**Net Practice**

The internet

The Internet (originally [ARPAnet](https://en.wikipedia.org/wiki/ARPAnet)) was developed as a network between government research laboratories and participating departments of universities. Other companies and organizations joined by direct connection to the [backbone](https://en.wikipedia.org/wiki/Internet_backbone), or by arrangements through other connected companies, sometimes using dialup tools such as [UUCP](https://en.wikipedia.org/wiki/UUCP). By the late 1980s, a process was set in place towards public, commercial use of the Internet. Some restrictions were removed by 1991,[[1]](https://en.wikipedia.org/wiki/Internet_service_provider" \l "cite_note-1) shortly after the introduction of the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web).[[2]](https://en.wikipedia.org/wiki/Internet_service_provider" \l "cite_note-2)

What do we use to connect to the internet?

To connect to the internet, several components and technologies work together. Here's a breakdown of the primary methods and equipment used to establish an internet connection:

### 1. ****Modem**** (Modulator-Demodulator):

A **modem** is a device that converts **digital data** from a computer or network into an **analog signal** that can travel over phone lines, coaxial cables, or fiber-optic cables. It also converts the incoming analog signal back into digital data that your computer or router can understand.

Modems are used in various types of broadband connections:

* **DSL (Digital Subscriber Line)**: Uses phone lines for data transmission.
* **Cable Modem**: Uses coaxial cables, typically provided by cable TV companies.
* **Fiber Optic Modem**: Used in fiber-optic broadband connections, providing faster and more reliable internet.

### 2. ****Router****:

A **router** is a device that routes data from your modem to multiple devices in your home or office. It typically connects to your modem and acts as the central hub for local area network (LAN) devices, allowing them to share the internet connection.

* **Wireless Router**: Provides Wi-Fi access for wireless devices like laptops, smartphones, and tablets.
* **Wired Router**: Provides wired Ethernet connections to devices like desktop computers, smart TVs, and gaming consoles.

### 3. ****Wi-Fi**** (Wireless Fidelity):

**Wi-Fi** is a wireless networking technology that uses radio waves to allow devices like smartphones, laptops, and tablets to connect to the internet without physical cables. It relies on a wireless router to broadcast the internet connection to nearby devices. Wi-Fi networks use different standards (e.g., **Wi-Fi 4, Wi-Fi 5, Wi-Fi 6**) that define the speed and range of the wireless signal.

### 4. ****Ethernet**** (Wired Connection):

**Ethernet** is a type of wired connection that allows devices to connect to the internet through a physical cable. It's typically faster and more reliable than wireless connections, providing a stable internet connection, especially for devices like desktop computers and game consoles. You plug an Ethernet cable into a router or modem to establish the connection.

5. ****Cellular Networks (Mobile Data)****:

For mobile devices like smartphones, **cellular networks** provide internet access using technologies such as **4G**, **5G**, or older standards like **3G**. These networks allow devices to connect to the internet via mobile towers that transmit and receive data signals. Cellular internet can be accessed either directly through your mobile device or by using a **mobile hotspot** device to share the connection with other devices.

### 6. ****Satellite****:

In rural or remote areas where wired or wireless broadband isn't available, **satellite internet** can be an option. It involves a satellite dish at the user's location that communicates with a satellite orbiting the Earth, which then connects to a ground station. Satellite internet is slower and has higher latency than other broadband options but is useful where other infrastructure is unavailable.

### 7. ****Fiber-Optic Internet****:

**Fiber-optic internet** is the fastest and most reliable type of broadband. It uses light signals transmitted through glass or plastic fibers to send data over long distances. Fiber-optic connections can provide speeds much faster than DSL or cable internet. Services like **FTTH** (Fiber To The Home) bring fiber directly to your residence, allowing for ultra-fast internet speeds.

### 8. ****DSL (Digital Subscriber Line)****:

**DSL** uses standard telephone lines to deliver internet service, but it does not interfere with voice calls. It's widely available in urban and suburban areas and can provide speeds ranging from hundreds of kbps to several Mbps, depending on the distance from the phone exchange.

### 9. ****Hotspots****:

**Public or mobile hotspots** allow devices to connect to the internet through a wireless connection in places like cafes, libraries, airports, and shopping centers. These are often powered by mobile network providers or private Wi-Fi networks, enabling users to access the internet anywhere there's a hotspot.

### 10. ****Tethering****:

**Tethering** allows a mobile phone with a cellular data connection to share its internet access with other devices like laptops, tablets, or other phones. This is typically done via Wi-Fi, Bluetooth, or USB cables.

### 11. ****Internet Service Provider (ISP)****:

Finally, to access the internet, you need an **Internet Service Provider (ISP)**. The ISP provides the actual connection to the internet, whether it's through fiber, cable, DSL, or satellite. Popular ISPs include companies like Comcast, AT&T, Verizon, or regional providers.

What is ISP **Internet service provider**?

