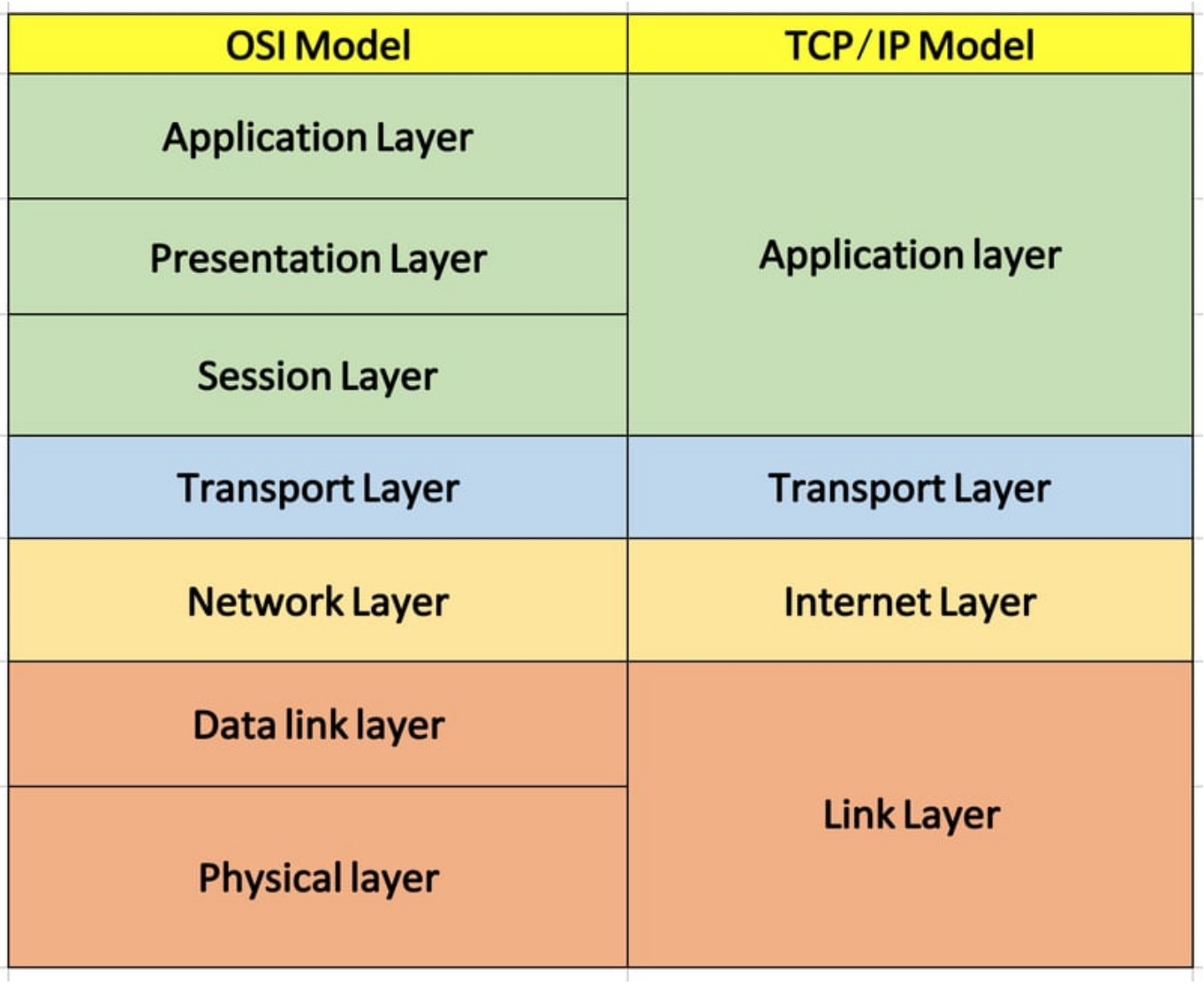
Tìm hiểu về Networking

I. Mô hình OSI, Giao thức TCP, HTTP, SSL

Mô tả các tầng OSI và TCP/IP



1. Mô hình OSI

OSI - Open Systems Interconnection (OSI): Mô hình kết nối các hệ thống mở là một khung khái niệm chia các chức năng truyền thông mạng thành 7 lớp.

Tóm tắt ngắn gọn mô hình OSI gồm 7 tầng:

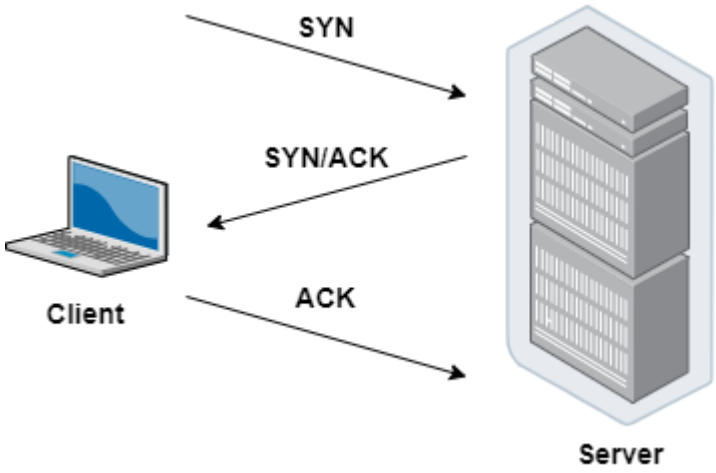
- Tầng Application: Cung cấp giao diện cho ứng dụng người dùng, thực hiện các dịch vụ như truyền tập tin, gửi email, và duyệt web.
- Tầng Presentation: Đảm bảo dữ liệu được trình bày, định dạng, và mã hóa sao cho ứng dụng có thể hiểu được. Mã hóa, nén, và định dạng dữ liệu để truyền dẫn và hiển thị dữ liệu tương thích với ứng dụng người dùng.
- Tầng Session: Quản lý kết nối phiên giữa các thiết bị, đồng bộ hóa việc truyền dữ liệu và quản lý phiên làm việc giữa các ứng dụng.

- Tầng Transport: Đảm bảo dữ liệu được chuyển đến đúng đích, kiểm soát lỗi và đánh dấu các gói tin. Phân biệt dịch vụ truyền dữ liệu đáng tin cậy (TCP) và không đáng tin cậy (UDP), kiểm soát luồng dữ liệu và đảm bảo tính toàn vẹn của dữ liệu.
- Tầng Network: Quản lý địa chỉ IP, định tuyến dữ liệu giữa các mạng. Định tuyến gói tin từ nguồn đến đích, kiểm soát luồng dữ liệu và quản lý giao diện mạng.
- Tầng Data link: Quản lý truy cập vào phương tiện truyền dẫn và kiểm soát lỗi truyền dẫn. Đóng gói dữ liệu thành các khung (frame), kiểm tra lỗi và điều khiển truy cập trong mạng đa truy cập.
- Tầng Physical: Định nghĩa các chuẩn về vật lý và điều khiển tín hiệu trên phương tiện truyền dẫn. Truyền dữ liệu thông qua phương tiện vật lý như cáp đồng, sợi quang, hoặc sóng vô tuyến.

2. Mô hình TCP/IP

TCP/IP - Transmission Control Protocol/Internet Protocol: Giao thức điều khiển truyền nhận/ Giao thức liên mạng. Đây là bộ các giao thức truyền thông được dùng để kết nối các thiết bị mạng trên internet với nhau.

Hoạt động TCP: TCP hoạt động theo tiến trình bắt tay 3 bước (3 way handshake). Tiến trình này hoạt động như sau:



- Máy khách gửi cho máy chủ một gói SYN — một yêu cầu kết nối từ port nguồn của nó đến port đích đến của máy chủ
- Máy chủ phản hồi bằng gói SYN/ACK, xác nhận việc nhận được yêu cầu kết nối
- Máy khách nhận gói SYN/ACK và trả lời bằng gói ACK của chính nó

Sau khi kết nối được thiết lập, TCP hoạt động bằng cách chia nhỏ dữ liệu đã truyền thành các segment (phân đoạn), mỗi segment được đóng gói thành một gói dữ liệu và được gửi đến đích của nó.

Tóm tắt ngắn gọn mô hình TCP/IP gồm 4 tầng:

- Tầng Application: Cung cấp cho các ứng dụng những trao đổi dữ liệu chuẩn hóa, giao tiếp dữ liệu giữa 2 máy khác nhau thông qua các dịch vụ mạng khác nhau
- Tầng Transport: Đảm bảo duy trì thông tin liên lạc từ đầu đến cuối trên toàn mạng là trách nhiệm của TCP. Giao thức này xử lý việc liên lạc giữa các máy chủ và cung cấp các tính năng kiểm soát luồng, ghép kênh và đảm bảo độ tin cậy
- Tầng Internet: Nhiệm vụ xử lý các gói tin mạng và kết nối các mạng độc lập, giúp vận chuyển các gói tin qua mạng

- Tầng Physical: Bao gồm các giao thức hoạt động trên một liên kết duy nhất – thành phần mạng kết nối các nút hoặc máy chủ trong mạng, chịu trách nhiệm truyền dữ liệu giữa hai thiết bị trong cùng một mạng

So sánh mô hình OSI và TCP/IP

Giống:

- Đều là mô hình logic để chuẩn hóa truyền thông mạng
- Xác định tiêu chuẩn cho các mạng máy tính
- Chia quá trình giao tiếp mạng thành nhiều tầng (layer) riêng biệt
- Cung cấp khuôn khổ để tạo và triển khai các tiêu chuẩn và thiết bị mạng

Khác:

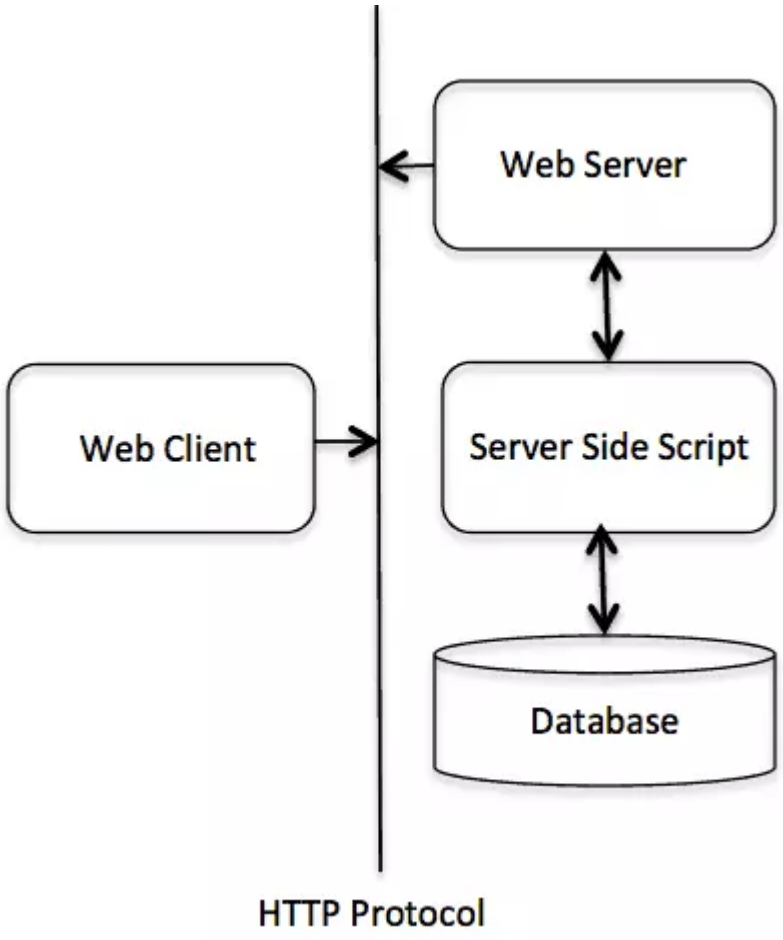
Phân loại	**TCP/IP**	**OSI**
Số lớp	4 lớp	7 lớp
Phổ biến	Nhiều sử dụng	Ít được sử dụng
Phương pháp tiếp cận	Chiều ngang	Chiều dọc
Cách giao tiếp các tầng	Kết hợp để thực hiện nhiệm vụ	Mỗi tầng 1 nhiệm vụ riêng biệt
Sự phụ thuộc	Phụ thuộc giao vào giao thức	Độc lập hoàn toàn
Sự phát triển	Phát triển giao thức trước - xây dựng mô hình sau	Xây dựng mô hình trước - giao thức dựng sau

3. Giao thức HTTP

Giao thức HTTP (Hypertext Transfer Protocol) được sử dụng rộng rãi trong việc truyền tải dữ liệu qua World Wide Web. HTTP cho phép trình duyệt của người dùng gửi yêu cầu đến máy chủ để truy cập nội dung trang web, từ đó tải về và hiển thị thông tin.

