Web Basics II

🌐 Disconnected

Imagine waking up in the morning and not being able to:

- 📱 **Check your phone** for messages, emails, or the latest news.
- 🛒 **Shop online** for the things you need.
- 📞 **Video call** your family or friends living far away.
- 🎬 **Watch Netflix** or binge your favorite shows.
- 🔍 **Google answers** to your questions or learn something new.

Sounds unthinkable, right? This is how deeply the Internet has embedded itself into our daily lives.

⚡ The Internet

The Internet is a global network that connects billions of computers and devices, allowing them to communicate and share information. It enables the instant transfer of data, ideas, and services across the globe, revolutionizing how we live and interact with the world.

- 🚖 **Book a cab:** Apps like Uber or Ola connect your location to nearby drivers in real-time.
- 📡 **Stream a movie:** Platforms like Netflix send data from servers thousands of kilometers away to your device in seconds.
- 🌍 **Run a business:** Expand globally and reach customers worldwide without physical barriers.

🕸️ **Think of the Internet as a giant spider web:**

Each thread represents a network, and every node is a device — like your smartphone, laptop, or even smart home devices. This intricate web ensures seamless communication and information flow between millions of users every second.

How the Internet Works

The Internet is a vast network in which many computers and devices are interconnected, and they communicate with each other using a common set of protocols.

This means that when you access something on the internet, like viewing a website or a video, your device exchanges data with other devices or servers, all of which happens through a certain set of rules (protocols). This allows all devices to communicate easily and share data.

Overview of how it works:

1. Devices and Connections

Devices:

- **Computers, Smartphones, Tablets, and Servers**: These are the main types of devices that access the Internet. Any device that can connect to the Internet, including laptops, desktop computers, mobile phones, tablets, and even game consoles, are all considered devices.

Internet Service Providers (ISPs):

- **Role of ISPs**: ISPs are companies that provide access to the Internet They act as the middlemen, connect users to the global network through technologies like fiber optics, DSL, satellite, or cellular networks. Examples of ISPs : Jio, Airtel, etc.

Technologies Used by ISPs

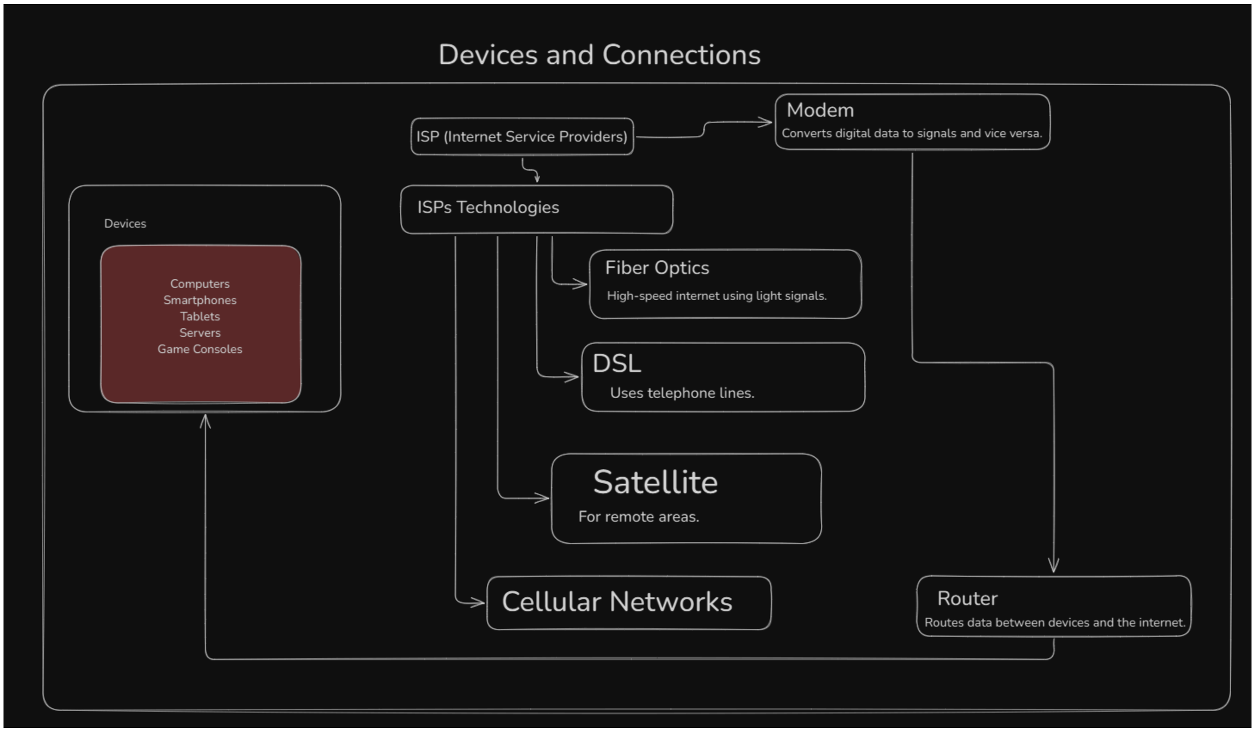
- **Fiber Optics**: High-speed internet is delivered through glass or plastic fibers that transmit data as light signals, providing faster and more reliable connections.

