the functionality for initiating the process of DNS Lookup which is also called DNS Resolution. By using the DNS Resolver, applications can easily access different websites and services present on the Internet by using domain names that are very much friendly to the user and that also resolves the problem of remembering IP Address.

What is MAC (Media Access Control) Address?

MAC Addresses are unique **48-bit** hardware numbers of a computer that are embedded into a network card (known as a [Network Interface Card](#)) during manufacturing. The MAC Address is also known as the [Physical Address](#) of a network device. In the IEEE 802 standard, the data link layer is divided into two sublayers:

1. Logical Link Control (LLC) Sublayer
2. Media Access Control (MAC) Sublayer

The MAC address is used by the Media Access Control (MAC) sublayer of the [Data-Link Layer](#). MAC Address is worldwide unique since millions of network devices exist and we need to uniquely identify each.



or



0: Unicast
1: Multicast

0: Globally unique
1: Locally administered

Format of MAC Address

To understand what is MAC address is, it is very important that first you understand the format of the MAC Address. So a MAC Address is a 12-digit hexadecimal number (48-bit binary number), which is mostly represented by Colon-Hexadecimal notation.

The First 6 digits (say 00:40:96) of the MAC Address identify the manufacturer, called the OUI (**Organizational Unique Identifier**). IEEE Registration Authority Committee assigns these MAC prefixes to its registered vendors.

Here are some OUI of well-known manufacturers:

CC:46:D6 - Cisco

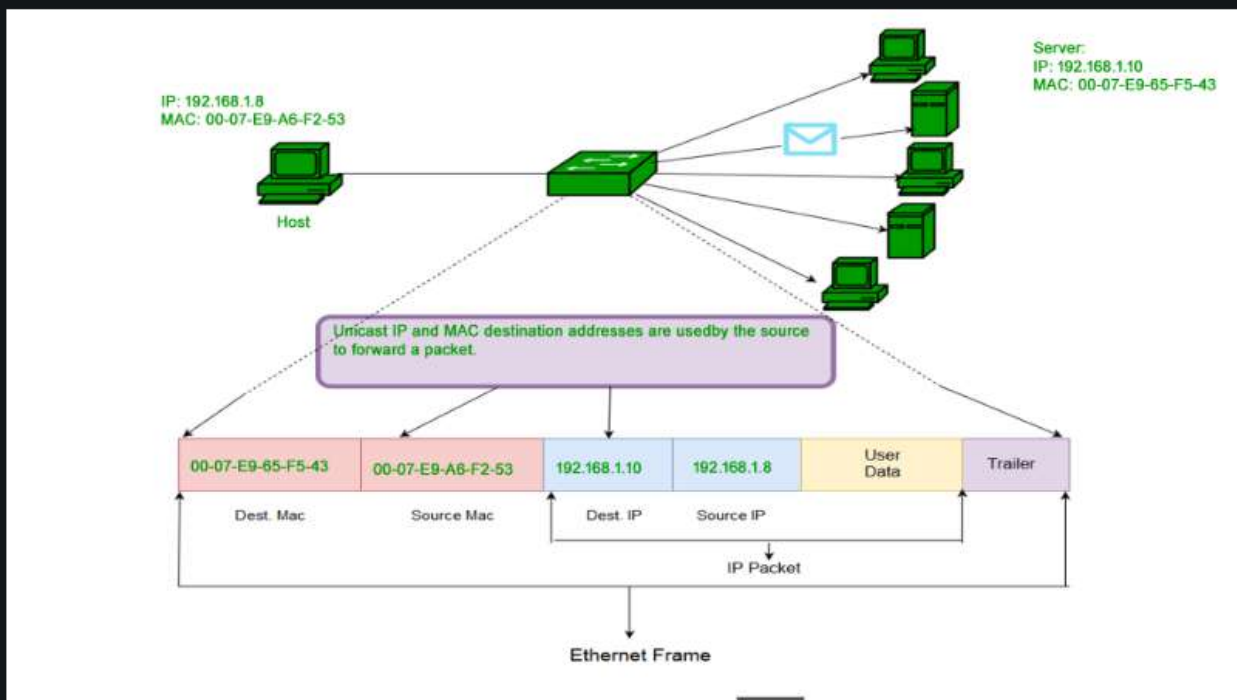
3C:5A:B4 - Google, Inc.