Cấu trúc hoạt động:

Giao thức HTTP còn được biết đến là một giao thức theo kiểu Yêu cầu – Phản hồi dựa trên cấu trúc Client – Server. Theo đó, Client và Server sẽ có xu hướng giao tiếp với nhau bằng cách trao đổi các message độc lập (điều này trái ngược hoàn toàn với một luồng dữ liệu). Các message này sẽ được gửi bởi Client, thông thường là qua một trình duyệt web. Các yêu cầu cũng như message sau đó sẽ được gửi lại bởi server như một sự trả lời, hay còn được gọi là phản hồi.



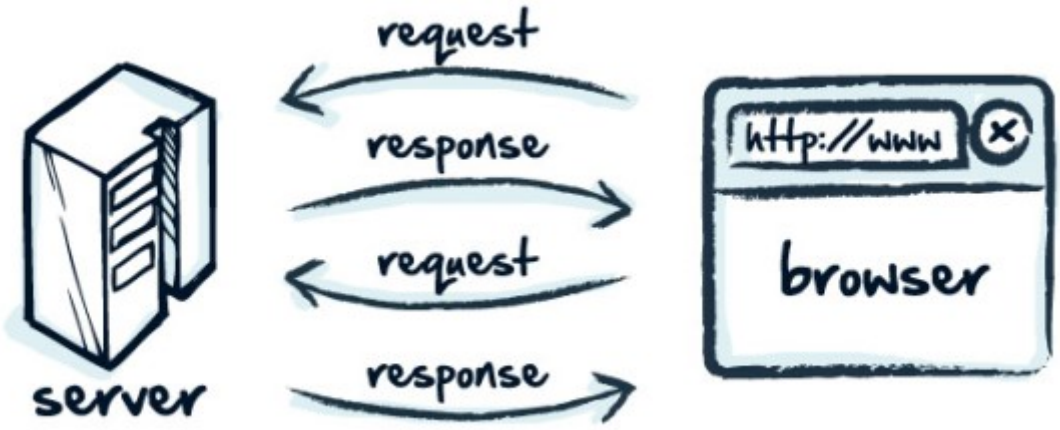
Phương thức HTTP (HTTP Methods)

HTTP hỗ trợ nhiều phương thức yêu cầu khác nhau, mỗi phương thức có mục đích riêng trong việc xử lý tài nguyên trên máy chủ:

- GET: Yêu cầu truy xuất tài nguyên từ máy chủ (ví dụ: tải một trang web)
- POST: Gửi dữ liệu đến máy chủ để tạo mới hoặc cập nhật tài nguyên
- PUT: Tương tự POST, nhưng thường dùng để cập nhật toàn bộ tài nguyên
- DELETE: Xóa tài nguyên trên máy chủ
- HEAD: Tương tự GET nhưng không trả về phần nội dung chính của tài nguyên, chỉ trả về các tiêu đề
- PATCH: Cập nhật một phần tài nguyên (khác với PUT cập nhật toàn bộ)
- OPTIONS: Lấy thông tin về các phương thức HTTP mà máy chủ hỗ trợ cho tài nguyên cụ thể

Phương thức kết nối của HTTP

HTTP là một giao thức mạnh mẽ nhờ vào khả năng xử lý linh hoạt các yêu cầu và phản hồi giữa máy khách và máy chủ. Quá trình này gồm các bước như sau:



3.1 Yêu cầu (Request)

Khi máy khách (client) gửi yêu cầu đến máy chủ (server) qua HTTP, yêu cầu này bao gồm:

- Phương thức yêu cầu: Các phương thức như GET (yêu cầu thông tin), POST (gửi dữ liệu), PUT (cập nhật dữ liệu), DELETE (xóa tài nguyên).
- URL tài nguyên: Địa chỉ của tài nguyên trên máy chủ.
- Dữ liệu biểu mẫu: Dữ liệu kèm theo trong các yêu cầu POST, PUT.
- Tiêu đề yêu cầu: Thông tin bổ sung như ngôn ngữ, định dạng dữ liệu mong muốn.

3.2 Xử lý yêu cầu (Request Processing)

Sau khi nhận được yêu cầu từ client, máy chủ xử lý nó bằng cách:

- Kiểm tra tính hợp lệ: Xem xét cú pháp và quyền truy cập.
- Truy xuất dữ liệu: Truy cập cơ sở dữ liệu hoặc các nguồn khác nếu cần.
- Thực hiện logic ứng dụng: Máy chủ xử lý các thao tác yêu cầu như truy vấn, đăng nhập, hoặc lưu trữ thông tin.

3.3 Phản hồi (Response)

Sau khi xử lý, máy chủ gửi phản hồi lại client. Phản hồi này bao gồm:

- Mã trạng thái: Thông báo kết quả của yêu cầu như 200 (OK), 404 (Not Found), 500 (Internal Server Error).
- Dữ liệu phản hồi: Nội dung trang web hoặc thông tin liên quan mà client yêu cầu.

3.4 Xử lý phản hồi (Response Processing)

Client nhận phản hồi từ máy chủ và xử lý nó bằng cách:

- Kiểm tra phản hồi: Xem xét mã trạng thái để xác định kết quả yêu cầu.
- Hiển thị nội dung: Nếu thành công, nội dung sẽ được hiển thị qua trình duyệt.
- Sử dụng dữ liệu: Client có thể dùng dữ liệu cho các mục đích khác, như cập nhật giao diện hoặc lưu trữ thông tin.

4. Giao thức SSL

SSL được viết tắt từ Secure Sockets Layer, đây là một tiêu chuẩn của công nghệ bảo mật, truyền thông mã hóa giữa trình duyệt và máy chủ web server. SSL hoạt động và đảm bảo rằng những dữ liệu được truyền tải giữa máy chủ và trình duyệt của bạn đều được toàn vẹn, riêng tư và bảo mật. Hiện nay, SSL được xem là tiêu chuẩn bảo mật cho đa số website trên thế giới, giúp dữ liệu truyền đi trên Internet được bảo vệ một cách an toàn.

4.1 Cách hoạt động SSL

HTTPS sử dụng giao thức SSL (Secure Sockets Layer) hoặc giao thức tiếp nối tầng bảo mật TLS để bảo mật thông tin liên lạc bằng cách truyền dữ liệu qua mạng Internet được mã hóa. SSL hoạt động dựa trên hai khái niệm chính: mã hóa bất đối xứng (asymmetric cryptography) và mã hóa đối xứng (symmetric cryptography).

Khi người dùng truy cập vào một dịch vụ hỗ trợ SSL, như một trang web, quá trình kết nối diễn ra như sau:

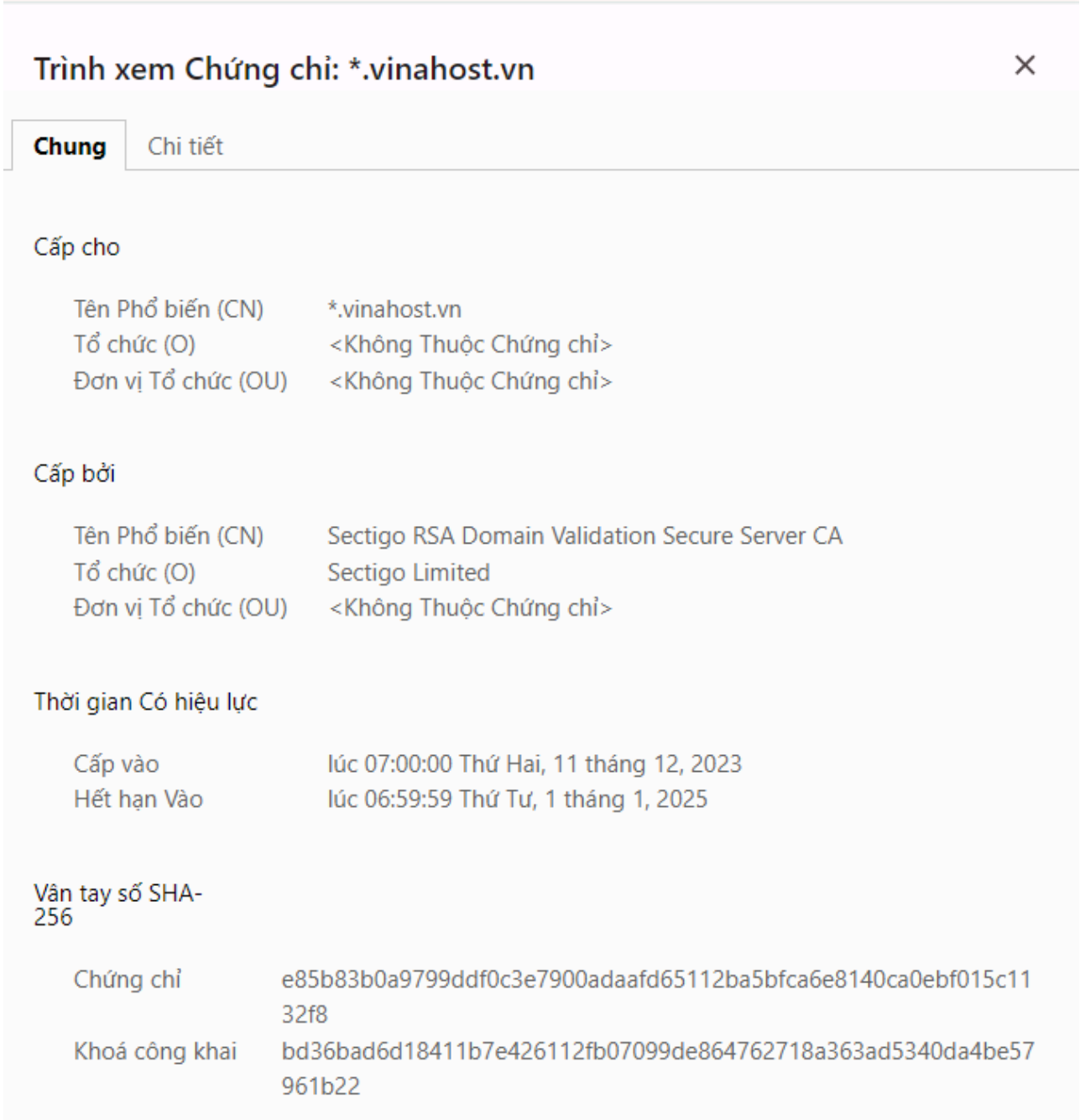
- Ứng dụng của người dùng yêu cầu khóa công khai từ máy chủ để trao đổi với khóa công khai của chính nó.
- Việc trao đổi khóa công khai này cho phép cả hai bên có thể mã hóa các tin nhắn mà chỉ bên đích mới có thể giải mã được.
- Khi người dùng gửi tin nhắn đến máy chủ, ứng dụng sử dụng khóa công khai của máy chủ để mã hóa tin nhắn.
- Máy chủ nhận tin nhắn từ người dùng và giải mã nó bằng khóa riêng của mình. Sau đó, tin nhắn được mã hóa và gửi trở lại trình duyệt của người dùng bằng cách sử dụng khóa công khai do ứng dụng của người dùng tạo ra.

