Giao thức và chồng giao thức

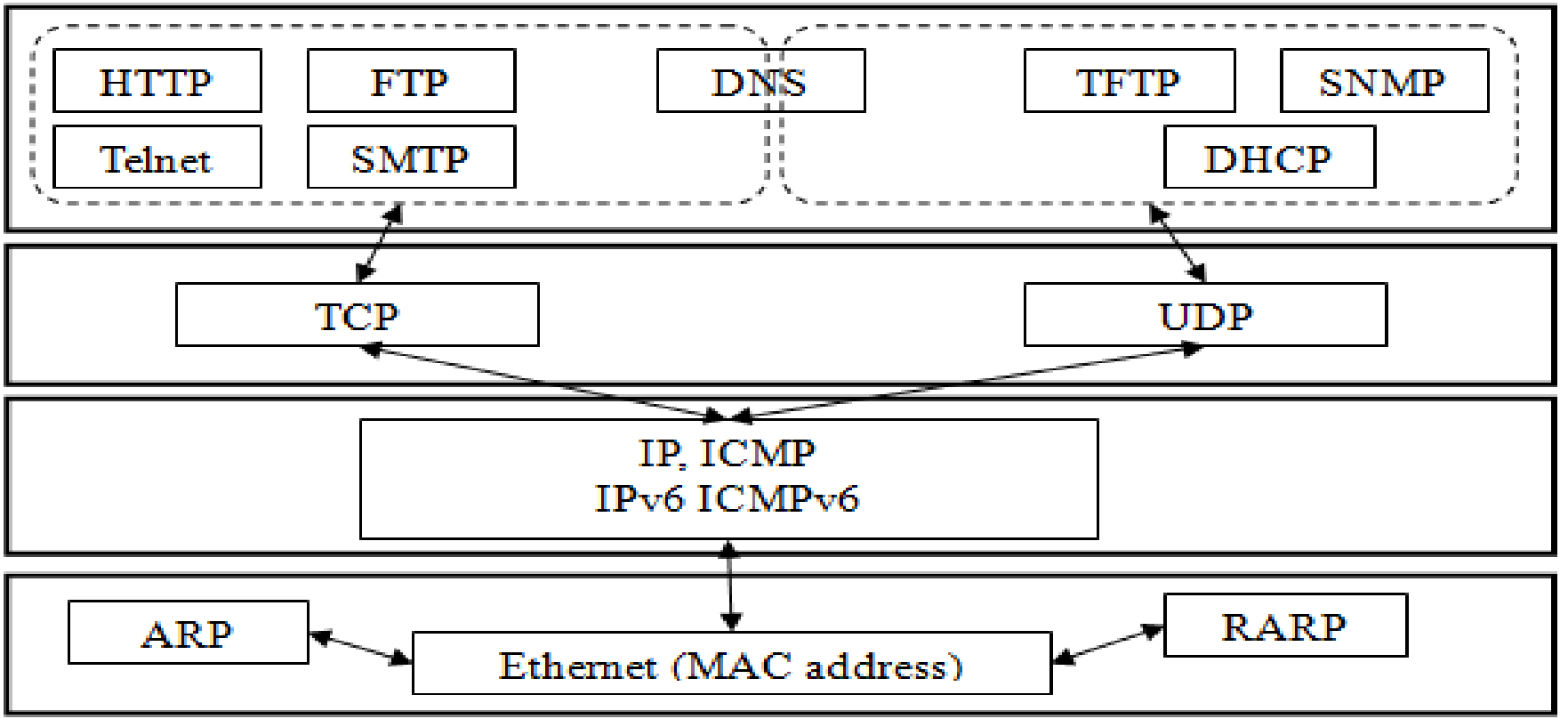