3C:D9:2B - Hewlett Packard

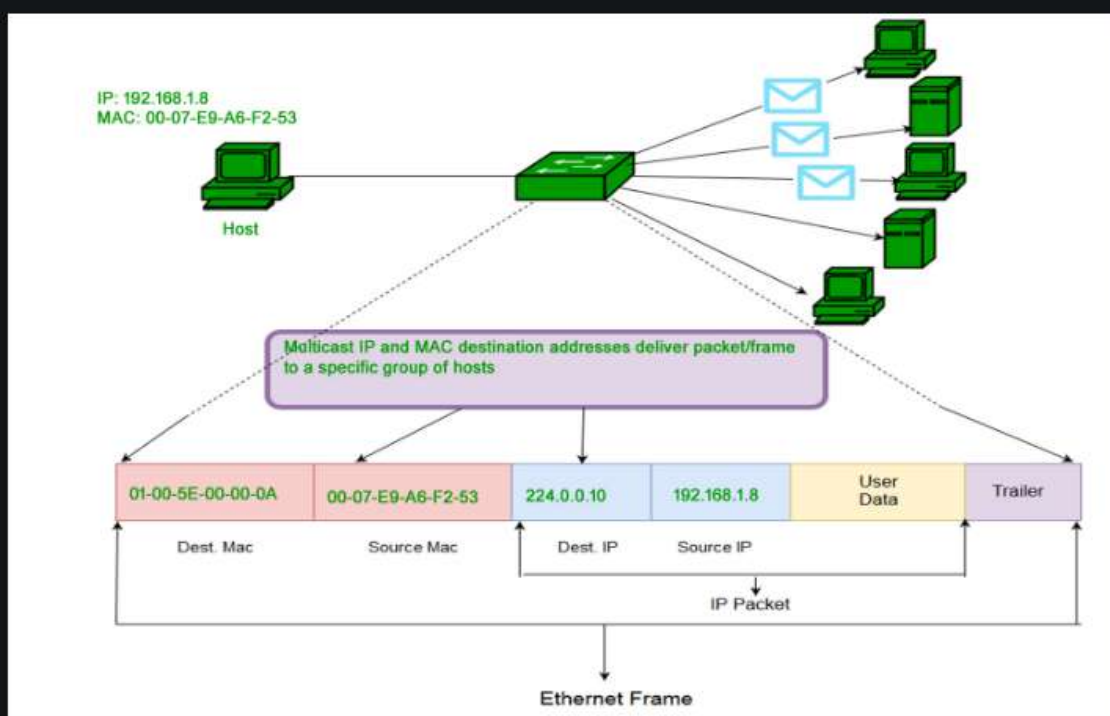
00:9A:CD - HUAWEI TECHNOLOGIES CO.,LTD

Types of MAC Address

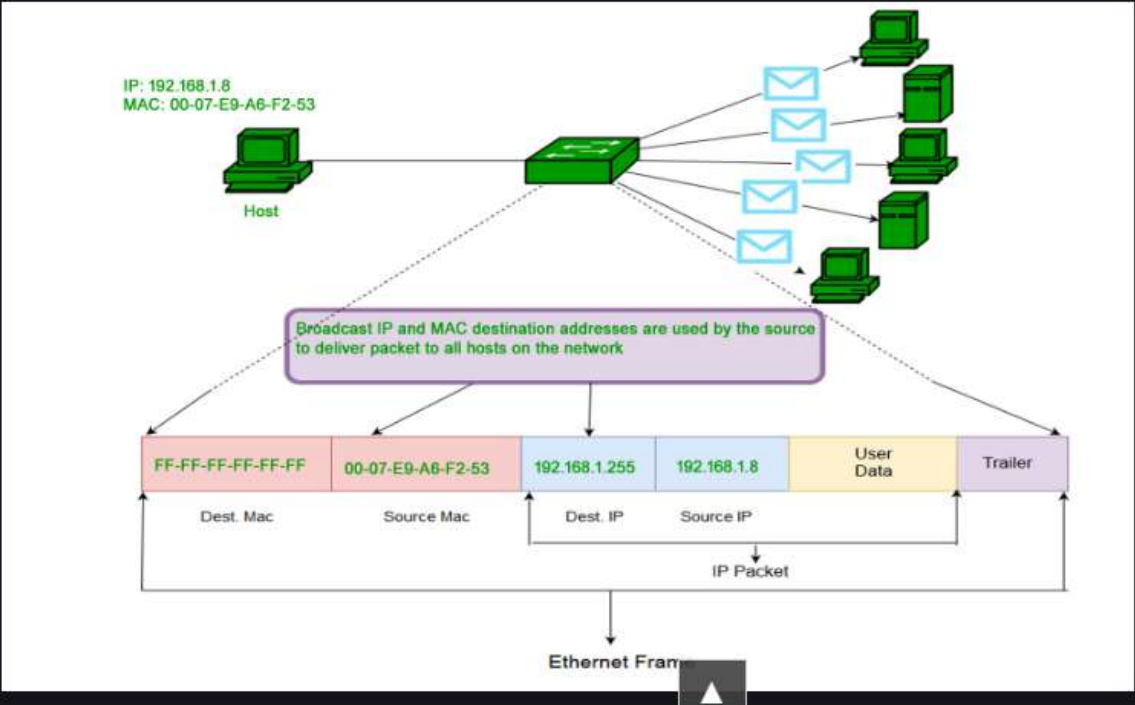
1. Unicast: A Unicast-addressed frame is only sent out to the interface leading to a specific NIC. If the LSB (least significant bit) of the first octet of an address is set to zero, the frame is meant to reach only one receiving NIC. The MAC Address of the source machine is always Unicast.



2. Multicast: The multicast address allows the source to send a frame to a group of devices. In Layer-2 (Ethernet) Multicast address, the LSB (least significant bit) of the first octet of an address is set to one. IEEE has allocated the address block 01-80-C2-xx-xx-xx (01-80-C2-00-00-00 to 01-80-C2-FF-FF-FF) for group addresses for use by standard protocols.



3. Broadcast: Similar to Network Layer, Broadcast is also possible on the underlying layer(Data Link Layer). Ethernet frames with ones in all bits of the destination address (FF-FF-FF-FF-FF-FF) are referred to as the broadcast addresses. Frames that are destined with MAC address FF-FF-FF-FF-FF-FF will reach every computer belonging to that LAN segment.



Reason to Have Both IP and MAC Addresses.

The reason for having both IP and MAC addresses lies in the way the Internet works, specifically in the structure of the OSI Model. This model is a conceptual framework that describes how data is sent and received over a network. It's divided into seven layers, each performing specific functions.


- **Layer 2** uses **MAC addresses** and is responsible for packet delivery from **hop to hop** .
- **Layer 3** uses **IP addresses** and is responsible for packet delivery from **end to end** .

Layer 2 ([Data Link Layer](#)) uses a **MAC (Media Access Control) address**. These are unique identifiers assigned to network interfaces for communications at the data link layer. The primary function of MAC addresses is to manage how data is transported from one network node to another on a direct, physical basis – this is also referred to as “hop to hop” delivery.

On the other hand, Layer 3 ([Network Layer](#)) uses an **IP (Internet Protocol) address**. These IP addresses are used to identify devices on a network and to route traffic between networks. The IP addresses ensure that the data gets from its original source reaches its final destination and it is also called “end-to-end” delivery of data.

When a computer sends data, it first wraps it in an IP header, which includes the source and destination IP addresses. This IP header, along with the data, is then encapsulated in a MAC header, which includes the source and destination MAC addresses for the current “hop” in the path.

As the data travels from one router to the next, the MAC address header is stripped off and a new one is generated for the next hop. However, the IP header, which was generated by the original computer, remains intact until it reaches the final destination. This process illustrates how the IP header manages the “end to end” delivery, while the MAC headers handle the “hop to hop” delivery.



PORT	Service	Description	Transport Protocol
7	Echo	Port just echoes whatever is sent to it. This feature can be used in many attacks, such as Smurf/Fraggle.	TCP and UDP
20 /21	File Transfer Protocol (FTP)	Port used by FTP protocol to send data to the client	TCP
22	Secure Shell (SSH)	Used as secure replacement protocol for Telnet	TCP and UDP
23	Telnet	Port used by Telnet to remotely connect to a workstation or server(unsecured)	TCP

25	Simple Mail Transfer Protocol (SMTP)	Used to send E-Mail over internet	TCP
53	Domain Name System (DNS)	Port for DNS requests, network routing, and zone transfers	TCP and UDP
67 /68	Dynamic Host Configuration Protocol (DHCP)	Used on networks that do not use static IP address assignment.	UDP
80	Hypertext Transfer Protocol (HTTP)	Used for browsing web-pages on a browser	TCP
110	Post Office Protocol (POP3)	Port used to retrieve complete contents of a server mailbox	TCP

TCP/IP Model

Last Updated : 08 May, 2025



The TCP/IP model (Transmission Control Protocol/Internet Protocol) is a four-layer networking framework that enables reliable communication between devices over interconnected networks. It provides a standardized set of protocols for transmitting data across interconnected networks, ensuring efficient and error-free delivery. Each layer has specific functions that help manage different aspects of network communication, making it essential for understanding and working with modern networks.

TCP/IP was designed and developed by the Department of Defense (DoD) in the 1970s and is based on standard protocols. The TCP/IP model is a concise version of the OSI model. It contains four layers, unlike the seven layers in the OSI model.

Role of TCP/IP

TCP/IP enables interoperability between diverse systems over various network types (e.g., copper, fiber, wireless). It ensures seamless communication across LANs, WANs, and the internet. Without TCP/IP, large-scale global networking would not be possible.

The main condition of this process is to make data reliable and accurate so that the receiver will receive the same information which is sent by the sender. To ensure that, each message reaches its final destination accurately, the TCP/IP model divides its data into packets and combines them at the other end, which helps in maintaining the accuracy of the data while transferring from one end to another end.



