*Tài liệu:*

[*https://developer.mozilla.org/en-US/docs/Learn\_web\_development/Howto/Web\_mechanics/How\_does\_the\_Internet\_work*](https://developer.mozilla.org/en-US/docs/Learn_web_development/Howto/Web_mechanics/How_does_the_Internet_work)

[*https://www.spiceworks.com/tech/networking/articles/what-is-the-internet/*](https://www.spiceworks.com/tech/networking/articles/what-is-the-internet/)

[*https://vi.wikipedia.org/wiki/Internet*](https://vi.wikipedia.org/wiki/Internet)

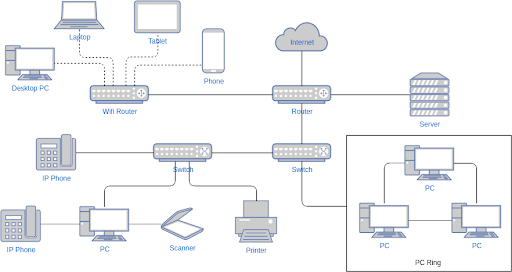
[*https://computer.howstuffworks.com/internet/basics/internet.htm*](https://computer.howstuffworks.com/internet/basics/internet.htm)

[*https://www.highspeedinternet.com/resources/how-the-internet-works*](https://www.highspeedinternet.com/resources/how-the-internet-works)

[*https://www.geeksforgeeks.org/how-does-the-internet-work/*](https://www.geeksforgeeks.org/how-does-the-internet-work/)

[*https://www.cloudflare.com/learning/network-layer/how-does-the-internet-work/*](https://www.cloudflare.com/learning/network-layer/how-does-the-internet-work/)

Qua bài viết này ta sẽ có 1 cái nhìn tổng quan về internet và cách hoạt động đằng sau 1 Website



# **I. Why the Internet?**

**Why was the Internet created?**

*Vào những năm 1960, trong thời kỳ Chiến tranh Lạnh, Bộ Quốc phòng Mỹ cần một hệ thống liên lạc bền vững, có khả năng hoạt động ngay cả khi một phần mạng bị phá hủy do chiến tranh hoặc sự cố. Điều này đã dẫn đến sự ra đời của ARPANET, tiền thân của Internet.*

Ngoài ra, việc giao tiếp qua các phương tiện truyền thống lúc bấy giờ thường mất nhiều thời gian, gây cản trở cho quá trình hợp tác giữa các nhà khoa học và nhà nghiên cứu. Vì vậy, họ cần một cách hiệu quả hơn để chia sẻ tài nguyên tính toán và dữ liệu qua khoảng cách xa. Việc kết nối các máy tính đã giải quyết vấn đề này, cho phép trao đổi thông tin mà không cần gặp trực tiếp.

# **II. What the Internet?**

1. **Định nghĩa**

Internet là một hệ thống **mạng (**network**)** lưới toàn cầu liên kết nhiều máy tính và thiết bị điện tử khác nhau, cho phép trao đổi thông tin giữa người dùng trên khắp thế giới.

**Network** là tập hợp các thiết bị (máy tính, máy chủ, thiết bị mạng) được kết nối với nhau để trao đổi dữ liệu.

* Internet là một hệ thống các mạng lưới toàn cầu được kết nối với nhau, bao gồm cả các mạng (**network)** riêng và mạng (**network)** công cộng, tạo thành một hệ sinh thái mạng phức tạp.

Một mạng có thể tồn tại độc lập mà không cần kết nối với Internet. Nhưng để truy cập Internet, thiết bị cần được kết nối vào một mạng (thường thông qua nhà cung cấp dịch vụ Internet - ISP) vd Viettel, VNPT, …

1. **Thành phần**
   1. **Phần cứng là nền tảng vật lý bao gồm:**

**Thiết bị kết nối**: Laptop, điện thoại (client), máy chủ (server), router, modem, switch,...

**Hạ tầng mạng**: Dây cáp quang, cáp đồng, vệ tinh, sóng di động, trạm phát sóng,...

**Hệ thống máy chủ**: Máy chủ DNS (Domain Name System), máy chủ email, máy chủ web, máy chủ ứng dụng,...

* 1. **Cơ sở hạ tầng, nhà cung cập dịch vụ ISP**

**Cơ sở hạ tầng mạng**: Mạng cục bộ LAN, mạng diện rộng WAN, mạng đô thị MAN, cáp quang, trạm phát sóng,...

**Nhà cung cấp dịch vụ Internet (ISP)**: Các công ty như Viettel, FPT, VNPT, hoặc các nhà cung cấp toàn cầu như AT&T, Verizon, cung cấp dịch vụ kết nối Internet cho người dùng cuối và tổ chức.

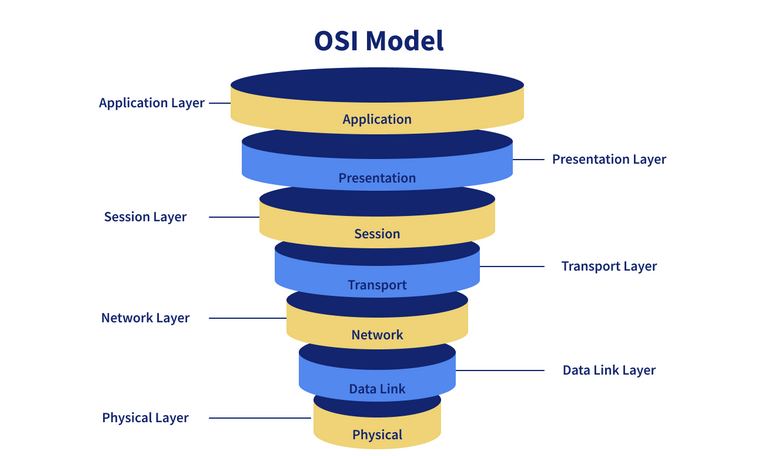
**Vai trò của ISP**: ISP chịu trách nhiệm kết nối thiết bị cá nhân và mạng nhỏ đến Internet, đồng thời quản lý băng thông và chất lượng kết nối.