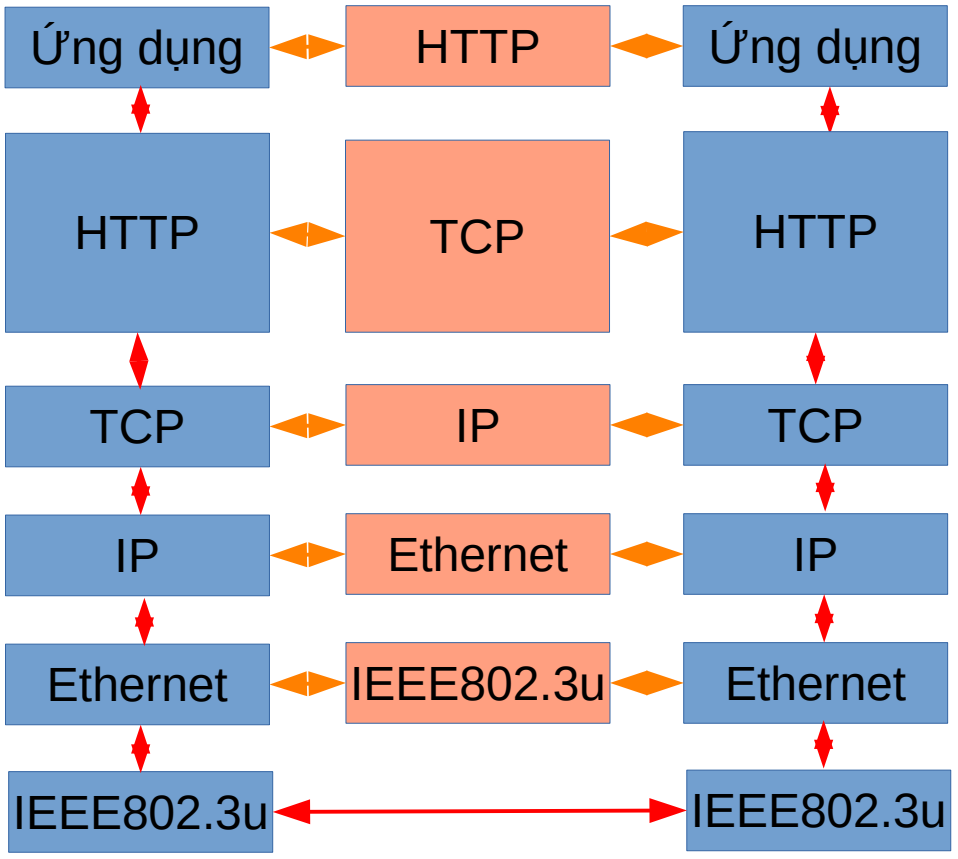
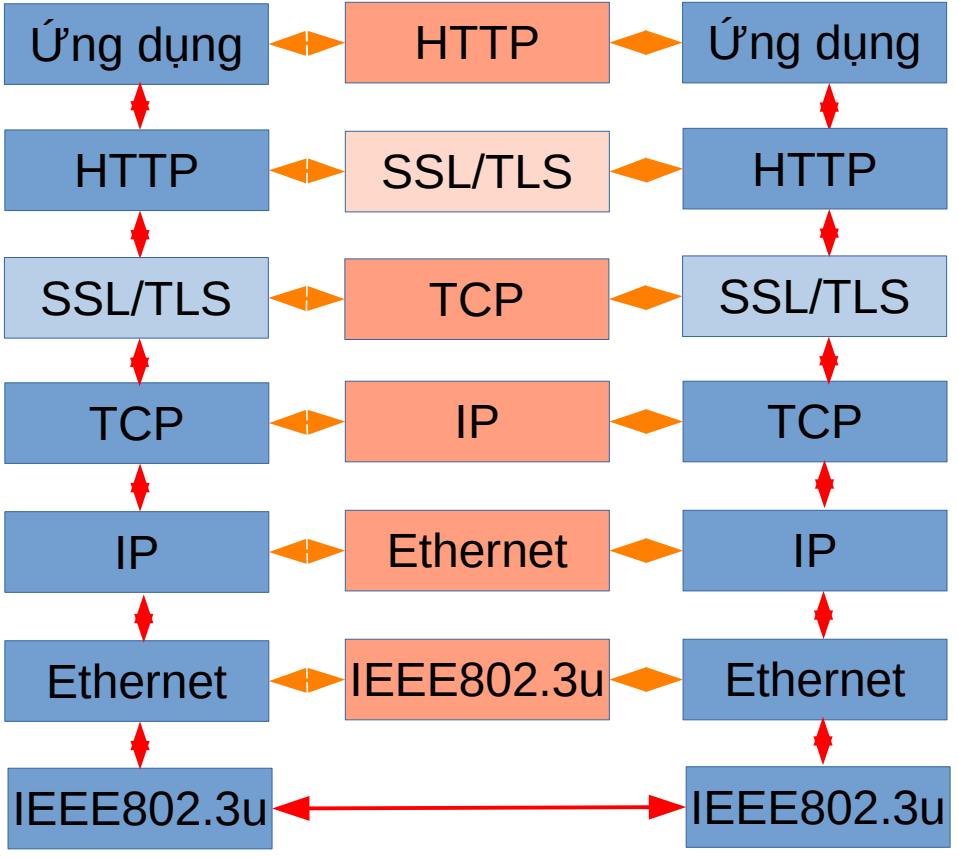