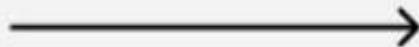
TCP/IP Layers



TCP/IP Layers



Sender

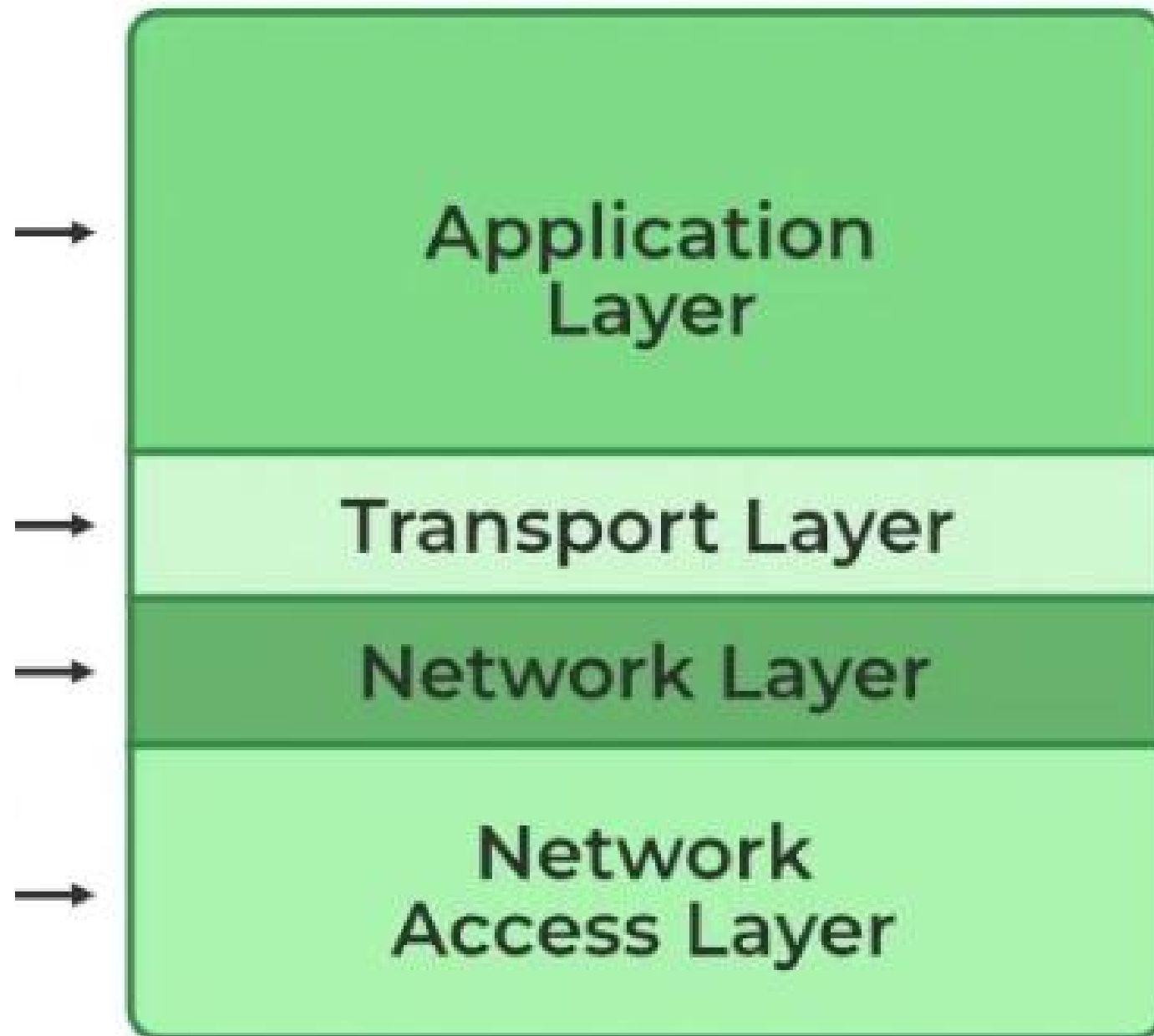


Server



Receiver

TCP/IP



1. Application Layer

The Application Layer is the closest to the end user and is where applications and user interfaces reside. It serves as the bridge between user programs and the lower layers responsible for data transmission.

- **Function:** Provides services and interfaces for end-user applications to access network resources.
- **Key responsibilities:**
 - Supports application protocols like HTTP, FTP, SMTP, DNS, etc.
 - Enables communication between software applications across networks.
 - Handles data formatting, encryption, and session management.

2. Transport Layer

This layer ensures data is delivered reliably and in the correct order between devices. The two main protocols in this layer are TCP (Transmission Control Protocol) and UDP (User Datagram Protocol).

- **Function:** Ensures reliable or unreliable delivery of data between hosts.
- **Key responsibilities:**
 - TCP (Transmission Control Protocol): Provides reliable, connection-oriented delivery with error checking, retransmission, and flow control.
 - UDP (User Datagram Protocol): Provides faster, connectionless transmission without guarantees.
 - Manages flow control and segmentation/reassembly of data.

3. Internet Layer

It handles the routing of data packets across networks. It uses the Internet Protocol (IP) to assign unique IP addresses to devices and decide the most efficient path for data to reach its destination.

- **Function:** Determines the best path for data to travel across networks.
- **Key responsibilities:**
 - IP (Internet Protocol): Provides addressing and routing.
 - Handles packet forwarding, fragmentation, and logical addressing (IP addresses).
 - Involves protocols like IP, ICMP (for diagnostics), and ARP (for address resolution).

4. Network Access Layer

This layer is the lowest layer in the model and responsible for the physical connection between devices within the same network segment.

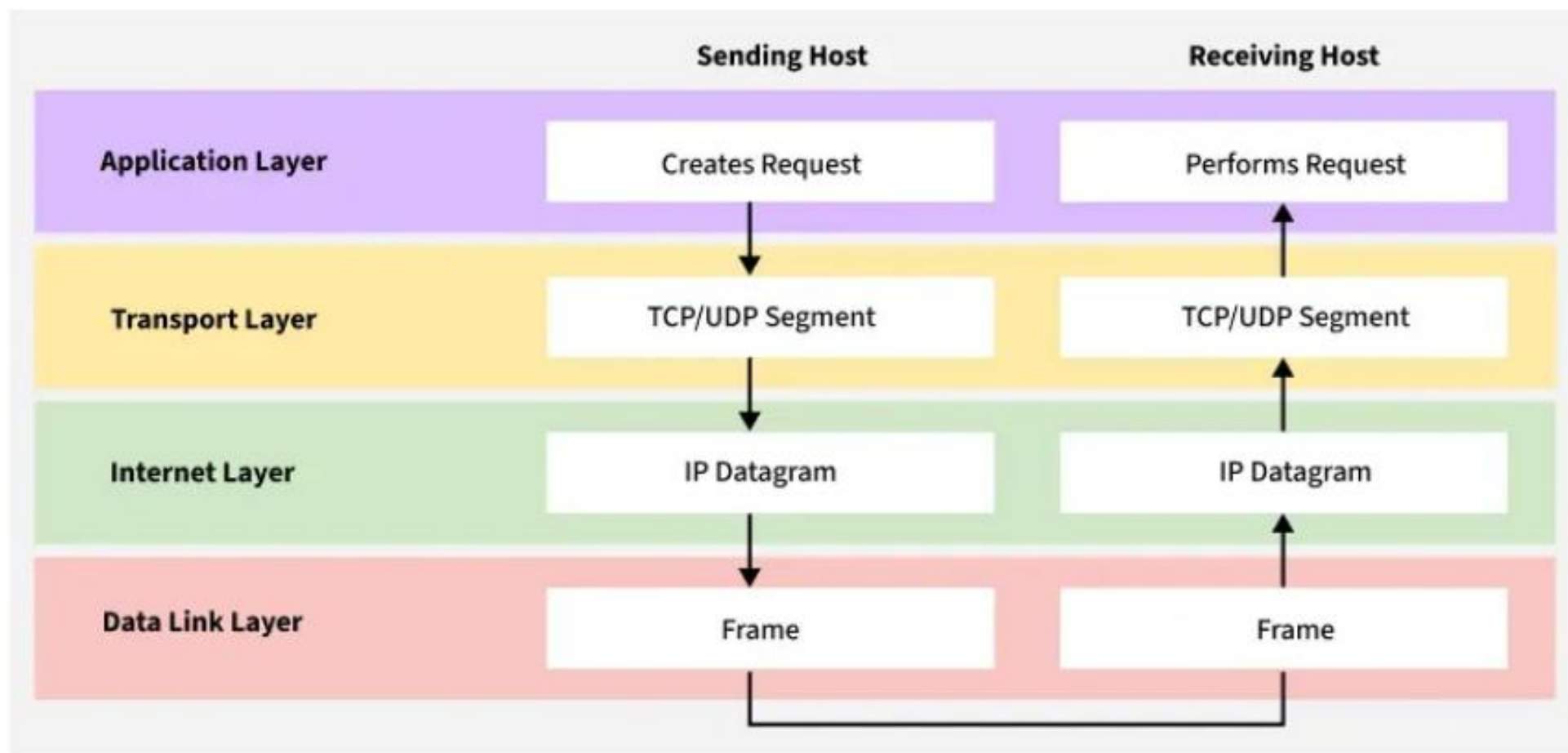
- **Function:** Manages the physical transmission of data over the network hardware.
- **Key responsibilities:**
 - Handles how data is physically sent over cables, Wi-Fi, etc.
 - Manages MAC addressing, framing, and error detection at the physical link.
 - Includes Ethernet, Wi-Fi, and other data link technologies.

