International Journal of Multimedia and Ubiquitous Engineering
Vol.9, No.7 (2014), pp.61-72
http://dx.doi.org/10.14257/ijmue.2014.9.7.06

Study of HTML5 WebSocket for a Multimedia Communication

Jin-tae Park[1], Hyun-seo Hwang[1], Jun-soo Yun[1] and Il-young Moon[1]

[1]*School of Computer Science and Engineering,*
Korea University of Technology and Education
{wlsxo05, smilebear1, yuntn55, iymoon}@koreatech.ac.kr

Abstract

Recently, with the advent HTML5, the performance of Web service has improved. It was difficult to configure the web services using only HTML. However, communication between the client and the server is now possible because of HTML5. HTML5 was designed to be platform independent, and can be used on an increasing number of mobile devices for creating both mobile websites and mobile applications. It is frequently cited as the primary solution for enabling effective cross-platform deployment onto various smart devices. HTML5 has many new features. One of the most powerful features is the WebSocket. Updated HTML5 specifications for web sockets, web browsers, and web servers allow sending and receiving data in real time, via an always-on TCP connection. The crucial differences between pre-HTML5 communication methods and HTML5 WebSocket are located in the new HTML5 protocol. WebSocket protocol uses HTTP to establish a connection, but the subsequent communication is performed by the WebSocket reader protocol. In addition, the feature header is very small, resulting in reduced communication overhead. The WebSocket aims to solve the problems of the conventional communication method, though it has several restrictions. Therefore, in this study, experiments were conducted to measure the performance analysis of WebSocket. We conclude by presenting the direction of future research in this field.

Keywords: *HTML5, WebSocket, Cross-Browser, Multi-media, Polling*

1. Introduction

HTML5 has attracted significant attention in the recent years. HTML5 is the next-generation standard proposed in HTML [1]. It is not possible to configure the Web services using only HTML. However, with the advent of HTML5, it is possible for the clients and servers to communicate. HTML5 was designed to be platform independent, and can be used on an increasing number of mobile devices for creating both mobile websites and mobile applications. It is frequently cited as the primary solution for enabling effective cross-platform deployment onto various smart devices. Some features of HTML5 may be provided to Web services without using external modules [2]. Market research firm Gartner, Inc. nominated HTML5 as one of the top 10 strategic technology trends for 2013. Gartner stated that many mobile development platforms exist in the market, but did not point out if any of them could sufficiently support development for multiple platforms. Therefore, claims that HTML5 can enable cross-platform implementation may exercise powerful influence, and could help it further increase its growing developer base [3]. According to a study by Vision Mobile, 52% of mobile application developers worldwide have used HTML5 as a development platform as of April 2013. This percentage increased by 2% from January 2013. For the same timeframe, 71% of mobile developers responded that they had used the Android

ISSN: 1975-0080 IJMUE

development platform while 57% responded that they had used iOS. These results demonstrate that utilization of the HTML5 development platform is increasing relative to Android and iOS. Figure 1 displays a graph of platform utilization ratio trends.

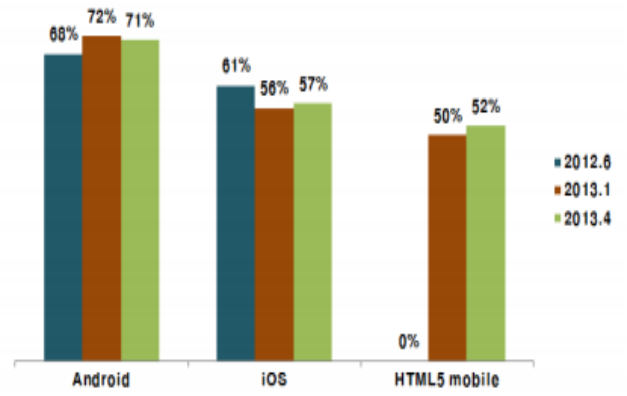


Figure 1. Proportional Changes in the Use of Application Development Platforms

Developers who have written a mobile application for a particular platform, commonly referred to as a native application, must perform a conversion process to implement the application on other platforms. Therefore, it is frequently necessary to re-design the application so it can be utilized on the new platform, often requiring a substantial time investment and additional funding [4]. In contrast, mobile developers can use HTML5 to write applications that can be implemented on a wide range of devices, in the same manner that web-based applications can be accessed through web browsers running on many different platforms. According to 58% of developers that were surveyed, reduced costs and development time for mobile applications were identified as the most significant advantage of HTML5.

