| **Giao thức** | **Khái niệm & công dụng** | **Quy trình hoạt động cơ bản** | **Mục đích sử dụng** | **Ưu điểm** | **Nhược điểm** |
| --- | --- | --- | --- | --- | --- |
| **TCP/IP** | “TCP/IP” là tên chung cho bộ giao thức (suite) chủ đạo trên Internet, bao gồm **IP (Internet Protocol)** và **TCP (Transmission Control Protocol)**.  • IP chịu trách nhiệm đánh địa chỉ (addressing), định tuyến (routing) các gói tin qua mạng.  • TCP chịu trách nhiệm đảm bảo truyền tải dữ liệu tin cậy: chia mảnh, bảo đảm thứ tự, kiểm lỗi, kiểm soát lưu lượng. | 1. Khi một ứng dụng (ví dụ trình duyệt, FTP, email) muốn gửi dữ liệu tới máy khác: truyền lên tầng TCP để chia dữ liệu thành các segment, đánh số thứ tự, thiết lập kết nối (three‑way handshake: SYN, SYN‑ACK, ACK).  2. Gói (segment) được gửi xuống tầng IP, đóng gói thêm địa chỉ nguồn & địa chỉ đích, sau đó IP xử lý việc định tuyến qua các router tới đích.  3. Tại máy đích, IP nhận, tách header, đưa lên tầng TCP; TCP kiểm tra thứ tự, tái sắp xếp, yêu cầu gửi lại các segment bị mất, sau đó giao dữ liệu cho ứng dụng. | Cho phép truyền dữ liệu tin cậy qua mạng không tin cậy; cơ sở hạ tầng để các giao thức khác (HTTP, FTP, DNS, v.v.) hoạt động được. | • Ưu điểm: tin cậy (reliable), đảm bảo thứ tự (ordered), kiểm lỗi, hỗ trợ chia mảnh & tái lập; phổ biến, tiêu chuẩn, rất linh hoạt.  • Khả năng tương thích lớn với nhiều loại mạng/hệ thống; có thể truyền qua mạng phức tạp. | • Nhược điểm: overhead (chi phí xử lý, thời gian) do phải thiết lập kết nối, kiểm lỗi, tái gửi nếu mất, thứ tự; có độ trễ hơn những giao thức nhẹ (như UDP).  • Không bảo mật tự động (không mã hóa) — lớp bảo mật phải thêm vào ngoài TCP/IP. |
| **HTTP (HyperText Transfer Protocol)** | Giao thức tầng ứng dụng dùng để truyền và nhận trang web: các yêu cầu (requests) từ client (trình duyệt) tới server, và trả về phản hồi (responses) gồm HTML, hình ảnh, CSS, script, etc. | Quy trình cơ bản:  1. Người dùng nhập URL hoặc click link → trình duyệt gửi yêu cầu HTTP tới server (qua TCP).  2. TCP thiết lập kết nối, sau đó HTTP gửi request (GET, POST, etc).  3. Server xử lý yêu cầu, gửi lại response (status code, headers, body).  4. Trình duyệt nhận response, hiển thị nội dung. | Dùng để hiển thị web, truyền nội dung trang web; cũng dùng cho API, tải nội dung web. | • Ưu điểm: đơn giản, phổ biến; tương thích trên tất cả trình duyệt; dễ debug; hỗ trợ caching, proxy, header tùy chỉnh.  • Có nhiều tùy chọn mở rộng (HTTP/2, HTTP/3) để tối ưu hóa hiệu suất. | • Nhược điểm: không bảo mật — dữ liệu gửi/nhận có thể bị chặn/chỉnh sửa nếu dùng HTTP (không mã hóa).  • Overhead nếu nhiều tài nguyên nhỏ; nếu không sử dụng các phiên bản mới (HTTP/2,3), hiệu năng có thể kém. |
| **HTTPS (HyperText Transfer Protocol Secure)** | Là phiên bản bảo mật của HTTP, dùng TLS (hoặc SSL trước đây) để mã hóa dữ liệu truyền giữa client và server, xác thực server (và đôi khi client), bảo đảm tính toàn vẹn dữ liệu. | Quy trình hoạt động thêm bước handshake TLS:  1. Client gửi yêu cầu HTTPS (qua TCP) tới server port 443.  2. TLS handshake: client và server trao đổi thông tin khả năng mã hóa, xác thực chứng chỉ (server certificate), thỏa thuận bộ mã hóa, tạo “session key” (khóa đối xứng) để dùng cho phiên giao tiếp.  3. Sau khi thiết lập TLS, HTTP request/response diễn ra qua kênh mã hóa. | Dùng khi muốn giữ thông tin riêng tư (ví dụ trang login, thanh toán, dữ liệu cá nhân), bảo vệ khỏi nghe lén (eavesdropping), giả mạo, tấn công trung gian. | • Ưu điểm: bảo mật (mã hóa), xác thực máy chủ, tính toàn vẹn dữ liệu; tăng độ tin cậy với người dùng; yêu cầu bảo vệ thông tin nhạy cảm. | • Nhược điểm: overhead handshake (thời gian thiết lập ban đầu), tiêu tốn tài nguyên máy chủ & client (CPU cho mã hóa/giải mã), nếu cấu hình chứng chỉ hoặc TLS sai có thể bị lỗi bảo mật; cũng có chi phí chứng chỉ (nhiều trường hợp). |