Qua quá trình này, thông tin truyền tải giữa người dùng và máy chủ được bảo mật bằng cách sử dụng mã hóa và khóa công khai/ riêng tư để đảm bảo chỉ những người có khóa riêng mới có thể giải mã và đọc được tin nhắn.

4.2 Các thành phần SSL

Các thành phần chung của chứng chỉ SSL như sau:

- CSR (Certificate Signing Request): là một tài liệu văn bản chứa thông tin về chủ sở hữu tên miền đã được mã hóa. Thông tin này được gửi đến nhà cung cấp dịch vụ SSL để xác minh và xác nhận.
- CRT (Certificate): Là thành phần được trả về sau khi CSR đã được xác nhận và chứa thông tin chứng chỉ SSL. Nếu CSR được tạo ra để nhà cung cấp SSL xác nhận tính tin cậy của website với thông tin đã được mã hóa trong nó, thì CRT là một tài liệu để trình duyệt web tin tưởng vào.
- Private key: Là một file mã hóa được tạo ra cùng với CSR. Để giải thích một cách đơn giản, hãy tưởng tượng rằng CRT là một phần mã hóa công khai mà trình duyệt web sử dụng để truy cập vào website của bạn. Khi dữ liệu đến đến website, nó cần một chìa khóa riêng để mở khóa thông tin đã được mã hóa trong CRT.
- CA (Certificate Authority hoặc Certification Authority): Là một tổ chức hoặc cơ quan cung cấp thông tin về chứng chỉ SSL.



4.3 Vai trò SSl

- Bảo mật thông tin bằng mã hóa

SSL giúp cho thông tin nhạy cảm trở nên an toàn khi chúng ta gửi qua Internet. SSL hoạt động bằng cách biến thông tin thành một loại "ngôn ngữ" chỉ có những người có chìa khóa mới có thể hiểu được. Khi bạn gửi thông tin trên Internet, thông tin đó di chuyển qua nhiều máy tính trước khi đến máy chủ đích. Nếu thông tin này không được bảo vệ bằng chứng chỉ SSL, bất kỳ máy tính nào ở giữa đường truyền đều có thể xem được. Điều này có nghĩa là thông tin như số thẻ tín dụng, tên đăng nhập và mật khẩu, cũng như các thông tin quan trọng khác có thể bị lộ ra ngoài.

- Cung cấp tính xác thực

Ngoài việc bảo mặt bằng cách biến thông tin thành ngôn ngữ bí mật, chứng nhận SSL còn đảm bảo rằng bạn đang gửi thông tin đến máy chủ đúng đích, chứ không phải tới một kẻ giả mạo đang cố gắng lừa đảo hoặc đánh cắp thông tin. Những nhà cung cấp SSL đáng tin cậy sẽ đặt điều kiện cho các công ty phải xác minh danh tính của họ trước khi nhận chứng chỉ SSL.

- Tăng uy tín website

Khi bạn truy cập một trang web an toàn, trình duyệt web thường sẽ hiển thị cho bạn những biểu tượng khóa hoặc một thành màu xanh lá cây để cho biết kết nối đang được bảo mật. Những dấu hiệu này giúp người dùng cảm thấy an tâm hơn và tin tưởng trang web.

- Tăng độ tin tưởng của người truy cập

HTTPS cũng giúp ngăn chặn các cuộc tấn công lừa đảo và các email gửi từ tội phạm giả mạo trang web của bạn. Trong các email này, thường sẽ có một liên kết dẫn bạn đến trang web của tội phạm hoặc họ có thể sử dụng chiêu thức "Man-in-the-middle" (tên tội phạm sẽ lừa người dùng gửi thông tin nhạy cảm trực tiếp cho họ) trên tên miền của trang web giả mạo.

- Bảo mật thanh toán (PCI Compliance)

Để cho phép người dùng nhập thông tin thẻ tín dụng trên trang web, bạn phải qua một loạt các kiểm tra để chứng minh rằng bạn tuân thủ các tiêu chuẩn an toàn khi thanh toán bằng thẻ, gọi là Payment Card Industry (PCI). Để đủ tiêu chuẩn, đương nhiên bạn cần sử dụng chứng chỉ SSL. Khi đó, các thông tin thẻ tín dụng của người dùng sẽ được bảo mật và an toàn khi họ thực hiện giao dịch trực tuyến.

- Tối ưu SEO

Google đã thông báo rằng việc sử dụng HTTPS sẽ ảnh hưởng đến việc xếp hạng trang web. Khi hiển thị kết quả tìm kiếm, các trang web có SSL sẽ được đặt ưu tiên hơn so với các trang web tương tự nhưng không có SSL. Điều này giúp cải thiện tính bảo mật và độ tin cậy của trang web, đồng thời tạo điều kiện tốt hơn cho người dùng khi tìm kiếm thông tin trực tuyến.

4.4 Nhược điểm SSl

- Tăng tải cho máy chủ: Mã hóa và giải mã dữ liệu trong quá trình truyền thông qua SSL có thể tốn thêm tài nguyên máy chủ, gây ra một chút tăng tải và làm chậm quá trình truyền dữ liệu.

- Chi phí: Một số chứng chỉ SSL có giá cao, đặc biệt là các chứng chỉ mở rộng như EV SSL. Điều này có thể tạo ra một chi phí đáng kể đối với các tổ chức nhỏ hoặc cá nhân muốn bảo mật trang web của mình.
- Đòi hỏi quá trình xác thực: Việc xác thực chứng chỉ SSL có thể đòi hỏi một số thủ tục phức tạp và mất thời gian, đặc biệt là đối với các loại chứng chỉ cao cấp như EV SSL.
- Khả năng tấn công trung gian: Một số hình thức tấn công trung gian như tấn công Man-in-the-Middle (MITM) có thể xảy ra trong quá trình thiết lập kết nối SSL nếu không được triển khai đúng cách. Điều này có thể đe dọa tính bảo mật của dữ liệu truyền qua SSL.
- Hạn chế của chứng chỉ tự ký: Chứng chỉ tự ký (self-signed) không được phê duyệt bởi một cơ quan xác thực đáng tin cậy, do đó, trình duyệt web sẽ cảnh báo người dùng về tính bảo mật không đáng tin cậy của chứng chỉ này.

II. Các lớp IP không gian Private IP Address

1. Các lớp IP

Bảng phân tích cấu tạo của địa chỉ IP

Các lớp địa chỉ IP	Mô tả
Lớp A	Bao gồm các địa chỉ IP có giá trị oc-tet đầu tiên nằm trong khoảng từ 1 đến 126 và dải địa chỉ trải dài từ 128.1.0.0 đến 191.254.0.0. Phân bổ chủ yếu cho các tổ chức lớn trên toàn cầu.
Lớp B	Bao gồm các địa chỉ IP có giá trị oc-tet đầu tiên nằm trong khoảng từ 128 đến 191. và dải địa chỉ trải dài từ 128.1.0.0 đến 191.254.0.0. Phân bổ chủ yếu cho các tổ chức tầm trung trên toàn cầu.
Lớp C	Bao gồm các địa chỉ IP có giá trị oc-tet đầu tiên nằm trong khoảng từ 192 đến 223 và dải địa chỉ trải dài từ 192.0.1.0 đến 223.255.254.0 Phân bổ chủ yếu cho các tổ chức nhỏ trên toàn cầu.
Lớp D	Bao gồm các địa chỉ IP có giá trị oc-tet đầu tiên trong khoảng từ 224 đến 239, với 4 bit đầu tiên cố định là 1110 và dải địa chỉ trải dài từ 224.0.0.0 đến 239.255.255.255. Dành riêng cho multicast hoặc broadcast.
Lớp E	Bao gồm các địa chỉ IP có giá trị oc-tet đầu tiên nằm trong khoảng từ 240 đến 255 và sải địa chỉ trải dài từ 240.0.0.0 đến 254.255.255.255. Chỉ dành cho việc nghiên cứu.

Trong thực tế, các địa chỉ IP lớp A B C được sử dụng phổ biến để thiết lập cho các thiết bị trong mạng. Địa chỉ lớp D thường dành riêng cho các ứng dụng truyền thông đa phương tiện. Còn địa chỉ lớp E vẫn đang được thử nghiệm và dự trữ cho các mục đích phát triển trong tương lai.

Ngoài ra, còn có một lớp đặc biệt gọi là Loopback, được đại diện bởi địa chỉ 127.x.x.x. Lớp này được sử dụng để kiểm tra vòng lặp quy hồi (loopback) và chỉ được sử dụng trong nội bộ của thiết bị.

1.2 Các phiên bản IP

Địa chỉ IP bao gồm 2 phiên bản là IPv4 và IPv6:

- IPv4 (Internet Protocol version 4) là một phiên bản của giao thức Internet Protocol (IP), được thiết kế để xác định và gửi dữ liệu giữa các thiết bị trên Internet. IPv4 là phiên bản chính thức đầu tiên của IP và là nền tảng cơ bản cho việc kết nối mạng trên toàn cầu.
- IPv6 (Internet Protocol version 6) là một phiên bản của giao thức Internet Protocol (IP), được thiết kế để thay thế và mở rộng IPv4 (Internet Protocol version 4), phiên bản IP trước đó. IPv6 được phát triển để giải quyết vấn đề cạn kiệt địa chỉ IP duy nhất của IPv4 do sự mở rộng nhanh chóng của Internet và sự gia tăng số lượng thiết bị kết nối.

1.3 Phân loại IP

Bảng phân loại IP

Phân loại IP	Mô tả
`IP Public`	IP public, còn được gọi là IP công cộng, là địa chỉ mạng được cung cấp bởi nhà cung cấp dịch vụ internet. Đây là địa chỉ mà các mạng gia đình hoặc doanh nghiệp sử dụng để kết nối với các thiết bị khác trên internet. IP public cho phép các thiết bị trong mạng truy cập vào web và giao tiếp trực tiếp với các máy tính khác.
`IP PRIVATE`	IP private, hay còn được gọi là IP riêng, được sử dụng trong mạng LAN nội bộ. Khác với public IP, private IP không thể kết nối với mạng Internet. Chỉ có các thiết bị máy tính, máy in,... trong mạng cục bộ mới có thể giao tiếp với nhau thông qua router. IP private có thể được cấp tự động bởi bộ định tuyến hoặc được thiết lập thủ công
`IP Static`	IP static (hay địa chỉ IP tĩnh), là cách đặt IP thủ công cho từng thiết bị một và không thay đổi theo thời gian. Điều này đảm bảo rằng địa chỉ IP của thiết bị sẽ không thay đổi và luôn được nhận dạng cố định trên mạng.
`IP Dynamic`	IP dynamic (hay địa chỉ IP động) là IP có thể thay đổi từ một địa chỉ này sang địa chỉ khác. Quá trình thay đổi này hoàn toàn tự động và được quản lý bởi máy chủ DHCP (Dynamic Host Configuration Protocol). Điều này cho phép tối ưu việc quản lý và phân phối IP trong mạng.

