REAL-TIME ANALYTICS

Karthika Devidasan, Liyamol Aliyar
Department of Computer Science and Engineering, ICET Muvattupuzha

Email:karthikadevidasan@gmail.com

Abstract: Internet communication provides the exchange of data between the client and the server.While doing this latency is a major issue faced by the internet communication.So inorder to overcome this we use Real-Time applications like polling,long polling,websockets. The most efficient communication can be done with websockets. This paper deals with the functioning of websockets and how we can overcome the limitations of polling and long polling. The Websocket specification defines an API establishing "socket" connections between a web browser and a server. There is a persistent connection between the client and the server and both parties can start sending data at any time.

Keywords:- Hypertext transfer protocol, websocket, long polling, transport layer protocol

I. INTRODUCTION

Websocket is a protocol providing full-duplex communications channel over a single tcp connection. Latency is a significant issue in applications such as networked control systems, where update frequencies of 10 to 500 milliseconds (ms) are required for adequate control of industrial processes. Closed-loop control over the Internet is possible by modeling the roundtrip delay and using UDP to consider only the most recent data, possibly discarding delayed packets. When an application must provide real-time data over an Internet connection in a peer-to-peer fashion, however (as when delivering real-time stock quotes or medical signals remotely for further processing), then latency becomes very important .HTTP polling is considered a good solution for delivering real-time information if the message delivery interval is known that is, when the data transmission rate is constant, as when transmitting sensor readings such as hourly temperature or water level. In such cases, the application developer can synchronize the client torequest data when it's known to be available.

When the rate increases, however, the overhead inherent to HTTP polling repeats significant header information, thus increasing latency. Earlier research posits that HTTP wasn't designed for real-time, full-duplex communication, due to the complexity of real-time HTTP Web designs applications. Thus, HTTP can simulate real-time communication only with a high price increased latency and high network traffic

.Long polling is a variation on HTTP polling that emulates the information push from a server to a client. The Comet Web application model ,for instance, was designed to push data from a server to a browser without a browser HTTP request, but is generally implemented using long polling to accommodate multiple browsers. Long polling isn't believed to provide any substantial improvement over traditional polling. The WebSocket protocol enables full-duplex communication between a client and a remote host over a single TCP socket .The WebSocketAPI is currently a W3C working draft, but the protocol is estimated to provide a three-to-one reduction in latency against half-duplex HTTP polling applications .Here,a comparison is done between the one-way transmission latency of WebSocket, long polling, and the best-case scenario for HTTP polling in a real-time application. The latency behavior at a 4-Hz rate for the low-volume communication (roughly 100 bytes per second of sensor data) typical of real-time sensor networks are experimentally validated.

II. RELATED WORK

In this section, introduce some fundamental backgrounds.

The windcomm web[1] application using the websocket protocol has three main components : the windsensor ,the base station computer and the client. The base station computer employs a jetty server running a web application called windcomm. This application communicates with the sensor and manages HTTP and websocket requests from clients. A client accesses the web application to see real-time wind sensor data using a web browser that supports the websocket protocol a HTML5's Canvas element. The Gill WindSonic is a robust, ultrasonic windsensor with no moving parts that measures win direction and speed . It connected the WindSonic to a base station computer through an RS232LS output cable connected to a USB serial port in the base station computer via an adapter. Simulated dynamic wind with an oscillating fan is used.

[2] Group communication is the exchange of information between groups of participants in a session. This can roughly be divided into two categories: synchronous and asynchronous. Asynchronous communication can easily be implemented by using HTML forms and back-end system including database and CGI-like application. Synchronous communication system also known as real-time Group communication is the exchange of information between groups of participants in a session. This can roughly be divided into two categories: synchronous and asynchronous. Asynchronous communication can easily be implemented by using HTML forms and back-end system including database and CGI-like application. Synchronous communication system also known as real-time communication system always includes a large number of passive recipient Ajax provides a mild salve to the HTTP communication model model by enabling web clients to asynchronously poll for server-side events. Comet introduced an even greater departure from the HTTP communications model by enabling "push"-style of communication over HTTP. HTML . WebSocket represents the next evolution of Comet and Ajax in an attempt to stand HTTP communications on its head. The HTML5 WebSocket specification defines a single-socket full-duplex (or bi-directional) connection for pushing and pulling information between the browser and server. Thus, it avoids the connection and portability issues of Comet and provides a more efficient solution than Ajax polling.