Working of TCP/IP Model

When Sending Data (From Sender to Receiver)

- **Application Layer**
 - A user sends data through an application (e.g., opening a website via a browser).
 - The application prepares data for transmission (e.g., using [HTTP](#), [FTP](#), [SMTP](#)).
- **Transport Layer (TCP/UDP)**
 - TCP breaks data into small segments, adds a header (with sequence numbers, source/destination ports).
 - Ensures reliable delivery (TCP) or fast, connectionless delivery (UDP).
- **Internet Layer (IP)**
 - Adds IP addresses to each packet (source and destination).
 - Determines the route the packet should take to reach the destination.
- **Link Layer (Network Access Layer)**
 - Converts packets into frames, adds MAC (physical) addresses.
 - Sends data as binary bits (0s and 1s) over the physical medium (e.g., Ethernet, Wi-Fi).





When Receiving Data (At the Destination)

- **Link Layer**
 - Receives bits and reconstructs frames.
 - Passes frames up to the Internet layer.
- **Internet Layer**
 - Reads the IP address to confirm it's the correct recipient.
 - Removes the IP header and sends the data to the Transport layer.
- **Transport Layer**
 - Reassembles TCP segments in the correct order.
 - Verifies data integrity using acknowledgments and checksums.
- **Application Layer**
 - The data is delivered to the appropriate application (e.g., browser displays a web page).

User Datagram Protocol (UDP) is a fundamental transport layer protocol in the Internet Protocol (IP) suite that enables applications to send messages, called datagrams, without the overhead and reliability guarantees of connection-oriented protocols like TCP. Below is an in-depth explanation of UDP, covering its design, header structure, operation, applications, comparison with TCP, role in network address translation (NAT) and security concerns, and some extended variants.

Overview and Design Philosophy

UDP is a connectionless protocol:

Unlike TCP, UDP does not establish a connection between the sender and receiver before data transmission. Each message, or datagram, is sent independently, without negotiation, session establishment, or acknowledgments. This minimalistic approach leads to very low latency and less protocol overhead, which is especially valuable for time-sensitive applications. However, this also means that UDP does not provide:

- **Reliability:** There is no guarantee that packets will arrive at the destination.
- **Ordering:** Packets may arrive in any order.
- **Flow or Congestion Control:** The protocol does not adjust its sending rate based on network conditions.

UDP's design is predicated on the idea that many modern applications (e.g., real-time streaming, voice over IP, online gaming) can tolerate some loss or reordering of packets if it means achieving faster transmission speeds and reduced latency. In such scenarios, the application itself must address any error correction or ordering if needed.

UDP Header Structure and Checksum

A UDP datagram is composed of a very simple and fixed-length header followed by the data payload. The header is only 8 bytes long and consists of four fields, each 16 bits in size:

1. Source Port:

Identifies the sending application's port. When not used (for example, when no response is expected), this field may be set to zero.

2. Destination Port:

Indicates the port number of the receiving application. Since ports are 16-bit numbers, the range spans from 0 to 65,535.

3. Length:

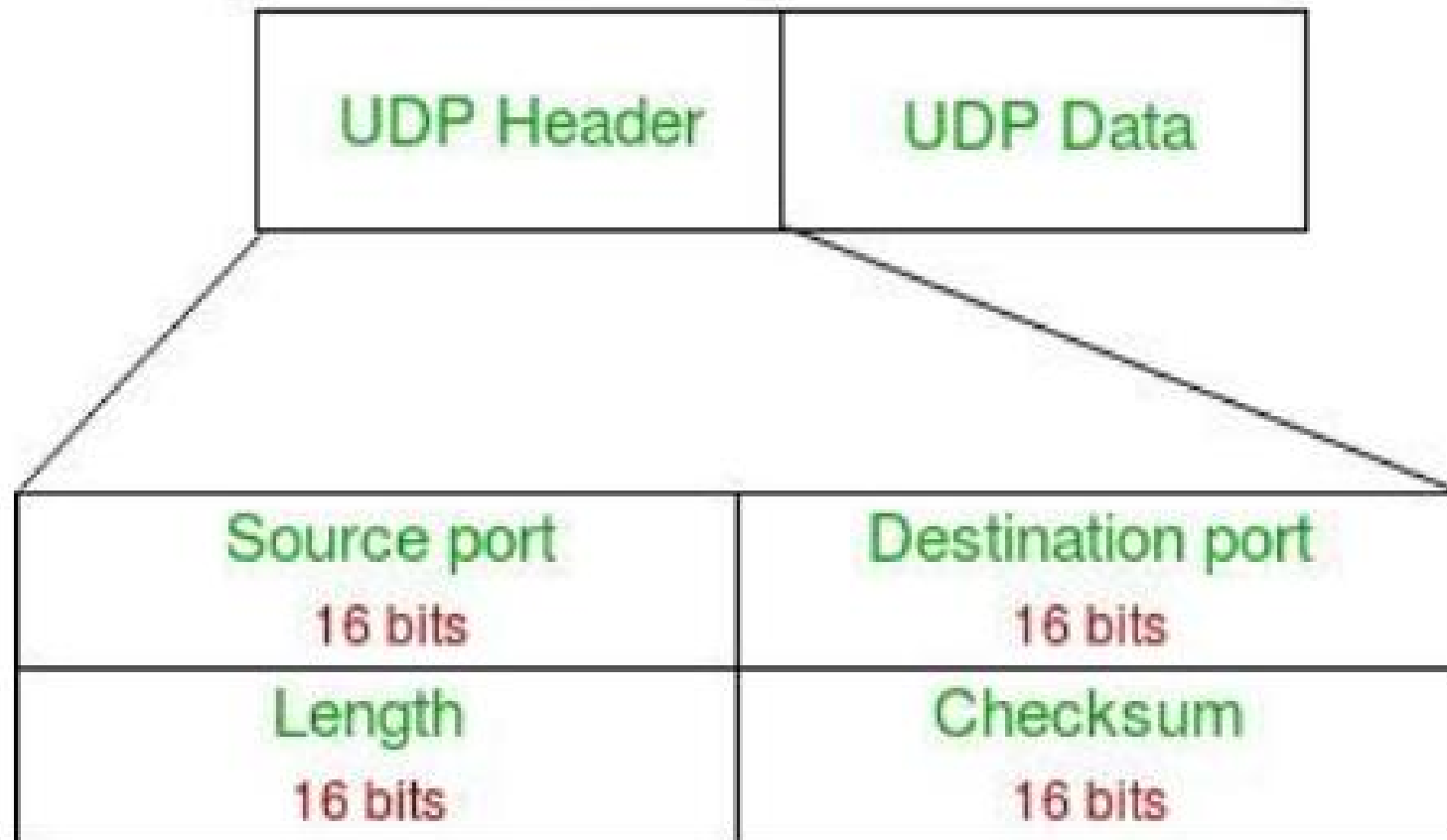
Specifies the total length of the UDP packet, including both header and data. This helps the receiver determine where the UDP packet ends.