III. Ý nghĩa của các công cụ

1. Ping

Công cụ kiểm tra kết nối mạng giữa hai thiết bị, dựa trên giao thức ICMP. Nó gửi gói tin ICMP Echo Request và nhận phản hồi ICMP Echo Reply. Kiểm tra xem một thiết bị từ xa có sẵn và có thể truy cập qua mạng hay

không, và đo độ trễ (latency).

```
root@47263:~# ping vinahost.vn
PING vinahost.vn (123.30.136.228) 56(84) bytes of data.
64 bytes from kingcorp.net (123.30.136.228): icmp_seq=1 ttl=63 time=0.325 ms
64 bytes from kingcorp.net (123.30.136.228): icmp_seq=2 ttl=63 time=0.431 ms
64 bytes from kingcorp.net (123.30.136.228): icmp_seq=3 ttl=63 time=0.380 ms
64 bytes from kingcorp.net (123.30.136.228): icmp_seq=4 ttl=63 time=0.426 ms
^C
--- vinahost.vn ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 0.325/0.390/0.431/0.042 ms
root@47263:~#
```

Trong đó:

- `PING vinahost.vn (123.30.136.228) 56(84) bytes of data` Dòng này cho biết bạn đang ping đến vinahost.vn, có địa chỉ IP là 123.30.136.228. Thông tin 56(84) bytes cho biết kích thước của gói tin được gửi đi (56 byte) và kích thước của toàn bộ dữ liệu được gửi đi (bao gồm cả header, 84 byte)
- `64 bytes from kingcorp.net (123.30.136.228) icmp_seq=1 ttl=63 time=0.325 ms` Dòng này hiển thị kết quả của gói tin đầu tiên được gửi đi.
    - 64 bytes from kingcorp.net: Cho biết gói tin đã được nhận lại, có kích thước 64 byte và đến từ kingcorp.net `Lưu ý` rằng kingcorp.net có thể là tên miền của máy chủ VinaHost mà bạn đang ping, hoặc là một máy chủ trung gian trên đường đi
    - icmp_seq=1: Số thứ tự của gói tin (gói tin đầu tiên)
    - ttl=128: Thời gian sống (Time To Live) của gói tin còn lại 128 lần nhảy (hop) giữa các router trước khi bị loại bỏ
    - time=6.26 ms: Thời gian tính bằng mili giây để gói tin đi và về (round-trip time - RTT). Các dòng tiếp theo giải thích tương tự
- `--- vinahost.vn ping statistics ---` Dòng này bắt đầu phần tóm tắt thống kê của quá trình ping.
- `4 packets transmitted, 4 received, 0% packet loss, time 3004ms -`
    - 4 packets transmitted: Tổng số gói tin được gửi đi là 4
    - 4 received: Tổng số gói tin nhận được phản hồi là 8 (tất cả các gói đều được nhận lại)
    - 0% packet loss: Tỷ lệ mất gói tin là 0%, nghĩa là không có gói tin nào bị mất trên đường đi
    - time3004ms: Tổng thời gian thực hiện lệnh ping là 7837 mili giây (khoảng 3.04 giây)
- `rtt min/avg/max/mdev = 0.325/0.390/0.431/0.042 ms`
    - rtt: Thời gian phản hồi (round-trip time)
    - min: Thời gian phản hồi thấp nhất là 0.325 mili giây
    - avg: Thời gian phản hồi trung bình là 0.390 mili giây
    - max: Thời gian phản hồi cao nhất là 0.431 mili giây
    - mdev: Độ lệch chuẩn của thời gian phản hồi là 0.042 mili giây

2. Telnet

Công cụ dùng để kiểm tra kết nối TCP đến một máy chủ cụ thể trên một cổng nào đó. Được sử dụng để kết nối tới các dịch vụ dựa trên TCP, chẳng hạn như HTTP, FTP, SMTP, hoặc kiểm tra xem một cổng có mở hay không.

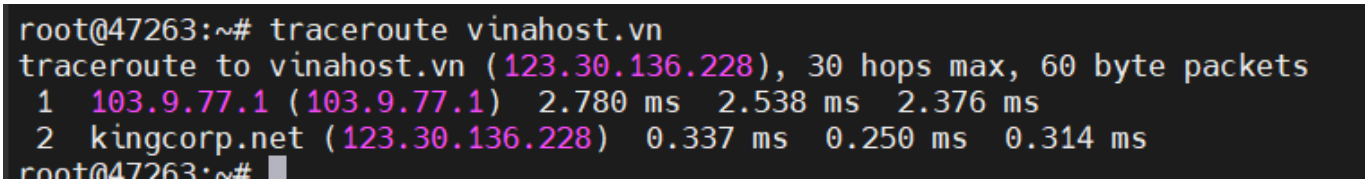
Cú pháp: `telnet <dia_chi_ip> 23`

3. Traceroute

Công cụ cho phép theo dõi hành trình của một gói tin từ thiết bị nguồn đến đích qua mạng, và xem các router trung gian mà gói tin đã đi qua. Chẩn đoán sự cố kết nối, phát hiện điểm tắc nghẽn hoặc các router bị hỏng trên đường đi.

Cú pháp:

- Trên `Windows`: tracert vinahost.vn
- Trên `Kali`: traceroute vinahost.vn

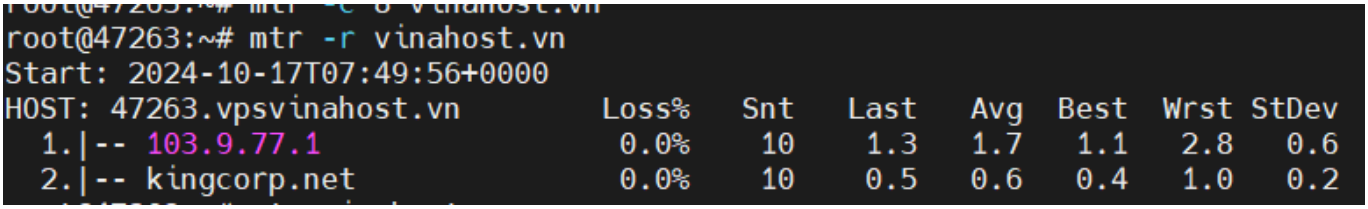


Trong đó:

- `traceroute to vinahost.vn (123.30.136.228), 30 hops max, 60 byte packets`
  - traceroute to vinahost.vn: Đây là lệnh được sử dụng để theo dõi đường đi đến vinahost.vn
  - (123.30.136.228): Đây là địa chỉ IP của máy chủ VinaHost
  - 30 hops max: Đây là số lần nhảy tối đa mà gói tin có thể thực hiện trước khi bị loại bỏ
  - 60 byte packets: Kích thước của các gói tin được sử dụng trong quá trình traceroute là 60 byte
- `1 103.9.77.1 (103.9.77.1) 2.780 ms 2.538 ms 2.376 ms`
  - 1: Đây là lần nhảy đầu tiên. Gói tin đã đến một máy chủ có địa chỉ IP 103.9.77.1
  - 103.9.77.1 (103.9.77.1): Tên miền (nếu có) và địa chỉ IP của máy chủ
  - 2.780 ms 2.538 ms 2.376 ms: Thời gian (tính bằng mili giây) cần thiết để gói tin đến đích ở lần nhảy này. Ba số đại diện cho ba lần gửi thử của gói tin, cho thấy độ trễ dao động nhẹ
- `2 kingcorp.net (123.30.136.228) 0.337 ms 0.250 ms 0.314 ms`
  - 2: Đây là lần nhảy thứ hai (đích đến). Gói tin đã đến máy chủ đích kingcorp.net, có địa chỉ IP 123.30.136.228. Đây cũng là địa chỉ IP của VinaHost mà bạn đang ping
  - kingcorp.net (123.30.136.228): Tên miền và địa chỉ IP của máy chủ đích
  - 0.337 ms 0.250 ms 0.314 ms: Thời gian phản hồi của gói tin ở lần nhảy này rất thấp, chỉ nằm trong khoảng từ 0.250 ms đến 0.337 ms. Điều này cho thấy kết nối đến máy chủ VinaHost là rất nhanh.

4. mtr

Một công cụ kết hợp cả ping và traceroute, hiển thị theo thời gian thực hành trình của các gói tin qua mạng và tính toán thông số hiệu suất như độ trễ, mất gói. Giám sát và phân tích hiệu suất mạng theo thời gian thực



Trong đó:

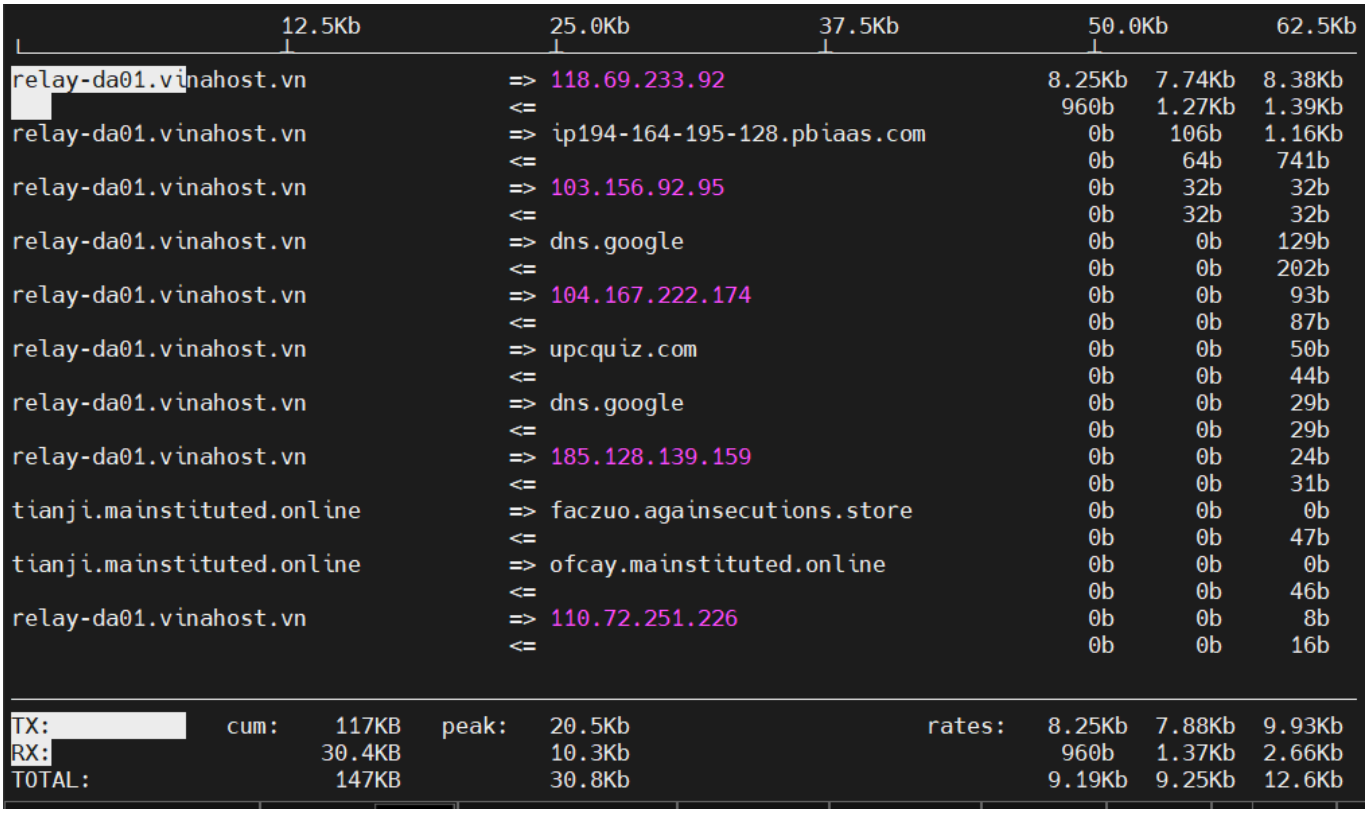
- `Start: 2024-10-17T07:49:56+0000`: Thời điểm bắt đầu thực hiện lệnh mtr. HOST: 47263.vpsvinahost.vn: Tên máy chủ của bạn (có thể là tên miền hoặc địa chỉ IP).

