ISSN: 1304-7981 Number: **12**, Year: **2016**, Pages: **01-08** **http://jnrs.gop.edu.tr**

Gaziosmanpasa University
Graduate School of Natural and Applied Sciences
Journal of New Results in Science

Received: 15.05.2016 *Invited Editor*: Mahmut Hekim
Accepted: 08.09.2016 *Area Editor*: Zafer Dogan

The Comparison of New Web Communication Method WebSocket with Traditional Methods

Onur COMERT[a,1] (onur.comert@gop.edu.tr)
Mahmut HEKIM[b] (mahmut.hekim@gop.edu.tr)
Kemal ADEM[c] (kemal.adem@gop.edu.tr)

[a] *Gaziosmanpasa University, Department of Computer Programming, 60150 Tokat*
[b] *Gaziosmanpasa University, Department of Electrical and Electronics Engineering, 60150 Tokat*
[c] *Gaziosmanpasa University, Department of Informatics, 60150 Tokat*

Abstract – The Internet has become an indispensable part of human life in the rapidly growing information age. In response to this demand the number of web-based applications is increasing as well. Web-based applications run in two parts, one on the client and other on the server side. Client-side usually is a web browser, and it requests content from the server. As a result, the server provides the necessary information to the client. To perform the communication between the server and the client, usually Polling, Long Polling and Comet web communication methods are used. In this study, WebSocket, which is the new web communication method, was compared with the Long Polling and Comet methods. For this, a web server and a client computer was used. On the server, three separate web pages were created to accept connection through the WebSocket, Long Polling, and Comet methods. The client computer, which uses a browser to view these web pages, downloads the data on the server into its own memory in pieces at regular intervals. In other experiments, the client computer sends data of different sizes to the server at regular intervals. The comparisons were made based on the bandwidth used. The advantages and disadvantages of the WebSocket method against the Long Polling and Comet methods were presented by comparing the results of the experiments.

Keywords -
Websocket, Long Polling, Comet, Ajax.

1. Introduction

The Internet has become an indispensable part of human life in the rapidly growing information age. More and more people use Internet for activities such as communication with others, playing games, providing content, learning, etc., and devote more time to these

[1] ***Corresponding Author***

activities gradually. In response to this new demand, the number of web-based applications is also increasing gradually.

Web-based applications run in two parts, one on the client and other on the server side. Client-side usually is a web browser, and it requests content from the server. The contents to be requested by the clients are stored in the server. In addition, most of the logical processes and procedures occur on the server side. As a result, the server provides the necessary information to the client.

The benefits of Web-based applications development are as follows: First, they are platform-independent. One web browser is sufficient. Second, it is easy to develop. Basically, the HTML (Hyper-Text Markup Language) is used for developing Web-based applications. Third, it is easy to distribute a new version. When users want to run the latest version of a web application, they can simply achieve this by connecting to the web server.

2. Web Communication Techniques

In addition to the above-mentioned advantages, web-based applications have numerous disadvantages. In the web communication, clients are required to establish an HTTP (Hyper Text Transfer Protocol) connection for server connection. But HTTP is a unidirectional communication protocol. That is, in this protocol, communication only occurs with the client's request. A large number of unnecessary requests are made due to continuous checking of updated data for the client [1]. Several methods have been developed in order to eliminate these disadvantages. AJAX-based HTTP Long Polling and Comet are the most important among them.

2.1. Long Polling

In the HTTP Long Polling method, the connection remains open until there is either a timeout or a change in data requested by the client. Thus, the updated contents are received without page refresh, and the server does not need to send empty messages. Polling every few seconds to see whether there is an updated data is the drawback of this technique, as in the HTTP polling [1,2,3].

2.2. Comet

In the Comet method, a Comet server doesn't respond immediately to client requests. One server waits for an event mechanism. When this event occurs, the server responds to the client with the requested information. However, the Comet server has to respond to this request within this period (usually 30 seconds) before the HTTP connection times out [4]. This demonstrates the necessity of sending a message to the client even when there is no change in information. Otherwise, the client needs to reconnect to server because of the disconnected connection [2].

2.3. Websocket

WebSocket [RFC6455] covers the JS API and IETF protocols set by the W3C (World Wide Web Consortium). The WebSocket API has not been standardized yet. Nevertheless, many web browsers have included it in its final form. It is a network protocol running on TCP (Transmission Control Protocol). It was designed to be run on web browsers and web servers, but can be used for other purposes as well. Although it is independent from the HTTP protocol, it has similarities in terms of the handshake process required to establish a connection. The WebSocket servers use the same port (TCP 80 or 443) with HTTP [4,5,6,10].