The browser manufacturers worldwide are cooperating in the development of HTML5. The most powerful feature is the ability of the WebSocket application programming interface (API) [5]. One of the early purposes of the Web was to link a document using hyperlinks and facilitate document transfer. The HTTP protocol for the network is a model that handles this purpose. However, with changing times and with developments in the environment, the purpose of the Web no longer focuses only on sharing longer documents. In the HTML5 initiative, many specifications have been developed under the banner of the Web application environment that are not standardized and are inconsistent with the available plug-ins, which was one of its main purposes [6]. Among them is the specification for real-time two-way communication in a pure Web environment. Updated HTML5 specifications for web sockets, web browsers, and web servers allow sending and receiving data in real time, via an always-on TCP connection [7]. Thus, the utilization of WebSockets, enabling full-duplex communication using TCP sockets, is now possible. Instead of evolving from HTTP communication, merely using an existing web socket shows significant improvements in properties [8]. In particular, in the event-based Web application, which is based on real-time communication, the effect is even greater. The HTML5 WebSocket technology may be able to receive information quickly via the push real time, and it may reduce the wait time for unnecessary network traffic. Therefore, in this paper, we conducted research on the HTML 5

WebSocket performance analysis by comparing the overhead depending on the number of concurrent users and the speed of the real-time multimedia communication.

2. Related Technology Trends

2.1. HTML5 WebSocket Method

As a Web application platform and a next-generation technology for productivity improvement of Web development, HTML5 is an open Web standard created to provide a better user experience. It has become possible through HTML5 to provide an excellent Web service rapidly [9]. The function of the WebSocket technology was to improve performance. Two-way data communication in real time actually corresponds to numerous concurrent connections. In this case, the real-time aspects specifically, WebSocket technology is often used. If the WebSocket technology is used, the unnecessary HTTP header data must be removed first in order to quickly send and receive pure data. This results in a reduction of the amount of data being transmitted and received; consequently, this reduces the load on the server and the network. Then, using the WebSocket Secure (WSS) protocol, security was enhanced with a unique algorithm for encryption / decryption of data. Figure 1 is a graph comparing the conventional communication method and new WebSocket system [10]. Thus, with the use of the WebSocket technology, connection-oriented full-duplex communication, such as TCP sockets, is possible [11]. Using features such as these, we were able to implement more effectively the development of applications that allow chatting from the Web, real-time games, and multimedia delivery.

Table 1. Description of WebSocket Servers

Server	Description
Pywebsocket	- WebSocket server developed in Python. - Independently sets the web server socket. - Can be installed on an Apache server. - Python must be installed on the server.
Phpwebsocket	- WebSocket server that operates in a php environment. - Google Chrome is the only supported browser.
jWebSocket	- Java-based WebSocket server - Can install on the server or on Apache Tomcat - Java Virtual Machine must be installed
web-socket-ruby	- Ruby-based WebSocket server - Client technologies developed
Socket.io	- Real-time application framework for Node.js

The crucial differences between pre-HTML5 communication methods and HTML5 WebSocket are located in the new HTML5 protocol. WebSocket protocol uses HTTP to establish a connection, but the subsequent communication is performed by the WebSocket reader protocol. In addition, the feature header is very small, resulting in reduced communication overhead. Because a persistent connection is assumed, it is possible for clients and servers to remain in a connected state. Accordingly, it is necessary to update existing TCP servers to meet the new WebSocket specification. WebSockets are present in a variety of servers. These servers may be implemented in

languages such as Java, Python, php, and Ruby. Table 1 provides descriptions for some selected WebSocket-enabled servers.