Các cột trong bảng:

- `HOST`: Địa chỉ IP của mỗi lần nhảy (hop) trên đường đi đến đích.
- `Loss%`: Tỷ lệ phần trăm gói tin bị mất tại mỗi lần nhảy. Trong trường hợp này, tất cả các gói tin đều đến đích (0% mất mát).
- `Snt`: Tổng số gói tin được gửi đến mỗi lần nhảy (thường là 10).
- `Last`: Thời gian phản hồi (RTT) của gói tin cuối cùng được gửi đến mỗi lần nhảy (tính bằng mili giây).
- `Avg`: Thời gian phản hồi trung bình của tất cả các gói tin được gửi đến mỗi lần nhảy.
- `Best`: Thời gian phản hồi nhanh nhất của một gói tin đến mỗi lần nhảy.
- `Wrst`: Thời gian phản hồi chậm nhất của một gói tin đến mỗi lần nhảy.
- `StDev`: Độ lệch chuẩn của thời gian phản hồi, cho biết mức độ dao động của các lần đo.

5. iftop

Công cụ giám sát lưu lượng mạng theo thời gian thực, hiển thị băng thông sử dụng của từng kết nối mạng. Giám sát lưu lượng mạng, giúp phát hiện các nguồn sử dụng nhiều băng thông hoặc các vấn đề mạng. Iftop sẽ hiển thị danh sách các tiến trình sử dụng băng thông mạng được cập nhật theo thời gian thực trung bình cứ sau 2, 10 và 40 giây.



6. iperf

Công cụ đo băng thông mạng giữa hai thiết bị, sử dụng TCP hoặc UDP. Đo tốc độ truyền dữ liệu và hiệu suất của kết nối mạng.

Cú pháp:

Trên server: `iperf -sp 2024`

- s: thiết lập server
- p: lắng nghe cổng kết nối là 2024

Trên client: `iperf -c 103.9.77.103 -p 2024`

- c: chế độ client
- 103.9.77.103: Ip server
- p: cổng 2024

```
root@47263:~# iperf -sp 2024
------------------------------------------------------------
Server listening on TCP port 2024
TCP window size:  128 KByte (default)
------------------------------------------------------------
[  1] local 103.9.77.103 port 2024 connected with 118.69.233.92 port 26682
[ ID] Interval       Transfer     Bandwidth
[  1] 0.0000-10.6098 sec  16.9 MBytes  13.3 Mbits/sec
```

Trong đó:

- `Server listening on TCP port 2024`: Máy chủ đang lắng nghe trên cổng TCP 2024, chờ nhận kết nối từ máy khách.
- `TCP window size: 128 KByte (default)`: Kích thước cửa sổ TCP mặc định là 128 KB.
- `[ 1] local 103.9.77.103 port 2024 connected with 118.69.233.92 port 26682:`
  - `[ 1]`: Đây là luồng kết nối đầu tiên (có thể có nhiều luồng nếu nhiều máy khách kết nối cùng lúc).
  - `local 103.9.77.103 port 2024`: Địa chỉ IP và cổng của máy chủ.
  - `connected with 118.69.233.92 port 26682:` Địa chỉ IP và cổng của máy khách đã kết nối.
- `[ ID] Interval Transfer Bandwidth`: Các cột này hiển thị thông tin về kết nối:
  - `ID`: Số thứ tự của luồng kết nối
  - `Interval`: Khoảng thời gian của cuộc kiểm tra (từ đầu đến cuối)
  - `Transfer`: Tổng lượng dữ liệu đã truyền trong khoảng thời gian đó
  - `Bandwidth`: Băng thông trung bình trong khoảng thời gian đó

Kết quả:

- Cuộc kiểm tra diễn ra trong khoảng thời gian 10.6098 giây. Tổng lượng dữ liệu truyền là 16.9 MB. Băng thông trung bình là 13.3 Mbits/sec.

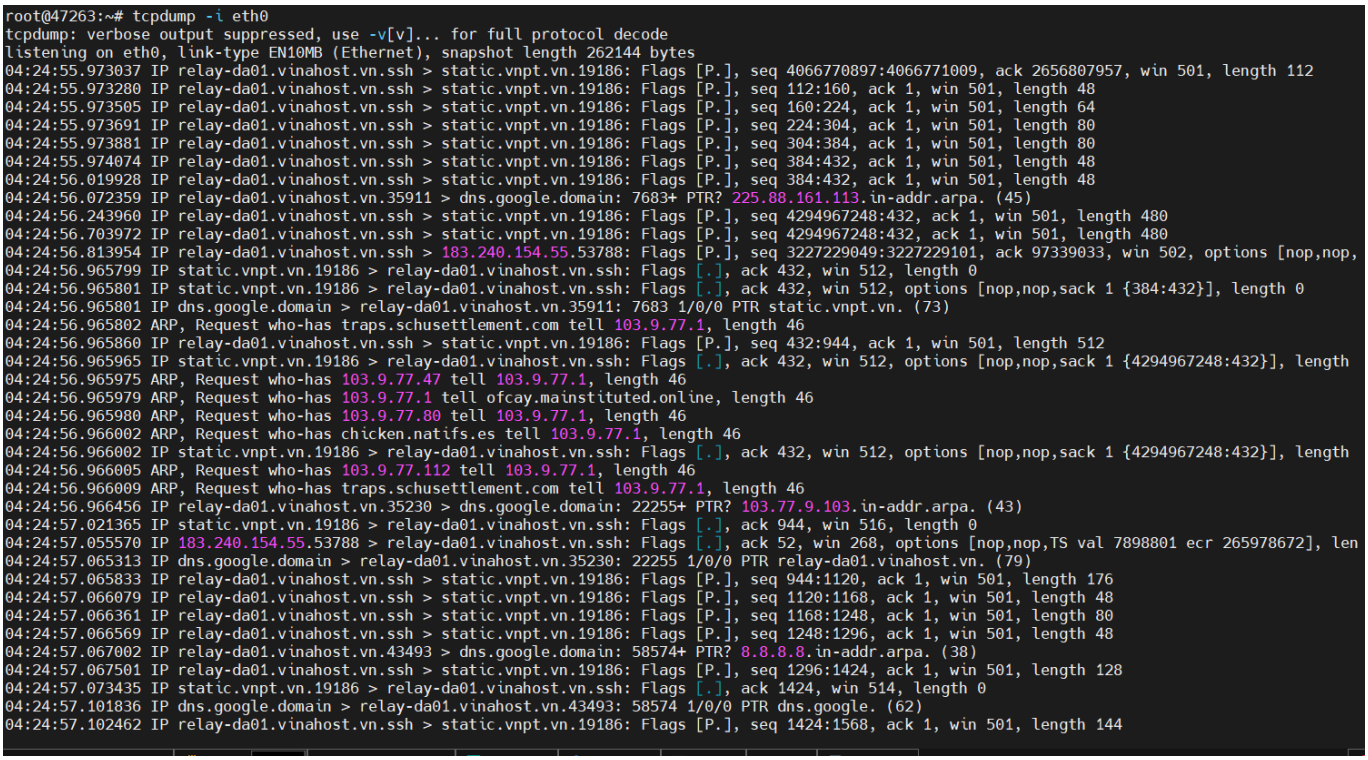
7. tcpdump

Công cụ bắt và phân tích gói tin trên mạng qua dòng lệnh. Nó cho phép xem chi tiết nội dung của các gói tin qua mạng. Gỡ lỗi kết nối, phân tích luồng mạng, hoặc kiểm tra lưu lượng dữ liệu.