WebSocket runs over a single TCP connection, and has similarities with the TCP connection method. First, the client makes an HTTP request to the server. This request has the "Upgrade WebSocket" header information. This means that the HTTP request is a handshake request with WebSocket. Then, the server receives the HTTP request, analyzes the information contained in the request, and generates a security key for the client. The WebSocket connection is established at this point, and now both parties (server and client) can transmit data to each other through this connection. After the connection is established, a session number will be sent to the server. This number shows only a specific Socket. The server maintains a list of session number in order to manage all the clients connected. When the client-side browser is closed, the server receives a warning message and then the browser and related resources are removed from client-list of the server [5,6,10].

The real reason behind the WebSocket is to replace Comet and HTTP Polling (and HTTP Long Polling), and improve the mediocre status of the real-time web applications. Compared with the common communication method on HTTP, the main difference of WebSocket is that it does not follow the traditional request/response standard. WebSocket needs few operating resources for data processing. WebSocket creates lesser network traffic in cases where small pieces of data need to be transferred periodically. WebSocket can reduce network traffic by 500:1 ratio, compared to ordinary HTTP method [5,7,9]. This is because, the size of each frame is 2 bytes in the Websocket connection. It exceeds 1 KB in Long Polling. In WebSocket, no new TCP connection is established for sending each HTTP messages. Hence, it has no latency needed for new connections [1].

WebSocket has a widespread use increasing day-by-day. In smart home systems, IoT (Internet of Things) devices send information collected from their sensors. WebSocket can be used for data transfer on these devices [9]. In addition, it can be used in web-browser-based multiplayer games with HTML5 [8], transmission of the energy measurement data [3], and web-based instant messaging and group communication [6,7].

3. Literature Survey

Many studies have been conducted to test the communication performance and applicability of the WebSocket protocol, and compare its performance with other methods of communication. Usually these studies have proposed WebSocket-based applications in various areas, and assessed the results of experiments performed on these applications.

In their study, Chen and Xu have designed a framework for game systems that use WebGL (Web-based Graphics Library) and WebSocket in order to investigate the real-time

performance of web browser based multiplayer games that use HTML5. As a result, they stressed that WebSocket and WebGL technologies will definitely support the development of browser-based multiplayer games [8].

In another study, Furukawa has developed a Web-based control software using WebSocket. Different browsers were used to access this application in the computer and mobile environments. Successful results suggested that the WebSocket protocol can be used in Web applications [4].

And, Swamy and Mahadevan have proposed a client/server communication system, and reported in their study carried out with the WebSocket protocol that the average latency has decreased from 150ms to 50ms, and WebSocket has better performance than the Long Polling in terms of bandwidth and scalability [1].

Later, Jiang and Duan have developed a WTC (Web Tree Component) based on web communication. The application has been tested by them using HTTP Long Polling, Comet and WebSocket. Compared to the other two methods, the program based on WebSocket technology has been found to have great advantages such as reduced server load, reduced network disconnections, and real-time connections. In addition, it has been stated that it provides an effective WTC communication, and may decrease the amount of node data transferred, hence increasing the rate of expansion of the tree structure, which is the basis of the WTC [2].

Similarly, Pimentel and Nickerson have implemented a web application to measure latency of unidirectional transfer of 4Hz real-time wind sensor in order to understand whether the WebSocket communication is faster than HTML Polling. They have compared WebSocket latency in HTTP Polling and Long Polling. As a result, they have found that the average latency in the Polling method was 2.3 to 4.5 times more than WebSocket or Long Polling. And, for the clients far away from the server, they found that the average latency is 3.8 to 4 times shorter in WebSocket compared to Long Polling [3]. These results also indicate that the WebSocket is more reasonable method for real-time Internet connections.

Zhangling and Mao have proposed a WebSocket-based group communication software in their study. The WebSocket-based communications module communicates with the client in an asynchronous and bidirectional manner, which increases the responsiveness of the system. A 100 byte message was sent continuously to the server in their tests. In total, 319 messages have been processed successfully on the server. These results showed that a WebSocket-based application like this can run on a low-cost server hardware easily [7].