Figure 1.2. Part of the TCP/IP protocol suite

Chồng giao thức HTTP và HTTPS





Giao thức bảo mật SSL/TLS

SSL (Secure Sockets Layer) và TLS (Transport Layer Security) là các giao thức mã hóa dữ liệu của kết nối theo phương thức khóa mã công khai. Một khóa riêng gọi là private key, có thể giải mã dữ liêu, từ khóa riêng này. Các khóa công khai (public key) được tạo ra để mã hóa dữ liệu, khóa này không thể giải mã dữ liệu đã được mã hóa bởi chính nó, vì vậy nó được công bố công khai và truyền thoải mái trên đường truyền mà vẫn đảm bảo bảo mật. Nó đảm bảo:

- 1) Kết nối là riêng tư. Mã khóa được tạo ra cho mỗi kết nối trong quá trình thiết lập, đảm bảo không bị lấy cắp (nghe lén) hay bị sửa đổi.
- 2) Giao thức hỗ trợ xác thực danh tính công khai bởi bên thứ 3, thường là một nhà cung cấp dịch vụ xác thực danh tính được công nhận. Chứng chỉ xác thực danh tính của các nhà cung cấp này phải được lưu trong HĐH (như Linux) hoặc bởi trình duyệt trước đó.
- 3) Kết nối là đáng tin.

Các phiên bản của SSL/TLS

Tên (phiên bản) giao thức	Năm công bố	Năm loại bỏ
SSL 1.0	Không dùng	Không dùng
SSL 2.0	1995	2011 (RFC 6176)
SSL 3.0 (RFC6101)	1996	2015 (RFC 7568) (Vẫn còn dùng ở một số nơi)
TLS 1.0 (RFC2246)	1999	2020 (không còn dùng)
TLS 1.1 (RFC4346)	2006	2020 (không còn dùng)
TLS 1.2 (RFC5246)	2008	Hiện dùng phổ biến
TLS 1.3 (RFC8446)	2018	Hiện dùng phổ biến

Giao thức internet/web

- HTTP (HyperText Transfer Protocol): được phát triển bởi Tim Berners-Lee ở CERN từ năm 1989. Ban đầu các đặc tả được phối hợp bởi IEFT(Internet Engineering Task Force) và W3C (World Wide Web Consortium). Sau đó thì do IEFT đảm nhiệm. Bản 0.9 ra đời năm 1990. bản 1.0 (RFC1945) ra đời năm 1996. Bản 1.1 (RFC2068) ra đời năm 1996, và bản sử đổi (RFC2616) ra đời 2 năm său là bản được dùng phổ biến ngày nay.