- **DSL (Digital Subscriber Line)**: This is a technology that uses telephone lines to transmit data. It’s slower than fiber optics but still commonly used in many homes.
- **Satellite**: In remote areas where cables can’t reach, satellite technology is used to deliver the internet. It uses signals sent to and from satellites orbiting Earth.
- **Cellular Networks**: ISPs also use cellular networks (like 4G, 5G) to provide mobile internet. This allows users to connect to the internet from their mobile phones or through mobile hotspots, providing mobility and flexibility.

Routers and Modems:

- Routers: The router is the device responsible for **directing** or **routing** the data between devices within your local network (like your computer, phone, and tablet) and the internet.
- Modem: The modem is the device that connects your home network to the ISP's network. It **converts** digital data from your devices into signals.

Together,
modems and **routers** form the backbone of your Internet connection, while **ISPs** provide access to the global network, and **devices** are the endpoints that you use to interact with the web.



2. Data Transmission

Packets

Data is divided into small chunks called packets. Each packet contains the sender's and receiver's IP addresses and the actual data being sent.

Packet Components:

- **Header**: Contains metadata like the sender's and receiver's IP addresses, protocol type (TCP or UDP), sequence number (for reordering), and checksum (for error detection).
- **Payload**: The **payload** is the actual **content** or **data** that you want to send (part of the original message or file).
- Footer: A **footer** is an optional part of the packet that is sometimes used for **error checking** and **validation**.

TCP/IP

TCP → Transmission Control Protocol

IP → Internet Protocol

TCP (Transmission Control Protocol) is an important protocol that makes data transmission reliable on the internet. Its job is to transfer data securely and accurately.

TCP and **IP** work together to ensure packets are correctly sent, routed, and reassembled.

- TCP is responsible for ensuring reliable and orderly transmission of data packets across the network.
- **Connection-Oriented**: TCP requires a connection to be established between the sender and receiver before data can be sent. This is done via a **three-way handshake.**
- **Packet Sequencing**: Each packet is assigned a **sequence number**. This allows the receiver to reorder packets if they arrive out of sequence, ensuring the data is reassembled in the correct order.
- **Error Checking**: TCP uses a **checksum** to verify that packets have been transmitted correctly. If a packet is corrupted during transmission, the receiver will discard it and request a retransmission from the sender.
- **Flow Control**: TCP uses a flow control mechanism. This comes into play when the sender is sending data too quickly, and the receiver needs time to process that data. This mechanism informs the sender when to stop sending data, preventing the receiver from being overloaded.
- **Congestion(Traffic) Control**: TCP uses congestion control when there is congestion on the network (e.g., when many devices are sending data simultaneously). It detects congestion and slows down the data transmission to ensure that the network's resources are properly managed.
- **Three-Way Handshake**:

To establish a connection, TCP uses a three-way handshake:

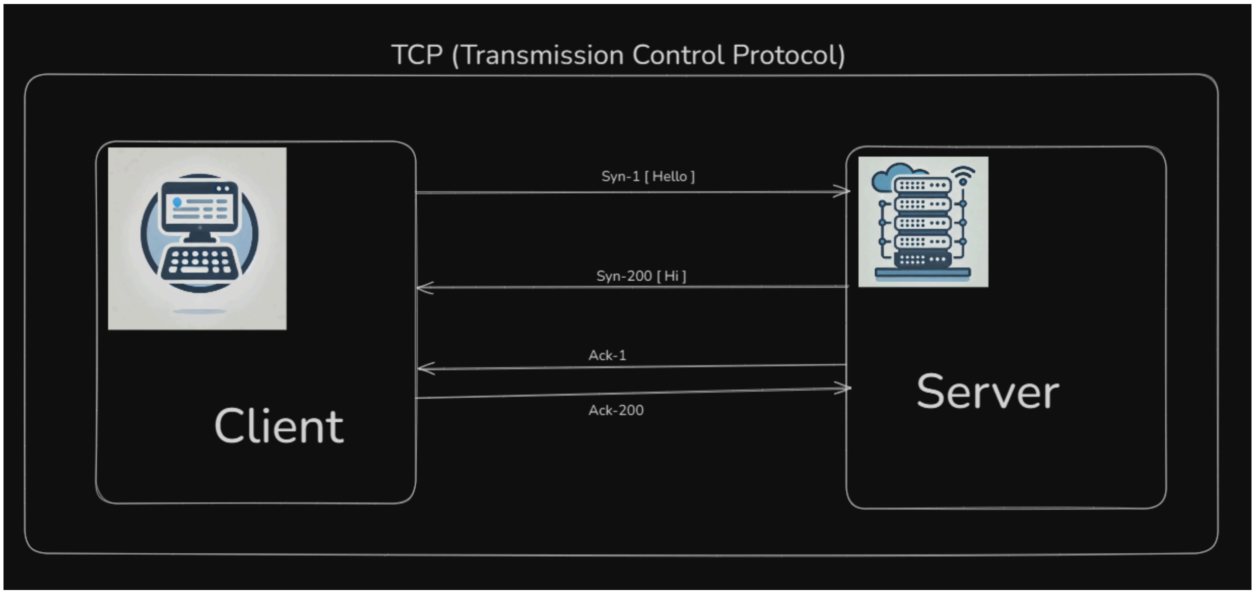
- **SYN**: The sender sends a synchronization (SYN) packet to the receiver, requesting a connection.
- **SYN-ACK**: The receiver responds with a synchronization acknowledgment (SYN-ACK) packet, confirming the request and agreeing to the connection.
- **ACK**: The sender sends an acknowledgment (ACK) packet back to the receiver, completing the connection setup

IP

- While TCP is responsible for reliable transmission, IP is responsible for routing packets between devices across networks.

How TCP and IP Work Together:

- **Layered Approach**: The two protocols work in layers, each with its own responsibilities. TCP operates at the **transport layer**, while IP operates at the **network layer**.
- **Collaboration**: When a sender wants to transmit data, TCP first breaks the data into packets, adds sequence numbers, and checksums for error handling. Then, IP adds the sender's and receiver's IP addresses, and the packet is sent to the network.



📦Real-World **Scenario**

Think of sending a **large package** (like a 🎁 gift box) to a friend:

- **Packets**: The large package is divided into smaller **boxes** (📦 data packets), each containing a part of the gift.
- **Header**: Each box has a **label** (📝 header) with the sender's and receiver's addresses (IP addresses), the box number (sequence number), and a tag for checking if the box is damaged (✅ checksum).
- **Payload**: The actual **gift** inside the box is the **payload** — the data you want to send.
- **Footer**: An optional **sticker** (🏷️) or seal that can be used for additional error checking or to verify the package's integrity.

🔌 TCP/IP:

- **TCP (Transmission Control Protocol)** ensures the **secure and reliable** delivery of data, just like a **delivery service** 📮 making sure the packages are delivered intact and in the correct order.
    - **Connection-Oriented**: Before sending the packages, the sender and receiver agree on the delivery process (like a handshake 🤝 between the postal service and the recipient).
    - **Packet Sequencing**: If the boxes arrive out of order, the receiver can reorder them using the sequence number 🔢.
    - **Error Checking**: If any box is damaged or lost, the receiver can ask for a replacement (📦🔄 retransmission).
    - **Flow Control**: If the sender is trying to send too many boxes at once, the system will slow things down to avoid overloading the receiver 🛑.
    - **Congestion Control**: If there's a traffic jam 🚦, the delivery service slows down the delivery to avoid network overload.
- **IP (Internet Protocol)**: Just like the postal system figures out the best route 🛣️ for the packages, **IP** ensures the packets are **routed correctly** through the network to the receiver's address.

🔄 How TCP and IP Work Together:

- **TCP** breaks the data into packets, ensures delivery, and handles retransmissions if necessary.
- **IP** routes the packets to the correct destination based on the address.

In summary, TCP ensures your data arrives intact and in order ✅, while IP makes sure it gets to the right place 📍.

