**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

\*\*\*



**BÁO CÁO BÀI TẬP LỚN**

**Môn: Mạng Máy Tính**

**Đề tài: Tìm hiểu các giao thức Http, Https, Http2**

**Giảng viên hướng dẫn: Nguyễn Đức Toàn**

**Thành viên nhóm:**

Nguyễn Tôn Tú - 20158412

Đào Mạnh Linh - 20158216

Nguyễn Mạnh Hải - 20158119

*Hà Nội 2018*

**Mục lục**

**I) Giao Thức HTTP:**

1) Tổng quan

2) Kết nối HTTP

3) Thông điệp HTTP

4) Gói tin HTTP

**II) Giao thức HTTPs**

1) Tại sao nên sử dụng HTTPs?

2) HTTPs là gì?

3) Chứng chỉ số SSL

4) SSL hoạt động như thế nào

5) Các loại SSL

6) Những lợi ích của HTTPs

**III) Giao thức HTTP 2**

1) HTTP 2 là gì?

2) Các đặc điểm của HTTP 2

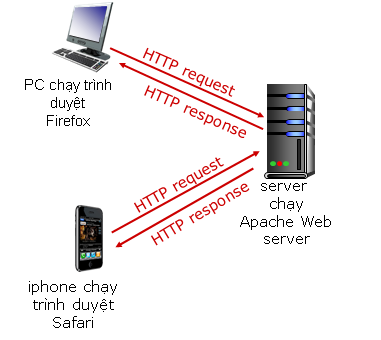
3) Vì sao HTTP 2 lại quan trọng?

**IV) Tài liệu tham khảo**

**I) Giao thức HTTP:**

**1) Tổng quan:**

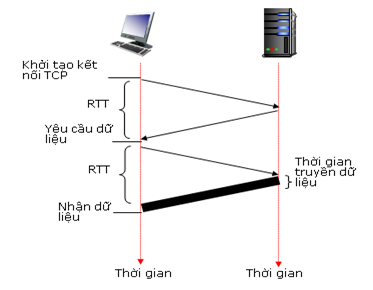
* Hypertext Transfer Protocol (HTTP) là giao thức thuộc lớp ứng dụng web. Sử dụng kết nối TCP cổng 80. HTTP được định nghĩa trong RFC 1945 (HTTP 1.0) và RFC 2616 (HTTP 1.1).
* HTTP hoạt động dựa trên mô hình client-server. Trình duyệt client thực hiện yêu cầu, nhận và hiển thị đối tượng web (gồm dữ liệu HTML, hình ảnh JPEG, Java applet, video, âm thanh, …). Trong khi, web server sẽ gửi trả lời khi nhận được yêu cầu từ client.



**2) Kết nối HTTP**

* Có hai loại kết nối **HTTP** là kết nối không bền vững và kết nối bền vững.
* Kết nối không bền vững: sau khi, server gửi đi một đối tượng thì kết nối TCP sẽ được đóng. Như vậy, mỗi kết nối TCP chỉ truyền được duy nhất một yêu cầu từ client và nhận lại một thông điệp trả lời từ server.
* Kết nối bền vững: server sẽ duy trì kết nối TCP cho việc gửi nhiều đối tượng. Như vậy, sẽ có nhiều yêu cầu từ client được gửi đến server trên cùng một kết nối. Thông thường kết nối TCP này sẽ được đóng lại trong một khoảng thời gian định trước.

