**So sánh các giao thức Internet phổ biến: HTTP, HTTPS, TCP/IP, FTP, DNS**

**1. Bảng so sánh chung**

| **Giao thức** | **Khái niệm & công dụng** | **Quy trình hoạt động cơ bản** | **Mục đích sử dụng** | **Ưu điểm** | **Nhược điểm** |
| --- | --- | --- | --- | --- | --- |
| **HTTP** (HyperText Transfer Protocol) | Giao thức ứng dụng dùng để truyền tài nguyên web (HTML, CSS, hình ảnh, script, …) giữa trình duyệt và máy chủ. | 1. Người dùng (client) nhập URL hoặc click link → trình duyệt gửi **HTTP request** (ví dụ GET, POST) tới máy chủ.  2. Máy chủ xử lý request, tìm tài nguyên, trả về **HTTP response** gồm mã trạng thái (ví dụ 200 OK, 404 Not Found) và nội dung.  3. Trình duyệt nhận response, hiển thị nội dung. | Truy cập trang web, xem nội dung web, gửi form qua web, tải tài nguyên từ server. | Đơn giản, phổ biến rộng, nhẹ hơn HTTPS vì không mã hoá; có rất nhiều công cụ & hỗ trợ. | Không an toàn (dữ liệu truyền dưới dạng văn bản rõ ràng, dễ bị nghe trộm/tấn công); không bảo vệ dữ liệu nhạy cảm; không phù hợp cho giao dịch cần bảo mật. |
| **HTTPS** (HTTP Secure) | Phiên bản bảo mật của HTTP: HTTP hoạt động trên kênh được mã hóa bằng SSL/TLS. | 1. Khi client muốn kết nối đến website HTTPS, client và server thực hiện **handshake SSL/TLS**: trao đổi chứng chỉ, khóa công khai, xác thực máy chủ (và có khi client).  2. Sau khi kết nối bảo mật được thiết lập, client gửi HTTP request qua kênh được mã hóa.  3. Server gửi response cũng mã hóa.  4. Kết thúc session, đóng kết nối. | Đảm bảo tính riêng tư và bảo mật cho các trang web: ngân hàng, thanh toán, dữ liệu cá nhân, ứng dụng nhạy cảm. | Bảo vệ dữ liệu người dùng, chống nghe trộm và sửa đổi dữ liệu; tạo độ tin cậy (người dùng nhìn thấy “https://”, biểu tượng ổ khóa). | Tiêu tốn thêm tài nguyên (CPU, xử lý mã hoá/giải mã); phức tạp hơn trong thiết lập chứng chỉ; đôi khi độ trễ cao hơn HTTP thông thường. |
| **TCP/IP** (Transmission Control Protocol / Internet Protocol) | Là bộ giao thức nền tảng cho Internet: *IP* đảm bảo định danh & định tuyến gói dữ liệu giữa các máy; *TCP* đảm bảo truyền dữ liệu tin cậy, sắp xếp, kiểm tra lỗi. | 1. TCP đặt kết nối giữa client và server (TCP handshake: SYN → SYN-ACK → ACK).  2. Dữ liệu được chia thành các segment, mỗi segment được gửi qua IP, có đánh số, kiểm tra lỗi.  3. IP định tuyến các packet qua nhiều router để tới đích.  4. Nếu có packet mất, TCP sẽ yêu cầu gửi lại.  5. Khi truyền xong, kết thúc kết nối. | Hầu hết các ứng dụng cần truyền dữ liệu tin cậy giữa hai máy, như web (HTTP/HTTPS), email, file transfer, streaming (khi cần đảm bảo đúng thứ tự). | Tin cậy cao, đảm bảo dữ liệu đến đầy đủ và đúng thứ tự; có kiểm soát lỗi và điều khiển luồng; phổ biến, được hỗ trợ rộng. | Giao thức phức tạp hơn so với giao thức không kết nối; đôi khi overhead cao; nếu mạng bị mất gói hoặc trễ cao, hiệu năng có thể giảm. |
| **FTP** (File Transfer Protocol) | Giao thức ứng dụng để truyền & quản lý file giữa client và server. | 1. Client kết nối tới server qua cổng điều khiển (control port, thường là 21).  2. Gửi lệnh (login, list directories, get file, put file…) qua control channel.  3. Khi truyền file, mở kênh dữ liệu (data connection; cổng 20 hoặc một cổng được thiết lập) để truyền nội dung file.  4. Khi xong, đóng kênh dữ liệu & kết thúc phiên làm việc. | Tải file lên/tải file xuống giữa máy của người dùng và server (ví dụ upload website, chia sẻ file). | Dễ sử dụng, nhiều Client & server hỗ trợ; quản lý file & thư mục; có hai chế độ (active và passive) để làm việc linh hoạt qua các firewall/NAT. | Dữ liệu và thông tin đăng nhập thường không mã hóa; có rủi ro an ninh nếu dùng FTP thông thường; cần quản lý quyền truy cập; đôi khi khó khăn khi băng qua tường lửa/NAT. |
| **DNS** (Domain Name System) | Hệ thống phân giải tên miền sang địa chỉ IP và ngược lại, giúp người dùng truy cập website qua tên dễ nhớ thay vì địa chỉ IP số. | 1. Client gửi truy vấn DNS (request) khi người dùng nhập tên miền (ví dụ www.example.com).  2. Truy vấn được gửi tới resolver DNS (máy chủ DNS gần nhất) nếu cache không có bản ghi.  3. Resolver có thể hỏi các máy chủ DNS cao cấp hơn để tìm bản ghi (root → TLD → authoritative).  4. Khi có IP tương ứng, trả lại cho client.  5. Client sử dụng IP đó để kết nối với máy chủ đích. | Giúp người dùng sử dụng tên miền thay vì nhớ địa chỉ IP; hỗ trợ hoạt động Internet thuận tiện: duyệt web, gửi email, truy cập dịch vụ. | Rất nhanh (truy vấn DNS thường chỉ mất vài mili-giây); khi có cache càng tốt; ẩn địa chỉ IP thực tế của máy chủ (người dùng không cần biết IP); làm cho hệ thống tên miền dễ quản lý & mở rộng. | Khi máy chủ DNS bị tấn công hoặc bị giả mạo (DNS spoofing) có thể làm lệch hướng người dùng; nếu mất DNS resolver hoặc server bị lỗi thì ảnh hưởng lớn truy cập web; tên miền mới có thể cần thời gian cập nhật (propagation). |