Cú pháp:

```
tcpdump -i eth0
```

- i: Tên card mạng (có thể dùng lệnh ip a kiểm tra)



```
root@47263:~# tcpdump -i eth0
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
04:24:55.973037 IP relay-da01.vinahost.vn.ssh > static.vnpt.vn.19186: Flags [P.], seq 4066770897:4066771009, ack 2656807957, win 501, length 112
04:24:55.973280 IP relay-da01.vinahost.vn.ssh > static.vnpt.vn.19186: Flags [P.], seq 112:160, ack 1, win 501, length 48
04:24:55.973505 IP relay-da01.vinahost.vn.ssh > static.vnpt.vn.19186: Flags [P.], seq 160:224, ack 1, win 501, length 64
04:24:55.973691 IP relay-da01.vinahost.vn.ssh > static.vnpt.vn.19186: Flags [P.], seq 224:304, ack 1, win 501, length 80
04:24:55.973881 IP relay-da01.vinahost.vn.ssh > static.vnpt.vn.19186: Flags [P.], seq 304:384, ack 1, win 501, length 80
04:24:55.974074 IP relay-da01.vinahost.vn.ssh > static.vnpt.vn.19186: Flags [P.], seq 384:432, ack 1, win 501, length 48
04:24:56.019928 IP relay-da01.vinahost.vn.ssh > static.vnpt.vn.19186: Flags [P.], seq 384:432, ack 1, win 501, length 48
04:24:56.072359 IP relay-da01.vinahost.vn.35911 > dns.google.domain: 7683+ PTR? 225.88.161.113.in-addr.arpa. (45)
04:24:56.243960 IP relay-da01.vinahost.vn.ssh > static.vnpt.vn.19186: Flags [P.], seq 4294967248:432, ack 1, win 501, length 480
04:24:56.703972 IP relay-da01.vinahost.vn.ssh > static.vnpt.vn.19186: Flags [P.], seq 4294967248:432, ack 1, win 501, length 480
04:24:56.813954 IP relay-da01.vinahost.vn.ssh > 183.240.154.55.53788: Flags [P.], seq 3227229049:3227229101, ack 97339033, win 502, options [nop,nop,
04:24:56.965799 IP static.vnpt.vn.19186 > relay-da01.vinahost.vn.ssh: Flags [.], ack 432, win 512, length 0
04:24:56.965801 IP static.vnpt.vn.19186 > relay-da01.vinahost.vn.ssh: Flags [.], ack 432, win 512, options [nop,nop,sack 1 {384:432}], length 0
04:24:56.965801 IP dns.google.domain > relay-da01.vinahost.vn.35911: 7683 1/0/0 PTR static.vnpt.vn. (73)
04:24:56.965802 ARP, Request who-has traps.schusettlement.com tell 103.9.77.1, length 46
04:24:56.965860 IP relay-da01.vinahost.vn.ssh > static.vnpt.vn.19186: Flags [P.], seq 432:944, ack 1, win 501, length 512
04:24:56.965965 IP static.vnpt.vn.19186 > relay-da01.vinahost.vn.ssh: Flags [.], ack 432, win 512, options [nop,nop,sack 1 {4294967248:432}], length
04:24:56.965975 ARP, Request who-has 103.9.77.47 tell 103.9.77.1, length 46
04:24:56.965979 ARP, Request who-has 103.9.77.1 tell ofcay.mainstituted.online, length 46
04:24:56.965980 ARP, Request who-has 103.9.77.80 tell 103.9.77.1, length 46
04:24:56.966002 ARP, Request who-has chicken.natifs.es tell 103.9.77.1, length 46
04:24:56.966002 IP static.vnpt.vn.19186 > relay-da01.vinahost.vn.ssh: Flags [.], ack 432, win 512, options [nop,nop,sack 1 {4294967248:432}], length
04:24:56.966005 ARP, Request who-has 103.9.77.112 tell 103.9.77.1, length 46
04:24:56.966009 ARP, Request who-has traps.schusettlement.com tell 103.9.77.1, length 46
04:24:56.966456 IP relay-da01.vinahost.vn.35230 > dns.google.domain: 22255+ PTR? 103.77.9.103.in-addr.arpa. (43)
04:24:57.021365 IP static.vnpt.vn.19186 > relay-da01.vinahost.vn.ssh: Flags [.], ack 944, win 516, length 0
04:24:57.055570 IP 183.240.154.55.53788 > relay-da01.vinahost.vn.ssh: Flags [.], ack 52, win 268, options [nop,nop,TS val 7898801 ecr 265978672], len
04:24:57.065313 IP dns.google.domain > relay-da01.vinahost.vn.35230: 22255 1/0/0 PTR relay-da01.vinahost.vn. (79)
04:24:57.065833 IP relay-da01.vinahost.vn.ssh > static.vnpt.vn.19186: Flags [P.], seq 944:1120, ack 1, win 501, length 176
04:24:57.066079 IP relay-da01.vinahost.vn.ssh > static.vnpt.vn.19186: Flags [P.], seq 1120:1168, ack 1, win 501, length 48
04:24:57.066361 IP relay-da01.vinahost.vn.ssh > static.vnpt.vn.19186: Flags [P.], seq 1168:1248, ack 1, win 501, length 80
04:24:57.066569 IP relay-da01.vinahost.vn.ssh > static.vnpt.vn.19186: Flags [P.], seq 1248:1296, ack 1, win 501, length 48
04:24:57.067002 IP relay-da01.vinahost.vn.43493 > dns.google.domain: 58574+ PTR? 8.8.8.8.in-addr.arpa. (38)
04:24:57.067501 IP relay-da01.vinahost.vn.ssh > static.vnpt.vn.19186: Flags [P.], seq 1296:1424, ack 1, win 501, length 128
04:24:57.073435 IP static.vnpt.vn.19186 > relay-da01.vinahost.vn.ssh: Flags [.], ack 1424, win 514, length 0
04:24:57.101836 IP dns.google.domain > relay-da01.vinahost.vn.43493: 58574 1/0/0 PTR dns.google. (62)
04:24:57.102462 IP relay-da01.vinahost.vn.ssh > static.vnpt.vn.19186: Flags [P.], seq 1424:1568, ack 1, win 501, length 144
```

Trong đó:

- `listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes`: Đang lắng nghe trên giao diện eth0, và mạng sử dụng loại kết nối Ethernet (EN10MB). Mỗi gói tin bắt được sẽ có độ dài tối đa là 262144 bytes
- Gói đầu tiên `04:24:55.973037 IP relay-da01.vinahost.vn.ssh > static.vnpt.vn.19186: Flags [P.], seq 4066770897:4066771009, ack 2656807957, win 501, length 112`:
  - `04:24:55.973037`: Thời gian gói tin được bắt
  - `IP`: Gói tin sử dụng giao thức IP
  - `relay-da01.vinahost.vn.ssh > static.vnpt.vn.19186`: Gói tin này được gửi từ máy chủ relay-da01.vinahost.vn sử dụng giao thức SSH đến máy static.vnpt.vn, trên cổng 19186
  - `Flags [P.]`: Cờ PUSH (P) được bật, nghĩa là dữ liệu đang được đẩy tới đích mà không cần đợi thêm dữ liệu khác. Dấu chấm (.) là cờ ACK (Acknowledgment) xác nhận nhận gói tin từ phía đối tác
  - `seq 4066770897:4066771009`: Đây là dải số sequence number của các byte dữ liệu trong gói tin này, từ 4066770897 đến 4066771009
  - `ack 2656807957`: Đây là ACK number xác nhận rằng máy gửi đã nhận được tất cả các byte dữ liệu từ phía đối tác với sequence number lên đến 2656807957
  - `win 501`: Đây là cửa sổ TCP (window size), cho biết số lượng byte dữ liệu mà máy gửi có thể nhận mà không cần đợi xác nhận thêm
  - `length 112`: Gói tin có 112 byte dữ liệu
- `Gói tiếp theo tương tự`: Các gói tin tiếp theo có cú pháp tương tự, chỉ khác về các giá trị thời gian, địa chỉ, cổng, cờ, số sequence, và số acknowledgment. Các cờ TCP có thể xuất hiện gồm:
  - SYN (S): Được sử dụng để khởi tạo một kết nối TCP
  - ACK (.): Xác nhận dữ liệu đã nhận
  - FIN (F): Yêu cầu ngắt kết nối
  - PUSH (P): Đẩy dữ liệu ngay lập tức
  - RST (R): Reset kết nối
  - URG (U): Dữ liệu khẩn cấp

- Gói tin truy vấn DNS `04:24:56.072359 IP relay-da01.vinahost.vn.35911 > dns.google.domain: 7683+ PTR? 225.88.161.113.in-addr.arpa. (45)`
  - `relay-da01.vinahost.vn.35911 > dns.google.domain`: Gói tin này từ máy chủ relay-da01.vinahost.vn với cổng 35911 đến máy chủ DNS của Google (8.8.8.8), sử dụng cổng domain (53)
  - `7683+ PTR? 225.88.161.113.in-addr.arpa.`: Đây là một truy vấn DNS, yêu cầu một bản ghi PTR để tra cứu tên miền ngược từ địa chỉ IP 113.161.88.225
- Gói tin ARP `04:24:56.965802 ARP, Request who-has traps.schusettlement.com tell 103.9.77.1, length 46`:
  - `ARP`: Gói tin này sử dụng giao thức ARP (Address Resolution Protocol)
  - `Request who-has traps.schusettlement.com tell 103.9.77.1`: Máy chủ 103.9.77.1 đang yêu cầu địa chỉ MAC của traps.schusettlement.com
  - `engths 46`: Gói tin ARP này có độ dài 46 byte
- `Các thông tin khác`:
  - SSH traffic: Giao thức SSH được sử dụng trong nhiều gói tin. Các cờ P. xuất hiện nhiều lần, có nghĩa là các dữ liệu đang được đẩy đi qua một phiên SSH đã được thiết lập
  - ACK và SACK: Một số gói tin chứa cờ ACK để xác nhận dữ liệu đã nhận, và có một số gói có SACK (Selective Acknowledgment) chỉ định các đoạn dữ liệu cụ thể được nhận

8. ss

Công cụ thay thế cho netstat, giúp hiển thị thông tin chi tiết về các kết nối socket (TCP, UDP) trên hệ thống. Theo dõi và quản lý các kết nối mạng hiện có, kiểm tra tình trạng của các socket và dịch vụ đang chạy

Cú pháp: `ss`

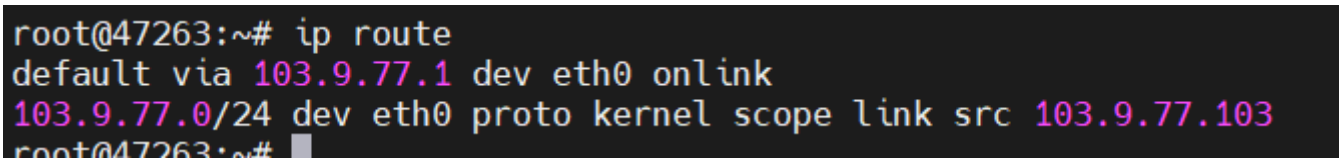
```
root@47263:~# ss
Netid       State       Recv-Q       Send-Q                          Local Address:Port                      Peer Address:Port          Process
u_dgr       ESTAB       0            0                        /run/systemd/notify 17916                              * 0
u_dgr       ESTAB       0            0                     /run/systemd/journal/dev-log 17942                        * 0
u_dgr       ESTAB       0            0                      /run/systemd/journal/socket 17944                        * 0
u_dgr       ESTAB       0            0                                          * 1278160                              * 17942
u_str       ESTAB       0            0                                          * 1278137                              * 1278138
u_str       ESTAB       0            0                                          * 19806                                * 19813
u_str       ESTAB       0            0                      /run/systemd/journal/stdout 18729                      * 18728
u_str       ESTAB       0            0                      /run/dbus/system_bus_socket 20091                      * 19810
u_str       ESTAB       0            0                                          * 1278074                              * 1278075
u_str       ESTAB       0            0                      /run/systemd/journal/stdout 18610                      * 18259
u_str       ESTAB       0            0                                          * 1278157                              * 1278156
u_str       ESTAB       0            0                                          * 20177                                * 20182
u_dgr       ESTAB       0            0                                          * 20159                                * 17944
u_str       ESTAB       0            0                      /run/systemd/journal/stdout 18611                      * 18285
u_str       ESTAB       0            0                                          * 1278039                              * 1278038
u_str       ESTAB       0            0                      /run/systemd/journal/stdout 19931                      * 19882
u_str       ESTAB       0            0                                          * 18728                                * 18729
u_str       ESTAB       0            0                                          * 93705                                * 93709
u_dgr       ESTAB       0            0                                          * 18966                                * 18967
u_str       ESTAB       0            0                      /run/systemd/journal/stdout 19873                      * 19865
u_dgr       ESTAB       0            0                                          * 18964                                * 18965
u_dgr       ESTAB       0            0                                          * 18344                                * 18343
u_str       ESTAB       0            0                                          * 93726                                * 93727
u_str       ESTAB       0            0                      /run/systemd/journal/stdout 20002                      * 19917
u_str       ESTAB       0            0                      /run/systemd/journal/stdout 19310                      * 19309
u_str       ESTAB       0            0                                          * 20087                                * 20088
u_str       ESTAB       0            0                                          * 1256135                              * 0
u_str       ESTAB       0            0                                          * 19899                                * 19982
u_dgr       ESTAB       0            0                                          * 19331                                * 19332
u_str       ESTAB       0            0                                          * 20337                                * 20338
u_dgr       ESTAB       0            0                                          * 20086                                * 17942
u_str       ESTAB       0            0                                          * 19309                                * 19310
u_str       ESTAB       0            0                      /run/dbus/system_bus_socket 20338                      * 20337
u_dgr       ESTAB       0            0                                          * 1256153                              * 1256154
u_dgr       ESTAB       0            0                                          * 19997                                * 17942
u_str       ESTAB       0            0                      /run/dbus/system_bus_socket 20089                      * 19760
u_str       ESTAB       0            0                      /run/dbus/system_bus_socket 20093                      * 20030
u_dgr       ESTAB       0            0                                          * 20236                                * 17942
```