Giao thức DNS

- DNS (Domain Name Service): Là dịch vụ chuyển đổi từ tên gợi nhớ dạng www.google.com sang IP khó nhớ và ngược lại.
- DNS được quản lý bởi ICANN (Internet Corporation for Assigned Names and Numbers), một tổ chức phi lợi nhuận. ICANN quản lý 13 máy chủ dịch vụ tên miền gốc (root server) có tên từ A đến M.
- Một số tên miền cấp 1(2) phổ biến:

  .COM : Commercial, dành cho các công ty.

  .NET : Network, dành cho các mạng riêng.

  .ORG : Organization, dành cho các tổ chức, thường là phi lợ nhuận, phi chính phủ.

  .GOV : Gorverment, Dành riêng cho các tổ chức chính quyền.

  .EDU : Education, dành riêng cho các tổ chức giáo dục đào tạo

  - Ngoài ra còn có các tên miền dành cho các quốc gia, vùng lãnh thổ như: .vn (Việt Nam), .ru (Rusia), .us (USA), uk … .asia (Asia), ..
  - Gần đây, một số tên miền dành cho các mục đích riêng biệt như .tv (Television), mobil, .. cũng đươc định nghĩa.

Khái niệm về tên URL

Khi bạn gõ vào trình duyệt một **URL**, ví dụ **http://moodle.hus.edu.vn/moodle**. URL này gồm ba phần:

- **http**: Phần này là giao thức, là phần đầu tiên được phân cách bởi dấu **://**
- **moodle.hus.edu.vn**: Tên máy chủ dịch vụ, cung cấp dịch vụ qua giao thức ở phần đầu (http). Phần này đầy đủ phải là moodle.hus.edu.vn:80, trong đó 80 là cổng cung cấp dịch vụ này. Tuy nhiên do cổng 80 là cổng mặc định của giao thức http nên phần này không cần thiết.
- **moodle**: Thư mục con trên máy chủ dịch vụ.

Khái niệm về tên tên miền

Trong tên của máy chủ dịch vụ, như ví dụ trên **moodle.hus.edu.vn**, gồm 4 phần được phân cách bởi dấu ‘**.**’:

- **vn** là tên miền cấp 1, tên miền này được dịch vụ bởi 13 máy chủ root DNS.
- **edu** là tên miền cấp 2, tên miền cấp 2 thường được dịch vụ bởi các máy chủ DNS cấp dưới.
- **hus** là tên miền cấp 3.
- **moodle** là tên máy.

Cách phân giải địa chỉ của DNS

Khi bạn đưa vào trình duyệt một tên miền, ví dụ **moodle.hus.edu.vn**, trình duyệt sẽ truy vấn địa chỉ IP của tên miền này từ máy chủ dịch vụ DNS cục bộ mặc định của nó, thường nhận được thông qua giao thức DHCP hoặc được khai báo bởi người quản trị. DNS server này:

- đầu tiên sẽ truy vấn tên miền cấp 1 **.vn** từ các máy chủ root DNS. Mọi máy chủ DNS phải lưu địa chỉ của 13 máy chủ này trong CSDL của nó. Máy chủ root DNS trả lại địa chỉ IP của máy chủ DNS phục vụ cho tên miền **.vn**.
- Său đó máy chủ DNS cục bộ tiếp tục truy vấn tên miền cấp 2 **.edu.vn** từ máy chủ DNS phục vụ cho tên miền **.vn**, để nhận được IP của máy chủ phục vụ cho tên miền cấp 2 này. Máy chủ DNS tiếp tục quá trình này để được tên miền cuối cùng **.hus.edu.vn**.
- Máy chủ DNS cục bộ truy vấn tên máy **moodle** từ máy chủ DNS dịch vụ cho tên miền **.hus.edu.vn** để được IP của máy **moodle.hus.edu.vn**.