An **Internet service provider** (**ISP**) is an organization that provides a myriad of services related to accessing, using, managing, or participating in the [Internet](https://en.wikipedia.org/wiki/Internet). ISPs can be organized in various forms, such as commercial, [community-owned](https://en.wikipedia.org/wiki/Community-owned), [non-profit](https://en.wikipedia.org/wiki/Non-profit), or otherwise [privately owned](https://en.wikipedia.org/wiki/Privately_owned).

Internet services typically provided by ISPs can include [internet access](https://en.wikipedia.org/wiki/Internet_access), [internet transit](https://en.wikipedia.org/wiki/Internet_transit), [domain name](https://en.wikipedia.org/wiki/Domain_name) registration, [web hosting](https://en.wikipedia.org/wiki/Web_hosting), and [colocation](https://en.wikipedia.org/wiki/Colocation_centre).

## **DSL: A quick definition**

The high-speed internet that you connect to via Wi-Fi or an ethernet cable through a modem is DSL internet. DSL stands for Digital Subscriber Line/Loop and it is a communication medium that receives data via a copper telephone landline.   
 DSL is the primary form of broadband internet access and uses existing telephone wiring to transmit data via a DSL modem, making the internet accessible to all.

What is FiOS?

FiOS (Fiber Optic Service) is an innovative communication technology utilizing fiber-optic cables for data transmission. It offers faster and more reliable internet compared to traditional services. With high bandwidth and low latency, FiOS enhances experiences like streaming, gaming, and video conferencing. The use of fiber optics ensures consistent speeds, making it an ideal choice for those prioritizing a seamless online connection. FiOS stands out for its future-proof design, adapting easily to evolving internet demands and providing a superior, reliable internet experience.

### How FiOS Works:

* **Fiber to the Home (FTTH)**: FiOS uses **FTTH** technology, meaning the fiber-optic cable runs directly into your home (as opposed to fiber-to-the-node or fiber-to-the-curb, where fiber is only brought to a certain point in your neighborhood). This ensures the highest performance possible.
* The fiber-optic cables carry digital data in the form of light, which is transmitted to and from your home through a device called an **optical network terminal (ONT)**. This device converts the fiber-optic signals into the electrical signals used by your router, TV, and phone.

### What is a Router?

A **router** is a network device that forwards data packets between computer networks, directing the data to its intended destination. It acts as a mediator or traffic manager between different devices on a network (such as computers, smartphones, or printers) and the broader internet or other networks.

1. **Data Transfer**:
   * When you try to access a website or send data over the internet, the data is first divided into small packets. Each packet contains information about its source, destination, and data content.
2. **Routing the Packets**:
   * The router reads the destination IP address in each packet and compares it with its **routing table** to determine where to send it next.
   * If you're accessing a website, the router sends your request to the ISP’s router, which forwards it across the internet to the web server. The server processes your request and sends the data back, using the same routing process to reach your router and deliver the response.
3. **Routing to Devices in a Local Network**:
   * When you access a website on your phone, for example, the router sends data between your phone (within your home network) and the internet. The router manages the data flow between all connected devices, ensuring they can communicate with each other or the internet.0

What is Dynamic Host Configuration Protocol?

**DHCP** (Dynamic Host Configuration Protocol) is a network protocol that automatically assigns **IP addresses** and other network configuration parameters to devices (also called **hosts**) on a network, allowing them to communicate with each other and access network resources. It eliminates the need for manual configuration of each device on a network.

### Key Functions of DHCP:

1. **Automatic IP Address Assignment**:
   * DHCP dynamically assigns a unique **IP address** to each device on a network. This process happens automatically, so you don’t have to manually assign IP addresses to devices (like computers, phones, printers, etc.).
   * When a device connects to a network (either wired or wireless), the DHCP server assigns it an available IP address from a **range of addresses** (called a **pool** or **scope**) specified by the network administrator.
2. **Providing Network Configuration Information**:
   * In addition to the IP address, DHCP also provides other essential information for network communication, such as:
     + **Subnet mask**: Defines the network's range and helps determine whether an IP address is within the local network.
     + **Default gateway**: The router or device that routes traffic from the local network to other networks (e.g., the internet).
     + **DNS servers**: Domain Name System servers that allow the device to resolve domain names (like www.example.com) to IP addresses.
3. **Lease Time**:
   * When a device gets an IP address from the DHCP server, it doesn’t own it permanently. The DHCP lease is typically valid for a certain period, known as the **lease time**.
   * When the lease expires, the device must request a new IP address from the DHCP server.

### How DHCP Works:

1. **DHCP Discovery**:
   * When a device connects to the network, it doesn’t have an IP address yet. The device sends a **DHCP Discover** message to locate a DHCP server on the network. This message is broadcast to all devices on the local network.
2. **DHCP Offer**:
   * The DHCP server responds with a **DHCP Offer** message, which includes an available IP address, subnet mask, gateway, DNS servers, and the lease time.
3. **DHCP Request**:
   * The device then sends a **DHCP Request** message, confirming that it would like to accept the offered IP address and other configuration settings.
4. **DHCP Acknowledgment**:
   * The DHCP server sends a **DHCP Acknowledgment** message to the device, finalizing the assignment. The device is now configured with the provided IP address and other network settings.

