How Does the Internet Work?

The internet is a vast global network that connects millions of devices, enabling them to communicate with each other. At its core, the internet relies on a system of interconnected networks that communicate using standard protocols, such as the Transmission Control Protocol (TCP) and the Internet Protocol (IP).

When you type a website address into your browser, your device sends a request across the network to a server hosting that website. This request is routed through a series of network devices (routers, switches, etc.) until it reaches the server, which responds by sending back the requested data (webpage, media, etc.).

Key components involved in the functioning of the internet include:

- **Devices** (computers, smartphones, routers)
- **IP addresses** to uniquely identify devices on the network
- **DNS** to resolve human-readable domain names into machine-readable IP addresses
- **Protocols** like TCP/IP and HTTP/HTTPS to enable communication and data exchange

---

DNS (Domain Name System) and How It Works

The **Domain Name System (DNS)** is like the internet's phonebook. It translates human-readable domain names (like www.example.com) into machine-readable IP addresses (like 192.168.1.1) that devices use to locate each other on the internet.

When you enter a website's URL in your browser:

1. The browser first checks if it already knows the IP address (cached).
2. If not, it sends a request to a DNS server to resolve the domain name.
3. The DNS server queries a series of other DNS servers until it finds the correct IP address.
4. The IP address is then returned to your browser, which uses it to connect to the web server hosting the website.

What is HTTP, HTTPS, and TCP?

- **HTTP (Hypertext Transfer Protocol)**: This is the foundational protocol for transferring data over the web. It is used to request and deliver web pages, images, and other resources. HTTP is stateless, meaning each request is independent of the previous one, and does not keep track of past interactions.
- **HTTPS (Hypertext Transfer Protocol Secure)**: HTTPS is the secure version of HTTP. It uses encryption via SSL/TLS (Secure Sockets Layer/Transport Layer Security) to ensure that data exchanged between the client and server is private and secure. HTTPS is crucial for protecting sensitive information, like login credentials or credit card numbers.
- **TCP (Transmission Control Protocol)**: TCP is a connection-oriented protocol that ensures reliable delivery of data between devices. It breaks data into smaller packets, ensures they reach their destination in the correct order, and requests retransmission if any packets are lost during transmission. This makes TCP ideal for applications like web browsing, email, and file transfers.

Core Networking Concepts

1. **IP Addressing**: Every device connected to the internet is assigned a unique identifier known as an IP address. There are two versions:
   - **IPv4** (e.g., 192.168.1.1) uses 32 bits, allowing for about 4.3 billion unique addresses.
   - **IPv6** (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334) uses 128 bits and can accommodate an almost infinite number of unique addresses.
2. **Subnets**: A subnet is a segment of a larger network that groups together devices with similar IP addresses. Subnetting helps optimize network performance and security by limiting broadcast traffic and dividing the network into smaller, manageable sections.
3. **DNS (Domain Name System)**: As described above, DNS translates domain names into IP addresses, allowing devices to locate each other on the internet.

4. **DHCP (Dynamic Host Configuration Protocol)**: DHCP automatically assigns IP addresses to devices on a network. Without DHCP, network administrators would have to manually configure each device's IP address, which can be time-consuming and error-prone.
5. **Routing**: Routers are responsible for directing traffic between different networks. They use routing tables and protocols (like BGP or OSPF) to determine the best path for data to travel.
6. **VLANs (Virtual Local Area Networks)**: VLANs are used to partition a single physical network into multiple logical networks. This is useful for improving security, managing traffic, and isolating different groups or departments within an organization.

REST API: Introduction and Details

A **REST API (Representational State Transfer API)** is a set of rules that allow applications to communicate with each other over HTTP. REST APIs are commonly used to enable web and mobile applications to interact with servers, databases, and other services.

Key principles of REST:

- **Stateless**: Each request from a client to a server is independent, meaning the server does not store any session information between requests.
- **Client-Server Architecture**: The client (e.g., a web browser) sends requests to the server (e.g., a web server), which processes those requests and sends back responses.
- **Uniform Interface**: RESTful APIs use standard HTTP methods (GET, POST, PUT, DELETE) and standard data formats like JSON or XML.
- **Resources**: In REST, the term "resource" refers to any object or data (e.g., user, product, order) that the API can manipulate.

Example of RESTful interactions:

- **GET /users**: Retrieve a list of users.
- **POST /users**: Create a new user.
- **PUT /users/1**: Update user with ID 1.
- **DELETE /users/1**: Delete user with ID 1.

REST API Authentication: Basic, OAuth, JWT, and SAML

Authentication is a critical aspect of REST APIs to ensure that only authorized users can access specific resources. Different methods of authentication include:

1. **Basic Authentication**: This method involves sending a username and password in the request header. It is simple but insecure because the credentials are transmitted in plaintext unless using HTTPS.
2. **OAuth**: OAuth is a more secure and flexible authentication framework. It allows third-party applications to access resources without exposing user credentials. OAuth works with access tokens that grant permissions to specific resources for a limited time. The user grants permission, and the third-party service exchanges an authorization code for an access token.
3. **JWT (JSON Web Token)**: JWT is an open standard used to securely transmit information between parties as a JSON object. It is typically used in token-based authentication. A JWT contains three parts: the header (algorithm), the payload (claims or data), and the signature (used for verification). JWT is stateless and does not require the server to store session information.
4. **SAML (Security Assertion Markup Language)**: SAML is an XML-based framework used for single sign-on (SSO). It allows users to authenticate once and gain access to multiple applications without needing to log in again. SAML is often used in enterprise environments for secure identity management.

Cookies

A **cookie** is a small piece of data that a web server sends to a user's browser, which stores it and sends it back on subsequent requests. Cookies are used for a variety of purposes, such as:

- **Session management**: Storing user authentication information (login state).
- **Personalization**: Remembering user preferences, themes, and settings.
- **Tracking and analytics**: Keeping track of user activity across websites for marketing and analysis.

Cookies can be **persistent** (stored for a set period) or **session-based** (deleted when the browser is closed). They are commonly used in conjunction with HTTPS for secure transmission of sensitive data.

There are also different types of cookies, such as:

- **First-party cookies**: Set by the website you're currently visiting.
- **Third-party cookies**: Set by external services like advertisers or analytics tools.