HTML5 has emerged as Server-Sent-Events (SSE) technology. This is a standard technique that has been proposed in HTML5 to implement server pushes in a web environment. However, SSE can also be utilized for socket communication, rather than as an aggressive communication method from the server to the client; for example, server push is a technology commonly employed by betting websites. HTML5 WebSockets provides the capability for pure two-way real-time communication on the web, and does not require browser plug-ins to be installed; this differentiates it from other technologies such as Java applets and ActiveX. Figure 2 depicts an HTML5-based WebSocket system.

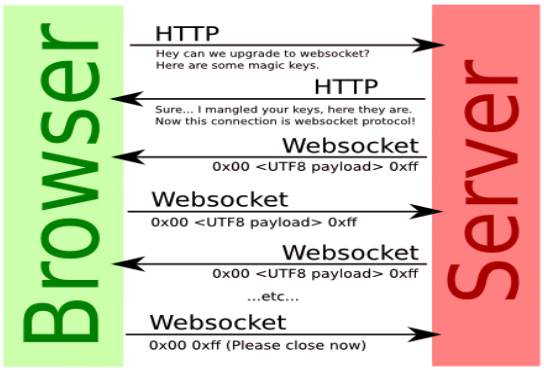


Figure 2. WebSocket Method

2.2. Polling Method

In general, HTTP is a method in which the client sends a request to the server, and subsequently the server responds. Unlike socket programming, HTTP is closer to one-way rather than two-way communication; it was not designed to enable servers to push messages to clients. However, as web communications have gradually become more complex, various technologies to push data from the server to the client have been designed [12]. This led to the invention of Ajax, a periodic polling method. However, Ajax's polling method is very inefficient, and is lacking in capability to transfer data in real time. Therefore, several other methods began to appear, although none of them really integrated seamlessly with HTTP. One resulting method is Comet, also known as Reverse Ajax. Comet employs a long polling method, using a script tag and XHR streaming system utilizing hidden HTML IFrames.

Streaming is a method in which the client and server maintain a connection. Each event represents the transmission of data to the client. When data is requested by the client, the server side remains connected and sends messages in succession without requiring a response. This method does not require client requests to continue, but requires simultaneous connections, and can be adversely affected by the number of connections.

In the long polling method, an HTTP request is sent and a response received, similar to the streaming method. However, after loading the page, it sends the request using a separate XMLHttpRequest (XHR) script; if the server does not immediately send a response to it, an event is generated on the server side [13]. A response is then transmitted to the client. By using this method, when an event occurs in the server, it is possible to transmit the data to the client in real time. However, after sending a response, the connection between the client and the server is disconnected. Fig 3 displays a comparison of Ajax, long polling, and streaming methods.

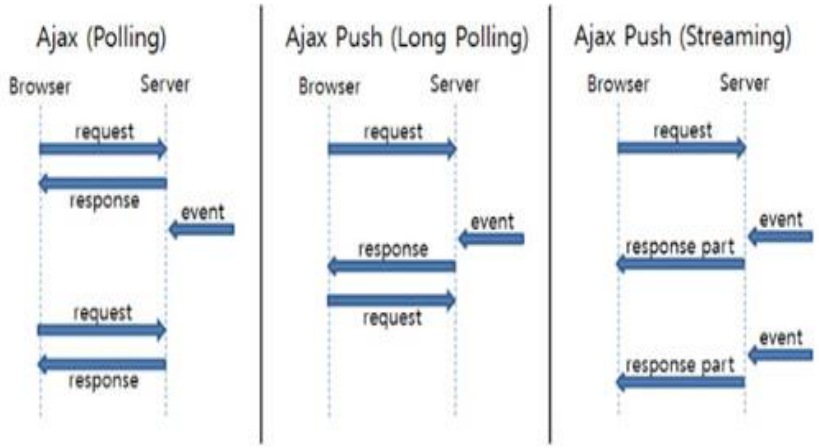


Figure 3. Polling Method

2.3. Comparative Analysis of the Polling Method and WebSocket System

Ajax and Comet with IFrames are polling methods. That is, a polling system is used instead of a push method for receiving data. The server sends messages to the client, the client sends a response to the server. The Comet technique has generally been the best choice to stay connected to avoid repeated requests. It must be re-connected to close the connection after a certain time period. This technique is reflected in HTML5, and was a factor leading to the design of WebSockets. WebSocket is a protocol that provides a bidirectional connection between the web server and the client. Unlike existing sockets, WebSocket provides actual two-way connection [14]. It begins with a HTTP-based handshake, then the WebSocket protocol takes over to facilitate communication. Because of its more advanced design compared to existing methods such as Comet, WebSocket establishes two-way connection in a different way; traffic is very light using WebSocket, and it demonstrates superior performance [15]. Figure 4 displays a comparison of polling methods and the WebSocket system.