4. Checksum:

Provides a simple error-checking mechanism to verify the integrity of both the header and data. In IPv4, the checksum is optional and may be omitted, whereas in IPv6 it is mandatory. The checksum is calculated using a one's-complement sum over the header, payload, and a pseudo header (which contains some fields from the IP header such as the source and destination addresses) to help ensure data integrity.



8 Bytes



UDP Header

How UDP Works

1. Packet Creation:

When an application wants to send data, it packages the data into a UDP datagram. The application sets the appropriate source and destination port numbers, calculates the length, and—if desired—computes the checksum.

2. Transmission (No Handshake):

The datagram is then passed to the IP layer for routing without any prior handshaking or session negotiation. Because UDP is stateless, each datagram is independent, and the sender does not wait for any acknowledgment from the receiver.

3. Routing:

Routers forward the IP packet based on the destination IP address. Since UDP does not maintain any state information about the transmission, it relies entirely on the underlying IP network for delivery; however, IP itself is a best-effort delivery mechanism.

4. Reception:

The receiving host processes the IP packet and extracts the UDP datagram. The operating system uses the destination port number to deliver the datagram to the correct application. If a checksum is present, the recipient can verify data integrity; however, if errors are detected, UDP does not provide a mechanism for retransmission.

Basis	Transmission Control Protocol (TCP)	User Datagram Protocol (UDP)
Type of Service	<p><u>TCP</u> is a connection-oriented protocol. Connection orientation means that the communicating devices should establish a connection before transmitting data and should close the connection after transmitting the data.</p>	<p><u>UDP</u> is the Datagram-oriented protocol. This is because there is no overhead for opening a connection, maintaining a connection, or terminating a connection. UDP is efficient for broadcast and multicast types of network transmission.</p>
Reliability	TCP is reliable as it guarantees the delivery of data to the destination router.	The delivery of data to the destination cannot be guaranteed in UDP.
Error checking mechanism	TCP provides extensive error-checking mechanisms. It is because it provides flow control and acknowledgment of data.	UDP has only the basic error-checking mechanism using checksums.
Acknowledgment	An acknowledgment segment is present.	No acknowledgment segment.
Sequence	Sequencing of data is a feature of Transmission Control Protocol (TCP). this means that packets arrive in order at the receiver.	There is no sequencing of data in UDP. If the order is required, it has to be managed by the application layer.
Speed	TCP is comparatively slower than UDP.	UDP is faster, simpler, and more efficient than TCP.

Retransmission	Retransmission of lost packets is possible in TCP, but not in UDP.	There is no retransmission of lost packets in the User Datagram Protocol (UDP).
Header Length	TCP has a (20-60) bytes variable length header.	UDP has an 8 bytes fixed-length header.
Weight	TCP is heavy-weight.	UDP is lightweight.
Handshaking Techniques	Uses handshakes such as SYN, ACK, SYN-ACK	It's a connectionless protocol i.e. No handshake
Broadcasting	TCP doesn't support Broadcasting.	UDP supports Broadcasting.
Protocols	TCP is used by HTTP , HTTPS , FTP , SMTP and Telnet .	UDP is used by DNS, DHCP, TFTP, SNMP , RIP, and VoIP.
Stream Type	The TCP connection is a byte stream.	UDP connection is a message stream.
Overhead	Low but higher than UDP.	Very low.
Applications	This protocol is primarily utilized in situations when a safe and trustworthy communication procedure is necessary, such as in email, on the web surfing, and in military services.	This protocol is used in situations where quick communication is necessary but where dependability is not a concern, such as VoIP, game streaming, video, and music streaming, etc.

Applications of UDP

Because of its low overhead and fast transmission capability, UDP is used in several real-world applications:

- **Real-Time Multimedia:**

Applications like VoIP, live video streaming, and online gaming use UDP because they can tolerate occasional packet loss better than the delay that would result from retransmissions (which could disrupt the real-time experience).

cloudns.net

- **Domain Name System (DNS):**

DNS queries and responses are typically short and require quick turnaround; using UDP minimizes the delay caused by connection setup.

- **Broadcast and Multicast:**

UDP supports broadcasting and multicasting, making it possible to send a single packet that multiple hosts can receive. This is particularly useful in applications like service discovery, streaming live events, or routing protocol updates (e.g., RIP).

geeksforgeeks.org

- **Network Time Protocol (NTP) and DHCP:**

Both NTP (for time synchronization) and DHCP (for IP address assignment) use UDP due to their tolerance for data loss and lower latency requirements.

- **Lightweight and Interactive Protocols:**

Many peer-to-peer (P2P) applications and NAT traversal techniques (like UDP hole punching) rely on UDP to establish direct communication paths between hosts behind network address translators.



Security Implications and Abuse

While UDP's simplicity and speed are beneficial for many applications, they also present some security challenges:

- **Lack of Built-in Authentication or Encryption:**

Since UDP does not establish a connection or maintain state, it offers no inherent way to authenticate packets. Applications requiring secure data transfer must implement their own security layers (for example, using protocols like Datagram Transport Layer Security or DTLS).

en.wikipedia.org

- **Vulnerability to DDoS Attacks:**

Because UDP does not have flow control or handshake mechanisms, it can be exploited for distributed denial-of-service (DDoS) attacks. Attackers can flood a target with UDP packets, overwhelming its capacity to respond, or use UDP amplification techniques (such as DNS amplification) to multiply the attack volume.

imperva.com

- **Firewall and NAT Filtering:**

The absence of a connection state can make UDP traffic more challenging to secure with traditional firewall rules, often leading to stricter filtering policies that may inadvertently block legitimate traffic.

Logical Addressing

Logical addressing is a function of the **Network** layer of the OSI Model (Layer-3), and provides a hierarchical structure to separate networks. Logical addresses are never hardcoded on physical network interfaces, and can be dynamically assigned and changed freely.

A logical address contains two components:

- **Network ID** – identifies which network a host belongs to.
- **Host ID** – uniquely identifies the host on that network.

Examples of logical addressing protocols include **Internetwork Packet Exchange (IPX)** and **Internet Protocol (IP)**. IPX was predominantly used on Novell networks, but is now almost entirely deprecated. **IP** is the most widely-used logical address, and is the backbone protocol of the Internet.

Internet Protocol (IP)

In the 1970's, the Department of Defense developed the **Transmission Control Protocol (TCP)**, to provide both Network and Transport layer functions. When this proved to be an inflexible solution, those functions were separated - with the **Internet Protocol (IP)** providing Network layer services, and TCP providing Transport layer services.

