| **Giao thức** | **Khái niệm / Tổng quan** | **Quy trình hoạt động chính** | **Mục đích sử dụng** | **Ưu điểm** | **Nhược điểm** |
| --- | --- | --- | --- | --- | --- |
| **TCP/IP** | Bộ giao thức nền tảng của Internet. “IP” là Internet Protocol chịu trách nhiệm phân phối gói dữ liệu (routing, định tuyến) qua các mạng. “TCP” là Transmission Control Protocol — cung cấp kênh truyền tin tin cậy, đảm bảo dữ liệu đến đích, đúng thứ tự, không bị mất. [MDN Web Docs+3Quantrimang+3GoAnywhere+3](https://quantrimang.com/cong-nghe/nhung-giao-thuc-mang-pho-bien-hien-nay-162843?utm_source=chatgpt.com) | **TCP**: thiết lập kết nối (three‑way handshake: SYN → SYN‑ACK → ACK), sau đó truyền dữ liệu có xác nhận, kiểm tra thứ tự, kiểm soát lỗi, sau xong đóng kết nối (4‑step FIN/ACK). “IP” định tuyến các gói, chia & đóng gói, chọn đường đi, xử lý khi đóng gói nhỏ/bị phân mảnh. [Wikipedia+3MDN Web Docs+3Quantrimang+3](https://developer.mozilla.org/en-US/docs/Glossary/TCP_handshake?utm_source=chatgpt.com) | Mọi ứng dụng mạng cần truyền dữ liệu đáng tin cậy qua mạng không đáng tin (Internet): web, email, truyền file, chat, streaming, v.v. | • Tin cậy cao: đảm bảo dữ liệu đến nơi, đúng thứ tự.   • Kiểm soát lỗi & hồi phục khi mất gói.   • Hỗ trợ phân mảnh, định tuyến qua nhiều mạng.  • Giao thức chuẩn, rộng rãi, tương thích tốt. | • Khởi tạo kết nối tốn thời gian (handshake, overhead).   • Tốn băng thông cho xác nhận, kiểm tra lỗi, retransmit.  • IP hiện tại (IPv4) có giới hạn địa chỉ; routing phức tạp.   • Khi độ trễ lớn hoặc mất gói, hiệu suất giảm. |
| **HTTP** | HyperText Transfer Protocol: giao thức lớp ứng dụng dùng để trao đổi siêu văn bản / tài nguyên web giữa client (trình duyệt) & server. Dựa trên mô hình request/response. Cổng mặc định là port 80. [Demo Site+3Quantrimang+3Tuhoc Network Security+3](https://quantrimang.com/cong-nghe/nhung-giao-thuc-mang-pho-bien-hien-nay-162843?utm_source=chatgpt.com) | Quy trình điển hình:  1. Client (browser) gửi HTTP Request đến server qua TCP connection (cơ bản là mở kết nối TCP).  2. Server nhận request, xử lý yêu cầu, gửi HTTP Response chứa HTML / tài nguyên (CSS, JS, hình ảnh…)  3. Connection có thể được đóng hoặc giữ mở nếu HTTP keep‑alive được hỗ trợ. Web hiện đại có HTTP/1.1, HTTP/2, HTTP/3 với các cải tiến về hiệu suất. [Demo Site+2Wikipedia+2](https://vpsttt.com/cac-giao-thuc-mang-duoc-su-dung-pho-bien-hien-nay/?utm_source=chatgpt.com) | • Dễ dùng, là nền tảng của Web.   • Hỗ trợ nhiều phương thức (GET, POST, etc.), header, cookies, sessions.   • Với HTTP/2, HTTP/3 có nhiều cải tiến tốc độ, nén header, multiplexing.   • Tương thích ngược. | • Không mã hóa theo mặc định → dữ liệu và thông tin nhạy cảm có thể bị nghe lén, tấn công trung gian (MITM).   • Một số request/response có overhead lớn nếu không tối ưu (ví dụ nhiều tài nguyên, nhiều kết nối nhỏ). |  |
| **HTTPS** | Là HTTP + bảo mật: HTTP over TLS/SSL. Sử dụng mã hóa để bảo vệ dữ liệu truyền tải giữa client và server, đảm bảo tính riêng tư, xác thực server, integrity. Cổng mặc định port 443. [Wikipedia+2Quantrimang+2](https://en.wikipedia.org/wiki/HTTPS?utm_source=chatgpt.com) | Quy trình: giống HTTP nhưng có giai đoạn thiết lập TLS trước khi gửi HTTP Request.  1. Client mở kết nối TCP tới server (port 443).  2. TLS handshake: trao đổi certificate, xác thực, thiết lập khóa mã hóa.  3. Sau khi TLS thiết lập, HTTP request/response được gửi qua kênh mã hóa.  4. Khi xong phiên, đóng kết nối TLS/TCP. | • Khi cần bảo mật: đăng nhập, thanh toán điện tử, truyền dữ liệu nhạy cảm, giao dịch ngân hàng, v.v.   • Đảm bảo người dùng biết đang kết nối đúng server qua certificate   • Ngăn MITM, nghe lén, sửa đổi nội dung. | • Overhead lớn hơn HTTP: TLS handshake, mã hóa/giải mã tốn CPU, độ trễ thêm.   • Chi phí quản lý certificate, nếu certificate không hợp lệ → lỗi.   • Cần cấu hình chính xác để tránh lỗ hổng bảo mật (TLS version, cipher suite). |  |
