



IJIRCCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 5, May 2021

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 7.488

 9940 572 462

 6381 907 438

 ijircce@gmail.com

 www.ijircce.com

Web Real-Time Communication Based on Web Socket

Harshada Kotinna, Yogeshchandra Puranik

PG Student, Department of MCA, P.E.S's Modern college of Engineering, Pune, India

Assistant Professor, Department of Master of Computer Application, Modern college of Engineering, Pune, India

ABSTRACT: In a nutshell, Web-sockets technology makes it possible to open a two-way real time communication session between the user's browser and a server via a persistent connection. If the server can keep track of the clients, we can also send messages between clients which you might have seen in case of chat applications.

KEYWORDS: Websocket, Full-Duplex, HTTP Streaming, Long-Polling.

I. INTRODUCTION

The Internet has been an indispensable part of people's life in the fast growing information age. People's requirements for the Internet have changed from information accessibility in the Web 1.0 era to information interaction in the Web 2.0 era, and to current instant interaction observed in an increasing number of pricing systems, e-commerce systems, and news announce systems.

Currently, the communication between client browser and server is based on Hypertext Transfer Protocol (HTTP), an Application Layer protocol which is request-response based and stateless. An HTTP client initiates a request. It establishes a Transmission Control Protocol (TCP) connection. After receiving the client's request message, a server sends back a message as a response and terminates the connection. Under this model, servers can not send real-time data to clients. Therefore, technologies, such as Flash, Comet, and Ajax long polling, have been applied to implement real-time communication between client and server. However, these technologies cannot accomplish real-time communications, because some of them need install plug-ins on browsers, some of them cause heavy load for server. The emergence of HTML5 and the Web Socket protocol realize the real-time data transmission in web-based system, so far, they are considered as the best solution to resolve this issue.

Traditionally, a two way communication was achieved through long polling where the client would keep the request session open till timeout and wait for the server to respond with data. The request would be sent at regular intervals, if the data is available the server would send a response. This is because the servers as the name suggests, were designed for only responding to requests, they cannot send any data to clients directly. This is the limitation that gave importance to developing a protocol which can let the servers also send data to clients. The same communication shown in the diagram shown above when handled using Web-sockets will not need any polling from the client side. The Client will send a request and will get a response once the response is ready.

II . REAL-TIME ETHERNET-REALTIME SOCKETS

The real-time communication takes place via fast Ethernet boards as well as network interfaces with gigabit controllers by Intel and RealTek, so that fast measurement value logging or image processing is possible. In order to achieve real-time capabilities, the network controllers are controlled directly. This way, the entire bandwidth of the connection is available and an immediate reaction to received data packets takes place in real-time. In contrast to common processes in operating systems during such implementations, copy operations are generally avoided. For data transfer as well as data reception the same memory as the network controller is used. This allows for the highest possible transfer rate at the shortest reaction time.

Functions for CRS calculation and Byteorder conversion are also provided. As usual, with the Kithara system the Socket application can be compiled by developers within the framework of the application in C/C++ or Delphi. Based on a DLL the real-time application can be embedded in a C#.NET application as well.

III.COMPARISON BETWEEN WEB SOCKETCONNECTIONS AND HTTP CONNECTIONS

Web Socket is a computer communications protocol, providing full-duplex communication channels over a single TCP connection. The WebSocket protocol was standardized by the IETF as RFC 6455 in 2011, and the WebSocket API in Web IDL is being standardized by the W3C.

WebSocket is distinct from HTTP. Both protocols are located at layer 7 in the OSI model and depend on TCP at layer 4. Although they are different, RFC 6455 states that WebSocket "is designed to work over HTTP ports 443 and 80 as well as to support HTTP proxies and intermediaries," thus making it compatible with HTTP. To achieve compatibility, the Web Socket handshake uses the HTTP Upgrade header^[1] to change from the HTTP protocol to the WebSocket protocol.