In their study conducted about the real-time communication in web applications, Chao et al. have conducted 8 different experiments on the WebSocket and Polling methods. As a result, they found that the performance of WebSocket is generally better in more advanced conditions that require real-time. In the experiments, they demonstrated that WebSocket generates 4.48 to 44.77 times less traffic to transmit the same data compared to Polling [6].

These examples reveal the superiority of the WebSocket protocol than the other methods in real-time applications.

4. Sample Application

In the study, a series of experiments were carried out using a server and client with the purpose of comparing the WebSocket communication method with the Long Polling and Commet methods in terms of the bandwidth used.

4.1. Materials

The hardware and software specifications of the server used in the experiments are as follows.

Operating System: Ubuntu 14.04.3
Processor: Intel(R) Xeon(R) CPU E7- 2850 @ 2.00GHz
Memory: 8 GB

Only the free software products were used on the server. Apache 2.8 Web Server Service was used to provide access to the server through the web pages. The web pages were prepared using PHP, HTML and JS languages.

Free Wireshark software was used for monitoring the packets and measuring the bandwidth in the experiments.

4.2. Application

In the study, data were sent both from server to client and client to server, and 18 different experiments were conducted in total. The experiments are as follows:

State-1**:** Every second, 1-byte-long data was sent.

State-2**:** Every second, 100-byte-long data was sent.

State-3**:** Every second, 1-kilobyte-long data was sent.

State-4**:** Every 25 seconds, 1-byte-long data was sent.

State-5**:** Every 25 seconds, 100-byte-long data was sent.

State-6**:** Every 25 seconds, 1-kilobyte-long data was sent.

State-7**:** Every 60 seconds, 1-byte-long data was sent.

State-8**:** Every 60 seconds, 100-byte-long data was sent.

State-9**:** Every 60 seconds, 1-kilobyte-long data was sent.

The results of the experiments are shown in the graphs below.