3. IP Addresses and Domain Names

IP Addresses

IP Addresses → Internet Protocol Address

IP Address: The Unique Identifier for Every Device

- An **IP address (Internet Protocol Address)** is a unique string of numbers that identifies each device connected to a network.

Format:

- **IPv4**: The most commonly used version. It has 4 sets of numbers separated by dots, e.g., `192.168.1.1`. Each set ranges from 0 to 255.
- **IPv6**: A newer version designed to handle the growing number of devices. It uses a combination of letters and numbers, e.g., `2001:0db8:85a3:0000:0000:8a2e:0370:7334`.

Why It's Important:

- Every device—whether it's a computer, smartphone, or server—needs an IP address to send and receive data.
- Without IP addresses, data wouldn't know where to go, just like you wouldn't receive mail without a home address.

Dynamic vs. Static IP:

- **Dynamic IP**: Assigned temporarily by an Internet Service Provider (ISP) and can change over time.

- **Static IP**: Fixed and does not change, often used for servers or specific devices.

Scenario:

📦 **Imagine you want to send a parcel**:

🏠 **IP Address = Home Address**

- Just like every house has a unique address, every device has a unique **IP address**.
- It ensures that data is delivered to the correct destination.

🔄 **Dynamic vs. Static IP**

- 🔄 **Dynamic IP**: Like a 🏚️ rental house address that changes occasionally (temporary).
- 📌 **Static IP**: Like a 🏡 permanent house address that never changes.

📨 **Data = Parcel**

- When you send data, both the sender 📤 and receiver 📥 need **IP addresses**.
- Just like sending a parcel requires both the sender's and receiver's addresses.

🚫 **Without an IP Address**

- ❌ Data could be delivered to the wrong place or might not reach at all.

Domain Name System (DNS)

The **Domain Name System (DNS)** is a system that translates human-readable domain names (like `google.com`) into machine-readable IP addresses.

Example:

- When you type `google.com` into your browser, the DNS converts it into an IP address like `142.250.190.78`, which your computer uses to connect to Google's servers

Why DNS Exists:

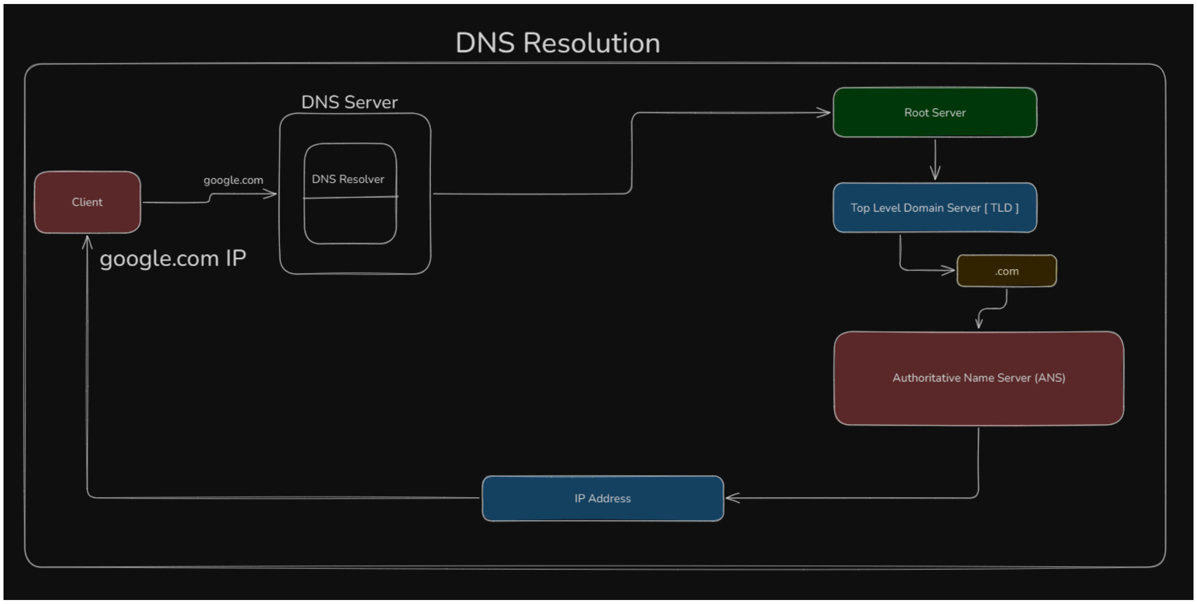
- IP addresses are difficult for humans to remember, especially IPv6 addresses.