* 1. **Giao thức Internet**

TCP/IP (Transmission Control Protocol/Internet Protocol): Là giao thức cốt lõi của Internet, định nghĩa cách dữ liệu được gửi và nhận qua mạng. TCP đảm bảo tính toàn vẹn của dữ liệu, trong khi IP chịu trách nhiệm định tuyến các gói dữ liệu đến đích..

1. **Các mô hình mạng phổ biến**
   1. **Mô hình OSI (Open Systems Interconnection Model)**

Là một mô hình tham khảo dùng để hiểu và thiết kế các hệ thống mạng. Nó chia quá trình truyền thông mạng thành 7 lớp (layers), mỗi lớp có nhiệm vụ riêng biệt:



**Lớp 1: Physical Layer (Lớp vật lý)**

* Đảm bảo truyền tải dữ liệu qua các phương tiện vật lý (dây cáp, sóng radio,...).
* Các thiết bị: Cáp quang, cáp đồng, switch, hub, modem.

**Lớp 2: Data Link Layer (Lớp liên kết dữ liệu)**

* + Quản lý việc truyền tải dữ liệu giữa các thiết bị trong cùng một mạng, xử lý lỗi truyền tải.
  + Các giao thức: Ethernet, PPP (Point-to-Point Protocol).

**Lớp 3: Network Layer (Lớp mạng)**

* Định tuyến dữ liệu giữa các mạng khác nhau, sử dụng địa chỉ IP để xác định đường đi.
* Giao thức: IP, ICMP (ping).

**Lớp 4: Transport Layer (Lớp vận chuyển)**

* Đảm bảo dữ liệu được truyền tải đáng tin cậy giữa các ứng dụng (ví dụ: thông qua TCP hoặc UDP).
* Giao thức: TCP, UDP.

**Lớp 5: Session Layer (Lớp phiên)**

* Quản lý các phiên làm việc, đảm bảo các kết nối và sự giao tiếp giữa các ứng dụng diễn ra liên tục.
* Giao thức: NetBIOS, RPC.

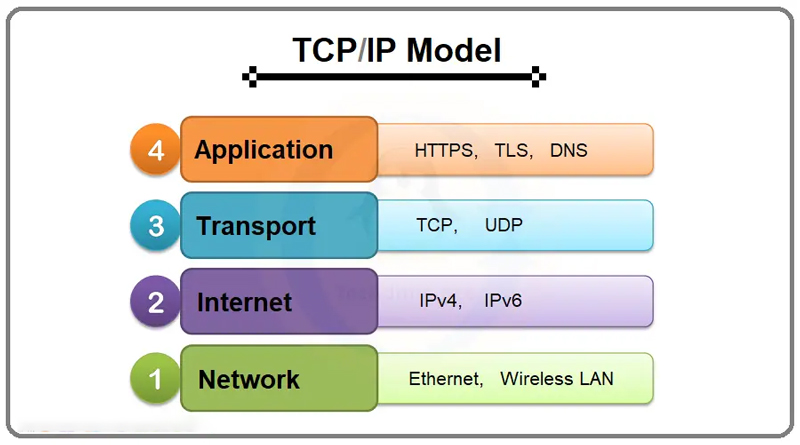
**Lớp 6: Presentation Layer (Lớp trình bày)**

* Đảm bảo dữ liệu được mã hóa và giải mã đúng định dạng mà người nhận hiểu được (ví dụ: mã hóa SSL).
* Giao thức: SSL/TLS, JPEG, GIF.

**Lớp 7: Application Layer (Lớp ứng dụng)**

* Là lớp mà người dùng cuối và các ứng dụng tương tác trực tiếp, cung cấp các dịch vụ mạng như duyệt web, gửi email.
* Giao thức: HTTP, FTP, SMTP, DNS.
  1. **Mô hình TCP/IP (Transmission Control Protocol/Internet Protocol)**

Là mô hình thực tế được sử dụng rộng rãi trong Internet và các mạng máy tính hiện nay. Nó đơn giản hơn mô hình OSI, chỉ có 4 lớp:



**Lớp 1: Link Layer (Lớp liên kết)**

* Tương tự như lớp Data Link trong OSI, bao gồm tất cả các công nghệ và giao thức dùng để truyền tải dữ liệu qua phương tiện vật lý.
* Các giao thức: Ethernet, PPP.

**Lớp 2: Internet Layer (Lớp mạng)**

* Tương tự lớp Network trong OSI, định tuyến và chuyển tiếp các gói dữ liệu qua các mạng.
* Giao thức: IP, ICMP.

**Lớp 3: Transport Layer (Lớp vận chuyển)**

* Đảm bảo việc truyền tải dữ liệu an toàn giữa các ứng dụng.
* Giao thức: TCP, UDP.

**Lớp 4: Application Layer (Lớp ứng dụng)**

* Tương tự lớp Application trong OSI, nơi các ứng dụng mạng như web và email hoạt động.
* Giao thức: HTTP, FTP, SMTP, DNS.