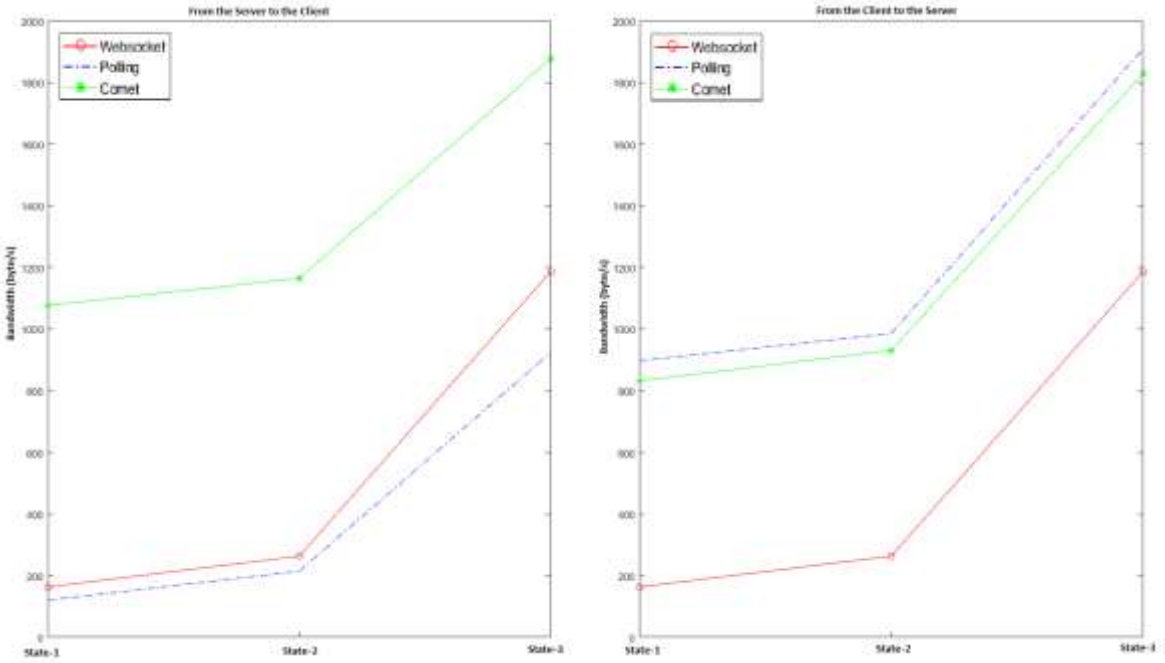

Figure 1: State-1, State-2 and State-3

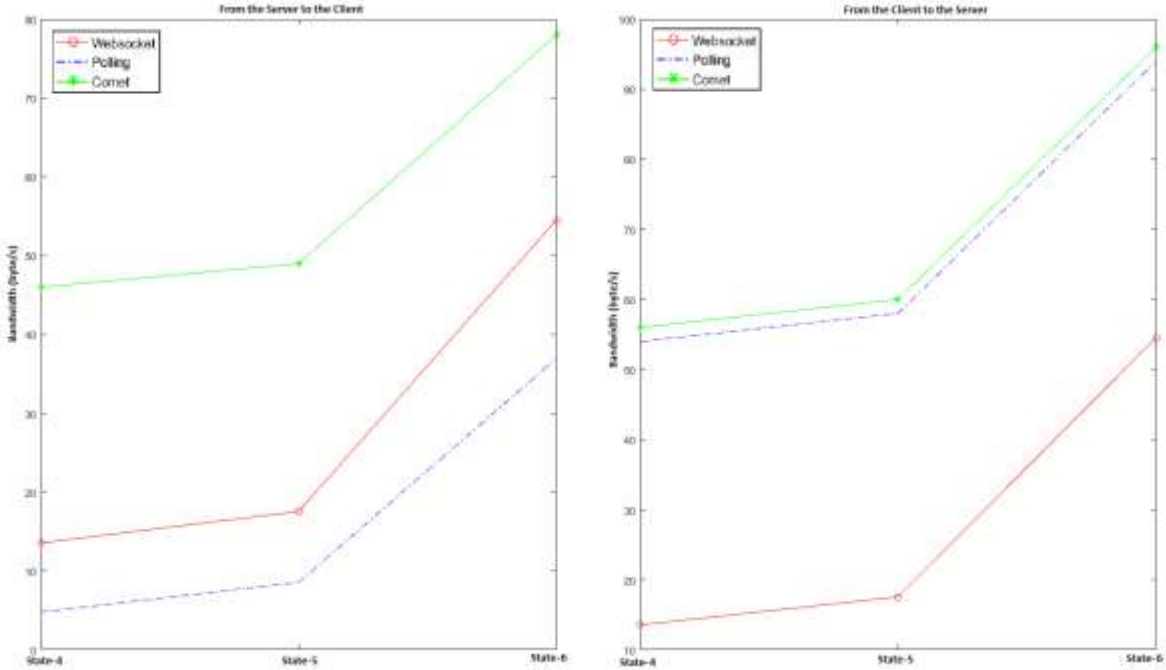

Figure 2: State-4, State-5 and State-6

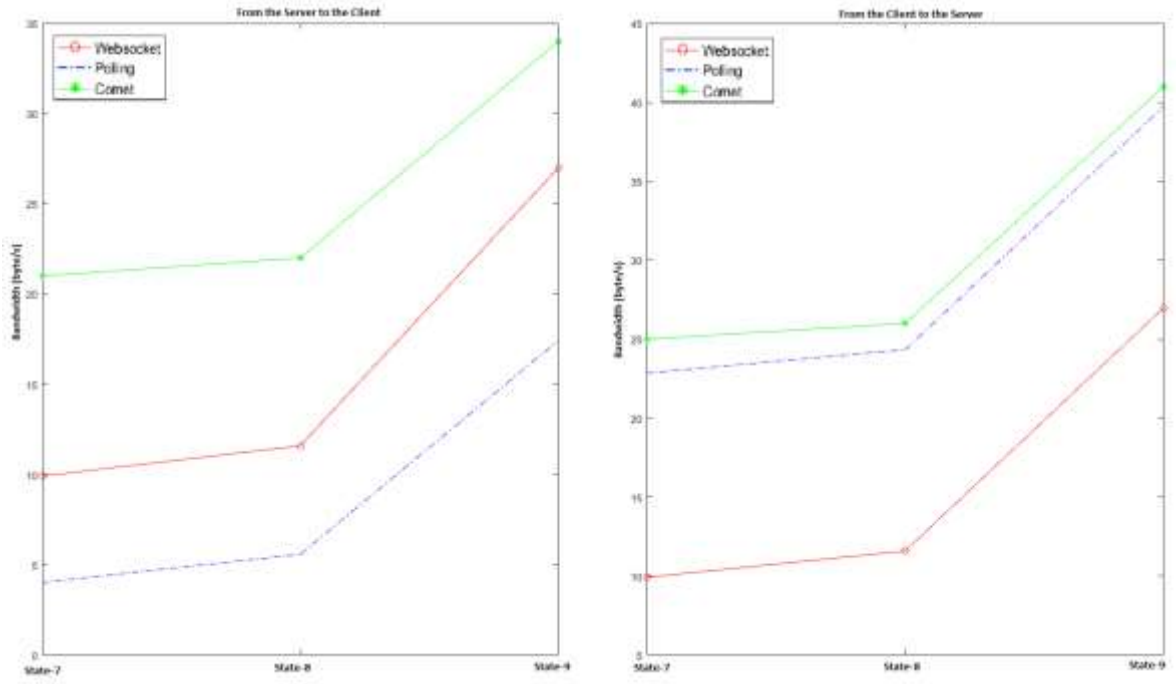

Figure 3: State-7, State-8 and State-9

In the graphs, Y-axis shows the total data received and sent per second in bytes. The X-axis shows the states. As shown in Figure 1, Figure 2, and Figure 3, the Long Polling method was more efficient compared to the WebSocket and Comet methods in terms of bandwidth used for the data sent from server to client. And, for the data sent from client to server, the WebSocket method was found to be more efficient than the Long Polling and Comet methods in terms of bandwidth used.

4. Conclusion

In this study, a series of experiments were carried out for comparing the new WebSocket method and traditional Long Polling method in terms of bandwidth used. According to the results of the experiments, the Long Polling method was observed to be more efficient for continuous data stream from server to client. This is because it uses a keep-alive management that has better performance compared to WebSocket and Comet methods. In the Long Polling method, the keep-alive message is not sent if the data are transmitted at intervals of less than 30 seconds, and delayed after a data transmission. In this way, less data traffic is created to maintain a persistent connection between a client and a server. And, in the WebSocket and Comet methods, this message is sent with 25- and 45-second intervals, whether there is data to send or not. This means extra bandwidth.

And, for the cases where data is sent from client to server continuously, the WebSocket method was found to be more efficient than the Long Polling and Comet methods in terms of bandwidth. This is because, in the Long Polling and Comet methods the client needs to send a connection request to the server constantly to sand data. This means extra data traffic. The bandwidth remains constant in the WebSocket method.