- DNS allows us to use easy-to-remember domain names instead of numerical IP addresses.

How DNS Works:

When you enter a domain name in your browser, the following steps happen:

1. **Local Cache Check**:
   - Your device or browser first checks if it already knows the IP address of the domain. If found, it skips further steps.
2. **Query to Recursive DNS Resolver**:
   - If not found in the local cache, the query goes to a **DNS resolver** (usually managed by your ISP or a public DNS service like Google DNS).
3. **Contact Root DNS Server**:
   - The resolver contacts a **root DNS server**, which directs it to the correct **Top-Level Domain (TLD) server**.
     - Example: For `google.com`, the root server points to the `.com` TLD server.
4. **TLD Server Query**:
   - The `.com` TLD server directs the resolver to the specific **authoritative name server** for `google.com`.
5. **Authoritative Name Server**:
   - The authoritative server for `google.com` provides the actual IP address (e.g., `142.250.190.78`).
6. **Connect to the IP Address**:
   - The resolver sends the IP address back to your browser, which then connects to the server at that IP.



Real-World Scenario

Imagine you want to send a letter to your friend who lives in another city, but you don’t know their address, only their name. Here are the steps you would follow to find their address:

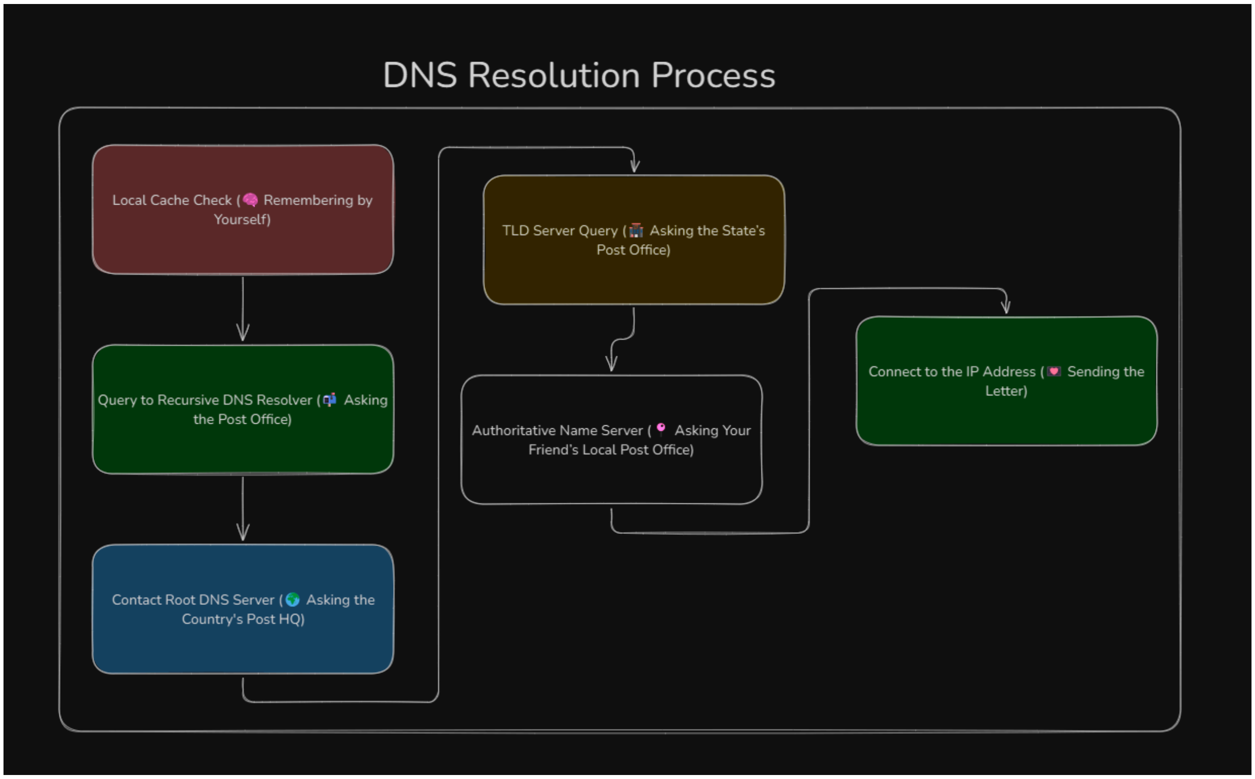
1. **Local Cache Check (🧠 Remembering by Yourself)**
    - **Scenario:** You first think if you already remember your friend’s address.
    - **DNS Equivalent:** Your device/browser first checks its memory (cache) to see if the domain’s IP address is already saved.
    - **Example:** If you’ve recently visited google.com, your browser’s cache saves its IP address, so it skips the further steps.
2. **Query to Recursive DNS Resolver (📫 Asking the Post Office)**
    - **Scenario:** If you don’t remember the address, you go to your local post office and ask them.
    - **DNS Equivalent:** Your ISP’s DNS resolver (or a public DNS like Google DNS) receives your query for google.com.
3. **Contact Root DNS Server (🌍 Asking the Country's Post HQ)**
    - **Scenario:** The local post office doesn’t know the address, so they direct you to the national post HQ.