#### **3.2.1 TCP (Transmission Control Protocol)**

* TCP chịu trách nhiệm chia nhỏ dữ liệu thành các gói tin, gửi đi, kiểm tra và đảm bảo dữ liệu được truyền đầy đủ và chính xác đến đích.
* TCP thiết lập một **kết nối** giữa hai thiết bị trước khi truyền dữ liệu (gọi là quá trình **bắt tay ba bước - 3-way handshake**).
* TCP đảm bảo rằng các gói tin bị mất trong quá trình truyền sẽ được gửi lại.

*Làm sao để TCP biết rằng gói tin nào bị mất ?*

* + TCP sử dụng một cơ chế gọi là **ACK (Acknowledgment)** và **Timeout** để phát hiện các gói tin bị mất trong quá trình truyền.

1. **ACK (Acknowledgment - Xác nhận nhận dữ liệu):**

**B1:** Khi một bên gửi dữ liệu qua TCP, nó chia dữ liệu thành các gói tin nhỏ (segment) và đánh số thứ tự cho từng gói tin (**sequence number**).

**B2:** Sau khi nhận được một gói tin, bên nhận sẽ gửi một thông báo xác nhận (**ACK**) về cho bên gửi, trong đó chứa số thứ tự của gói tin tiếp theo mà nó mong muốn nhận (**ACK number**).

**B3:** Nếu gói tin bị mất, bên gửi sẽ không nhận được thông báo xác nhận (ACK) cho gói tin bị mất trong một khoảng thời gian quy định (**timeout**). Khi hết thời gian chờ mà không nhận được ACK, bên gửi sẽ **gửi lại gói tin bị mất.**

1. **Timeout (Thời gian chờ)**

TCP sử dụng một bộ hẹn giờ (timer) cho mỗi gói tin được gửi đi. Nếu không nhận được ACK trước khi timer hết hạn, TCP sẽ giả định rằng gói tin đã bị mất và gửi lại gói tin đó.

**Timeout được tính toán như thế nào?**

TCP dựa trên thời gian trễ trung bình (**RTT - Round Trip Time**) để ước tính thời gian gói tin đi từ bên gửi đến bên nhận và ACK quay lại.

Nếu RTT cao (mạng chậm), TCP sẽ tăng thời gian chờ. Ngược lại, nếu mạng nhanh, TCP giảm thời gian chờ.

1. **Fast Retransmit**

Ngoài timeout, TCP còn sử dụng cơ chế Fast Retransmit (Truyền lại nhanh) để phát hiện mất gói tin nhanh hơn:

Nếu bên nhận phát hiện một gói tin bị mất, nó sẽ liên tục gửi các ACK lặp lại (duplicate ACKs) cho gói tin cuối cùng mà nó nhận được đúng.

Nếu bên gửi nhận được 3 ACK lặp lại liên tiếp cho cùng một gói tin, nó sẽ ngay lập tức gửi lại gói tin tiếp theo mà nó nghĩ là đã bị mất, thay vì chờ timeout.

1. **Ví dụ**

**Dữ liệu gốc (giả sử chia thành 4 gói tin):**

Gói tin 1: Sequence number = 1

Gói tin 2: Sequence number = 2

Gói tin 3: Sequence number = 3

Gói tin 4: Sequence number = 4

**Quá trình gửi và nhận:**

Bên gửi gửi gói tin 1, 2, 3, 4.

Bên nhận nhận được gói tin 1, gửi ACK = 2 (mong muốn gói tin tiếp theo là 2).

Bên nhận nhận được gói tin 2, gửi ACK = 3.

**Phát hiện mất gói tin:**

Giả sử gói tin 3 bị mất, bên gửi không nhận được ACK = 4 trong thời gian chờ.

Hoặc bên gửi nhận được 3 ACK = 3 liên tiếp (bên nhận đang chờ lại gói tin 3).

**Gửi lại:**

Bên gửi nhận thấy gói tin 3 bị mất, lập tức gửi lại gói tin 3.

Khi bên nhận nhận được gói tin 3, gửi ACK = 4.

#### **3.2.2 IP (Internet Protocol)**

IP chịu trách nhiệm **định tuyến (routing)** các gói dữ liệu từ nguồn đến đích qua mạng dựa trên **địa chỉ IP** chi tiết ở phần sau.

#### **3.2.3 UDP (User Datagram Protocol)**

Là một giao thức tầng Transport trong mô hình TCP/IP, tương tự TCP, nhưng có những điểm khác biệt quan trọng về cách hoạt động và mục đích sử dụng.

**Đặc điểm chính:**

**Không kết nối:** UDP không thiết lập kết nối trước khi gửi dữ liệu. Các gói tin (datagram) được gửi trực tiếp từ bên gửi đến bên nhận mà không cần xác nhận.

**Không đảm bảo độ tin cậy**: UDP **không kiểm tra** xem gói tin có được nhận bởi bên nhận hay không. Các gói tin bị mất hoặc nhận sai thứ tự **sẽ không được gửi lại**.

**Không kiểm tra thứ tự gói tin** : UDP không đảm bảo rằng các gói tin đến đích theo đúng thứ tự mà chúng được gửi đi.

**Tốc độ cao:** UDP nhanh hơn TCP vì nó không có cơ chế kiểm tra hoặc gửi lại gói tin, không cần thiết lập kết nối và không quản lý trạng thái.

**Cách hoạt đông:**

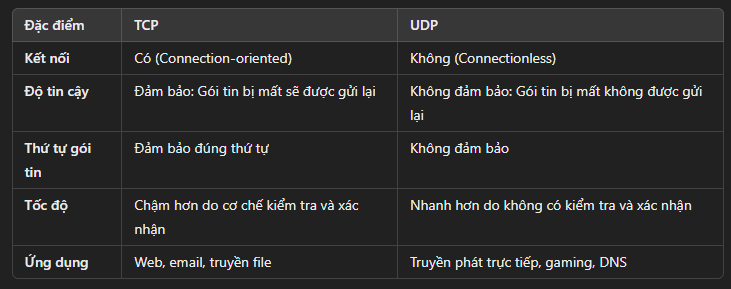
**B1:** **Gửi dữ liệu**

Bên gửi đóng gói dữ liệu vào các datagram. Mỗi datagram chứa:

* Địa chỉ IP của máy nhận.
* Port của ứng dụng trên máy nhận.
* Dữ liệu thực tế.