### Advantages of DHCP:

1. **Convenience**:
   * DHCP simplifies network management by automating the process of IP address assignment. Users and administrators don’t need to manually configure each device's IP address.
2. **Efficient IP Address Allocation**:
   * DHCP ensures that IP addresses are efficiently assigned and reused. When a device disconnects, its IP address is returned to the DHCP pool and can be reassigned to another device.
3. **Consistency**:
   * Since DHCP assigns IP addresses dynamically, it ensures that devices are consistently configured with the correct settings (e.g., gateway, DNS servers).
4. **Reduced Risk of Conflicts**:
   * By automatically managing IP addresses, DHCP reduces the risk of **IP address conflicts**, which can occur if two devices are manually assigned the same IP address.

### DHCP Lease Process:

* When a device receives its IP address from the DHCP server, the lease time determines how long the device can use that address.
* Before the lease expires, the device may request a **renewal** for the same IP address to keep using it.
* If the device doesn't renew the lease or disconnects, the DHCP server will eventually return the IP address to the pool to be reassigned to another device.

### Example:

Let’s say you connect your laptop to a Wi-Fi network in your home:

1. **Your laptop** sends a **DHCP Discover** message asking for an IP address.
2. **The router (DHCP server)** responds with a **DHCP Offer** that includes an IP address (e.g., 192.168.1.10), subnet mask, and other details.
3. **Your laptop** sends a **DHCP Request** to accept the offered settings.
4. **The router** sends a **DHCP Acknowledgment** message, confirming that the laptop is now assigned the IP address 192.168.1.10.

Now your laptop can communicate with other devices on the network and access the internet (if configured correctly).

### Summary:

* **DHCP** automatically assigns IP addresses and network configuration settings to devices on a network.
* It simplifies network management, reduces the likelihood of errors, and ensures devices can communicate without manual configuration.
* The process involves discovery, offer, request, and acknowledgment steps between devices and the DHCP server.

DHCP is commonly used in home networks, corporate networks, and any environment with multiple devices that need an efficient way to manage network addresses.

What is DNS - Domain Name System

**DNS** stands for **Domain Name System**, and it is a critical component of how the internet works. Essentially, DNS is like the **phonebook of the internet**. It translates human-readable domain names (like www.example.com) into IP addresses (like 192.168.1.1), which are required for computers and devices to locate each other on the network.

### How DNS Works:

When you type a web address into your browser (e.g., www.google.com), the DNS system helps translate that domain name into an IP address that your device can use to connect to the server hosting the website. This process involves several steps:

1. **User Enters a Domain Name**:
   * When you enter a domain name into your browser, such as www.example.com, the browser needs to find out the corresponding IP address to send a request to the correct web server.
2. **DNS Resolver Query**:
   * The browser sends a request to a **DNS resolver** (usually provided by your ISP or a public DNS provider like Google or Cloudflare). The resolver's job is to look up the IP address for the domain name.
3. **Recursive Query**:
   * The DNS resolver may not have the IP address cached (stored from previous requests). If it doesn't, it performs a **recursive query** by asking other DNS servers, starting with the **root DNS server**. These servers help point the resolver in the right direction.
4. **Root DNS Server**:
   * The **root DNS server** is the first point of contact in the DNS hierarchy. It doesn’t store the actual IP address of the domain but provides information on where to find the **Top-Level Domain (TLD) servers** (e.g., .com, .org, .net).
5. **TLD DNS Server**:
   * Once the resolver knows the TLD server for .com (or whichever TLD the domain belongs to), it asks the **TLD DNS server** for the domain. The TLD server directs the resolver to the **authoritative DNS server** for that specific domain.
6. **Authoritative DNS Server**:
   * The **authoritative DNS server** holds the actual IP address for the domain (e.g., www.example.com → 93.184.216.34).
   * The authoritative server replies with the IP address back to the DNS resolver.
7. **Final Resolution**:
   * The resolver sends the IP address back to your browser, which can now contact the web server using that address to load the website.

### DNS Structure:

DNS operates hierarchically, with several types of servers at different levels:

1. **Root DNS Servers**:
   * These are the highest level in the DNS hierarchy and provide information about TLD servers (like .com, .org, .edu).
2. **TLD DNS Servers**:
   * These servers manage domains that end in specific TLDs. For example, the .com TLD server handles domains like google.com, example.com, etc.
3. **Authoritative DNS Servers**:
   * These servers contain the actual records for the domain name, such as A records (for IPv4 addresses), AAAA records (for IPv6 addresses), MX records (for email servers), and more.