Trong đó:

- `Netid (Network ID)`: Giao thức kết nối mạng cục bộ, thường được sử dụng bởi các ứng dụng hệ thống.
    - `u_str`: Unix Stream Socket
    - `u_dgr`: Unix Datagram Socket
- `State`: Trạng thái của kết nối. Các trạng thái thông thường bao gồm:
    - `ESTAB (Established)`: Kết nối đã được thiết lập và đang hoạt động
- `Recv-Q (Receive Queue)`: Số lượng byte trong hàng đợi nhận của socket. Nếu số lượng này lớn hơn 0, có thể hệ thống đang gặp vấn đề xử lý dữ liệu đến chậm
- `Send-Q (Send Queue)`: Số lượng byte trong hàng đợi gửi của socket. Giá trị này lớn hơn 0 có thể cho thấy hệ thống không thể gửi dữ liệu đủ nhanh
- `Local Address`: Địa chỉ cục bộ và cổng của kết nối
- `Peer Address`: Địa chỉ và cổng của đối tác từ xa mà hệ thống của bạn kết nối đến. Khi địa chỉ này là *, nó biểu thị rằng đây là một socket cục bộ, không có kết nối từ xa
- `Process`: Tên và ID của tiến trình liên quan đến kết nối (nếu có)
    - `/run/systemd/journal/stdout`: Đây là socket được sử dụng bởi systemd, dịch vụ quản lý hệ thống của Linux, để ghi log
    - `/run/dbus/system_bus_socket`: Socket của DBus, hệ thống liên lạc giữa các tiến trình

9. ip route

Công cụ này quản lý bảng định tuyến (routing table) của hệ thống. Hiển thị và quản lý bảng định tuyến, thêm, sửa hoặc xóa các tuyến đường trong mạng.



Cú pháp: `ip route`

Trong đó:

- `default via 103.9.77.1 dev eth0 onlink`
    - `default`: Đây là tuyến đường mặc định (default route), nghĩa là khi hệ thống không tìm thấy tuyến đường cụ thể cho một địa chỉ IP đích, nó sẽ gửi lưu lượng đến tuyến đường này
    - `via 103.9.77.1`: Địa chỉ của cổng (gateway) mà hệ thống sẽ sử dụng để gửi các gói tin đi. Trong trường hợp này, cổng có địa chỉ IP là 103.9.77.1
    - `dev eth0`: Chỉ định giao diện mạng eth0 để truyền dữ liệu. Đây là giao diện vật lý hoặc ảo mà gói tin sẽ đi qua
    - `onlink`: Tùy chọn này cho hệ điều hành biết rằng cổng (103.9.77.1) có thể tiếp cận được trực tiếp từ giao diện này, ngay cả khi hệ thống không tìm thấy cổng đó trong ARP (Address Resolution Protocol). Nó có thể dùng để tạo ra tuyến đường không yêu cầu ARP hoặc các thông tin khác từ router
- `103.9.77.0/24 dev eth0 proto kernel scope link src 103.9.77.103`
    - `103.9.77.0/24`: Đây là một tuyến đường con, tức là phạm vi mạng con này bao gồm tất cả các địa chỉ IP từ 103.9.77.0 đến 103.9.77.255 (với mặt nạ mạng /24 nghĩa là 255.255.255.0)
    - `dev eth0`: Tất cả các gói tin gửi đến mạng con 103.9.77.0/24 sẽ đi qua giao diện mạng eth0

- `proto kernel`: Điều này cho biết rằng tuyến đường này đã được thêm bởi hệ điều hành (kernel) khi giao diện eth0 được cấu hình với một địa chỉ IP
- `scope link`: Phạm vi của tuyến đường này là "link-local", nghĩa là nó chỉ hợp lệ cho các máy nằm trên cùng một mạng vật lý hoặc cục bộ, tức là không cần phải đi qua cổng (gateway)
- `src 103.9.77.103`: Địa chỉ IP nguồn của giao diện eth0. Khi gửi gói tin từ giao diện này, hệ thống sẽ sử dụng địa chỉ IP 103.9.77.103 làm nguồn

10. ip -s link

Công cụ hiển thị các thông tin chi tiết về giao diện mạng (network interface), bao gồm số lượng byte đã truyền và nhận, số lượng gói tin đã bị lỗi (errors), và các thông tin thống kê khác. Đây là một công cụ có sẵn trên hầu hết các hệ điều hành Linux

Cú pháp: `ip link show [tên_card_mạng]`

```
root@47263:~# ip -s link show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT gr
oup default qlen 1000
    link/ether 00:16:3e:e3:97:b5 brd ff:ff:ff:ff:ff:ff
    RX:  bytes packets errors dropped  missed   mcast
     468131263 6066012      0       0       0    4018
    TX:  bytes packets errors dropped carrier collsns
     156543231  962197      0       0       0       0
    altname enp0s18
    altname ens18
```

Trong đó:

- `2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000`:
  - `2`: eth0: Số thứ tự của giao diện mạng và tên giao diện (eth0). Trong hệ thống có thể có nhiều giao diện mạng, và mỗi giao diện sẽ được đánh số và đặt tên khác nhau (eth0, eth1, wlan0,...)
  - `<BROADCAST, MULTICAST, UP, LOWER_UP>`: Các thuộc tính của giao diện mạng:
    - `BROADCAST`: Giao diện này hỗ trợ truyền broadcast (gửi gói tin tới tất cả thiết bị trong mạng)
    - `MULTICAST:` Hỗ trợ truyền multicast (gửi gói tin tới một nhóm thiết bị trong mạng)
    - `UP`: Giao diện này đang hoạt động
    - `LOWER_UP`: Có tín hiệu kết nối vật lý (có thể là kết nối cáp Ethernet hoặc mạng không dây đang hoạt động)
  - `mtu 1500`: Maximum Transmission Unit, kích thước tối đa của một gói tin có thể truyền qua giao diện này (1.500 byte). Đây là giá trị phổ biến cho Ethernet
  - `qdisc fq_codel`: Loại thuật toán hàng đợi (queueing discipline) được sử dụng cho việc quản lý hàng đợi gói tin. Ở đây là fq_codel (Fair Queuing Controlled Delay), một thuật toán chống tắc nghẽn hàng đợi hiệu quả, giúp giảm độ trễ
  - `state UP`: Trạng thái của giao diện là "UP", có nghĩa là nó đang hoạt động
  - `mode DEFAULT`: Chế độ hoạt động của giao diện mạng là mặc định
  - `group default`: Giao diện này thuộc nhóm "default"
  - `qlen 1000`: Độ dài hàng đợi gói tin (queue length), tức là số lượng gói tin có thể chờ được xử lý trước khi bị loại bỏ. Ở đây, độ dài hàng đợi là 1.000 gói tin.
- Thông tin về địa chỉ MAC `link/ether 00:16:3e:e3:97:b5 brd ff:ff:ff:ff:ff:ff`:

    - `link/ether`: Đây là địa chỉ MAC của giao diện mạng, một định danh vật lý của thiết bị mạng. Địa chỉ MAC này là 00:16:3e:e3:97
    - `brd ff:ff:ff:ff:ff`:Địa chỉ broadcast, địa chỉ này được dùng để gửi dữ liệu tới tất cả các thiết bị trong mạng LAN. Địa chỉ broadcast của Ethernet luôn là ff:ff:ff:ff:ff:ff
- `RX (Receive)`: Đây là thông tin về số liệu gói tin nhận được trên giao diện này
    - `bytes`: Số byte dữ liệu đã nhận qua giao diện này, ở đây là 468,131,263 byte (~468 MB)
    - `packets`: Tổng số gói tin đã nhận, ở đây là 6,066,012 gói tin
    - `errors`: Số lượng gói tin bị lỗi khi nhận, ở đây là 0 (không có lỗi nào)
    - `dropped`: Số gói tin bị loại bỏ (drop) do lỗi bộ đệm hoặc quá tải, ở đây là 0
    - `missed`: Số gói tin bị bỏ lỡ (missed) trong quá trình nhận do card mạng không xử lý kịp, ở đây là 0
    - `mcast`: Số gói tin multicast nhận được, ở đây là 4,018 gói
- `TX (Transmit)`: Đây là thông tin về số liệu gói tin đã được gửi đi qua giao diện này. Thông số tương tự
    - `carrier`: Các lỗi liên quan đến đường truyền vật lý (carrier), chẳng hạn như lỗi tín hiệu. Số này cũng là 0, nghĩa là không có lỗi carrier.
    - `collsns`: Số va chạm (collision) trên giao diện. Trên các mạng hiện đại, số lượng này thường là 0, do các mạng dùng switch Ethernet không còn xảy ra va chạm nhiều như trước.
- `altname enp0s18 và altname ens18`: Đây là các tên khác (alternative names) của giao diện eth0, thường được cấu hình theo kiểu định danh dựa trên vị trí phần cứng hoặc cấu trúc định danh mới của hệ thống Linux, thay thế tên "eth0". Hệ thống có thể sử dụng các tên khác này thay cho eth0

11. ethtool

Công cụ để lấy thông tin chi tiết về trạng thái và các chỉ số (metrics) của card mạng, bao gồm các thống kê liên quan đến Physical Layer (tầng vật lý)

Cú pháp: `ethtool -S [tên_card_mạng]`

```
root@47263:~# ethtool -S eth0
NIC statistics:
     rx_packets: 6105402
     tx_packets: 981335
     rx_bytes: 495359251
     tx_bytes: 159174404
     rx_broadcast: 5213528
     tx_broadcast: 1
     rx_multicast: 4018
     tx_multicast: 109
     rx_errors: 0
     tx_errors: 0
     tx_dropped: 0
     multicast: 4018
     collisions: 0
     rx_length_errors: 0
     rx_over_errors: 0
     rx_crc_errors: 0
     rx_frame_errors: 0
     rx_no_buffer_count: 0
     rx_missed_errors: 0
     tx_aborted_errors: 0
     tx_carrier_errors: 0
     tx_fifo_errors: 0
     tx_heartbeat_errors: 0
     tx_window_errors: 0
     tx_abort_late_coll: 0
     tx_deferred_ok: 0
     tx_single_coll_ok: 0
     tx_multi_coll_ok: 0
     tx_timeout_count: 0
     tx_restart_queue: 0
     rx_long_length_errors: 0
     rx_short_length_errors: 0
     rx_align_errors: 0
     tx_tcp_seg_good: 141
     tx_tcp_seg_failed: 0
     rx_flow_control_xon: 0
     rx_flow_control_xoff: 0
     tx_flow_control_xon: 0
     tx_flow_control_xoff: 0
     rx_long_byte_count: 495359251
     rx_csum_offload_good: 0
     rx_csum_offload_errors: 0
     alloc_rx_buff_failed: 0
     tx_smbus: 0
     rx_smbus: 0
     dropped_smbus: 0
```

Trong đó:

- `Thống kê gói tin nhận (RX - Receive)` gồm:
  - `rx_packets: 6105402`: Tổng số gói tin đã được nhận qua giao diện eth0 là 6,105,402 gói
  - `rx_bytes: 495359251`: Số lượng byte dữ liệu đã nhận được là 495,359,251 byte (~495 MB)
  - `rx_broadcast: 5213528`: Số lượng gói tin broadcast đã nhận là 5,213,528. Các gói tin broadcast được gửi tới tất cả các thiết bị trong mạng
  - `rx_multicast: 4018`: Số lượng gói tin multicast đã nhận là 4,018. Gói tin multicast được gửi tới một nhóm cụ thể trong mạng
  - `rx_errors: 0`: Số lượng lỗi khi nhận gói tin là 0, nghĩa là không có lỗi nào xảy ra khi nhận gói
  - `rx_length_errors: 0`: Không có lỗi gói tin do chiều dài không đúng (gói tin bị quá ngắn hoặc quá dài)

    - `rx_over_errors: 0`: Không có lỗi do bộ đệm của giao diện mạng bị tràn (buffer overrun), tức là card mạng luôn xử lý kịp gói tin mà không bị tràn bộ đệm
    - `rx_frame_errors: 0`: Không có lỗi khung (frame errors), lỗi này có thể xảy ra khi kích thước gói tin vượt quá giới hạn hoặc không đồng bộ với mạng
    - `rx_no_buffer_count: 0`: Không có lỗi liên quan đến thiếu bộ đệm khi nhận dữ liệu
    - `rx_missed_errors: 0`: Không có gói tin nào bị bỏ lỡ do card mạng không xử lý kịp
    - `rx_long_length_errors 0` : Không có lỗi do gói tin có chiều dài vượt quá mức quy định
    - `rx_short_length_errors: 0`: Không có lỗi do gói tin có chiều dài nhỏ hơn mức quy định
    - `rx_align_errors: 0`: Không có lỗi do gói tin không được căn chỉnh đúng với giao thức mạng
    - `rx_flow_control_xon và rx_flow_control_xoff: 0`: Không có gói tin liên quan đến điều khiển luồng (flow control) được nhận. Điều khiển luồng là kỹ thuật điều chỉnh lưu lượng giữa các thiết bị để tránh tắc nghẽn
    - `rx_long_byte_count: 495359251`: Tổng số byte dữ liệu đã nhận với gói tin có chiều dài lớn hơn là 495,359,251 byte
    - `rx_csum_offload_good: 0`: Không có gói tin nào được kiểm tra checksum tốt bởi card mạng (offload kiểm tra checksum)
    - `rx_csum_offload_errors: 0`: Không có lỗi kiểm tra checksum nào được phát hiện khi card mạng thực hiện offload kiểm tra checksum
- `Thống kê gói tin gửi (TX - Transmit)` gồm:
    - `tx_packets: 981335`: Tổng số gói tin đã được gửi qua giao diện eth0 là 981,335 gói
    - `tx_bytes: 159174404`: Số lượng byte dữ liệu đã gửi qua giao diện là 159,174,404 byte (~159 MB)
    - `tx_broadcast: 1`: Chỉ có 1 gói tin broadcast đã được gửi đi, điều này cho thấy giao diện chủ yếu gửi dữ liệu điểm-điểm
    - `tx_multicast: 109`: Tổng số gói tin multicast đã được gửi là 109
    - `tx_errors: 0`: Không có lỗi nào xảy ra khi gửi gói tin
    - `tx_dropped: 0`: Không có gói tin nào bị loại bỏ khi gửi
    - `tx_aborted_errors: 0`: Không có lỗi gói tin bị hủy bỏ khi gửi
    - `tx_carrier_errors: 0`: Không có lỗi liên quan đến tín hiệu mạng khi gửi gói tin
    - `tx_heartbeat_errors: 0`: Không có lỗi liên quan đến nhịp tim (heartbeat) của mạng, điều này thường chỉ áp dụng cho mạng Ethernet cũ
    - `tx_window_errors: 0:` Không có lỗi cửa sổ (window errors), lỗi này thường liên quan đến việc xử lý gói tin quá nhanh so với khả năng của hệ thống
    - `tx_abort_late_coll: 0`: Không có lỗi do va chạm muộn khi truyền dữ liệu (late collision)
    - `tx_deferred_ok: 0`: Không có lần truyền nào bị hoãn lại do phải chờ tới lượt
    - `tx_single_coll_ok và tx_multi_coll_ok: 0`: Không có gói tin nào gặp va chạm đơn lẻ hoặc nhiều va chạm khi gửi. Điều này chỉ thường thấy ở các mạng sử dụng Ethernet cũ
    - `tx_timeout_count: 0`: Không có lỗi do hết thời gian chờ khi gửi gói tin
    - `tx_restart_queue: 0`: Không có sự kiện nào yêu cầu khởi động lại hàng đợi truyền tin
    - `tx_tcp_seg_good: 141`: Có 141 gói tin TCP được phân mảnh thành công bởi card mạng (TCP segmentation offload)
    - `tx_tcp_seg_failed: 0`: Không có gói tin TCP nào bị phân mảnh thất bại
    - `tx_flow_control_xon và tx_flow_control_xoff: 0`: Không có gói tin điều khiển luồng nào được gửi
- `multicast: 4018`: Số gói tin multicast đã được nhận là 4,018, thông tin này trùng với thống kê ở phần RX.

- `collisions: 0`: Không có va chạm gói tin trên mạng. Điều này cho thấy mạng đang hoạt động ổn định mà không có xung đột giữa các thiết bị.
- `alloc_rx_buff_failed: 0`: Không có sự cố nào khi cấp phát bộ đệm để nhận gói tin
- `tx_smbus và rx_smbus: 0`: Không có gói tin nào liên quan đến giao thức SMBus (System Management Bus) được gửi hoặc nhận
- `dropped_smbus: 0`: Không có gói tin SMBus nào bị loại bỏ

IV. Phân biệt Bandwidth (Băng thông) và Data Transfer (Truyền tải dữ liệu)

1. Băng thông - Bandwidth

Băng thông (Bandwidth) là một thuật ngữ trong lĩnh vực mạng và viễn thông, dùng để chỉ lượng dữ liệu có thể được truyền tải qua một kết nối mạng trong một khoảng thời gian nhất định, thường được đo bằng đơn vị bps (bits per second) hoặc Mbps (megabits per second).

Nói cách khác, băng thông biểu thị khả năng của đường truyền trong việc truyền tải dữ liệu, băng thông càng cao thì khả năng truyền tải dữ liệu càng lớn.

Ví dụ: Nói một cách dễ hiểu, băng thông có thể được ví như kích thước của ống nước – ống càng lớn thì lượng nước chảy qua trong một khoảng thời gian càng nhiều.

Phân loại băng thông

- `Băng thông nội địa`: Được dùng để tương tác & giao tiếp giữa các server trong cùng một đất nước. Loại band-width này thích hợp cho việc sử dụng trong mạng nội bộ của bạn
- `Băng thông quốc tế`: Thường được sử dụng để giao tiếp và tương tác giữa các máy chủ ở nhiều quốc gia khác nhau. Khi cáp quốc tế bị đứt, bạn có thể gặp khó khăn trong việc truy cập vào các trang web quốc tế hoặc tốc độ truy cập có thể bị giảm đáng kể

2. Truyền tải dữ liệu - Data Transfer

Data Transfer (truyền tải dữ liệu) là một khái niệm rất quan trọng trong kỹ thuật máy tính và các ngành công nghệ liên quan. Nó được sử dụng để chỉ quá trình truyền tải dữ liệu từ một thiết bị hoặc hệ thống đến một thiết bị hoặc hệ thống khác. Quá trình truyền tải này có thể được thực hiện qua nhiều phương thức khác nhau, bao gồm cáp mạng, sóng vô tuyến và đường truyền qua Internet.

Các phương pháp truyền tải dữ liệu:

- `Data Transfer qua cáp mạng`
- `Data Transfer qua sóng vô tuyến`
- `Data Transfer qua đường truyền qua Internet`
- `Data Transfer qua Bluetooth`
- `Data Transfer qua USB`

Phân biệt băng thông và truyền tải dữ liệu

Giống nhau:

- Yếu tố quan trọng của việc truyền dữ liệu trên mạng

- Đều ảnh hưởng đến trải nghiệm mạng
- Đều bị giới hạn bởi nhà cung cấp dịch vụ
- Ảnh hưởng đến cước phí Internet

Khác nhau:

Phân loại	Bandwidth	Datatransfer
`Định nghĩa`	Là tốc độ tối đa mà dữ liệu có thể được truyền qua mạng	Là tổng lượng dữ liệu thực tế được truyền qua mạng trong một khoảng thời gian
`Thời gian và tốc độ đơn vị đo`	Được tính trong mỗi giây (liên tục). Đo tốc độ truyền tải dữ liệu (số bit truyền đi mỗi giây). Đơn vị thường bps, Mbps, Gbps (bits per second)	Được tính trong khoảng thời gian dài (tháng, tuần, ngày). Đo tổng lượng dữ liệu được truyền đi trong khoảng thời gian dài. Đơn vị KB, MB, GB, TB (bytes)
`Giới hạn`	Giới hạn tốc độ truyền tải	Giới hạn dung lượng truyền tải
`Chi phí`	Băng thông cao có thể đắt hơn (tốc độ nhanh hơn, phí cao hơn)	Lượng dữ liệu cao có thể dẫn đến phí phụ trội nếu vượt giới hạn
`Phương pháp tối ưu hóa`	Tăng băng thông để cải thiện tốc độ truyền tải	Giảm lượng dữ liệu tiêu thụ để tránh vượt giới hạn
`Vai trò`	Cung cấp khả năng truyền dữ liệu nhanh chóng và ổn định	Giới hạn tổng lượng dữ liệu được sử dụng mà không gây thêm chi phí
`Ứng dụng`	Ảnh hưởng đến tốc độ tải xuống/tải lên dữ liệu trên mạng	Ảnh hưởng đến lượng dữ liệu bạn có thể sử dụng trong một khoảng thời gian
`Ví dụ sử sử dụng mạng`	Tốc độ truy cập website, xem video mượt mà hay giật lag	Lượng dữ liệu tiêu thụ khi xem video, tải file trong tháng

===> `Bandwidth` và `Data Transfer` đều liên quan đến truyền dữ liệu qua mạng và ảnh hưởng trực tiếp đến trải nghiệm của người dùng

QA: Theo bạn khi truy cập một website `https://vinahost.vn` trên trình duyệt, những điều gì đã xảy ra?

Phân giải tên miền (DNS resolution) là quá trình chuyển đổi một tên miền (ví dụ: vinahost.vn) thành địa chỉ IP tương ứng (ví dụ: 123.30.136.228). Quá trình này là cần thiết vì các hệ thống mạng, bao gồm Internet, dựa vào các địa chỉ IP để định vị và kết nối các thiết bị với nhau.

`Bước 1`: Phân giải DNS: Tìm địa chỉ IP của máy chủ từ tên miền

Bước 2: Kết nối TCP và bắt tay TLS: Thiết lập kết nối an toàn giữa trình duyệt và máy chủ

Bước 3: Gửi yêu cầu HTTP trình duyệt gửi yêu cầu lấy tài nguyên từ máy chủ

Bước 4: Máy chủ xử lý yêu cầu và gửi lại dữ liệu trang web

Bước 5: Trình duyệt tải trang web, phân tích HTML, CSS, và JavaScript

Bước 6: Trình duyệt hiển thị trang web để người dùng tương tác

ATM

Happy hacking, happy life !