   - **DNS Equivalent:** The resolver contacts the root DNS server, which has information about all Top-Level Domains (TLDs).
   - **Example:** The root server tells the resolver to ask the .com TLD server for details about google.com.

4. **TLD Server Query (🏤 Asking the State's Post Office)**
   - **Scenario:** The national HQ directs you to the state post office that handles addresses for your friend's area.
   - **DNS Equivalent:** The .com TLD server directs the resolver to the authoritative name server for google.com.
5. **Authoritative Name Server (📍 Asking Your Friend's Local Post Office)**
   - **Scenario:** The state office finally gives you the exact address of your friend's house.
   - **DNS Equivalent:** The authoritative name server provides the actual IP address of google.com (e.g., 142.250.190.78).
6. **Connect to the IP Address (💌 Sending the Letter)**
   - **Scenario:** Now that you have your friend's address, you send the letter, and it reaches their house.
   - **DNS Equivalent:** Your browser uses the provided IP address to connect to the google.com server.

Without DNS?

Imagine if you had to remember every friend's address instead of just their name. DNS simplifies the process, just like knowing someone's name and letting the system automatically find the address for you! 😊


DNS Resolution Process
Local Cache Check (🧠 Remembering by Yourself)
Query to Recursive DNS Resolver (📬 Asking the Post Office)
Contact Root DNS Server (🌍 Asking the Country's Post HQ)
TLD Server Query (🏣 Asking the State's Post Office)
Authoritative Name Server (📍 Asking Your Friend's Local Post Office)
Connect to the IP Address (💌 Sending the Letter)

4. 🌐 Routing Data

When data is transmitted over the internet, it doesn't travel in a straight line. Instead, it moves through various **routers** and **switches**:

1. **Router** 🛣️:
    - Think of a router as a **traffic director** for data.
    - It **directs packets** between different networks (e.g., from your home network to your ISP, then to a global network).
    - Routers ensure that data moves closer to its destination by finding the next best hop.
2. **Switch** 🔄:
    - A switch operates **within a single network** (e.g., within your home or office).

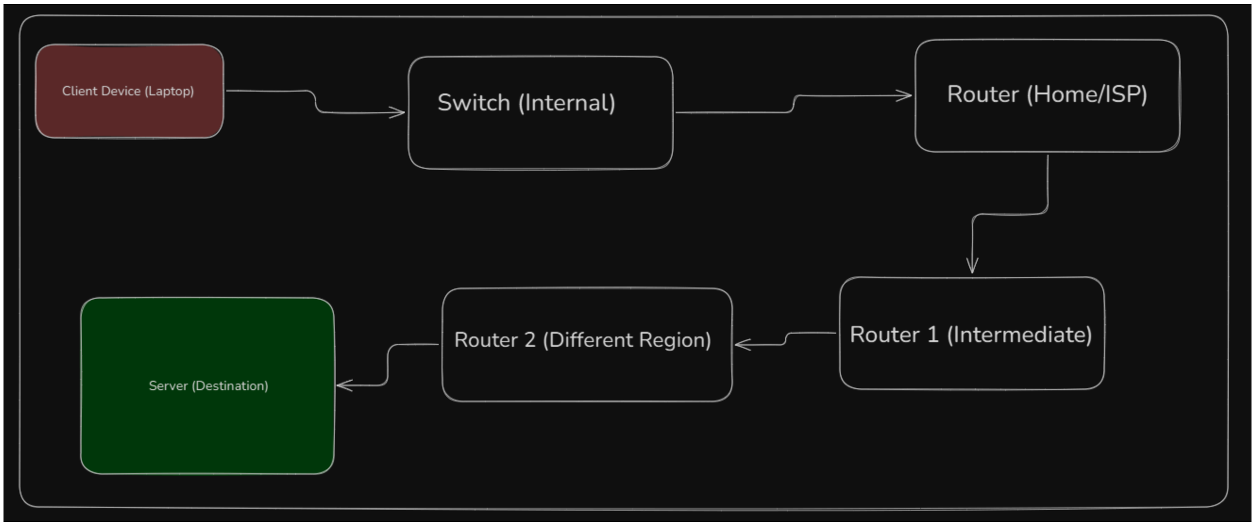
- It **routes packets internally** between devices like your laptop, printer, and phone.