**2. Ví dụ cụ thể về quy trình hoạt động (HTTP / HTTPS / DNS)**

**Ví dụ**: Khi bạn mở trình duyệt nhập https://www.example.com/page.html

1. Trình duyệt gửi truy vấn DNS để tìm IP của www.example.com.
2. DNS trả IP (ví dụ 93.184.216.34).
3. Trình duyệt (HTTP client) khởi tạo kết nối HTTPS tới IP đó (qua cổng 443) — thực hiện handshake TLS.
4. Sau khi kênh bảo mật được thiết lập, client gửi HTTP request mã hóa (GET /page.html).
5. Server nhận, xử lý, trả về HTTP response (status + nội dung HTML, CSS, hình ảnh…) qua kênh bảo mật mã hóa.
6. Trình duyệt giải mã, hiển thị nội dung.

**3. Ưu điểm & nhược điểm tổng hợp**

* **HTTP**
  + Ưu: nhanh, nhẹ, dễ thiết lập, phù hợp nội dung không nhạy cảm.
  + Nhược: không có bảo mật, dễ bị nghe trộm, giả mạo, man-in-the-middle.
* **HTTPS**
  + Ưu: bảo mật, tính riêng tư, tăng tính tin cậy, bảo vệ người dùng.
  + Nhược: độ trễ hơi cao khi handshake; chi phí chứng chỉ SSL/TLS; cần quản lý chứng chỉ.
* **TCP/IP**
  + Ưu: tin cậy, phổ biến, có đầy đủ các chức năng từ lớp mạng đến lớp ứng dụng; có khả năng xử lý lỗi và đảm bảo thứ tự.
  + Nhược: overhead (băng thông & tài nguyên) khi kết nối & kiểm soát lỗi; có thể chậm hơn UDP cho ứng dụng cần tốc độ cao và có thể chịu mất mát nhỏ.
* **FTP**
  + Ưu: hỗ trợ chia sẻ file & quản lý file, có tính năng đa dạng (upload/download, di chuyển, xoá, liệt kê thư mục).
  + Nhược: bảo mật thấp nếu dùng FTP thường; dễ bị tấn công; đôi khi cấu hình phức tạp khi có firewall/NAT.
* **DNS**
  + Ưu: rất quan trọng để người dùng không phải nhớ IP; có cache giúp nhanh; hệ thống phân tán, khắp nơi.
  + Nhược: là mục tiêu tấn công (DNS spoofing, DNS cache poisoning); nếu server DNS bị tấn công hoặc sập thì ảnh hưởng lớn.