The WebSocket protocol enables interaction between a web browser (or other client application) and a web server with lower overhead than half-duplex alternatives such as HTTP polling, facilitating real-time data transfer from and to the server. This is made possible by providing a standardized way for the server to send content to the client without being first requested by the client, and allowing messages to be passed back and forth while keeping the connection open. In this way, a two-way ongoing conversation can take place between the client and the server. The communications are usually done over TCP port number 443 (or 80 in the case of unsecured connections), which is beneficial for environments that block non-web Internet connections using a firewall. Similar two-way browser-server communications have been achieved in non-standardized ways using stopgap technologies such as Comet. Most browsers support the protocol, including Google Chrome, Firefox, Microsoft Edge, Internet Explorer, Safari and Opera. Unlike HTTP, WebSocket provides full-duplex communication.^{[2][3]} Additionally, WebSocket enables streams of messages on top of TCP. TCP alone deals with streams of bytes with no inherent concept of a message. Before WebSocket, port 80 full-duplex communication was attainable using Comet channels; however, Comet implementation is nontrivial, and due to the TCP handshake and HTTP header overhead, it is inefficient for small messages. The WebSocket protocol aims to solve these problems without compromising the security assumptions of the web.

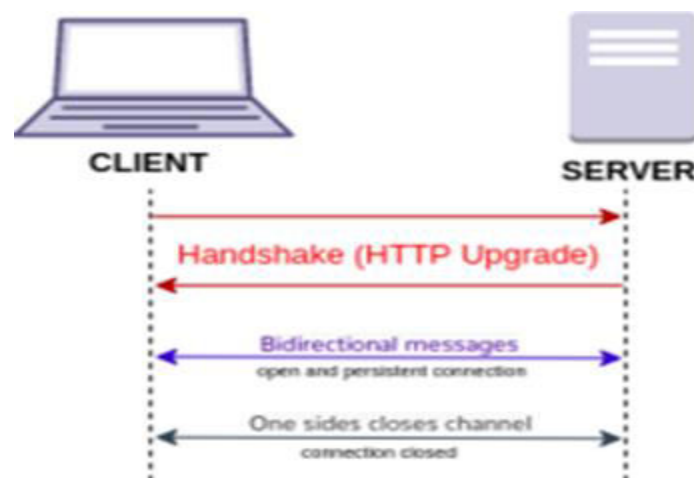


Figure 1. Handshake Protocol

Protocol handshake

To establish a WebSocket connection, the client sends a WebSocket handshake request, for which the server returns a WebSocket handshake response, as shown in the example below.

Client request (just like in HTTP, each line ends with `\r\n` and there must be an extra blank line at the end

```
GET/chatHTTP/1.1
Host: server.example.com
Upgrade:websocket
Connection: Upgrade
Sec-WebSocket-Key: x3JJHMbDL1EzLkh9GBhXDw==
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 13
```

Origin: <http://example.com>

Server response:

HTTP/1.1101Switching Protocols

Upgrade:websocket

Connection: Upgrade

Sec-WebSocket-Accept: HSmrc0sMIYUkAGmm5OPpG2HaGwk=

Sec-WebSocket-Protocol: chat

The handshake starts with an HTTP request/response, allowing servers to handle HTTP connections as well as WebSocket connections on the same port. Once the connection is established, communication switches to a bidirectional binary protocol which does not conform to the HTTP protocol.

In addition to Upgrade headers, the client sends a Sec-WebSocket-Key header containing base64-encoded random bytes, and the server replies with a hash of the key in the Sec-WebSocket-Accept header. This is intended to prevent a caching proxy from re-sending a previous WebSocket conversation,^[35] and does not provide any authentication, privacy, or integrity. The hashing function appends the fixed string 258EAF5-E914-47DA-95CA-C5AB0DC85B11 (a UUID) to the value from Sec-WebSocket-Key header (which is not decoded from base64), applies the SHA-1 hashing function, and encodes the result using base64.