🌍 How Packets Travel:

- **Fastest Path**: Data packets always take the **fastest available route**, which might not be the most direct one.
- **Multiple Stops (Hops)**: Packets may pass through several intermediate devices (routers) spread across the globe before reaching their destination.
- Example: Sending an email from India to the US might involve hops through routers in Europe, Asia, and North America.

🚦 Real-World Scenario:

- **Router**: Like highway exits directing traffic to different cities 🛣️.
- **Switch**: Like local streets directing cars to houses 🏠.
- **Hops**: The stops (checkpoints) a vehicle makes before reaching the destination.



5 Secure Connections

HTTP/HTTPS

HTTP → Hypertext Transfer Protocol

Protocol → Set of rules [Protocol used to transfer Hypertext]

Hypertext → Text docs → Hyperlink [These hyperlinks connect one document to other documents, resources, or information.]

- It's the protocol used by web browsers to request and load web pages.
- When you enter a URL in your browser (like `http://example.com`), your browser uses HTTP to communicate with the server that hosts the web page.
- HTTP defines how messages are formatted and transmitted over the web, as well as how web servers and browsers should respond to various commands.

HTTPS → Hypertext Transfer Protocol Secure

- It is a more secure version of HTTP.
- **HTTPS** encrypts the data exchanged between your browser and the server, making it secret.
- **Encryption** happens using SSL/TLS (Secure Sockets Layer/Transport Layer Security), which secures the data.
- When you share **sensitive information** (like passwords or credit card numbers), you'll notice the website's URL starts with **"HTTPS"** (e.g., `https://example.com`).

In simple terms: HTTPS means your data is **safe** and cannot be seen by anyone in between.

TLS

🔒 TLS → Transport Layer Security

TLS (Transport Layer Security) is a **cryptographic protocol** that ensures secure communication over a network, especially the internet. It is the successor to **SSL (Secure Sockets Layer)** and is widely used for securing connections between web browsers and servers.

Key Functions of TLS:

1. **Encryption** 🔐:

   - TLS encrypts the data exchanged between a client (like your browser) and a server, ensuring that anyone trying to intercept the communication cannot read the information.

2. **Authentication 🛡️**:
   - TLS ensures that the server you're communicating with is actually the intended server, preventing man-in-the-middle attacks.
   - It uses **digital certificates** issued by trusted Certificate Authorities (CAs) to authenticate the server's identity.

3. **Integrity ✅**:
   - TLS ensures the data hasn't been tampered with during transmission. It uses **hash functions** and **checksums** to verify data integrity.

How TLS Works:

1. **Handshake 🤝**:
   - When a client (e.g., web browser) connects to a server (e.g., website), they perform a **handshake** to establish a secure connection.
   - During the handshake, they agree on encryption algorithms, exchange keys, and authenticate each other.

2. **Session Key Creation 🔑**:
   - After the handshake, the client and server generate a **session key**, which is used to encrypt the actual data exchanged during the session.

3. **Secure Communication 📡**:
   - Once the secure connection is established, the data is transmitted over the encrypted channel, ensuring confidentiality, integrity, and authenticity.