| **FTP** | File Transfer Protocol: giao thức lớp ứng dụng để chuyển (upload/download) file giữa client và server qua mạng sử dụng TCP/IP. Có kênh điều khiển (control connection) và kênh dữ liệu (data connection). Port mặc định: 21 (control), 20 (data) trong nhiều trường hợp. [Wikipedia+2Tuhoc Network Security+2](https://en.wikipedia.org/wiki/File_Transfer_Protocol?utm_source=chatgpt.com) | Quy trình cơ bản:  1. Client kết nối tới server qua port control (21), đăng nhập (username/password) nếu cần.  2. Sau khi xác thực, client gửi lệnh (ví dụ “LIST”, “GET”, “PUT”).  3. Dữ liệu chuyển qua kết nối dữ liệu (data connection) – danh sách file hoặc nội dung file.  4. Khi xong, đóng kết nối dữ liệu; tiếp tục dùng control nếu gửi thêm lệnh; cuối cùng client đóng control. FTP có chế độ active/passive để thiết lập data connection tùy theo firewall/định tuyến. [Wikipedia+2Tuhoc Network Security+2](https://zh.wikipedia.org/wiki/%E6%96%87%E4%BB%B6%E4%BC%A0%E8%BE%93%E5%8D%8F%E8%AE%AE?utm_source=chatgpt.com) | • Chuyển file lớn hiệu quả, hỗ trợ resume trong nhiều trường hợp.   • Có nhiều lệnh quản lý file (traverse thư mục, xóa, rename, etc.)   • Phổ biến, được hỗ trợ bởi nhiều client và server. | • Không mã hóa mặc định → username/password, dữ liệu có thể bị nghe lén.   • Sẽ gặp vấn đề với firewall / NAT do data‑connection phụ thuộc vào nhiều kết nối (active/passive).   • Độ phức tạp cao hơn nếu cấu hình bảo mật.   • Hiệu suất có thể kém nếu nhiều kết nối nhỏ hoặc overhead quản lý nhiều phiên. |  |
| **DNS** | Domain Name System: hệ thống dịch tên miền thành địa chỉ IP — giúp máy tính / người dùng sử dụng tên dễ nhớ thay vì nhớ địa chỉ số. Là dịch vụ application‑layer, dùng UDP chủ yếu (có khi TCP) port 53. [lamth.github.io+2TechTarget+2](https://lamth.github.io/Report-MDT/Network-Fun/docs/16.DNS-DHCP-HTTP-FTP-Telnet.html?utm_source=chatgpt.com) | Quy trình phân giải tên miền (DNS resolution):  1. Client (máy người dùng hoặc trình duyệt) gửi yêu cầu tới local resolver (máy chủ DNS do ISP hoặc cấu hình)  2. Resolver kiểm tra cache xem đã có mapping (domain → IP chưa)  3. Nếu không, resolver gửi truy vấn tới Root DNS servers để tìm máy chủ cấp TLD (top-level domain) tương ứng (.com, .org, .vn...)  4. Resolver tiếp tục tới máy chủ TLD để tìm máy chủ chính xác cho domain  5. Resolver tới máy chủ DNS có thẩm quyền (authoritative) để lấy record thực sự  6. Kết quả được trả về, lưu vào cache; sau đó trả cho client  7. Client sử dụng IP để thiết lập kết nối (ví dụ HTTP/HTTPS). [Tuhoc Network Security+3GeeksforGeeks+3Huawei Support+3](https://www.geeksforgeeks.org/how-does-dns-resolve-domain-names-to-ip-addresses/?utm_source=chatgpt.com) | • Giúp truy cập dễ dàng qua tên miền thay vì nhớ địa chỉ IP   • Caching giúp giảm độ trễ, giảm tải mạng   • Hệ phân cấp, phân phối làm tăng khả năng mở rộng, chịu lỗi cao   • Hỗ trợ nhiều loại record (A, AAAA, MX, CNAME, etc.) | • Nếu DNS bị tấn công (cache poisoning, spoofing) → sai thông tin, người dùng bị chuyển hướng nguy hiểm   • Trễ nếu các bước phân giải phức tạp hoặc máy chủ bị chậm, không có cache   • Khi bị chặn hoặc bị giả mạo, tên miền có thể không được phân giải   • UDP bị giới hạn kích thước gói; nếu response lớn phải dùng TCP hoặc phân mảnh. |  |