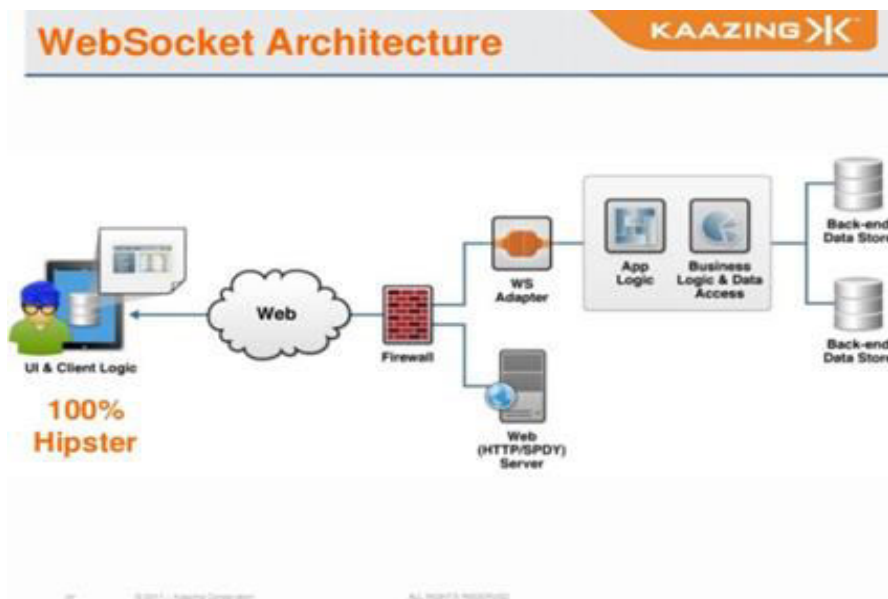


Figure 2. WebSocket Architecture.

Websockets is a technology for bidirectional communication over single (TCP) socket, a type of PUSH technology. It is a protocol which provides bidirectional, full duplex communication channels over a single TCP socket. It is designed to be implemented in web browsers and web servers, but it can be used by any client or server application.

IV. BENEFITS OR ADVANTAGES OF WEBSOCKETS OVER HTTP

Following are the benefits or **advantages of Web sockets over HTTP:**

- ➔ It supports duplex communication.
- ➔ Using websockets, one can send and receive data immediately faster than HTTP. Moreover they are faster than AJAX.



- ➔ Cross origin communication (however this poses security risks).
- ➔ Cross platform compatibility (web, desktop, mobile)
- ➔ HTTP takes upto 2000 bytes of overhead where as websocket takes only 2 bytes.
- ➔ Replace long polling
- ➔ Websockets are data typed but AJAX calls can only send string datatypes.

V.DRAWBACKS OR DISADVANTAGES OF WEBSOCKETS

Following are the drawbacks or **disadvantages of Websockets**:

- ➔ Web browser must be fully HTML5 compliant.
- ➔ Websockets has no success functions like AJAX.
- ➔ Intermediary/Edge caching is not possible with websockets unlike HTTP.
- ➔ To build even simple protocol of your own, one can not be able to use friendly HTTP statuses, body etc.
- ➔ If application does not require a lot of dynamic interaction, HTTP is much simpler to implement.

VI.CONCLUSION

Real-time data transmission will be an inevitable trend for web-based information system. Web Socket considered as the next generation of Ajax will be widely used in the Internet. Currently, the most popular browser IE8 and its lower versions still not support Web Socket. However, Kaazing Company has been developing an intelligent gateway which can convert Ajax polling and Comet used in lower version browser to Web Socket instant communication. Web Socket protocol and Web Socket API are still being updated. Probably, Web Socket will become the perfect solution for the “C10K” issue in the near future.

REFERENCES

1. Wikipedia, “WebSockets,” 2010. <http://en.wikipedia.org/wiki/WebSockets>
2. W3C, “The Web Sockets API,” 2009. <http://www.w3.org/TR/2009/WD-websockets-20091222>
3. https://www.tutorialspoint.com/websockets/websockets_api.html



INNO  SPACE
SJIF Scientific Journal Impact Factor

Impact Factor:
7.488

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

 9940 572 462  6381 907 438  ijircce@gmail.com



www.ijircce.com

Scan to save the contact details