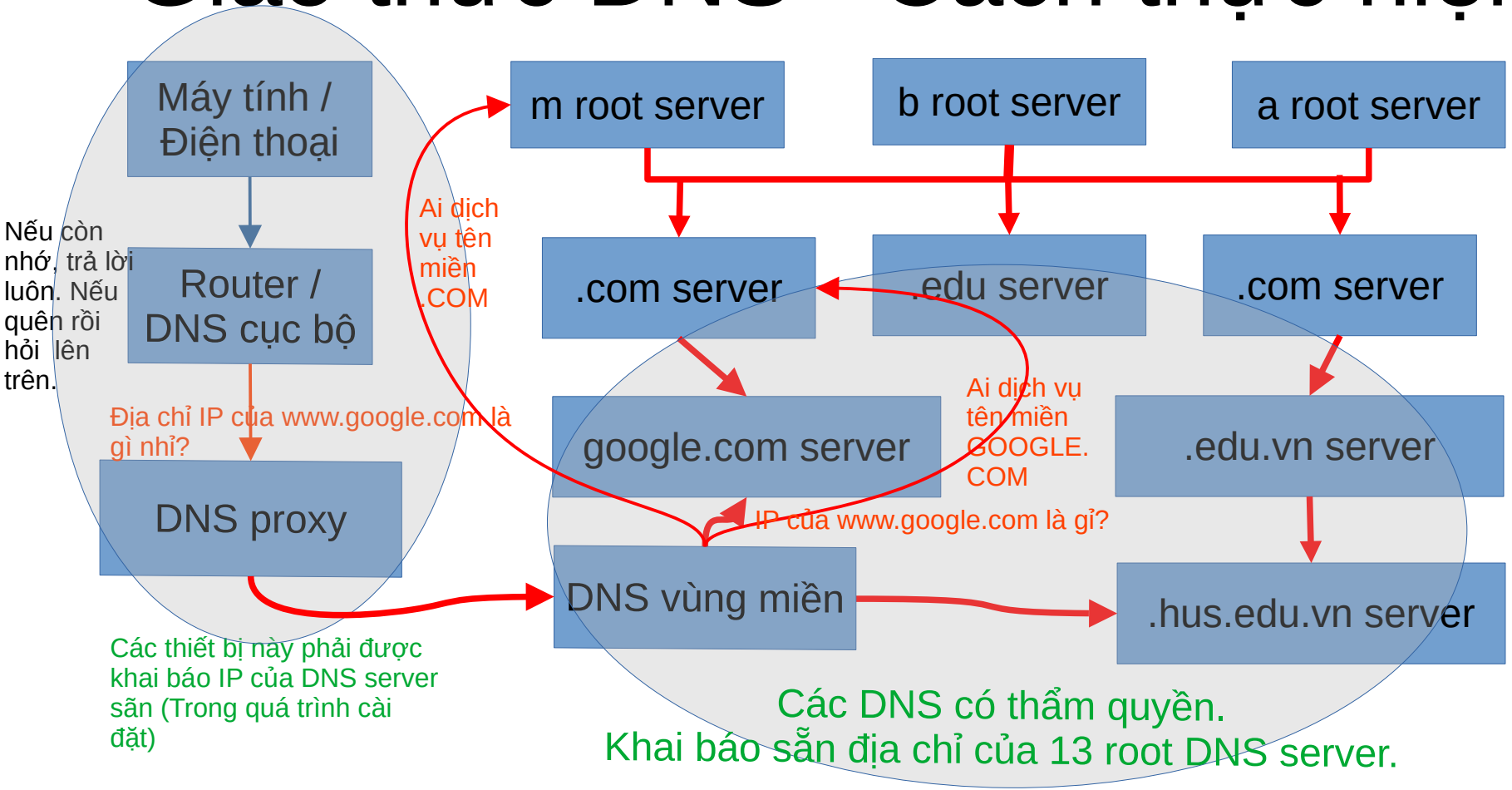
Chú ý khi khai báo DNS cục bộ

Một số người thích dùng DNS server có IP 8.8.8.8 do google cung cấp, đây không phải là ý tưởng tốt, vì:

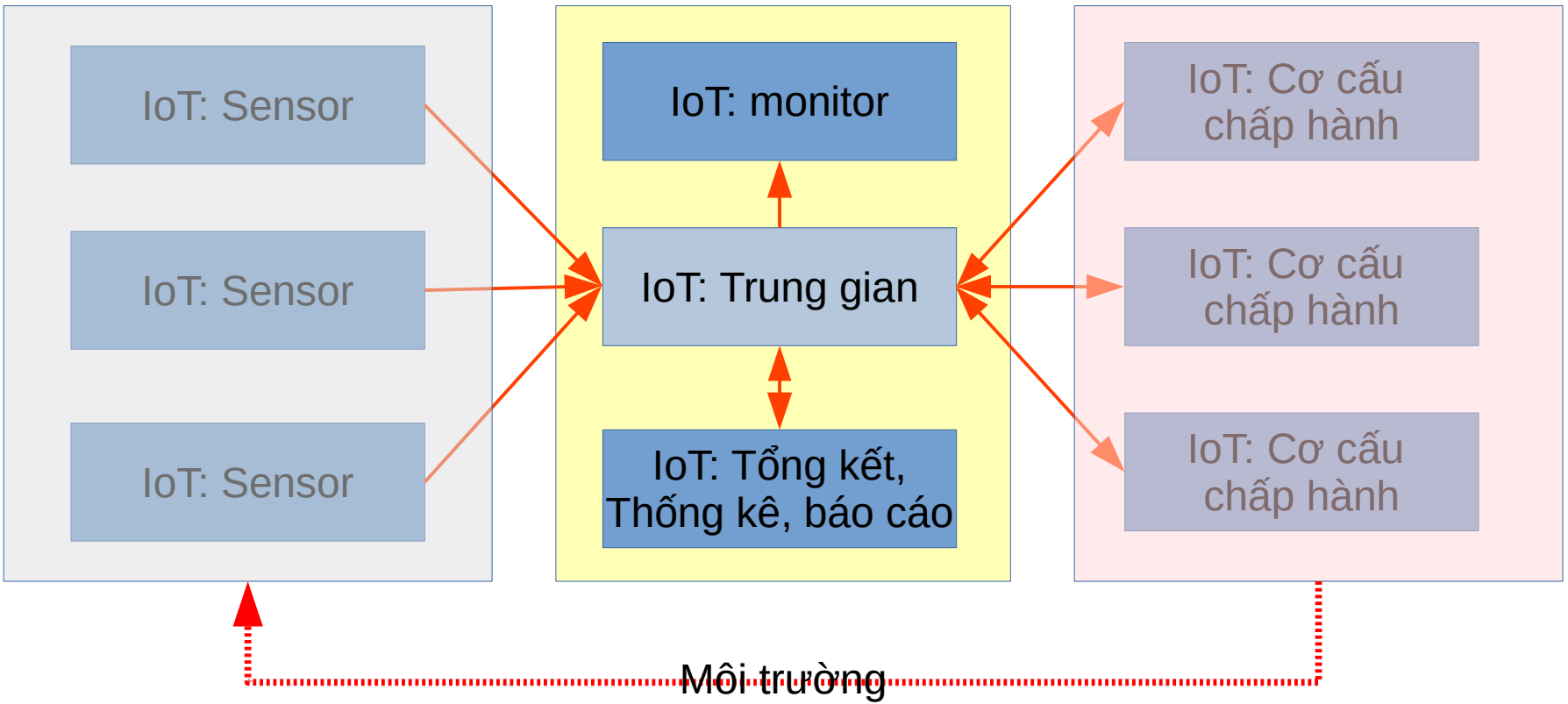
- Hầu hết các trang web bạn dùng là trang web ở Việt Nam (VN), như vậy các DNS server của các tên miền này là ở VN.
- Mọi truy vấn của bạn sẽ phải chuyển sang Mỹ, rồi từ Mỹ, DNS server hầu như lại phải truy vấn các DNS đặt ở VN, cuối cùng kết quả mới chuyển về VN cho bạn, như vậy độ trễ rất cao, đồng thời chiếm dụng thông lượng đường truyền quốc tế.
- Mặc dù các gói tin DNS khá nhỏ, nhưng số lượng truy vấn lại rất lớn nên nó chiếm dụng khá nhiều đường truyền, cũng như độ trễ sẽ rất lớn (bạn phải chờ lâu hơn).