| **FTP (File Transfer Protocol)** | Giao thức ứng dụng để truyền file giữa máy khách và máy chủ qua mạng TCP/IP. Cho phép upload, download, duyệt thư mục, đôi khi quản lý quyền. | Quy trình cơ bản:  1. Client kết nối tới server (cổng mặc định 21) để thiết lập “control connection”.  2. Sau khi xác thực (nếu cần), client gửi các lệnh điều khiển (list, get, put) qua control connection.  3. Khi truyền dữ liệu (file upload/download), một kết nối dữ liệu (“data connection”) được mở (cổng 20 hoặc các cổng khác tùy mode: active vs passive).  4. Dữ liệu truyền qua data connection, khi xong đóng kết nối dữ liệu; control connection có thể tiếp tục để gửi thêm lệnh hoặc đóng sau. | Dùng khi cần gửi/nhận file lớn hoặc nhiều file, quản trị máy chủ, backup, chia sẻ file giữa server-client. | • Ưu điểm: chuyên cho file transfer; nhiều lệnh điều khiển; có chế độ active/passive để ứng phó firewall/NAT.  • Có hỗ trợ chương trình khách (client) và máy chủ (server) rõ ràng; có thể upload & download. | • Nhược điểm: an ninh thấp nếu không dùng phiên bản bảo mật (FTP truyền username/password & dữ liệu dạng plaintext).  • Kém phù hợp cho mạng NAT/firewall; overhead trong thiết lập nhiều kết nối.  • Không tối ưu hóa cho việc truyền file nhỏ hoặc nhiều file nhỏ (nhiều overhead). |
| **DNS (Domain Name System)** | Giao thức tầng ứng dụng dùng để chuyển đổi tên miền (domain name) dễ nhớ sang địa chỉ IP, cũng hỗ trợ bản ghi khác như bản ghi mail, bản ghi alias, bản ghi IPv6. Là hệ thống phân tán & phân cấp. | Quy trình hoạt động cơ bản:  1. Client (máy tính, trình duyệt) khi cần truy cập tên miền gửi truy vấn tên (domain name) tới DNS resolver (thường do ISP hoặc dịch vụ bên ngoài).  2. Resolver kiểm tra cache; nếu không có, sẽ truy vấn lên các máy chủ DNS phân cấp: root → máy chủ TLD → máy chủ có thẩm quyền cho domain.  3. Máy chủ cung cấp bản ghi IP (A hoặc AAAA) trả về resolver, rồi resolver trả lại cho client.  4. Client dùng địa chỉ IP đó để tiếp tục kết nối HTTP/HTTPS hay các giao thức khác.  5. Bản ghi DNS được lưu cache để giảm độ trễ về sau. | Giúp con người không cần nhớ địa chỉ IP, dễ dàng truy cập website thông qua tên miền; linh hoạt thay đổi IP máy chủ mà không ảnh hưởng người dùng; hệ thống phân tán giúp bền vững. | • Ưu điểm: dễ nhớ tên miền, hiệu quả; giảm gánh nặng quản lý IP; phân cấp và phân phối giúp chịu tải & đạt độ tin cậy cao.  • Thời gian đáp ứng nhanh nếu cache hiệu quả.  • Có thể hỗ trợ nhiều loại bản ghi (IP, mail, alias,…). | • Nhược điểm: nếu DNS server bị lỗi hoặc bị tấn công (DNS spoofing, cache poisoning) thì người dùng bị hướng lầm; độ trễ ban đầu nếu DNS resolver không cache & phải truy vấn nhiều tầng; thiếu bảo mật nếu không dùng các bản mở rộng như DNSSEC.  • Giao thức truy vấn DNS truyền thống dùng UDP không mã hóa / không bảo vệ tính riêng tư (gần đây có các phương pháp như DNS over TLS, DoH). |