Together, TCP and IP provide the core functionality for the **TCP/IP** or **Internet protocol suite**.

IP provides two fundamental Network layer services:

- **Logical addressing** – provides a unique address that identifies both the *host*, and the *network* that host exists on.
- **Routing** – determines the *best path* to a particular destination network, and then *routes* data accordingly.

IP was originally defined in RFC 760, and has been revised several times. IP Version 4 (**IPv4**) was the first version to experience widespread deployment, and is defined in RFC 791. IPv4 will be the focus of this guide.

IPv4 employs a **32-bit address**, which limits the number of possible addresses to 4,294,967,296. IPv4 will eventually be replaced by IP Version 6 (**IPv6**), due to a shortage of available IPv4 addresses. IPv6 is covered in great detail in another guide.

IPv4 Addressing

A core function of IP is to provide logical addressing for hosts. An **IP address** provides a hierarchical structure to both uniquely identify a *host*, and what *network* that host exists on.

An IP address is most often represented in **decimal**, in the following format:

158.80.164.3

An IP address is comprised of four **octets**, separated by periods:

First Octet	Second Octet	Third Octet	Fourth Octet
158	80	164	3

Each octet is an **8-bit** number, resulting in a **32-bit IP address**. The smallest possible value of an octet is 0, or *00000000* in binary. The largest possible value of an octet is 255, or *11111111* in binary.

The above IP address represented in binary would look as follows:

First Octet	Second Octet	Third Octet	Fourth Octet
10011110	01010000	10100100	00000011

The Subnet Mask

Part of an IP address identifies the *network*. The other part of the address identifies the *host*. A **subnet mask** is required to provide this distinction:

158.80.164.3 255.255.0.0

The above IP address has a subnet mask of 255.255.0.0. The subnet mask follows two rules:

- If a binary bit is set to a **1** (or *on*) in a subnet mask, the corresponding bit in the address identifies the **network**.
- If a binary bit is set to a **0** (or *off*) in a subnet mask, the corresponding bit in the address identifies the **host**.

Looking at the above address and subnet mask in binary:

IP Address:	10011110.01010000.10100100.00000011
Subnet Mask:	11111111.11111111.00000000.00000000

The first 16 bits of the subnet mask are set to *1*. Thus, the first 16 bits of the address (158.80) identify the *network*. The last 16 bits of the subnet mask are set to *0*. Thus, the last 16 bits of the address (164.3) identify the unique *host* on that network.

The network portion of the subnet mask must be **contiguous**. For example, a subnet mask of 255.0.0.255 is not valid.

The Subnet Mask (continued)

Hosts on the same logical network will have *identical* network addresses, and can communicate freely. For example, the following two hosts are on the same network:

Host A:	158.80.164.100	255.255.0.0
Host B:	158.80.164.101	255.255.0.0

Both share the same network address (*158.80*), which is determined by the *255.255.0.0* subnet mask. Hosts that are on *different* networks cannot communicate without an intermediating device. For example:

Host A:	158.80.164.100	255.255.0.0
Host B:	158.85.164.101	255.255.0.0

The subnet mask has remained the same, but the network addresses are now different (*158.80* and *158.85* respectively). Thus, the two hosts are *not* on the same network, and cannot communicate without a **router** between them. **Routing** is the process of forwarding packets from one network to another.

Consider the following, trickier example:

Host A:	158.80.1.1	255.248.0.0
Host B:	158.79.1.1	255.248.0.0

The specified subnet mask is now *255.248.0.0*, which doesn't fall cleanly on an octet boundary. To determine if these hosts are on separate networks, first convert everything to binary:

Host A Address:	10011110.01010000.00000001.00000001
Host B Address:	10011110.01001111.00000001.00000001
Subnet Mask:	11111111.11111000.00000000.00000000

Remember, the **1** (or **on**) bits in the subnet mask identify the *network* portion of the address. In this example, the first *13 bits* (the 8 bits of the first octet, and the first 5 bits of the second octet) identify the network. Looking at only the first 13 bits of each address:

Host A Address:	10011110.01010
Host B Address:	10011110.01001

Clearly, the network addresses are *not* identical. Thus, these two hosts are on separate networks, and require a router to communicate.

IP Address Classes

The IPv4 address space has been structured into several **classes**. The value of the **first octet** of an address determines the class of the network:

<i>Class</i>	<i>First Octet Range</i>	<i>Default Subnet Mask</i>
Class A	1 - 127	255.0.0.0
Class B	128 - 191	255.255.0.0
Class C	192 - 223	255.255.255.0
Class D	224 - 239	-

Class A networks range from **1** to **127**. The *default* subnet mask is 255.0.0.0. Thus, by *default*, the first octet defines the network, and the last three octets define the host. This results in a maximum of **127** Class A networks, with **16,777,214** hosts per network!

Example of a Class A address:

Address:	64.32.254.100
Subnet Mask:	255.0.0.0

Class B networks range from **128** to **191**. The *default* subnet mask is 255.255.0.0. Thus, by *default*, the first two octets define the network, and the last two octets define the host. This results in a maximum of **16,384** Class B networks, with **65,534** hosts per network.

Example of a Class C address:

Address:	207.79.233.6
Subnet Mask:	255.255.255.0

Class D networks are reserved for **multicast** traffic. Class D addresses do not use a subnet mask.

CIDR (Classless Inter-Domain Routing)

Classless Inter-Domain Routing (CIDR) is a simplified method of representing a subnet mask. CIDR identifies the number of binary bits set to a **1** (or *on*) in a subnet mask, preceded by a slash.

For example, a subnet mask of *255.255.255.240* would be represented as follows in binary:

11111111.11111111.11111111.11110000

The first 28 bits of the above subnet mask are set to *1*. The CIDR notation for this subnet mask would thus be */28*.

The CIDR mask is often appended to the IP address. For example, an IP address of *192.168.1.1* and a subnet mask of *255.255.255.0* would be represented as follows using CIDR notation:

192.168.1.1 /24

Subnet and Broadcast Addresses

On *each* IP network, two host addresses are reserved for special use:

- The **subnet** (or **network**) address
- The **broadcast** address

Neither of these addresses can be assigned to an actual host.

The **subnet** address is used to identify **the network itself**. A routing table contains a list of known networks, and each network is identified by its subnet address. Subnet addresses contain **all 0 bits in the host portion** of the address.

For example, *192.168.1.0/24* is a subnet address. This can be determined by looking at the address and subnet mask in binary:

IP Address:	11000000.10101000.00000001.00000000
Subnet Mask:	11111111.11111111.11111111.00000000

Note that all host bits in the address are set to 0.

The **broadcast** address identifies *all* hosts on a particular network. A packet sent to the broadcast address will be received and processed by every host on that network. Broadcast addresses contain **all 1 bits in the host portion** of the address.

For example, *192.168.1.255/24* is a broadcast address. Note that all host bits are set to 1:

IP Address:	11000000.10101000.00000001.11111111
Subnet Mask:	11111111.11111111.11111111.00000000

Private vs. Public IPv4 Addresses

The rapid growth of the Internet resulted in a shortage of available IPv4 addresses. In response, a specific subset of the IPv4 address space was designated as *private*, to temporarily alleviate this problem.

A **public address** can be routed on the Internet. Thus, hosts that must be Internet-accessible must be configured with (or *reachable* by) public addresses. Allocation of public addresses is governed by the Internet Assigned Numbers Authority (IANA).

A **private address** is intended for internal use within a home or organization, and can be freely used by anyone. However, private addresses can *never be routed* on the Internet. In fact, Internet routers are configured to immediately drop traffic with private addresses.