Vì vậy, nếu bạn có đủ hiểu biết để tự khai báo DNS thì nên sử dụng các DNS của chính nhà cung cấp dịch vụ internet cho bạn (ISP - Internet Service Provider)(như là FTP, Viettel, VNPT,..), như vậy độ trễ sẽ giảm nhỏ rất nhiều, do kết nối từ bạn đến ISP là ngắn nhất. Các IP của DNS server này có thể lấy được từ các trang web của ISP.

Giao thức DNS - Cách thực hiện



Sơ đồ mạng IoT



Kiến trúc hệ thống IoT

Hệ thống IoT gồm nhiều thành tố như các cảm biến, các cơ cấu chấp hành, bộ phận trung chuyển, lưu trữ dữ liệu, hệ thống theo dõi, quan sát, quản lý,.... Trong đó có 3 thành phần chính cốt lõi để hệ thống vận hành trowng tru:

- Các **cảm biến**: định kỳ đo đạc các tham số môi trường rồi chuyển về trung tâm xử lý.
- **Broker** là trung tâm của bộ phận xử lý, nó sẽ nhận các dữ liệu môi trường từ các cảm biến rồi chuyển cho các bộ phận khác cần dữ liệu này - nó không lưu trữ hay xử lý dữ liệu. Các bộ phận khác nếu muốn nhận được một dữ liêu nào đó thì phải đăng ký nhận dữ liệu này trên broker.
- **Cơ cấu chấp hành**: Nhận dữ liệu từ các cảm biến thông qua broker, rồi hành xử theo các dữ liệu mà nó nhận được.

Kiến trúc hệ thống IoT

Kiến trúc này giúp cho hệ thống vận hành an toàn hơn, do các broker, trung tâm trung chuyển dữ liệu - bộ phận mà nếu nó ngừng hoạt động, toàn bộ hệ thống sẽ sụp đổ, không phải xử lý quá nhiều giúp giảm tải cho nó. Như vậy broker không đòi hỏi phải dùng các vi xử lý mạnh, không cần nhiều năng lượng nên không tỏa quá nhiều nhiệt. Từ đó không cần các hệ thống tản nhiệt phức tạp mà vẫn có thể đảm bảo nhiệt độ hoạt động cho hệ thống.

Các cảm biến hay các cơ cấu chấp hành cũng hầu như không phải xủa lý gì - chủ yếu là định kỳ gửi hoặc nhận dữ liệu. Yêu cầu xử lý lớn nhất là mã hóa dữ liệu (SSL/TLS hoặc E2E), nên cũng không cần phải xử lý nhiều.

Như vậy, các thành phần chính của IoT, chủ yếu được đặt ở thực địa, nơi thường là không có điều kiện ổn định, an toàn, đều không đòi hỏi bộ tản nhiệt phức tạp mà vẫn có thể hoạt động liên tục được.

Các bộ phận khác như theo dõi, quản lý, lưu trữ dữ liệu đòi hỏ năng lực xử lý mạnh, đều có thể được đặt trong các trung tâm dữ liệu nơi có điều kiện rất tốt.