**B2: Nhận dữ liệu**

* Bên nhận đọc datagram đến dựa trên địa chỉ IP và port.
* Nếu có lỗi trong datagram, UDP không sửa lỗi mà chỉ bỏ qua gói tin.



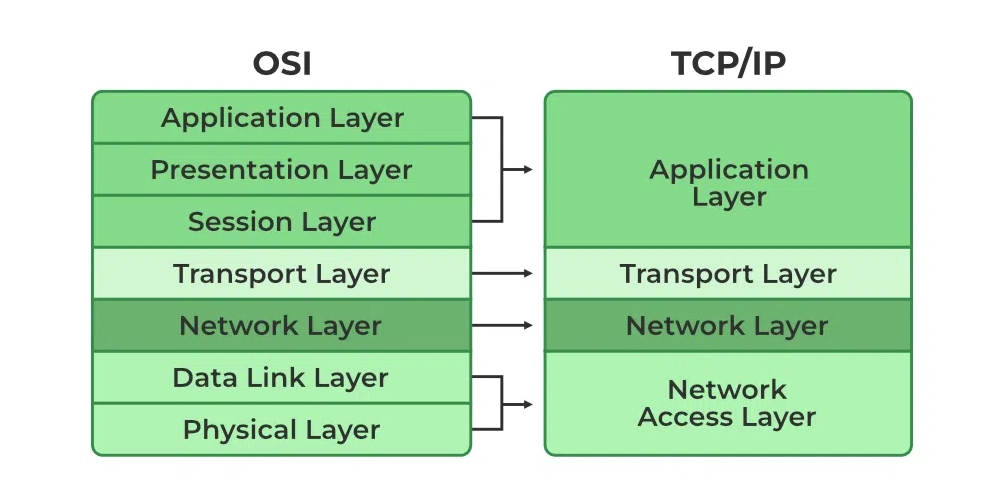
#### **3.2.4 Kết luận**

**IP** đưa gói tin từ nơi này đến nơi khác thông qua **địa chỉ IP**.

**TCP** đảm bảo rằng những gói tin đó được truyền chính xác, đúng thứ tự, và không bị mất.

Nói cách khác, **IP** giống như người đưa thư lo việc vận chuyển gói hàng đến địa chỉ, còn **TCP(UDP**) là người kiểm tra và đảm bảo rằng hàng hóa đến đúng nơi, đầy đủ, và không bị hư hại.

* 1. **So sánh TCP/IP và OSI**



* 1. **Mô hình Client-Server**

Trong mô hình này, các thiết bị client (người dùng) yêu cầu dịch vụ từ các máy chủ, ví dụ như duyệt web hoặc gửi email.

* 1. **Mô hình Peer-to-Peer (P2P)**

Mô hình này cho phép các thiết bị giao tiếp trực tiếp với nhau mà không cần máy chủ trung gian, ví dụ như trong chia sẻ tệp tin (torrent)

1. **Chi tiết hơn về Giao thức Internet (IP)**
   1. **Ipv4 (Giao thức Internet Phiên bản 4)**

IPv4 sử dụng địa chỉ dài 32 bit, được chia thành 4 nhóm số, mỗi nhóm gồm 8 bit (hoặc một byte), và mỗi nhóm được phân cách bằng dấu chấm, IPv4 đang được sử dụng chủ yếu.

IPv4 sử dụng các địa chỉ kiểu lớp (classful addressing) với các loại địa chỉ như **unicast** (địa chỉ đơn), **multicast** (địa chỉ đa điểm), và **broadcast** (phát sóng). Ngoài ra, IPv4 còn chia địa chỉ thành các lớp như A, B, C tùy theo quy mô mạng.   
Ví dụ: **192.168.0.1**.

Số lượng địa chỉ **IPv4** có thể cung cấp là 223, khoảng 4,3 tỷ địa chỉ. IPv4 không tích hợp bảo mật trong giao thức, mặc dù có thể bổ sung bảo mật qua các giao thức khác như IPsec (tuy nhiên, điều này không bắt buộc). Sự phát triển bùng nổ của Internet đã dẫn đến tình trạng cạn kiệt địa chỉ IPv4, đặc biệt là vào năm 2011, khi nhóm phân bổ địa chỉ **IPv4** toàn cầu (IANA) thông báo rằng địa chỉ IPv4 đã gần như hết.  
 **Giải pháp**: Để giải quyết vấn đề cạn kiệt địa chỉ, IPv4 thường sử dụng kỹ thuật **NAT (Network Address Translation)**, cho phép nhiều thiết bị trong mạng nội bộ chia sẻ một địa chỉ IP công cộng. Nhờ NAT, một địa chỉ IP công cộng có thể được "chia sẻ" và sử dụng cho nhiều địa chỉ IP trong mạng nội bộ.

* 1. **IPv6 (Giao thức Internet Phiên bản 6)**

IPv6 được phát triển để thay thế IPv4 nhằm giải quyết vấn đề cạn kiệt địa chỉ, cũng như cải thiện khả năng bảo mật và hiệu suất mạng.