### Types of DNS Records:

1. **A Record**:
   * **A (Address) record** maps a domain name to an **IPv4 address**. For example, www.example.com → 93.184.216.34.
2. **AAAA Record**:
   * Maps a domain to an **IPv6 address**. For example, www.example.com → 2001:0db8:85a3:0000:0000:8a2e:0370:7334.
3. **MX Record**:
   * **MX (Mail Exchange) record** specifies the mail servers responsible for receiving email for the domain (e.g., mail.example.com).
4. **CNAME Record**:
   * A **CNAME (Canonical Name) record** allows one domain to point to another domain. For example, blog.example.com can point to www.example.com.
5. **NS Record**:
   * **NS (Name Server) record** specifies the DNS servers that are authoritative for a particular domain.
6. **TXT Record**:
   * **TXT records** store arbitrary text, often used for things like domain verification or email security (e.g., SPF records to fight spam).

### Why DNS is Important:

* **Ease of Use**: Without DNS, we would need to remember numeric IP addresses for every website we visit, which is impractical. DNS simplifies this by letting us use human-readable domain names.
* **Efficiency**: DNS helps quickly resolve domain names into IP addresses, speeding up the process of connecting to websites and other services.
* **Distributed and Scalable**: DNS is a distributed system, meaning it's spread across multiple servers worldwide. This makes it scalable and resilient to failures, as no single server is responsible for all DNS requests.

### DNS Caching:

* **Caching** improves performance by temporarily storing DNS query results in the browser, operating system, or DNS resolver. Once a DNS resolver gets an IP address for a domain, it keeps that information cached for a specified period (known as **TTL**—Time To Live) to avoid repeatedly querying the authoritative server for the same domain.

### Example of a DNS Query:

1. **User enters** www.example.com in the browser.
2. The browser queries the **DNS resolver** (e.g., your ISP's DNS).
3. The resolver queries the **root DNS servers**, then the **TLD DNS servers**, and finally the **authoritative DNS server** for example.com.
4. The authoritative DNS server returns the IP address for www.example.com.
5. The browser connects to the IP address and loads the website.

### DNS and Security:

While DNS is essential for internet functionality, it can also be a target for malicious attacks:

1. **DNS Spoofing**: Attackers can manipulate DNS responses to direct users to fraudulent websites (phishing attacks).
2. **DDoS Attacks on DNS**: Distributed Denial of Service (DDoS) attacks can target DNS servers, making websites unavailable.

To mitigate these risks, security features like **DNSSEC** (DNS Security Extensions) and **DNS filtering** are used to prevent unauthorized DNS responses.

What is TCP - Transmition Control Protocol?

How TCP/IP addressing works?

TCP/IP (Transmission Control Protocol/Internet Protocol) addressing is fundamental for devices to communicate over a network, particularly the internet. Here's how it works in a simplified manner:

### 1. ****IP Addressing****:

Every device connected to a network (such as a computer, router, or phone) is assigned an IP address. This unique identifier is necessary for data to be routed correctly between devices. There are two main types of IP addressing:

* **IPv4**: The most common form, consisting of a 32-bit address, represented as four octets (each ranging from 0 to 255). For example: 192.168.1.1.
  + **Structure**: X.X.X.X where each X is an 8-bit number (ranging from 0 to 255).
* **IPv6**: The newer version, consisting of a 128-bit address, allowing for a much larger address space. It's written as eight groups of four hexadecimal digits separated by colons (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334).

### 2. ****Subnetting****:

An IP address is divided into two parts: the **Network** and **Host** portions. The subnet mask helps in this division and defines how much of the IP address is used for the network and how much is used for identifying the device (host).

* **Subnet Mask**: For example, in IPv4, a common subnet mask is 255.255.255.0, which means that the first three octets represent the network, and the last one represents the host within that network.
* **Subnetting Example**: If you have the address 192.168.1.25/24, the /24 means that the first 24 bits are for the network part (192.168.1), and the remaining 8 bits are used for host addresses within the 192.168.1.x range.

### 3. ****MAC Addresses****:

While IP addresses are used to route packets across networks (from one device to another, across the internet), the **MAC (Media Access Control) address** is used on a local level (within a local network). A MAC address is a hardware address burned into network interfaces and is used for communication on the Data Link layer (Layer 2 of the OSI model).

### 4. ****How TCP/IP Communication Works****:

* **Sending Data**: When a device wants to send data to another device, the application (like a web browser) sends the data through the transport layer (TCP) to ensure reliability.
* **IP Layer**: The data is then passed to the Internet Layer, where the destination IP address is used to route the data packet across the network. Routers use the destination IP address to forward the packet towards its destination.
* **Routing**: Routers examine the IP header and decide where to send the packet next based on the destination address. They use routing tables and the destination IP address to determine the next hop towards the destination network.
* **TCP**: At the receiving end, the TCP protocol ensures that the data is reassembled correctly. TCP provides error checking, retransmission of lost packets, and ensures that packets arrive in the correct order.

### 5. ****Port Numbers****:

In addition to IP addresses, each communication session over TCP/IP is identified by a **port number**. This is a 16-bit number that helps differentiate between different services or applications on the same device.