Three private address ranges were defined in RFC 1918, one for each IPv4 class:

- Class A - **10.x.x.x /8**
- Class B - **172.16.x.x /12**
- Class C - **192.168.x.x /24**

It is possible to *translate* between private and public addresses, using **Network Address Translation (NAT)**. NAT allows a host configured with a private address to be *stamped* with a public address, thus allowing that host to communicate across the Internet. It is also possible to translate multiple privately-addressed hosts to a single public address, which conserves the public address space.

Reserved IPv4 Addresses

In addition to the three private IPv4 ranges, several other addresses and ranges are reserved for specific purposes:

- The **0.0.0.0 /0** network is used to identify **all networks**, and is referred to as the **default route**. If a default route exists in a routing table, it will be used only if there is *not* a more *specific* route to a particular destination. Routing and default routes are covered extensively in another guide.
- The **0.0.0.0 /8** range is used to identify hosts on the *local* network. Addresses in this range can only be used as a *source* address. The most commonly used address in this range is **0.0.0.0 /32**, which a host will use when dynamically attempting to learn its IP address via Dynamic Host Configuration Protocol (DHCP). DHCP is covered extensively in another guide.
- The entire **127.x.x.x /8** range is reserved for diagnostic purposes. The most commonly used address in this range is **127.0.0.1**, which identifies the local host, and is referred to as the **loopback** or **localhost** address.
- The **169.254.x.x /16** range is reserved for Automatic Private IP Addressing (APIPA). A host assigns itself an address in this range, if it cannot dynamically obtain an address from a DHCP server.
- The **224.x.x.x – 239.x.x.x** ranges are reserved for **multicast**, and are referred to as **Class D** addresses.

What is Static IP Address

A [Static IP address](#) is an IP address that does not change frequently or constantly; it is reserved for a specific computer or device. This type of IP address does not dynamically change with time, but will only change through an action done by the user or the network administrator. Assigning a static IP address is common in servers, network devices or any device that has to have a fixed address that can be accessed from a distance.

How to Get a Static IP Address

In order to obtain a static IP address, there is a possibility of applying for it from the [Internet Service Provider](#) (ISP). This may be accompanied with the extra expense of getting the ISPs to provide static IP addresses since these often come with an added cost. For those who have their own physical infrastructure of the network, static IP addresses can be assigned in the settings through the device.

When Static IPs are Needed

Static IP addresses are especially important in cases where a device has to be quickly found over the internet on a permanent basis.

- **Web Servers:** A website must have one or more static IP addresses to be assigned to the domain always point to the correct server.
- **Remote Access:** Some of the devices that require a remote connection like the CCTV cameras or a [VPN](#) are preferable to be as static as possible.
- **Hosting Servers:** Game or email servers that are in constant use also need a static IP so that the services running in the background remain undisturbed.



Advantages of Static IP Address

- **Stable Connection:** A static IP address stays the same, so your device or server always has the same address.
- **Easy Remote Access:** It makes it easier to connect to your device or server from anywhere because the address doesn't change.
- **Website Hosting:** If you're hosting a website or email server, a static IP ensures it's always found at the same address.
- **Better Security:** It's easier to set up security like firewalls for a device with a static IP since the address is fixed.

What is Dynamic IP Address

A [Dynamic IP address](#) is an IP address which is changed from time to time. In contrast to the static IP, an [IPv6](#) address is obtained by DHCP server – (Dynamic Host Configuration Protocol) automatically. In the [DHCP](#), a host receives an available IP address from the DHCP server for some period of time referred to as the lease time and the IP address given to the host may change. Dynamic IPs are more common for home and commercial appliances and other electronic devices for which it is not necessary to have a fixed (static) IP address.

How to Get a Dynamic IP Address

Dynamic IP addresses are those that are assigned to you by the ISP on a random and on a connecting basis. Dynamic IPs can be obtained without any specific request or change as most of the ISP's offer them by default. Whenever one launches a connection to some web, automatically the DHCP server doles out an IP address from a series of IP addresses.

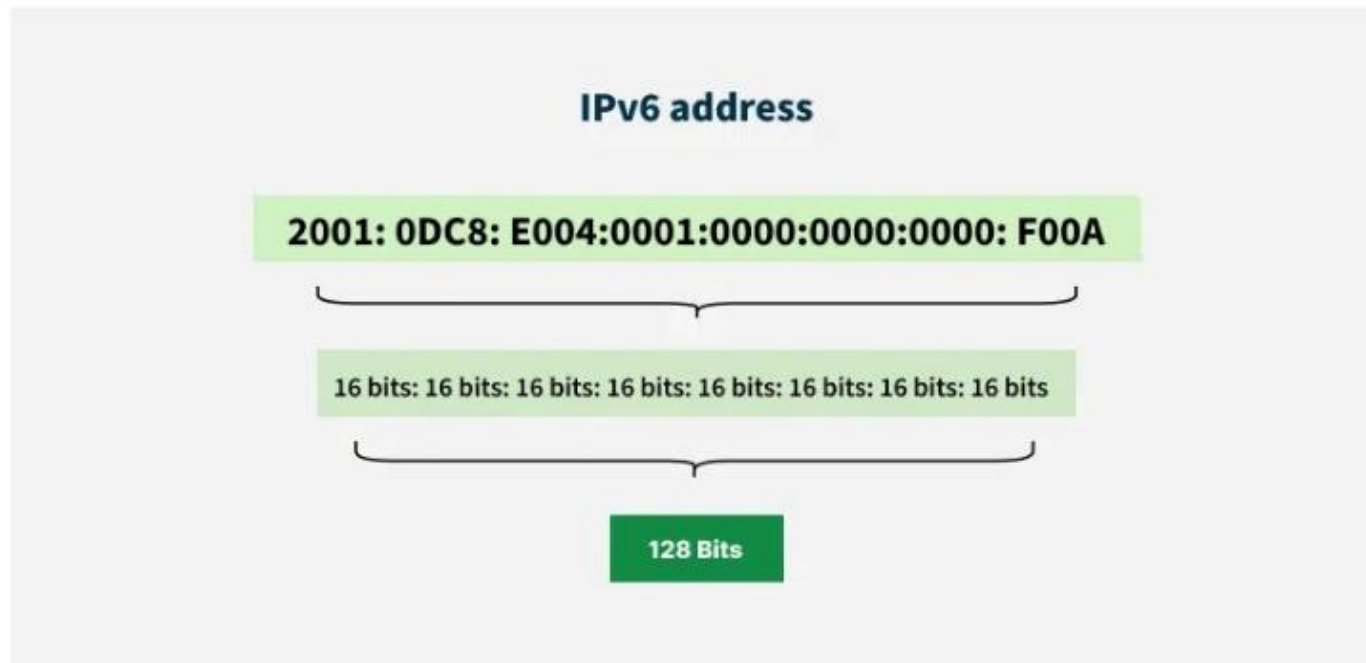
Advantages of Dynamic IP Address

- **More Available and Cheaper:** Since there are more dynamic IPs and they cost less, they are more economical than static ones.
- **Better Security:** Dynamic IPs change each time you connect, making it harder for hackers to target your device.
- **Easy to Use:** You don't have to worry about managing or setting up the IP, which makes it great for home use or people who aren't tech-savvy.