Giao thức IoT: MQTT

- Ra đời năm 1999 bởi IBM
- Phiên bản đầu tiên được đệ trình lên OASIS năm 2013 (3.1) với điều kiện chỉ các thay đổi nhỏ được chấp nhận.
- Phương thức: publish–subscribe
- Version mới nhất: 5
  http://docs.oasis-open.org/mqtt/mqtt/v5.0/cs01/mqtt-v5.0-cs01.html

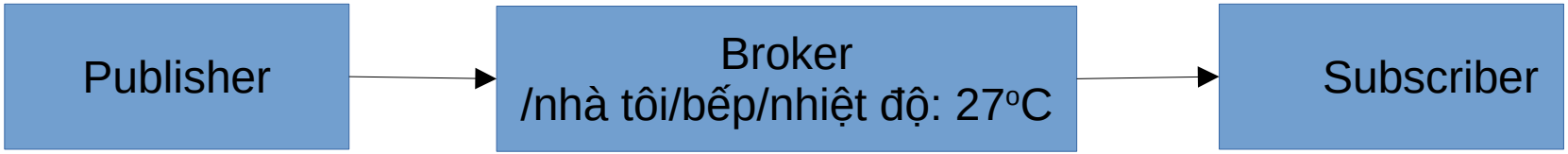
Giao thức IoT: MQTT

- Dữ liệu được trình bày dưới dạng các topic.
- Broker nhận dữ liệu từ các publishing rồi chuyển cho các subscribers đã đăng ký topic đó rồi loại bỏ dữ liệu (trừ khi dữ liệu được đặt cờ phải nhớ).
- Các publishing có thể đặt message mặc định sẽ được chuyển cho các subscribers nếu nó bị ngắt khỏi mạng.
- Có thể dùng TLS với chứng chỉ cả hai phía (client/broker)
- 3 mức QoS: Đã gửi, Ít nhất nhận được 1, Chắc chắn nhận được 1.

Giao thức IoT: MQTT

- Dễ dàng mở rộng quy mô.
- Quản lý và theo dõi tất cả các trạng thái kết nối của thiết bị, bao gồm thông tin xác thực và chứng chỉ bảo mật.
- Giảm thông lượng mạng mà không ảnh hưởng đến bảo mật.

Mô hình broker



- Broker nhận thông tin từ publisher rồi gửi ngay cho subscriber, thông tin sau đó bị loại bỏ khỏi bộ nhớ (trừ khi yêu cầu phải nhớ), giúp giảm nhỏ yêu cầu về bộ nhớ của broker.
- Broker chỉ nhận rồi gửi luôn, không thực hiện bất kỳ quá trình xử lý nào, giúp giảm yêu cầu về năng lực của vi xử lý, đặc biệt các vi xử lý này chỉ cần xử lý vào ra, không cần các xử lý toán học (dấu chấm động - FPU) hay các xử lý hiển thị - GPU, nên nó có thể dùng các bộ xử lý chuyên biệt (thực ra là đơn giản nhất - chỉ CPU), giúp giảm gía thành, tiết kiệm năng lượng, không tỏa nhiều nhiệt, ít hỏng hóc, ..

Giao thức IoT: AMQP

- Là một chuẩn mở, ra đời năm 2003 bởi JPMorgan.
- Phiên bản đầu tiên được đệ trình lên OASIS năm 2011 (1.0).

Giao thức IoT: AMQP

- Dữ liệu có cấu trúc tự mô tả.
- Cho phép mã hóa đầu cuối.
- Có thể dùng TLS.
- 3 mức QoS: Đã gửi, Ít nhất nhận được 1, Chắc chắn nhận được 1.

Giao thức IoT: CoAP

- Ra đời 2014 – RFC 7252
- Là giao thức ở lớp 6.
- Thiết kế tương tự HTTP để dễ dàng chuyển đổi dữ liệu sang dạng Web.
- Thiết kế cho các thiết bị ít tài nguyên.
- Dùng giao thức UDP (DTLS)

Tài liệu tham khảo

- https://standards.iso.org/ittf/PubliclyAvailableStandards/s020269_ISO_IEC_7498-1_1994(E).zip
- IEFT: https://tools.ietf.org/html/