* **Well-known Ports**: Ports like 80 (HTTP), 443 (HTTPS), and 22 (SSH) are reserved for specific services.
* **Dynamic or Private Ports**: These are typically used for client-side applications and are assigned dynamically.

### 6. ****DNS (Domain Name System)****:

IP addresses are hard for humans to remember, so the **DNS** system was created to map human-readable domain names (like example.com) to IP addresses. When you type a domain name into a browser, a DNS query is sent to resolve the IP address of the server hosting that domain.

### 7. ****Address Resolution Protocol (ARP)****:

On a local network, when a device knows the destination IP address but not the MAC address, it uses ARP to find the corresponding MAC address. ARP sends a broadcast message requesting the MAC address of the device associated with the IP address, and the device responds with its MAC address.

### Summary:

* **IP Addresses** are used to uniquely identify devices on a network.
* **Subnetting** allows networks to be divided into smaller segments.
* **MAC Addresses** identify devices within a local network.
* **TCP/IP** ensures data is sent and received reliably across networks.
* **Port Numbers** help to differentiate between services on the same device.
* **DNS** converts domain names to IP addresses, and **ARP** helps map IP addresses to MAC addresses in local networks.

This layered approach ensures that data can travel across different devices, networks, and the internet efficiently.

**TCP** (Transmission Control Protocol) is one of the core protocols of the **Internet Protocol Suite** (often referred to as **TCP/IP**), which is the set of protocols that governs how data is transmitted over the internet and other networks.

TCP is a **connection-oriented protocol**, meaning it establishes a reliable, end-to-end communication channel between two devices before data can be transferred.

### Key Characteristics of TCP:

1. **Reliable Data Transmission**:
   * TCP ensures that data is delivered **accurately** and in the correct **order**. If any data is lost or corrupted during transmission, TCP will **retransmit** it.
2. **Connection-Oriented**:
   * TCP requires a **handshake** to establish a connection before any data is transmitted. This ensures that both sender and receiver are ready to communicate.
3. **Error Checking**:
   * TCP uses checksums to ensure the integrity of the data being transmitted. If a packet arrives with errors, it is discarded, and a request is made to resend it.
4. **Flow Control**:
   * TCP uses a mechanism called **flow control** to prevent the sender from overwhelming the receiver with too much data at once. This is managed through the **sliding window protocol**, where the receiver specifies how much data it is willing to accept at a time.
5. **Congestion Control**:
   * TCP also implements **congestion control** mechanisms to avoid network congestion. If too much data is being sent and the network is congested, TCP adjusts the rate of transmission to reduce the chance of packet loss.

### How TCP Works (The 3-Way Handshake):

Before TCP can start transmitting data, it must establish a connection between the sender and receiver using a **3-way handshake**:

1. **SYN** (Synchronize) - **Client to Server**:
   * The client sends a **SYN** message to the server to start the connection. This message indicates the client wants to establish a connection and specifies the initial sequence number.
2. **SYN-ACK** (Synchronize-Acknowledge) - **Server to Client**:
   * The server responds with a **SYN-ACK** message, acknowledging the receipt of the **SYN** request and providing its own initial sequence number.
3. **ACK** (Acknowledge) - **Client to Server**:
   * The client sends an **ACK** message, confirming the receipt of the server's **SYN-ACK**. At this point, the connection is established, and data transmission can begin.

### Data Transmission:

Once the connection is established, the data is sent in **segments**. Each TCP segment contains:

* **Sequence Number**: To ensure the data arrives in the correct order.
* **Acknowledgment Number**: To confirm the receipt of data.
* **Window Size**: Specifies the amount of data the receiver is willing to accept.
* **Checksum**: To check for data integrity.

The sender continuously monitors the data being sent, and if the receiver fails to acknowledge the receipt of a segment (e.g., due to packet loss), the sender retransmits the segment.

### Closing the Connection:

Once the data transfer is complete, TCP uses a **four-way handshake** to close the connection:

1. **FIN** (Finish) - **Client to Server**:
   * The client sends a **FIN** message to indicate it is done sending data.
2. **ACK** - **Server to Client**:
   * The server acknowledges the **FIN** with an **ACK** message.
3. **FIN** (Finish) - **Server to Client**:
   * The server sends a **FIN** message to the client to indicate it is done sending data.
4. **ACK** - **Client to Server**:
   * The client acknowledges the server's **FIN** with an **ACK** message, completing the connection termination.

### Advantages of TCP:

1. **Reliability**:
   * TCP ensures that data is delivered correctly, without errors, and in the right order.
2. **Error Recovery**:
   * If data is lost, TCP automatically requests that it be retransmitted.
3. **Flow Control**:
   * The sender and receiver communicate to manage the flow of data, preventing overload and ensuring that the receiver can handle the data.
4. **Congestion Control**:
   * TCP can adjust the transmission speed based on network conditions to prevent congestion.