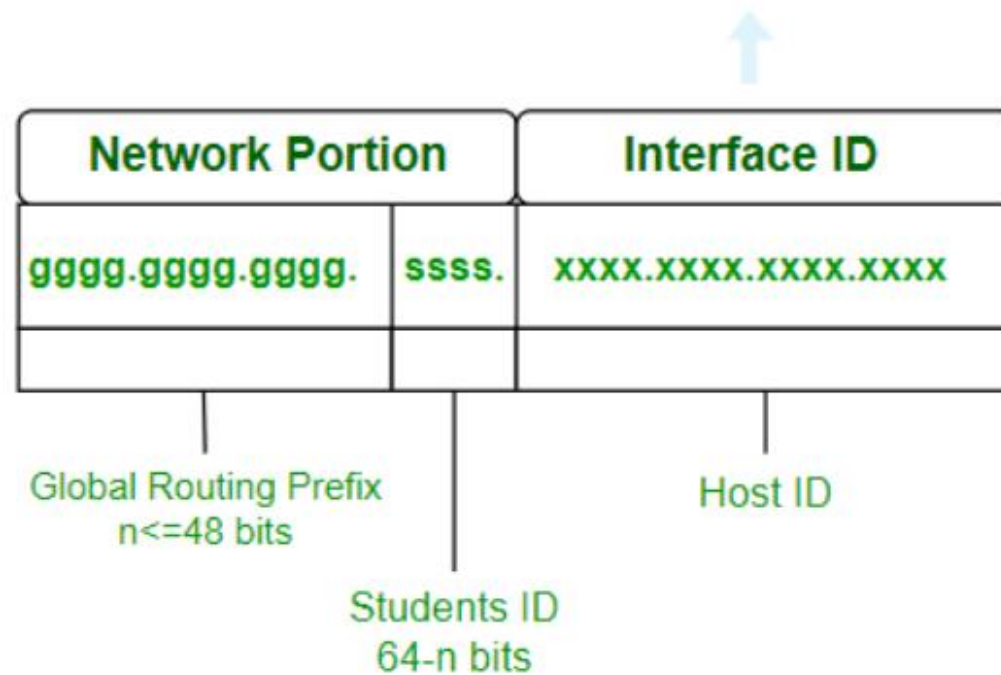
IPv6

The next generation Internet Protocol (IP) address standard, known as IPv6, is meant to work in cooperation with [IPv4](#). To communicate with other devices, a computer, smartphone, home automation component, Internet of Things sensor, or any other Internet-connected device needs a numerical IP address. Because so many connected devices are being used, the original IP address scheme, known as IPv4, is running out of addresses. This new IP address version is being deployed to fulfil the need for more Internet addresses. With 128-bit address space, it allows 340 undecillion unique address space. IPv6 support a theoretical maximum of 340, 282, 366, 920, 938, 463, 463, 374, 607, 431, 768, 211, 456.



Representation of IPv6

An IPv6 address consists of eight groups of four hexadecimal digits separated by ' . ' and each Hex digit representing four bits so the total length of IPv6 is 128 bits. Structure given below.



IPV6-Representation

gggg.gggg.gggg.ssss.xxxx.xxxx.xxxx.xxxx

The first 48 bits represent Global Routing Prefix. The next 16 bits represent the student ID and the last 64 bits represent the host ID. The first 64 bits represent the network portion and the last 64 bits represent the interface id.

- **Global Routing Prefix:** The Global Routing Prefix is the portion of an IPv6 address that is used to identify a specific network or subnet within the larger IPv6 internet. It is assigned by an ISP or a regional internet registry (RIR).
- **Student Id:** The portion of the address used within an organization to identify subnets. This usually follows the Global Routing Prefix.
- **Host Id:** The last part of the address, is used to identify a specific host on a network.

PORT	Service	Description	Transport Protocol
7	Echo	Port just echoes whatever is sent to it. This feature can be used in many attacks, such as Smurf/Fraggle.	TCP and UDP
20 /21	File Transfer Protocol (FTP)	Port used by FTP protocol to send data to the client	TCP
22	Secure Shell (SSH)	Used as secure replacement protocol for Telnet	TCP and UDP
23	Telnet	Port used by Telnet to remotely connect to a workstation or server(unsecured)	TCP
25	Simple Mail Transfer Protocol (SMTP)	Used to send E-Mail over internet	TCP
53	Domain Name System (DNS)	Port for DNS requests, network routing, and zone transfers	TCP and UDP
67 /68	Dynamic Host Configuration Protocol (DHCP)	Used on networks that do not use static IP address assignment.	UDP
80	Hypertext Transfer Protocol (HTTP)	Used for browsing web-pages on a browser	TCP

What are Application Layer Protocols?

Application layer protocols are those protocols utilized at the application layer of the OSI (Open Systems Interconnection) and TCP/IP models. They facilitate communication and data sharing between software applications on various network devices. These protocols define the rules and standards that allow applications to interact and communicate quickly and effectively over a network.

Application Layer Protocol in Computer Network

1. TELNET

Telnet stands for the TELEtype NETwork. It helps in terminal emulation. It allows Telnet clients to access the resources of the Telnet server. It is used for managing files on the Internet. It is used for the initial setup of devices like switches. The telnet command is a command that uses the Telnet protocol to communicate with a remote device or system. The port number of the telnet is 23.

Command

```
telnet [\\RemoteServer]  
\\RemoteServer  
: Specifies the name of the server  
to which you want to connect
```

2. FTP

FTP stands for [File Transfer Protocol](#). It is the protocol that actually lets us transfer files. It can facilitate this between any two machines using it. But FTP is not just a protocol but it is also a program. FTP promotes sharing of files via remote computers with reliable and efficient data transfer. The Port number for FTP is 20 for data and 21 for control.

Command

```
ftp machinename
```


5. SMTP

It stands for [Simple Mail Transfer Protocol](#). It is a part of the TCP/IP protocol. Using a process called "store and forward," SMTP moves your email on and across networks. It works closely with something called the Mail Transfer Agent (MTA) to send your communication to the right computer and email inbox. The Port number for SMTP is 25.

Command

```
MAIL FROM:<mail@abc.com?
```

8. SNMP

It stands for [Simple Network Management Protocol](#). It gathers data by polling the devices on the network from a management station at fixed or random intervals, requiring them to disclose certain information. It is a way that servers can share information about their current state, and also a channel through which an administrator can modify pre-defined values. The Port number of SNMP is 161(TCP) and 162(UDP).

Command

```
snmpget -mALL -v1 -cp public snmp_agent_ip_address sysName.0
```

9. DNS

It stands for [Domain Name System](#). Every time you use a domain name, therefore, a DNS service must translate the name into the corresponding IP address. For example, the domain name www.abc.com might translate to 198.105.232.4. The Port number for DNS is 53.

Command

```
ipconfig /flushdns
```



10. DHCP

It stands for [Dynamic Host Configuration Protocol](#) (DHCP). It gives IP addresses to hosts. There is a lot of information a DHCP server can provide to a host when the host is registering for an IP address with the DHCP server. Port number for DHCP is 67, 68.

Command

```
clear ip dhcp binding {address | * }
```

11. HTTP/HTTPS

HTTP stands for [Hypertext Transfer Protocol](#) and HTTPS is the more secured version of HTTP, that's why HTTPS stands for Hypertext Transfer Protocol Secure. This protocol is used to access data from the World Wide Web. The Hypertext is the well-organized documentation system that is used to link pages in the text document.

- HTTP is based on the client-server model.
- It uses TCP for establishing connections.
- HTTP is a stateless protocol, which means the server doesn't maintain any information about the previous request from the client.
- HTTP uses port number 80 for establishing the connection.