*a) Quy trình hoạt động của kết nối HTTP không bền vững:*



* Hình trên thể hiện chi tiết các bước hoạt động trong kết  nối HTTP không bền vững giữa HTTP client (Firefox, Chrome,…) và HTTP server (ví dụ [www.hutech.edu.vn](http://www.hutech.edu.vn)):
* Bước 1: client khởi tạo kết nối TCP bằng việc gửi yêu cầu đến server. Server nhận được yêu cầu và chấp nhận kết nối bằng việc gửi trả lời về cho client. Nếu sau khoảng thời gian RTT mà không nhận được trả lời từ phía server thì client sẽ gửi lại yêu cầu.
* Bước 2: sau khi kết nối được thiết lập, client sẽ gửi thông điệp yêu cầu chứa tên đường dẫn của các đối tượng ( ví dụ: www.hutech.edu.vn/homepage/index.php) đến server. Server nhận được thông điệp yêu cầu và tiến hành lấy ra các đối tượng được yêu cầu. Sau đó, các đối tượng được đóng gói thành thông điệp trả lời và gửi đến client.
* Bước 3: server đóng kết nối TCP (Lưu ý: server chỉ đóng kết nối TCP khi chắc chắn rằng client nhận được thông điệp trả lời)
* Bước 4: client nhận thông điệp trả lời chứa tập tin HTML và hiển thị các đối tượng
* Bước 5: lặp lại các bước từ 1-4 cho các đối tượng khác.
* Lưu ý: Trong kết nối HTTP không bền vững cần có một RTT (Round Trip Time) để khởi tạo kết nối TCP. Ngoài ra, cần có một RTT cho thông điệp HTTP yêu cầu và một vài byte đầu tiên của thông điệp HTTP  trả lời được trả về.Tổng thời gian truyền tập tin = 2RTT+ thời gian truyền.
* Thời gian đáp ứng RTT làthời gian gửi một gói tin cơ bản từ client đến server rồi quay ngược lại. RTTbao gồmđộ trễtruyền gói tin và hàng đợi, độ trễtrongcác bộ định tuyếntrung gian, chuyển mạch và xử lýgói tin.

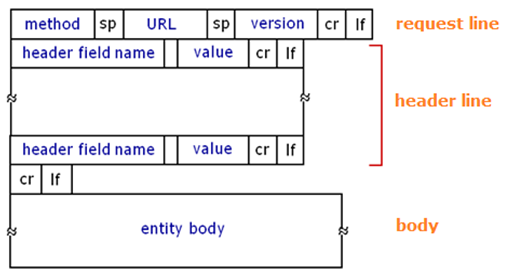
*b) Qui trình hoạt động của kết nối HTTP bền vững:*

* Kết nối bền vững không có pipelining:
* Client phát ra yêu cầu mới, chỉ khi đáp ứng trước đó đã nhận xong.
* RTT cho mỗi đối tượng tham chiếu.
* Kết nối bền vững có pipelining:
* Mặc định có trong HTTP 1.1.
* Client gửi yêu cầu ngay sau khi gặp một đối tượng tham chiếu.
* Ít nhất 1 RTT cho tất cả đối tượng tham chiếu.

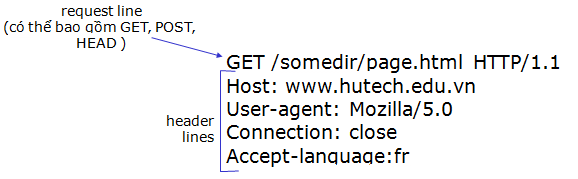
### 3) Thông điệp HTTP

* Thông điệp HTTP được viết bằng văn bản ASCII thông thường, do vậy người dùng có thể đọc được. Có 2 dạng thông điệp HTTP: thông điệp HTTP yêu cầu và thông điệp HTTP trả lời.

*a) Thông điệp HTTP yêu cầu:*

[](http://2.bp.blogspot.com/-0ZPjAnbecDs/U51YluTsjdI/AAAAAAAAAFA/5Nw_dsqsiH0/s1600/2.PNG)

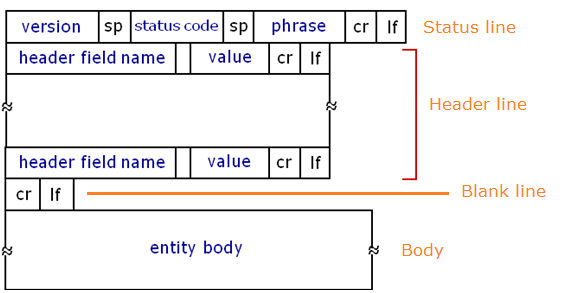
* **Request line**: dòng đầu tiên của thông điệp HTTP yêu cầu. **Request line** bao gồm có 3 trường như: cách thức (method), URL, phiên bản (version). Trường cách thức (method) có thể chứa các giá trị khác nhau, bao gồm **GET, POST, HEAD**. Phần lớn các thông điệp HTTP yêu cầu điều sử dụng phương thức GET. Các phương thức GET được sử dụng khi trình duyệt yêu cầu một đối tượng được xác định trong trường URL. Phương thức POST là những phương thức mà HTTP client sử dụng khi người dùng điền vào một biểu mẫu (form) nào đó, chẳng hạn như người dung muốn nhập một từ khóa nào đó vào google.com đề tìm kiếm thông tin. Nếu phương thức là POST thì Entity body của thông điệp yêu cầu sẽ chứa thông tin mà người dùng đã điền. Phương thức HEAD tương tự như GET, nhưng Khi server nhận được yêu cầu bằng phương thức HEAD, nó sẽ trả về thông điệp HTTP và không chứa đối tượng được yêu cầu.
* Header line: là các dòng tiếp theo.
* Sp: bao gồm các giá trị về khoảng trống.
* Blank line: bao gồm các giá trị điều khiển trở về đầu dòng, xuống hang (cr,lf).
* Entity Body (nếu có): là phần thân của thông điệp HTTP yêu cầu.
* Sau đây là một ví dụ về thông điệp HTTP yêu cầu:

[](http://3.bp.blogspot.com/-fyJE2vQzAIU/U51YlhJChEI/AAAAAAAAAFc/QAlgMWD5FrE/s1600/3.PNG)

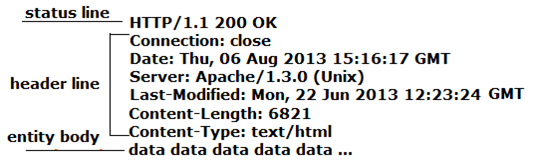
* GET: Trong ví dụ này, trình duyệt đang yêu cầu các đối tượng /somedir/page.html.
* www.hutech.edu.vn: chỉ địnhcác servermà trên đó cácđối tượng được lưu trữ.
* Connection: close trình duyệt báo cho máy chủ là nó không muốn sử dụng kết nối liên tục và muốn máy chủ đóng kết nối sau khi gửi các đối tượng được yêu cầu.
* -agent : chỉ thị loại trình duyệt gửi yêu cầu đến server, ở đây dùng trình duyệt Mozilla/5.0.
* Accept-language (fr): chỉ thị người dùng muốn nhận được phiên bản tiếng Pháp của đối tượng chứa trên server, nếu không server sẽ gửi phiên bản mặc định của đối tượng.

*b) Thông điệp HTTP trả lời*

* Thông điệp **HTTP** trả lời có ba phần: dòng trạng thái (**status line**), dòng tiêu đề (**header lines**), thân thông điệp (**entity body**). Thân (Body) là thành phần chính của thông điệp. Status line có 3 trường: phiên bản của giao thức (version), mã trạng thái (status code), trạng thái tương ứng (phrase) và các giá trị khoảng trống (sp), điều khiển trở về đầu dòng, xuống hàng (cr,lf).

[](http://4.bp.blogspot.com/-6MzisogRE1I/U51YmWRQalI/AAAAAAAAAFI/AlztIUBFZFw/s1600/4.PNG)

* Sau đây là ví dụ về thông điệp HTTP trả lời

[](http://2.bp.blogspot.com/-NM7NpzilqIs/U51YmbqlpDI/AAAAAAAAAFQ/bS-vrghy590/s1600/5.PNG)

* Status line thể hiện server đang sử dụng HTTP/1.1.
* Header line bao gồm:
* Connection: close báo cho client rằng server sẽ đóng kết kết TCP sau khi gửi thông điệp.
* Date: cho biết thời gian mà thông điệp HTTP trả lời được tạo và gửi bởi server. Đó là thời gian mà server lấy đối tượng từ hệ thống tập tin của nó, chèn vào thông điệp và gửi cho client.
* Server: cho biết đây là Apache Web server, tương tự như User-agent trong thông điệp yêu cầu.
* Last-Modified: cho biết thời gian đối tượng được tạo hay sửa đổi lần cuối
* Content-Length: cho biết số bytes của đối tượng được gửi.
* Content-Type: cho biết đối tượng trong phần entity body là HTML.
* Entity body chứa đối tượng được đại diện bởi các *data data data data……….*

*c) Mã trạng thái trả lời:*

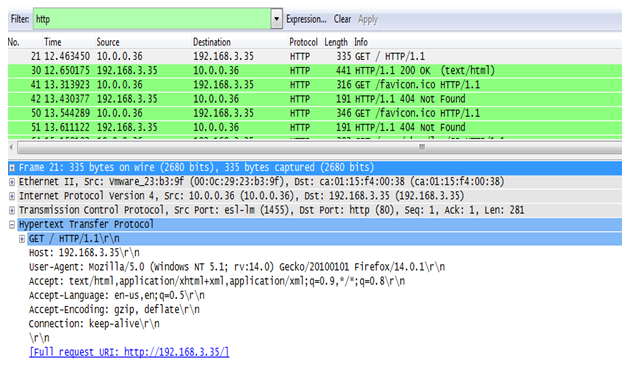
Các giá trị chữ số đầu tiên của mã trang thái (status code) có 5 giá trị:

* 1xx\_Thông tin: khôngđược sử dụng, chỉ dự phòngtrong tương lai.
* 2xx\_Thành công: hành độngđã nhận đượcthành công và đươc chấp nhận.
* 3xx\_Chuyển hướng: hành động tiếp theo phải được thực hiện để hoàn tất yêu cầu.
* 4xx\_Clien lỗi: chứa cú pháp sai có thể không thực hiện được.
* 5xx\_ Server lỗi: các máy server không thực hiện một yêu cầu rõ ràng đối với các yêu cầu hợp lệ.

Một vài mã trạng thái thông dụng thường gặp:

* 200 OK: Yêu cầu thành công
* 301 Moved Permanently: đối tượng yêu cầu đã được chuyển.
* 400 Bad Request: server không hiểu được thông điệp yêu cầu.
* 404 Not Found: đối tượng được yêu cầu không có trong server.
* 505 HTTP Version Not Supported: server không hỗ trợ phiên bản giao thức HTTP này.

**4) Gói tin Http**

[[](http://1.bp.blogspot.com/-ivOeSoUcwIk/U51YmjOn3-I/AAAAAAAAAFY/dqfcBkAlFpc/s1600/6.PNG)](http://1.bp.blogspot.com/-ivOeSoUcwIk/U51YmjOn3-I/AAAAAAAAAFY/dqfcBkAlFpc/s1600/6.PNG)

Hình wireshark trên cho thấy HTTP đang sử dụng phiên bản HTTP 1.1 dùng phương thức GET.

-          Có các mã trạng thái hồi đáp như:

* 200 OK
* 404 Not Found
* Địa chỉ IP nguồn: 10.0.0.36
* Địa chỉ IP đích: 192.168.3.35

-          Cổng HTTP: 80

* Sử dụng phương thức TCP
* Trang web đăng nhập là (host) [http://192.168.3.35](http://192.168.3.35/) không sử dụng DNS để phân giải địa chỉ IP 192.168.3.35.
* User-agent: loại trình duyệt gửi yêu cầu đến server ở đây dùng trình duyệt Mozilla/5.0.
* Accept-language: sử dụng ngôn ngữ tiếng anh (en-us).

**II) Giao thức Https:**

**1) Tại sao nên sử dụng HTTPS?**

* Hãy thử trừu tượng hóa một chút. Khi sử dụng giao thức HTTP tức là bạn đang phô diễn mọi thức về bạn một cách trần trụi nhất, nhìn vào người ta sẽ biết bạn có gì. Còn giao thức HTTPS là lúc đó bạn đã mặc lên bộ quần áo che khuất tất cả, dấu đi thông tin cá nhân của bạn không cho người khác biết.
* HTTPS – chữ “S” thừa ra ở đây là viết tắt của Secure – an toàn. Khi kết nối bằng giao thức HTTPS, người khác sẽ khó có thể biết bạn đang làm gì.

**2)** **HTTPS là gì?**

* HTTPS là viết tắt của “Hypertext Transfer Protocol Secure”, Nó là một sự kết hợp giữa giao thức HTTP và giao thức bảo mật SSL hay TLS cho phép trao đổi thông tin một cách bảo mật trên Internet. Giao thức HTTPS thường được dùng trong các giao dịch nhậy cảm cần tính bảo mật cao.
* Netscape Communications tạo ra HTTPS vào năm 1994 cho trình duyệt web Netscape Navigator . Ban đầu, HTTPS đã được sử dụng với SSL mã hóa. Phiên bản hiện hành của HTTPS được chính thức chỉ định bởi RFC 2818 Tháng 5 năm 2000.
* HTTPS đã có lịch sử phát triển khá dài, thậm chí có thể nói rằng nó sinh cùng thời với Web. Nhưng cho đến nay, giao thức này vẫn chỉ được sử dụng chủ yếu bởi những trang web có hoạt động chuyển tiền, trang web của các ngân hàng hoặc các cửa hàng trực tuyến. Nếu như có những trang web khác sử dụng HTTPS, thì họ cũng chỉ dùng nó cho một vài tác vụ nhất định mà thôi.