### TCP vs. UDP (User Datagram Protocol):

While **TCP** is reliable and guarantees that data is delivered in order and without errors, there is another protocol called **UDP** that is different:

* **UDP** is **connectionless** and doesn't guarantee delivery or order. It's faster and has lower overhead than TCP, but it's not suitable for applications where reliability is crucial (like web browsing, file transfers, etc.).
* **TCP** is preferred for applications that require reliability (e.g., HTTP, FTP, email), while **UDP** is used for applications that prioritize speed over reliability (e.g., video streaming, online gaming, VoIP).

### Use Cases for TCP:

* **Web Browsing**: Websites use **HTTP** or **HTTPS**, both of which rely on TCP for reliable communication.
* **File Transfers**: Protocols like **FTP** and **SFTP** use TCP to ensure file integrity during transfer.
* **Email**: Protocols like **SMTP**, **POP3**, and **IMAP** use TCP for sending and receiving email messages.
* **Remote Access**: **SSH** (Secure Shell) and **Telnet** use TCP to securely connect to remote servers.

### Summary:

**TCP (Transmission Control Protocol)** is a connection-oriented protocol that ensures reliable, ordered, and error-free transmission of data between devices over a network. It achieves this through a process that includes:

* **Connection establishment** via a 3-way handshake,
* **Data transmission** with error detection, flow control, and congestion control,
* **Connection termination** via a 4-way handshake.

TCP is widely used for applications that require guaranteed delivery, such as web browsing, file transfer, and email.

What are IP adress Classes?

**IP address classes** are a way of dividing the **IP address space** (specifically for **IPv4**) into different ranges that are suitable for different types of networks. The classes help determine the size of a network and how the addresses are allocated for different purposes, such as private networks, public networks, or special-use networks.

### Overview of IPv4 Address Structure:

IPv4 addresses are **32 bits** long, usually written in **dotted decimal notation**, where each of the four octets is represented by a number between 0 and 255. For example: 192.168.1.1.

The 32-bit address can be divided into two parts:

1. **Network portion**: Identifies the network to which the device belongs.
2. **Host portion**: Identifies the specific device or host within that network.

### IPv4 Address Classes:

There are **five main IP address classes**, but only **A**, **B**, and **C** are used for regular network addressing. The other two classes (D and E) are reserved for special uses.

#### 1. ****Class A****:

* **Range**: 0.0.0.0 to 127.255.255.255
* **First octet**: 0 to 127 (The first bit of the first octet is always 0)
* **Network Size**: **16,777,216 addresses** (or **2^24** addresses)
* **Default Subnet Mask**: 255.0.0.0 or /8
* **Network Size**: Class A can support **16 million hosts** per network, which makes it suitable for very large networks (like large ISPs or multinational corporations).
* **Reserved for**: Class A addresses are primarily used for **very large organizations** or **publicly routable addresses**.

**Example**:

* **IP range**: 10.0.0.0 to 10.255.255.255
* **Private IP Range for Class A**: 10.0.0.0 to 10.255.255.255 (often used in large internal networks)

#### 2. ****Class B****:

* **Range**: 128.0.0.0 to 191.255.255.255
* **First octet**: 128 to 191 (The first two bits of the first octet are 10)
* **Network Size**: **65,536 addresses** (or **2^16** addresses)
* **Default Subnet Mask**: 255.255.0.0 or /16
* **Network Size**: Class B can support **65,536 hosts** per network, suitable for medium-to-large-sized networks.
* **Reserved for**: Class B is used for **medium-sized organizations**, universities, and other entities requiring a significant number of IP addresses.

**Example**:

* **IP range**: 172.16.0.0 to 172.31.255.255
* **Private IP Range for Class B**: 172.16.0.0 to 172.31.255.255

#### 3. ****Class C****:

* **Range**: 192.0.0.0 to 223.255.255.255
* **First octet**: 192 to 223 (The first three bits of the first octet are 110)
* **Network Size**: **256 addresses** (or **2^8** addresses)
* **Default Subnet Mask**: 255.255.255.0 or /24
* **Network Size**: Class C is typically used for **small networks** or **small businesses**, supporting up to **254 hosts** per network.
* **Reserved for**: Class C is often used for **small organizations** and for creating **smaller subnets**.

**Example**:

* **IP range**: 192.168.1.0 to 192.168.1.255
* **Private IP Range for Class C**: 192.168.0.0 to 192.168.255.255 (commonly used in home or small business networks)

#### 4. ****Class D (Multicast Addresses)****:

* **Range**: 224.0.0.0 to 239.255.255.255
* **First octet**: 224 to 239 (The first four bits of the first octet are 1110)
* **Purpose**: Class D addresses are **reserved for multicast** communication, where data is sent to multiple destinations simultaneously. These are not used for typical IP addressing but for efficient data distribution across networks.
* **Reserved for**: **Multicast communication** (e.g., streaming video or large-scale data delivery to multiple devices).