Ví dụ: 2001:0db8:85a3:0000:0000:8a2e:0370:7334

IPv6 có một cách phân chia địa chỉ rõ ràng hơn với các phần như Global Unicast Address (GUA) và Link-local Address. Đặc biệt, IPv6 sử dụng các địa chỉ kiểu **Link-local** để giao tiếp trong mạng cục bộ mà không cần router.

Số lượng địa chỉ IPv4 có thể cung cấp là  2128 khoảng **340.282 tỷ tỷ tỷ tỷ tỷ tỷ** , một con số rất lớn, đủ để đáp ứng nhu cầu của Internet trong nhiều thế kỷ tới.

Với số lượng địa chỉ cực kỳ lớn, IPv6 không cần sử dụng NAT. Mỗi thiết bị trong mạng có thể được gán một địa chỉ IP công cộng riêng biệt.

IPv6 được thiết kế với **bảo mật tích hợp**. Giao thức bảo mật IPsec là một phần không thể tách rời của IPv6, giúp bảo vệ dữ liệu khỏi các cuộc tấn công và nghe lén.



* 1. **IPv5**

Điều gì xảy ra với IPv5

**IPv5** thực tế không phải là một phiên bản giao thức IP chính thức được sử dụng trên Internet. Điều này có thể gây nhầm lẫn vì trong hệ thống đánh số các phiên bản giao thức IP, sau **IPv4** người ta đã kỳ vọng sẽ có **IPv5**, nhưng thực tế là **IPv5 không bao giờ được sử dụng rộng rãi** trong các giao thức Internet.

Lý do:

**IPv5 là một giao thức thử nghiệm**

Mặc dù có dự định là một phiên bản giao thức IP, nhưng IPv5 không được triển khai rộng rãi vì nó không đáp ứng được nhu cầu phát triển mạng Internet khi các tiêu chuẩn mới như IPv6 ra đời.

Sau khi IPv4 đã phát triển và tồn tại trong một thời gian dài, các kỹ sư và nhà phát triển đã nhận thấy sự cần thiết phải phát triển một giao thức mới để đáp ứng yêu cầu về địa chỉ IP lớn hơn, bảo mật tốt hơn, và khả năng mở rộng cao hơn. Chính vì vậy, **IPv6** được ra đời vào năm 1998 thay thế các giải pháp như IPv5 hoặc các giao thức thử nghiệm khác.

Mặc dù IPv5 (ST-II) có khả năng hỗ trợ các ứng dụng streaming, nhưng nó không được triển khai rộng rãi vì không phù hợp với nhu cầu phát triển của Internet thời kỳ đó.

1. **Phân bổ địa chỉ IP động và tĩnh: DHCP (Dynamic Host Configuration Protocol)**

**DHCP** là một giao thức mạng được sử dụng để tự động phân phối và quản lý các thông tin cấu hình mạng cho các thiết bị kết nối vào mạng.

**5.1. Phân bổ địa chỉ IP động và tĩnh**

1. Phân bổ địa chỉ IP động (Dynamic IP Allocation)

DHCP tự động phân bổ địa chỉ IP cho các thiết bị kết nối.

Thiết bị nhận được địa chỉ IP từ DHCP server thay vì được cấu hình thủ công;

VD: Laptop ở công ty

2. Phân bổ địa chỉ IP tĩnh (Static IP Allocation)

Địa chỉ IP được cấu hình thủ công và không thay đổi.

Thường được sử dụng cho các thiết bị ổn định như máy chủ web.

VD: Desktop ở công ty

**5.2.** **Cách thức hoạt động của DHCP**

* DHCP Discover: Thiết bị gửi yêu cầu tìm kiếm DHCP server.
* DHCP Offer: DHCP server trả lời với một địa chỉ IP.
* DHCP Request: Thiết bị chấp nhận đề xuất.
* DHCP Acknowledge: DHCP server xác nhận giao dịch

1. **Tên miền và Hệ thống tên miền (DNS)**

**6.1. Tên miền (Domain Name)**

Tên miền và Hệ thống tên miền (DNS - Domain Name System) là một hệ thống cho phép thiết lập tương ứng giữa địa chỉ IP và tên miền trên Internet.

Thay vì phải nhớ địa chỉ ip máy chủ 142.250.197.238 của google thì ta chỉ cần nhớ google.com

**6.2 Hệ thống tên miền (DNS – Domain name system)**

Là một hệ thống phân tán, có nhiệm vụ dịch các tên miền dễ nhớ thành các địa chỉ IP số mà máy tính và các thiết bị có thể sử dụng để xác định nhau trên mạng. Nó có thể coi như một "sổ điện thoại" của Internet.

**DNS hoạt động giống như một danh bạ điện thoại:** Bạn không cần nhớ số điện thoại của người bạn muốn gọi, chỉ cần nhớ tên của họ. Mỗi khi bạn nhập tên (tên miền) vào trình duyệt, DNS server sẽ tìm ra số điện thoại (địa chỉ IP) và kết nối bạn tới người đó (máy chủ web).

**DNS** đóng vai trò quan trọng trong việc làm cho việc duyệt web và sử dụng các dịch vụ trên Internet trở nên dễ dàng hơn, bằng cách tự động xử lý việc ánh xạ giữa các tên miền dễ nhớ và các địa chỉ IP khó nhớ.

# **III. How do Network/Internet work?**

## **1. Internet work**

Internet hoạt động ở quy mô toàn cầu, sử dụng mô hình TCP/IP và các giao thức để kết nối hàng triệu mạng với nhau:

1. Kết nối thông qua ISP:

* Một thiết bị muốn truy cập Internet phải được kết nối với ISP (như Viettel, VNPT, FPT).
* ISP định tuyến dữ liệu từ thiết bị đến các mạng khác trên Internet.