**3) Chứng chỉ số SSL là gì?**

* SSL là viết tắt của Secure Sockets Layer, một giao thức mạng máy tính nhằm đảm bảo tất cả các dữ liệu trao đổi giữa máy chủ web và trình duyệt mang tính riêng tư, bảo mật và an toàn.
* Cha đẻ của SSL là tiến sĩ Taher Elgamal, làm việc tại Netscape Communications. Phiên bản SSL 1.0 ra đời năm 1994 nhưng chỉ được lưu hành nội bộ và sử dụng trên trình duyệt Netscape. Phiên bản 2.0 được công bố tháng 2 năm 1995 nhưng chứa những lỗ hổng an ninh nghiêm trọng. Phiên bản 3.0 được thiết kế lại hoàn toàn mới vào năm 1996.
* Tổ chức IETF đã chuẩn hoá SSL và đặt lại tên là TLS (Transport Layer Security) vào tháng 1 năm 1999, tuy nhiên thuật ngữ SSL vẫn được sử dụng phổ biến hơn. Trên thực tế TLS chỉ là một phiên bản mới của SSL với sự khác biệt không đáng kể (phiên bản TLS 1.0 tương đương với phiên bản SSL 3.1). Phiên bản cập nhật của TLS là 1.1 ra đời vào tháng 4 năm 2006 và 1.2 ra đời tháng 8 năm 2008. Từ đó đến nay tiêu chuẩn SSL/TLS ngày càng trở nên phổ biến.
* Mozilla đã phát triển phiên bản TLS 1.3 cho trình duyệt của hãng từ tháng 2 năm 2017. Google cũng đã từng thử nghiệm TLS 1.3 nhưng sau đó phải gỡ bỏ vì thấy chưa tương thích. Chính vì vậy phiên bản này hiện nay vẫn chỉ đang hoạt động ở dạng thử nghiệm.

#### **4) SSL hoạt động như thế nào?**

* Để truy cập được một trang web sử dụng giao thức HTTPS thì đương nhiên thiết bị của bạn phải cài đặt một trình duyệt web có hỗ trợ SSL. Và tin vui là tất cả các trình duyệt web mới nhất hiện nay đều đã hỗ trợ TLS.
* Khi truy cập vào 1 trang web sử dụng https thì:
* Trình duyệt sẽ gửi yêu cầu cho máy chủ web cung cấp thông tin xác nhận danh tính.
* Máy chủ web gửi lại cho trình duyệt chứng chỉ SSL mà nó đã được cấp.
* Trình duyệt kiểm tra thông tin chứng chỉ. Nếu đúng, nó thông báo lại cho máy chủ web là SSL đã được chấp nhận
* Máy chủ web gửi ngược lại chữ ký số sẽ dùng để mã hóa và giải mã trong suốt quá trình giao tiếp sau đó.
* Mọi thông tin trao đổi giữa trình duyệt và máy chủ sẽ được mã hóa.

#### **5)** **Các loại SSL**

|  |  |
| --- | --- |
| **DV SSL** Domain Validation | Đây là loại chứng chỉ số cấp cho tên miền. Website của bạn sẽ được hiển thị với 1 biểu tượng ổ khóa màu xanh lá. |
| **OV SSL** Organization Validation | Loại chứng chỉ cấp cho tên miền và hiển thị thêm thông tin về doanh nghiệp đang sở hữu tên miền đó. |
| **EV SSL** Extended Validation | Là loại chứng chỉ cao cấp nhất dành cho các đối tượng là doanh nghiệp hoặc tổ chức có giấy phép hoạt động. Loại chứng chỉ này cho phép hiển thị đầy đủ tên công ty hoặc tổ chức sở hữu tên miền trên thanh địa chỉ của trình duyệt web. |
| **Wildcard SSL** Wildcard SSL Certificate | Là loại chứng chỉ bảo mật có hiệu lực với tất cả các tên miền phụ (subdomain). Wildcard SSL khác với các loại SSL bình thường là có thể chạy cho không giới hạn tên miền phụ với chỉ một chứng chỉ ssl duy nhất. |
| **UCC/SAN SSL** | Là loại chứng chỉ sử dụng được cho nhiều tên miền (tối đa là 40), được thiết kế riêng cho các dịch vụ của Microsoft như Exchange Server hay Office Communications. |