III. OBJECTIVES

1.Model

In this paper the system builds a javascript based application that can be plugged into any website.This is a webbased online shopping where users can use the site in realtime. An administrator will be entering all transfers into database in realtime through an admin application. All users visiting website will get a realtime update.This project uses the protocol websocket .Technology used are node.js at the backend server and angularjs at the frontend technology. The database used will be mongodb. SocketIO will be used for implementing websocket. The websocket protocol has two parts. The handshake consists of a message from the client and the handshake response from the server. The second part is data transfer. Jetty's implementation of the WebSocket API is fully integrated into the Jetty HTTP server and servlet containers. Thus, a Jetty servlet can process and accept a request to upgrade an HTTP connection to a WebSocket connection. Socketio library is used for websocket implementation. Socketio makes ws implantation across all browsers possible.For browsers not supporting wssocketio will fall back to a supported protocol like long polling. To evaluate the Internet's effectiveness for real-time data exchange, a comparison is done between Web-Socket communication with HTTP. Other Internet protocols, such as UDP is not considered, because they're designed for streaming real-time data when the newest data is more important and allowing older information to be dropped.

HTTP polling consists of a sequence of request response messages. The client sends a request toa server. Upon receiving this request, the server responds with a new message, if there is one or with an empty response if no new message is available for that client. After a short time , called the polling interval, the client polls the server again to see if any new messages are available. Various applications including chat, online games, and text messaging use HTTP polling.

One weakness associated with polling is the number of unnecessary requests made to theserver when it has no new messages for a client. Long polling emerged as a variation on thepolling technique that efficiently handles the information push from servers to clients. With long polling, the server doesn't send an emptyresponse immediately after realizing that no new messages are available for a client. Instead, the server holds the request until a new message is available or a timeout expires. This reduces the number of client requests when no new messages are available.

With continuous polling, an application must repeat HTTP headers in each request from the client and each response from the server. Depending on the application this can lead to increased communication overhead. The websocket protocol provides a full-duplex, bidirectional communication channel that operates through a single socket over the web and can help build scalable , real-time web applications. The websocket protocol has two parts. The handshake consists of a message from the client and the handshake response from the server. The second part is data transfer. Jetty's implementation of the WebSocket API is fully integrated into the Jetty HTTP server and servlet containers. Thus, a Jetty servlet can process and accept a request to upgrade an HTTP connection to a WebSocket connection.

2. Goals

The main security goals for our proposed protocol are: 1) Real-time tracking 2) device demographics 3) geo-location4) real-time message delivery

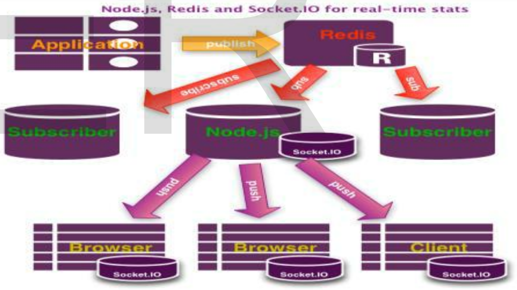
The paper real-time analytics deals with a real-time dashboard for customer analyyics and customer engagement.Site visitors can be tracked at realtime.It also tracks key parameters such as active users device demographics,geo-location and other user details.The action table tracks active users and rule action table tracks alerts based on predefined rules.

OUR PROPOSED PROTOCOL

The Proposed Protocol consists of five processes:

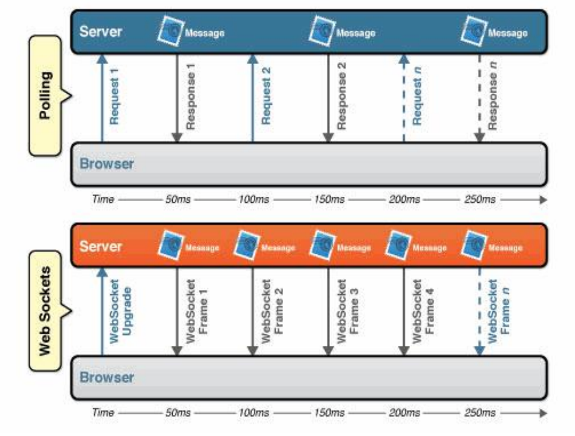
- Responsive by design
- Widgets are configurable
- Lightweight and perfomant
- Easy integration

The websocket protocol provides a full-duplex, bidirectional communication channel that operates through a single socket over the web and can help build scalable , real-time web applications.The websocket protocol has two parts. The handshake consists of a message from the client and the handshake response from the server. The second part is data transfer. Jetty's implementation of the WebSocket API is fully integrated into the Jetty HTTP server and servlet containers. Thus, a Jetty servlet can process and accept a request to upgrade an HTTP connection to a WebSocket connection. So considering these fators the dashboard gives a quick response to the users without allowing them to wait at all.