#### 5. ****Class E (Reserved for Experimental Use)****:

* **Range**: 240.0.0.0 to 255.255.255.255
* **First octet**: 240 to 255 (The first four bits of the first octet are 1111)
* **Purpose**: Class E addresses are **reserved for experimental purposes** and are not generally used in public networks.
* **Reserved for**: **Research and experimental use** by organizations like IETF (Internet Engineering Task Force).

What is CIDR notation?

**CIDR** (Classless Inter-Domain Routing) notation is a method for specifying IP addresses and their associated network prefix. CIDR notation is used to represent the range of IP addresses in a more flexible and efficient way than traditional **IP address classes** (A, B, C). It allows for more precise control over the division of networks and subnets.

### CIDR Notation Format:

CIDR notation combines the **IP address** with a **slash** (/) followed by the **prefix length**, which indicates how many bits of the IP address are dedicated to the network portion of the address. The rest of the bits are used for the host portion.

* **IP Address**: The usual IPv4 or IPv6 address (e.g., 192.168.1.0).
* **Prefix Length**: The number of bits used for the network portion (e.g., /24).

### Example of CIDR Notation:

* 192.168.1.0/24
  + This means that the first **24 bits** are used for the network portion, and the remaining **8 bits** (since an IPv4 address is 32 bits long) are used for hosts.
  + This represents a subnet where the IP addresses range from 192.168.1.0 to 192.168.1.255 (a total of **256 addresses**).

### Understanding the Prefix Length:

The prefix length tells you how many bits are reserved for the network address. Here’s how it works:

* **Network Portion**: The first **N bits** (determined by the prefix length) are used to identify the network.
* **Host Portion**: The remaining **(32 - N)** bits are used for individual hosts on that network.

### Common Prefix Lengths:

* **/8**: 255.0.0.0 (Class A network, 16,777,216 addresses)
* **/16**: 255.255.0.0 (Class B network, 65,536 addresses)
* **/24**: 255.255.255.0 (Class C network, 256 addresses)
* **/30**: 255.255.255.252 (Subnet for 4 addresses, commonly used for point-to-point links)
* **/32**: 255.255.255.255 (A single IP address, often used for specifying a host)

### How CIDR Works:

CIDR notation allows for more flexible subnetting by not being restricted to predefined network sizes (like the old classful IP addressing). For example:

* A **/24** subnet (255.255.255.0) allows for 256 IP addresses.
* A **/22** subnet (255.255.252.0) allows for 1,024 IP addresses.

### Subnetting with CIDR:

CIDR allows for subnetting into smaller or larger blocks as needed, without having to follow the fixed boundaries imposed by classful addressing. For example:

* With a **/24** prefix, the network is split into smaller subnets using a longer prefix (e.g., **/26**, **/27**), which can create smaller groups of IP addresses.
* Similarly, **/22** would cover a larger network than **/24**, and the network size would include more hosts.

### Advantages of CIDR:

1. **Efficient IP Address Allocation**:
   * CIDR reduces the waste of IP addresses compared to classful addressing. It allows for flexible allocation of IP ranges, helping optimize the use of available address space.
2. **Simplified Routing**:
   * CIDR allows for **route aggregation**, where multiple IP ranges can be combined into a single routing table entry. This reduces the number of entries in global routing tables and helps optimize routing.
3. **Better Utilization of IPv4 Address Space**:
   * Instead of using rigid Class A, B, or C address blocks, CIDR allows more granular control over the allocation of IP addresses, helping conserve IPv4 space.

### CIDR Example:

Let’s break down the example 192.168.1.0/24:

* **IP Address**: 192.168.1.0
* **Prefix Length**: /24
  + This means that the first **24 bits** of the address are the **network portion**.
  + The last **8 bits** (from the 32-bit address) are for **host addresses**.

This results in a subnet with the following:

* **Network Address**: 192.168.1.0 (this is the address used to identify the network itself).
* **First Usable Address**: 192.168.1.1 (this can be assigned to the first device in the network).
* **Last Usable Address**: 192.168.1.254 (this can be assigned to the last device in the network).
* **Broadcast Address**: 192.168.1.255 (this address is used to send messages to all devices on the network).

So, the **range** of usable IP addresses for hosts in this subnet is from **192.168.1.1** to **192.168.1.254**, totaling **254 usable addresses**.