**6) Những lợi ích của https:**

* Như đã nói ở trên, giao thức https sinh ra để thay thế cho http, mục đích là để bảo mật tốt hơn
* Giúp chứng thực doanh nghiệp bạn là tồn tại, tránh giả mạo website.
* Làm tăng niềm tin, sự an tâm cho khách hàng truy cập website.
* Góp phần cải thiện thứ hạng của trang web trên công cụ tìm kiếm Google.
* Tất cả các nhà cung cấp SSL đều có chính sách bảo hiểm. Tức là họ sẽ bồi thường thiệt hại cho bạn trong trường hợp cấp sai chứng chỉ SSL để xảy ra tổn thất, lừa đảo, lộ thông tin thẻ tín dụng,… của khách hàng khi giao dịch trên website có cài https.

**III) Giao thức HTTP 2**

1**) Giao thức HTTP 2 là gì?**

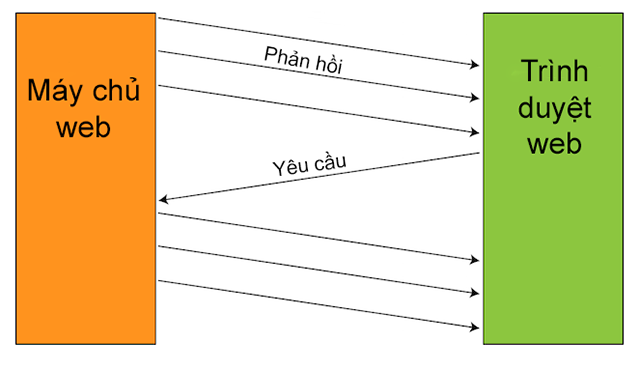
* HTTP 2 là cuộc cách mạng Hypertext Transfer Protocol (HTTP) mới nhất tính đến thời điểm này. HTTP là giao thức mạng được sử dụng để yêu cầu và nhận page cùng dữ liệu trên môi trường World Wide Web. Công nghệ mới này sẽ dần thay thế chuẩn HTTP 1.1 đã được sử dụng rộng rãi trong hơn hai thập kỷ gần đây
* **HTTP 2** hay HTTP phiên bản 2 là phiên bản chính thức thứ hai của giao thức truyền tải siêu văn bản HTTP, được xây dựng trên hàng loạt những công nghệ tiên tiến nhằm cải thiện đáng kể hiệu suất website. HTTP 2 được phát triển dựa trên công nghệ SPDY bởi bộ phận Hypertext Transfer Protocol thuộc Internet Engineering Task Force.

## **2) Các đặc điểm của HTTP 2**

* **Dữ liệu truyền tải dạng nhị phân**
* HTTP 2 truyền dữ liệu ở dạng nhị phân thay vì dạng text như của HTTP 1.1, giúp tác vụ được thực hiện hiệu quả và tiêu tốn ít thời gian hơn. Trình duyệt sẽ không phải tốn thời gian xử lý dữ liệu bằng cách dịch văn bản sang một định dạng khác để hiểu.
* **Nén dữ liệu được gửi đi**
* Khác với HTTP 1.1, các Header gói tin sẽ được nén trước khi gửi đi. Thông tin được gửi đi kèm với truy vấn mô tả dữ liệu, nguồn gốc, kiểu, độ dài… của dữ liệu đó.
* **Giải quyết phản hồi ưu tiên**
* Trong HTTP 1.1, server phải gửi phản hồi theo cùng trật tự nhận truy vấn. HTTP 2 thì giải quyết bất đồng bộ, nên các truy vấn nhỏ hơn hoặc nhanh hơn có thể được xử lý sớm hơn. Đồng thời, cho phép trình duyệt có thể sắp xếp thứ tự ưu tiên tải về cho các tài nguyên nào quan trọng dùng để hiển thị website.
* **Thiết lập kết nối liên tục**
* HTTP 2 cho phép xử lý nhiều truy vấn giữa máy chủ và trình duyệt trên một kết nối TCP duy nhất. Cụ thể, HTTP 2 giúp tạo ra kết nối liên tục giữa máy chủ và trình duyệt. Trong khi đó, HTTP 1.1 thiết lập một kết nối mới mỗi lần trình duyệt truy vấn thông tin đồng thời chỉ xử lý 1 truy vấn duy nhất trên một kết nối TCP
* **Gửi trước dữ liệu Server Push**
* HTTP 2 cho phép server có thể gửi gói dữ liệu trước khi nó được yêu cầu. Ví dụ, bạn có thể “reference” một script ở cuối page. Trong HTTP 1.1, trình duyệt sẽ tải, phân giải HTML rồi chạy JavaScript khi đến đoạn tag script. Với HTTP 2, server có thể gửi file JavaScript đến trước khi trình duyệt yêu cầu tài nguyên này.