As a result, the WebSocket method has shown a higher performance compared to the Long Polling and Comet methods, on average. This indicates that WebSocket method can be preferred in certain applications in the future, despite being a new method.

References

[1] R. Swamy, G. Mahadevan, *Event driven architecture using HTML5 web sockets for wireless sensor networks*, Planetary Scientific Research Center, 2011.

[2] F. Jiang, H. Duan, *Application research of websocket technology on web tree component*, In: Proceeding of International Symposium on Information Technology in Medicine and Education (ITME 2012) IEEE, 978-1-4673-2108-2/12, pp: 889-892, 2012.

[3] V. Pimentel, B. Nickerson, *Communicating and displaying real-time data with websocket*, IEEE Computer Society, 1089-7801/12, pp:45-53, 2012.

[4] Y. Furukawa, *Web-based control application using websocket*, In: Proceedings of ICALEPCS2011, Grenoble, France, WEMAU010, pp: 673-675, 2011.

[5] J. Erkkilä, *Websocket security analysis*, Aalto University T-110.5291 Seminar on Network Security, 2012.

[6] Y. Chao, L. Ming, L. Yan, *Review on real-time communications technology in web application*, Advanced Materials Research Vols. 1044-1045, pp:1309-1314, 2014.

[7] Y. Zhangling, D. Mao, *A real-time group communication architecture based on websocket*, International Journal of Computer and Communication Engineering, 1(4) 408-411, 2012.

[8] B. Chen, Z. Xu, *A Framework for browser-based multiplayer online games using webGL and websocket*, IEEE, 978-1-61284-774-0/11, pp:471-474, 2011.

[9] K. Ma, R. Sun, *Introducing websocket-based real-time monitoring system for remote intelligent buildings,* Hindawi Publishing Corporation International Journal of Distributed Sensor Networks, 2013.

[10] I.B. Castillo, J.M.V Versatica, V.P. Quobis, *The websocket protocol as a transport for the session initiation protocol (SIP),* Internet Engineering Task Force (IETF), ISSN: 2070-1721, 2014.