1. Định tuyến qua các mạng trung gian:

* Dữ liệu được truyền qua nhiều router và mạng trung gian để đến được thiết bị đích.
* Các giao thức như TCP/IP, HTTP/HTTPS đảm bảo dữ liệu đến đúng nơi.

1. Tra cứu địa chỉ (DNS):

* Khi truy cập một website, Internet sử dụng hệ thống DNS để chuyển đổi tên miền (ví dụ: google.com) thành địa chỉ IP của máy chủ.

## **2. Network work**

Dựa trên mô hình 4 lớp của **TCP/IP** mà network hoạt động như sau :

**Tại máy gửi:**

* Lớp **Application**: Trình duyệt của bạn gửi yêu cầu HTTP đến server Google.
* Lớp **Transport**: Dữ liệu HTTP được đóng gói trong TCP/UDP, gắn số thứ tự và cổng (port).
* Lớp **Internet**: TCP/UDP được đóng gói tiếp trong một gói IP, thêm địa chỉ IP nguồn và đích.
* Lớp **Link**: Gói tin IP được chuyển thành khung dữ liệu và truyền qua cáp hoặc không dây.

**Trên đường truyền:**

* Gói tin đi qua các router. Các router đọc địa chỉ IP đích, định tuyến gói tin qua các mạng trung gian để đến đúng máy chủ.

**Tại máy nhận:**

* Lớp **Link**: Nhận các khung dữ liệu từ mạng.
* Lớp **Internet**: Lấy gói IP, đọc địa chỉ và chuyển tiếp lên lớp trên.
* Lớp **Transport**: Ghép lại các gói tin, kiểm tra lỗi và chuyển dữ liệu đến đúng ứng dụng dựa trên cổng.
* Lớp **Application**: Hiển thị kết quả cho người dùng (trang web, file tải xuống, v.v.).

## **3. Mô hình mạng nội bộ LAN**

* Thiết bị kết nối với nhau thông qua switch hoặc router.
* Dữ liệu truyền tải nhanh chóng vì tất cả thiết bị nằm trong cùng một mạng.
* Không yêu cầu Internet để hoạt động.
* Ví dụ: Các máy tính trong văn phòng chia sẻ một máy in qua mạng LAN

## **4. Mô hình mạng diện rộng WAN**

* Kết nối các mạng LAN ở những địa điểm khác nhau thông qua router.
* Cần ISP để truyền tải dữ liệu giữa các mạng ở xa.
* Ví dụ: Chi nhánh ngân hàng ở các tỉnh kết nối với hệ thống trung tâm

## **5. Kết luận**

**Network** hoạt động ở phạm vi nhỏ hơn và có thể độc lập, tập trung vào việc chia sẻ tài nguyên giữa các thiết bị trong một khu vực cụ thể.

**Internet** kết nối hàng triệu mạng trên toàn cầu, sử dụng các giao thức giống như network nhưng ở quy mô lớn hơn và cần ISP để định tuyến dữ liệu..

=> Hai hệ thống này có mối quan hệ chặt chẽ: **Internet là tập hợp của rất nhiều network** được kết nối với nhau.

**Ví Dụ Minh Họa**

1. **Mạng nội bộ (Network):**

Trong một công ty, các máy tính và máy in được kết nối với nhau qua mạng LAN. Nhân viên có thể in tài liệu hoặc chia sẻ file mà không cần Internet.

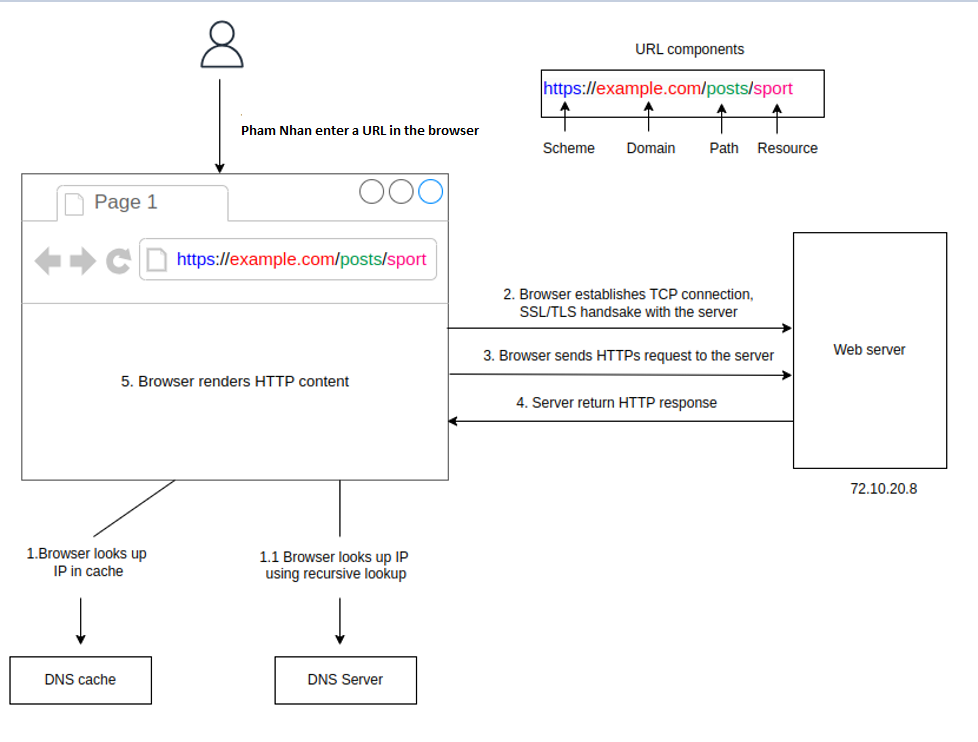
Nếu công ty muốn kết nối với chi nhánh khác, họ sử dụng mạng WAN.