### **Subnet Mask Chart**

| **Subnet Mask** | **CIDR** | **Binary Notation** | **Network Bits** | **Host Bits** | **Available Addresses** |
| --- | --- | --- | --- | --- | --- |
| 255.255.255.255 | /32 | 11111111.11111111.11111111.11111111 | 32 | 0 | 1 |
| 255.255.255.254 | /31 | 11111111.11111111.11111111.11111110 | 31 | 1 | 2 |
| 255.255.255.252 | /30 | 11111111.11111111.11111111.11111100 | 30 | 2 | 4 |
| 255.255.255.248 | /29 | 11111111.11111111.11111111.11111000 | 29 | 3 | 8 |
| 255.255.255.240 | /28 | 11111111.11111111.11111111.11110000 | 28 | 4 | 16 |
| 255.255.255.224 | /27 | 11111111.11111111.11111111.11100000 | 27 | 5 | 32 |
| 255.255.255.192 | /26 | 11111111.11111111.11111111.11000000 | 26 | 6 | 64 |
| 255.255.255.128 | /25 | 11111111.11111111.11111111.10000000 | 25 | 7 | 128 |
| 255.255.255.0 | /24 | 11111111.11111111.11111111.00000000 | 24 | 8 | 256 |
| 255.255.254.0 | /23 | 11111111.11111111.11111110.00000000 | 23 | 9 | 512 |
| 255.255.252.0 | /22 | 11111111.11111111.11111100.00000000 | 22 | 10 | 1024 |
| 255.255.248.0 | /21 | 11111111.11111111.11111000.00000000 | 21 | 11 | 2048 |
| 255.255.240.0 | /20 | 11111111.11111111.11110000.00000000 | 20 | 12 | 4096 |
| 255.255.224.0 | /19 | 11111111.11111111.11100000.00000000 | 19 | 13 | 8192 |
| 255.255.192.0 | /18 | 11111111.11111111.11000000.00000000 | 18 | 14 | 16384 |
| 255.255.128.0 | /17 | 11111111.11111111.10000000.00000000 | 17 | 15 | 32768 |
| 255.255.0.0 | /16 | 11111111.11111111.00000000.00000000 | 16 | 16 | 65536 |
| 255.254.0.0 | /15 | 11111111.11111110.00000000.00000000 | 15 | 17 | 131072 |
| 255.252.0.0 | /14 | 11111111.11111100.00000000.00000000 | 14 | 18 | 262144 |
| 255.248.0.0 | /13 | 11111111.11111000.00000000.00000000 | 13 | 19 | 524288 |
| 255.240.0.0 | /12 | 11111111.11110000.00000000.00000000 | 12 | 20 | 1048576 |
| 255.224.0.0 | /11 | 11111111.11100000.00000000.00000000 | 11 | 21 | 2097152 |
| 255.192.0.0 | /10 | 11111111.11000000.00000000.00000000 | 10 | 22 | 4194304 |
| 255.128.0.0 | /9 | 11111111.10000000.00000000.00000000 | 9 | 23 | 8388608 |
| 255.0.0.0 | /8 | 11111111.00000000.00000000.00000000 | 8 | 24 | 16777216 |

Subnetting Table

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Subnet | 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 |
| Host | 256 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| Subnet Mask | /24 | /25 | /26 | /27 | /28 | /29 | /30 | /31 | /32 |

Links with explanation:

Internet-CS50s Undersansing Tehnology

https://www.youtube.com/watch?v=n\_KghQP86Sw&t=1235s

1. Routing Tables | CCNA - Explained

https://www.youtube.com/watch?v=CGmTvukObOw

2. IP addresses and subnet masks

<https://www.youtube.com/watch?v=hbdT_Q9DM8w&t=185s>

3. Subnetting Explained: Networking Basics

<https://www.youtube.com/watch?v=s_Ntt6eTn94>

<https://www.youtube.com/watch?v=ecCuyq-Wprc&t=4s>

https://www.youtube.com/watch?v=ecCuyq-Wprc

Example for create a 3 separate networks or subnets for a Cofee shop, from a network ID:

Original networkID: 192.168.4.0/24

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Network ID | Subnet Mask | Host ID Range | #of Usable Host | Broadcast ID |
| 192.168.4.0 | /26 | 192.168.4.1-192.168.4.62 | 62 | 192.168.4.63 |
| 192.168.4.64 | /26 | 192.168.4.65-192.168.4.126 | 62 | 192.168.4.127 |
| 192.168.4.128 | /26 | 192.168.4.129-192.168.4.190 | 62 | 192.168.4.191 |
| 192.168.4.192 | /26 | 192.168.4.193-192.168.4.254 | 62 | 192.168.4.255 |

Resolving ID:

From the the table we are focusing jus on column with 4 Subnet(becouse we must create three subnet) : 4, 64, /26.

The first ID is the original: 192.168.4.0

Second ID is 192.168.4.64 (resulting from first is 0 + 64 from our Host column)

Third ID is 192.168.4.128 (resulting from second 64 + 64 = 128)

Four ID is 192.168.4.192 (resulting from 128 + 64 = 192)

For usable host ID we found out also from 64, so we have 64 available adress, just that the first adress 192.168.4.64 is used for host and the last is used for broadcasting.  
The number of usable host ID is 64 - 2, wich is 62.

A subnet’s Host ID range is any ID’s between its network ID and broadcast ID.

For the first subnet, 1-62 is between network ID 0 and broadcast ID 63

Second between 65-126, broadcast ID 127

Third between 129 -190, broadcast ID 191

Four between 193-254, broadcast ID 255

**Subnetting a subnet**

https://www.youtube.com/watch?v=aVTEZHC2wdA