**3) Vì sao HTTP 2 lại quan trọng?**

* Như đã nói ở trên, vì HTTP 1.1 được phát triển cách đây quá lâu nên nó bắt đầu bộc lộ những nhược điểm của sự "già cỗi" đó. Các trang web ngày nay không chỉ có những đoạn mã HTML đơn giản, ngoài ra nó còn được kết hợp với các công cụ dùng cho việc trang trí - thiết kế (CSS), các đoạn mã thực thi (JavaScript), hình ảnh, video, thậm chí là cả nội dung Flash nữa. Để truyền tải lượng dữ liệu đó, trình duyệt phải tạo ra nhiều kết nối đến máy chủ, mỗi kết nối chứa rất nhiều thông tin về nguồn lấy là ai, lấy ở đâu, lấy nội dung, lấy ra sao gì và rất nhiều thứ khác nữa. Tất cả những thứ này đã tạo ra một khối lượng công việc rất lớn lên trình duyệt trong máy tính (hoặc thiết bị di động) của chúng ta cũng như máy chủ nơi đặt trang web.



* Các kết nối trên, kết hợp với sức mạnh xử lý cần để thiết lập, giải quyết yêu cầu thông tin cho mỗi kết nối, có thể làm cho việc tải trang web của chúng ta bị chậm đi rất nhiều. Website chứa càng nhiều thứ thì tình trạng này càng trở nên nghiêm trọng hơn. Với người dùng chúng ta, đòi hỏi tốc độ ngày càng tăng lên khi mà đường truyền Internet của chúng ta cũng tốt dần, thế nên chỉ vài giây trì hoãn khi tải trang web cũng có thể khiến ta bực bội. Còn với các doanh nghiệp, website tải chậm có nghĩa là mất tiền (hãy nghĩ đến các trang mua sắm trực tuyến, trang chậm thì chẳng ai mua mà người ta sẽ chuyển qua dùng dịch vụ của đối thủ). Nó cũng làm cho trải nghiệm người dùng trở nên kém đi rất nhiều.
* Người ta đã cố gắng tìm nhiều cách để tăng tốc Internet từ thời mà kết nối dial-up còn phổ biến chứ chưa có ADSL. Một trong những kĩ thuật phổ biến nhất đó là sử dụng bộ đệm (cache, động từ là caching). Với kĩ thuật này, một số nội dung sẽ được lưu sẵn trên thiết bị của người dùng, khi đó trình duyệt chỉ việc tải những phần nội dung mới thay vì phải tải lại mọi thứ từ đầu. Ngoài ra người ta còn nghĩ đến việc giảm dung lượng ảnh và video, một số khác thì dành nhiều thời gian để tối ưu hóa các đoạn mã nguồn để cắt giảm từng mili giây thời gian tải trang.
* Những thứ nói trên khá hữu ích, tuy nhiên nó cũng chỉ giống như một miếng băng gạc băng lại cho “đỡ đau” chứ không giải quyết được cội nguồn vấn đề. Thế nên Google và một số công ty khác mới quyết định sẽ làm mới lại HTTP/1.1 để rồi tạo ra [SPDY](https://tinhte.vn/tags/spdy/) (đọc là speedy), một công nghệ truyền tải mới nhằm thu được kết quả tốt hơn. Nhìn tổng quan, việc liên lạc giữa server và trình duyệt bằng SPDY nhanh hơn rất nhiều ngay cả khi các công cụ mã hóa được áp dụng. Ở mức thấp nhất, tốc độ truyền tải dữ liệu bằng SPDY có thể nhanh hơn khoảng 10%, và trong một số trường hợp khác thì có thể cải thiện đến 40%.

**IV) Tài liệu tham khảo:**

* **Slide môn mạng máy tính**
* [**https://quantrimang.com/**](https://quantrimang.com/)
* **Tìm kiểm trên Google**
* [**https://vi.wikipedia.org/wiki/**](https://vi.wikipedia.org/wiki/)
* [**https://vnreview.vn**](https://vnreview.vn)