1. **Internet:**

Khi nhân viên trong công ty truy cập một trang web bên ngoài (ví dụ: Google), dữ liệu phải đi qua router, ISP, và nhiều mạng trung gian trước khi đến được máy chủ Google.

# **IV. How do Website work?**

Hằng ngày, chúng ta truy cập các trang web để tìm kiếm thông tin. Đã bao giờ bạn tự hỏi điều gì thực sự xảy ra khi chúng ta nhập một URL vào thanh tìm kiếm của trình duyệt chưa? Sau đây hãy cùng thảo luận về điều đó nhé.



**Quá trình từ nhập URL đến hiển thị trang web**

Khi bạn nhập một URL vào trình duyệt và nhấn Enter, rất nhiều quá trình phức tạp diễn ra để đưa trang web từ máy chủ đến màn hình của bạn. Dưới đây là các bước chi tiết:

## **1. Tra cứu DNS**

Khi bạn nhập URL (ví dụ: www.example.com), trình duyệt không biết trực tiếp địa chỉ IP của máy chủ. Do đó, nó cần thực hiện **tra cứu DNS** (Domain Name System) để chuyển đổi tên miền thành địa chỉ IP public (ví dụ: www.example.com -> 72.10.2.3).

Hãy tưởng tượng DNS giống như một **danh bạ điện thoại** trên Internet. Bạn không cần nhớ địa chỉ IP phức tạp của máy chủ, mà chỉ cần tên miền dễ nhớ, và DNS sẽ làm phần còn lại.

Quá trình tra cứu DNS diễn ra qua các bước sau:

* **Kiểm tra bộ nhớ cache:**  
  Trình duyệt kiểm tra xem địa chỉ IP đã được lưu trong bộ nhớ cache hay chưa. Bộ nhớ cache này có thể nằm ở:
  + Trình duyệt.
  + Hệ điều hành (file như /etc/hosts).
  + Router cục bộ trong mạng.
  + Máy chủ DNS của nhà cung cấp dịch vụ Internet (ISP).
* **Truy vấn máy chủ DNS:**  
  Nếu không tìm thấy trong cache, trình duyệt gửi truy vấn tới máy chủ DNS để lấy địa chỉ IP. Các bản ghi DNS phổ biến gồm:
  + **A record:** Chuyển tên miền thành địa chỉ IPv4.
  + **CNAME record:** Chuyển hướng tên miền sang một tên miền khác.
  + **MX record:** Dùng cho email.

Quá trình này thường mất vài mili giây, và kết quả là trình duyệt nhận được địa chỉ IP của máy chủ cần truy cập.

## **2. Thiết lập kết nối TCP**

Sau khi có địa chỉ IP, trình duyệt bắt đầu thiết lập kết nối TCP (Transmission Control Protocol) với máy chủ để truyền tải dữ liệu.

Quá trình này bao gồm:

* **Bắt tay 3 bước (TCP handshake):**
  1. Trình duyệt gửi gói tin **SYN** để yêu cầu kết nối.
  2. Máy chủ trả lời bằng gói **SYN-ACK** để xác nhận.
  3. Trình duyệt gửi gói **ACK** cuối cùng để hoàn tất kết nối.

Nếu URL sử dụng **HTTPS**, một bước bổ sung sẽ diễn ra:

* **SSL/TLS handshake:**  
  Trình duyệt và máy chủ đồng ý về các phương pháp mã hóa, đảm bảo rằng dữ liệu truyền đi được bảo mật và không bị tin tặc đánh cắp. Điều này dựa vào chứng chỉ SSL/TLS do các tổ chức đáng tin cậy cấp.

## **3. Trình duyệt gửi yêu cầu HTTP**

Khi kết nối TCP đã được thiết lập, trình duyệt gửi một yêu cầu HTTP hoặc HTTPS đến máy chủ. Yêu cầu này bao gồm:

* **Phương thức yêu cầu:**
  + GET: Yêu cầu lấy tài nguyên (ví dụ: trang web).
  + POST: Gửi dữ liệu đến máy chủ.
  + PUT, DELETE: Các phương thức khác để thao tác dữ liệu.
* **Tiêu đề (Headers):**
  + Cung cấp thông tin như loại trình duyệt (User-Agent), cookie, hoặc dữ liệu bộ nhớ cache.
* **Đường dẫn:**
  + Ví dụ: GET /index.html HTTP/1.1.

## **4. Phản hồi từ máy chủ**

Sau khi nhận yêu cầu, máy chủ xử lý và gửi phản hồi lại trình duyệt. Phản hồi bao gồm:

* **Dòng trạng thái:** Cho biết tình trạng của yêu cầu. Ví dụ:
  + 1xx: Thông tin (hiếm khi sử dụng)
  + 2xx: Thành công

200 OK: Y4:0

201 Created: Tạo mới thành công

* + 3xx: Chuyển hướng

301 Moved Permanently: Đã chuyển dời vĩnh viễn

* + 4xx: Lỗi máy khách

400 Bad Request: Y4:0

404 Not Found: Không tìm thấy

* + 5xx: Lỗi máy chủ

500 Internal Server Error: Lỗi máy chủ nội bộ

**Các phản hồi thông dụng**

200 OK: Phản hồi thành công cho yêu cầu hợp lệ

404 Not Found: Không tìm thấy tài nguyên được yêu cầu

500 Internal Server Error: Lỗi máy chủ nội bộ

400 Bad Request: Yêu cầu không hợp lệ hoặc có lỗi cú pháp

401 Unauthorized: Cần xác thực để truy cập

403 Forbidden: Truy cập bị từ chối

* **Headers phản hồi:**
  + Chứa thông tin như loại dữ liệu (Content-Type: text/html), độ dài (Content-Length), hoặc chỉ dẫn bộ nhớ cache.
* **Body:**
  + Tài nguyên được yêu cầu, chẳng hạn: HTML, CSS, JavaScript, hình ảnh, hoặc video.