Web Socket is designed to be implemented in web browsers and web servers, but it can be used by any client or server application. The Web Socket protocol makes possible more interaction between a browser and a web site, facilitating live content and the creation of real-time applications. In Web Socket, a two-way(bi-directional) ongoing conversation can take place between a browser and the server. A similar effect has been archived in non-standardized ways using stop-gap technologies such as Comet. However, a comet is not trivial to implement reliably, and due to the TCP handshake and HTTP header overhead, it may inefficient for small messages. The WebSocket protocol aims to solve these problems without compromising security assumptions of the web.

PROBLEM FORMULATION



When the rate increases, however, the overhead inherent to HTTP polling repeats significant header information, thus increasing latency. Earlier research posits that HTTP wasn't designed for real-time, full-duplex communication, due to the complexity of real-time HTTP Web designs applications. Thus, HTTP can simulate real-time communication only with a high price increased latency and high network traffic.Long polling is a variation on HTTP polling that emulates the information push from a serverto a client. The Comet Web application model,for instance, was designed to push data from a server to a browser without a browser HTTP request, but is generally implemented using long polling to accommodate multiple browsers. Long polling isn't believed to provide any substantial improvement over traditional polling.TheWebSocket protocol enables full-duplex communication between a client and a remote host over a single TCP socket.The WebSocketAPI is currently a W3C working draft, but the protocol is estimated to provide a three-to-one reduction in latency against half-duplex HTTP polling applications.Here,a comparison is done between the one-way transmission latency of WebSocket, long polling, and the best-case scenario for HTTP polling in a real-time application.The latency behavior at a 4-Hz rate for the low-volume communication (roughly 100 bytes per second of sensor data) typical of real-time sensor networks are experimentally validated.

IV. CONCLUSION

Internet communication provides a convenient, hyperlinked, stateless exchangeof information, but can be problematic when real-time data exchange is needed. The WebSocket protocol reduces Internet communication overhead and provides efficient, stateful communication between Web servers and clients. To determine whether WebSocket communication is faster than HTTP polling, a Web application was built to measure the one way transmissions latency of sending real-time wind sensor data at a rate of 4 Hz. Thus, implemented a Jetty servlet to upgrade an HTTP connection to a WebSocket connection. Here, the comparison of Websocket protocol latency to HTTP polling and long polling is done.It can also be used for many online applications like online shopping and online tutorials ,to make them happen at a faster rate than the present long polling technique.The faster the data transmission the more the efficiency and less the time consumption.

REFERENCES

1. J. Åkerberg, M.M. Gidlund, and M. Björkman, "Future Research Challenges in Wireless Sensor and Actuator Networks Targeting Industrial Automation," Proc. 9th IEEE Int'l Conf. Industrial Informatics (INDIN 11), IEEE Press, 2011, pp. 410–415.

2. W. Hu, G.-P. Liu, and D. Rees, "Networked Predictive Control over the Internet Using Round-Trip Delay Mea- surement," IEEE Trans. Instrumentation and Measure- ment, vol. 57, no. 10, 2010, pp. 2231–2241.

3. P. Lubbers, B. Albers, and F. Salim, Pro HTML5 Programming: Powerful APIs for Richer Internet Appli- cation Development, Apress, 2010.

4. R. Gravelle, “Comet Programming: Using Ajax to Simulate Server Push,” Webreference tutorial, Mar. 2009; www.webreference.com/programming/javascript/rg28/ index.html.

5. P. Lubbers and F. Greco, “HTML5 Web Sockets: A Quantum Leap in Scalability for the Web,” SOA World Magazine, Mar. 2009; http://soa.sys-con.com/ node/1315473.

6. I. Fette and A. Melnikov, “The Websocket Protocol,” IETF Internet draft, work in progress, Dec. 2008.

7. I. Hickson, “The Websocket API,” W3C candidate rec- ommendation, Dec. 2008; www.w3.org/TR/websockets.

8. A.S. Tanenbaum, Computer Networks, 4th ed., Prentice Hall, 2003.

9. V. Pimentel and B.G. Nickerson, Web Display of Real- Time Wind Sensor Data, tech. report TR11-214, Faculty of Computer Science, Univ. of New Brunswick, Dec. 2002.

10. D. Mills et al., Network Time Protocol Version 4: Protocol and Algorithms Specification, IETF RFC 5905, June 2002; http://tools.ietf.org/html/rfc5905.