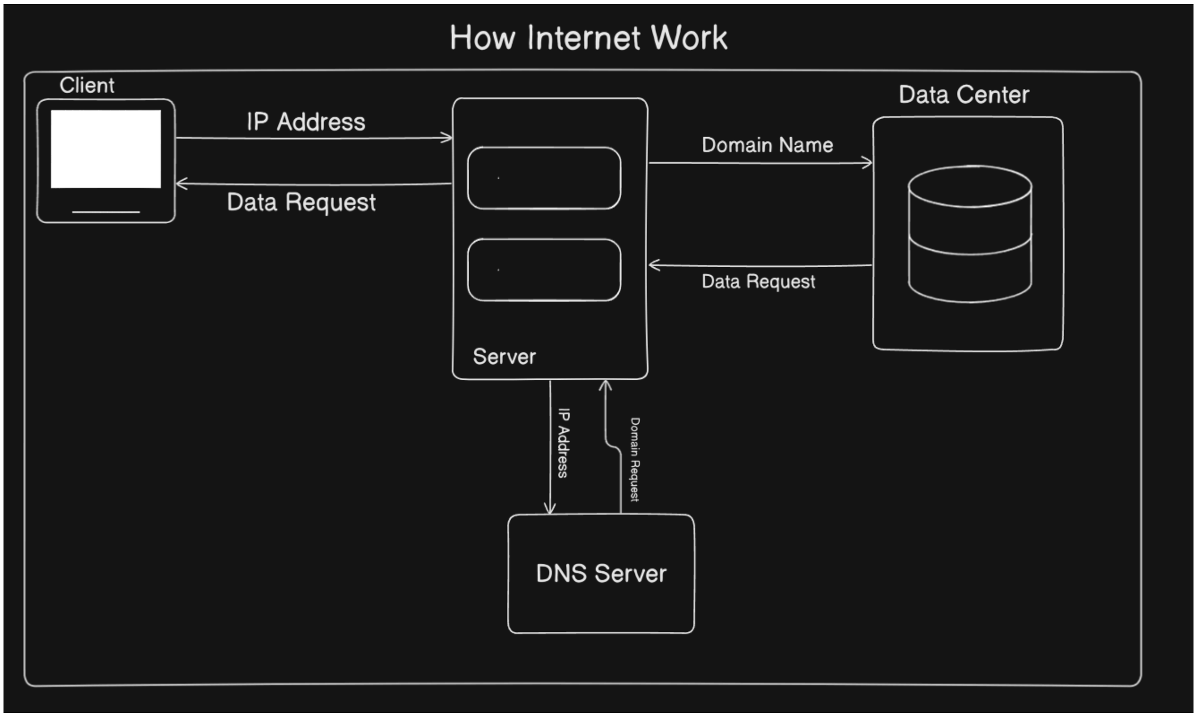
🌐 Real-World Scenario:

Online banking

1. **You Visit Your Bank's Website 🏦**:
   - You open your browser and type in your bank's website address (e.g., `https://mybank.com`).

- The "https" indicates that TLS is being used to secure your connection.

2. **TLS Handshake Begins** 🤝:
   - When your browser connects to the bank's server, a **TLS handshake** happens in the background.
   - The bank's server sends a **digital certificate** to your browser to prove its identity (authentication).
   - Your browser checks the certificate against trusted **Certificate Authorities (CAs)** to make sure it’s legitimate.
   - Both your browser and the bank’s server agree on the encryption algorithms they’ll use and exchange **session keys** to ensure the communication remains secure.
3. **Encrypted Data Transmission** 🔐:
   - Once the handshake is complete, a **secure encrypted channel** is established.
   - Now, when you log in with your account details (username and password), your sensitive information is **encrypted** using the session key.
   - Even if a hacker tries to intercept the data while it’s being sent through the internet, the encryption ensures they can’t read the sensitive data.
4. **Data Integrity and Protection** ✅:
   - As you perform transactions (e.g., transferring money), TLS ensures that none of the data has been tampered with.
   - If anyone tries to alter the transaction data, it would break the encryption and be detected.
5. **Session Ends Securely** 🔒:
   - After you’ve completed your banking tasks, you log out, and the TLS connection is closed securely.



Conclusion

The Internet is a transformative technology that has turned the world into a connected network, revolutionizing every aspect of our daily lives. Whether it's video streaming, online shopping, or real-time communication, the Internet has enabled seamless interaction and information sharing.

Its functioning is based on a complex yet well-coordinated system that efficiently transfers data through devices, ISPs, routers, packets, TCP/IP protocols, and DNS. Your points have explained the working of the Internet in a beginner-friendly manner, making it clear and impactful for any reader.

Key Highlights:

- The role of **devices** and **ISPs** is well explained, covering technologies like fiber optics, DSL, and cellular networks.
- The process of **reliable and secure data transfer** using data packets and TCP/IP protocols is crystal clear.
- Practical examples of **DNS** and **IP addressing** have simplified complex topics, making them relatable.

- The explanation of **routing** and **secure connections (HTTP/HTTPS, TLS)** highlights the core importance of Internet safety and reliability.

In today's world, the Internet is not just a tool but a necessity, forming the foundation for education, business, entertainment, and communication.