

Tìm hiểu và so sánh giao thức HTTP và WebSockets

1. Giao thức HTTP

a) Khái niệm

HTTP là giao thức tầng ứng dụng để truyền tải siêu văn bản như HTML. Nó được dùng để truyền tải dữ liệu giữa trình duyệt web và máy chủ web. HTTP hoạt động theo mô hình Client – Server, là giao thức không trạng thái

b) Cách thức hoạt động

Khi máy khách khởi tạo yêu cầu, máy chủ nhận được yêu cầu rồi gửi lại trạng thái đến máy khách kèm theo thông điệp (thông tin máy khách yêu cầu, thông báo lỗi, thông tin khác)

c) Các thành phần chính của giao thức HTTP

- HTTP – Requests: là phương thức để yêu cầu một hành động mong muốn trên tài nguyên đã xác định. Cấu trúc một dòng yêu cầu gồm: phương thức (GET, POST, HEAD,...), URL yêu cầu và phiên bản HTTP
- HTTP – Response: là phương thức để trả về kết quả mà máy khách đã yêu cầu máy chủ. Cấu trúc một dòng phản hồi gồm: phiên bản HTTP, mã trạng thái (1xx: thông tin, 2xx: thành công, 3xx: điều hướng lại, 4xx: lỗi phía máy khách, 5xx: lỗi phía máy chủ) và trạng thái

2. Giao thức WebSocket

a) Khái niệm Socket

Là giao diện lập trình ứng dụng mạng được dùng để truyền và nhận dữ liệu trên internet. Giữa hai chương trình chạy trên mạng cần có một liên kết giao tiếp hai chiều để kết nối hai tiến trình trò chuyện với nhau. Điểm cuối của liên kết này được gọi là socket

b) Cách thức hoạt động của Socket

- Chức năng của Socket là kết nối giữa máy khách và máy chủ là thông qua UDP, TCP/IP để truyền, nhận dữ liệu qua internet

- Hai ứng dụng nằm trên cùng một máy hoặc trên hai máy khác nhau. Nếu hai ứng dụng nằm trên cùng một máy thì hiệu số cổng bắt buộc không được trùng nhau

c) *Khái niệm về WebSocket*

Là công nghệ hỗ trợ giao tiếp hai chiều giữa máy khách và máy chủ bằng cách sử dụng dụng TCP socket để tạo kết nối

d) *Ứng dụng WebSocket*

WebSocket được sử dụng trong ứng dụng thời gian thực (hiển thị biến động giá chứng khoán liên tục được máy chủ đưa đến máy khách bằng websocket), ứng dụng trò chơi (dữ liệu được máy chủ liên tục nhận và không cần làm mới giao diện, giao diện sẽ được làm mới tự động mà không thiết lập kết nối mới), ứng dụng trò chuyện (sử dụng lại kết nối websocket để gửi và nhận tin nhắn và gửi tin nhắn 1-1)

3. So sánh HTTP và WebSocket

a) *Giống nhau*

HTTP và WebSocket đều là giao thức hỗ trợ truyền tải thông tin giữa máy khách và máy chủ

b) *Khác nhau*

| Nội dung | HTTP | WebSocket |
|--------------------|---|--|
| Quá trình vận hành | Giao thức 1 chiều, sau khi kết nối thực hiện xong và phản hồi lại kết quả, quá trình kết thúc và đóng lại | Giao thức 2 chiều, sau khi kết nối thực hiện xong và phản hồi lại kết quả và kết nối vẫn được sử dụng lại cho các kết nối sau cho đến khi kết nối được đóng bởi máy khách hoặc máy chủ |
| Ứng dụng thực tế | Hầu hết ứng dụng thời gian thực đều sử | Ứng dụng vào các dịch vụ ứng dụng |

| | | |
|------------------------------|---|---|
| | dùng websocket để truyền và nhận dữ liệu trên 1 kênh liên lạc duy nhất | RESTful đơn giản, chủ yếu là nhận thông tin 1 chiều về server xử lý rồi phản hồi và đóng kết nối |
| Ứng dụng theo mức độ ưu tiên | Đối với các ứng dụng cần thường xuyên hoặc liên tục cập nhật thông tin, websocket là ưu tiên hàng đầu vì kết nối nhanh và độ trễ thấp so với HTTP | Khi người dùng muốn giữ lại kết nối trong thời gian cụ thể hoặc chỉ để truyền dữ liệu thì HTTP được ưu tiên hơn |

4. Kết luận

Bài báo cáo đã đi tìm hiểu về khái niệm, cách thức hoạt động của HTTP và WebSocket. Đồng thời đưa ra so sánh giữa 2 phương thức này để thấy được ưu điểm và nhược điểm từ đó cho thấy WebSocket đang dần trở nên phổ biến và được ưu tiên hơn nhưng HTTP vẫn chưa thể bị thay thế hoàn toàn.