## **5. Trình duyệt hiển thị trang web**

Sau khi nhận phản hồi, trình duyệt bắt đầu hiển thị nội dung trang web theo các bước sau:

1. **Phân tích HTML:**  
   Trình duyệt xây dựng một cấu trúc cây gọi là **DOM (Document Object Model)**, đại diện cho các phần tử trên trang.
2. **Tải CSS và JavaScript:**
   * Nếu HTML tham chiếu đến các file CSS, trình duyệt sẽ tải và áp dụng các quy tắc để tạo kiểu dáng cho trang.
   * Nếu có JavaScript, trình duyệt thực thi để thêm chức năng tương tác, như hiển thị popup hoặc cập nhật nội dung động.
3. **Tải thêm tài nguyên:**
   * Hình ảnh, video, phông chữ hoặc các tài nguyên khác cũng được tải để hoàn thiện giao diện trang web.
4. **Hiển thị:**  
   Sau khi tất cả các tài nguyên được tải và xử lý, trình duyệt hiển thị nội dung cuối cùng trên màn hình.

## **6. Trình duyệt hiển thị trang web**

* **Bộ nhớ cache:**  
  Để tăng tốc độ tải trong lần truy cập sau, trình duyệt có thể lưu các tài nguyên như hình ảnh, CSS, JavaScript.
* **Kỹ thuật tối ưu hóa:**
  + **Lazy loading:** Chỉ tải tài nguyên khi cần thiết.
  + **Minify:** Nén HTML, CSS, và JavaScript để giảm kích thước.
  + **CDN (Content Delivery Network):** Phân phối tài nguyên từ máy chủ gần nhất với người dùng.

## **7. Kết luận**

Tất cả các bước trên chỉ mất vài giây (hoặc mili giây) để hoàn thành, mang đến trải nghiệm truy cập nhanh chóng và mượt mà cho người dùng. Mỗi bước đều đóng vai trò quan trọng, từ việc định vị máy chủ qua DNS, bảo mật kết nối bằng HTTPS, đến việc tối ưu hóa hiển thị nội dung trên trình duyệt.

Nhờ có sự phối hợp chặt chẽ giữa trình duyệt, máy chủ, và mạng Internet, chúng ta có thể truy cập hàng triệu trang web chỉ với một cú nhấp chuột.

# **V. Talk**

Đặt câu hỏi :   
*Trước khi có internet, con người giao tiếp và chia sẻ thông tin như thế nào?*

*Nếu một ngày không có Internet?*

Ví dụ so sánh **Internet**, **Network**, và **Website** bằng ví dụ trực quan:

* **Internet** giống như một con đường lớn, kết nối mọi nơi với nhau.
* **Network** là các con đường nhỏ hơn nối từ con đường lớn đó đến các địa phương khác nhau.
* **Website** là các cửa hàng, nhà dân, cây xăng... dọc theo các con đường nhỏ đó, đại diện cho các dịch vụ hoặc thông tin có sẵn mà người dùng có thể truy cập.
* Trên con đường **(Internet)** , nơi tất cả các phương tiện (dữ liệu) di chuyển từ nơi này đến nơi khác. Trên con đường này, có nhiều loại **phương tiện** khác nhau, tượng trưng cho các **giao thức mạng** (**Network Protocols**)sử dụng để truyền tải dữ liệu :
* **TCP (Transmission Control Protocol)** - Giống như một chiếc **xe tải lớn**, luôn đảm bảo rằng mọi thứ được vận chuyển an toàn và đúng địa điểm. Xe tải này sẽ dừng lại kiểm tra và xác nhận nếu có bất kỳ thứ gì bị mất trong quá trình vận chuyển, và sẽ quay lại để lấy lại nếu cần.
* **UDP (User Datagram Protocol)** - Là một chiếc **xe máy** nhanh chóng và không dừng lại kiểm tra. Xe máy này sẽ di chuyển với tốc độ cao, không có sự đảm bảo rằng mọi thứ sẽ đến đích nguyên vẹn, nhưng nếu bạn cần di chuyển nhanh, UDP là lựa chọn tốt.
* **HTTP/HTTPS (Hypertext Transfer Protocol/Secure)** - Là những chiếc **ô tô** vận chuyển hàng hóa đặc biệt, thường là các **website** và **dữ liệu trang web**. HTTPS giống như chiếc ô tô có bảo vệ an toàn, giúp dữ liệu (chẳng hạn như thông tin cá nhân) được bảo mật khi di chuyển.
* **FTP (File Transfer Protocol)** - Giống như một **xe tải chở hàng hóa**, nhưng xe này chuyên chở các tập tin lớn (như hình ảnh, video, tài liệu). FTP giúp di chuyển các file giữa các máy tính, giống như các tài xế vận chuyển đồ đạc giữa các cửa hàng.
* **ICMP (Internet Control Message Protocol)** - Là chiếc **ô tô kiểm tra đường** dùng để đo lường tình trạng của con đường (kiểm tra tình trạng mạng) và gửi thông báo nếu có sự cố. Ví dụ, khi bạn thực hiện lệnh **ping**, ô tô này sẽ kiểm tra xem đường có thông suốt không và gửi thông tin về trạng thái.