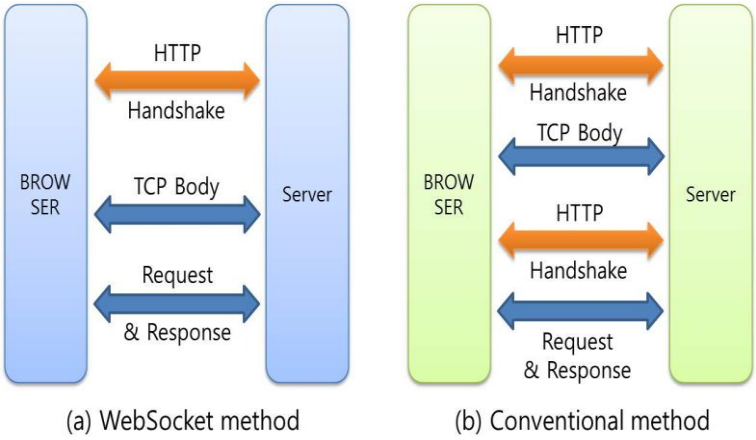


Figure 4. Comparison of WebSocket and Conventional Polling Methods

3. Design and Implementation of the Experiment

3.1. Design

The following assumptions are made:

- WebSocket technology is required.
- Real-time two-way data communication is required.
- There are many concurrent connections.
- It is necessary to extend the TCP-based communication browser.
- The developer is required to use the easy-to-use API net reason.
- It is necessary to extend the SOA beyond environments such as the cloud, and Web.

Therefore, in this paper, our experimental design functions through the polling conventional manner with real-time two-way data communication using the WebSocket methodology through the transfer delay time. We compare the overhead corresponding to the number of concurrent users, and analyze its performance. To ensure that the experiment is accurate, we implemented the Web to execute the polling method and the WebSocket technology to execute the multimedia data communication system. For the client polling method, we use the setInterVal () method, and transmit the HTTP requests at regular intervals through the browser.

3.2. Experiment Environment

In this paper, we propose a HTML5-based client/server environment for the experiment. The client and the server will utilize two-way communication to send and receive multimedia data. We conducted experiments using WebSocket methods and polling methods to connect the client and server. The WebSocket and polling system source code for both client and server is displayed in the following section. The client requests a WebSocket connection on the server with JavaScript (JS), included in the HTML to be sent to the HTTP server. Therefore, it transmits the HTML by constructing an HTTP server node.js.

```
var clientScript = function() {
  var ws = new WebSocket("ws://localhost:3000/", ["test", "chat"]);
  ws.onopen = function() {
    console.log(ws);
    ws.send("Test");
    ws.onmessage = function(message) {
      console.log(message.data);
    };
  }
}

var server = http.createServer(function(req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  var html = '<html><head><title>Web Socket 서버</title>' +
    '<script type="text/javascript">' +
    '('+ clientScript + ')();' +
    '</script>' +
    '</head>' +
```



```
    '<body>WebSocketTest</body>' +
    '</html>';
  res.end(html);
});
```

- Source of the WebSocket server system

```
var ws = new WebSocket("ws://localhost:3000/", ["test", "chat"]);
ws.onopen = function() {
  ws.send("test");
  ws.onmessage = function(message) {
    console.log(message.data);
  };
}
```

- Source of the WebSocket client system

```
<script
  language='javascript' src='http://ajax.googleapis.com/ajax/libs/jquer
  y/1.4.2/jquery.min.js'></script>
   <script>
   $(document).ready(function(){
   jQuery.ajax({
    type:"GET",
    url:"test.html",
    success:function(msg){
    alert(msg);
   },error: function(xhr,status,error){
   alert(error);
   }
  });
  });
  </script>
```

- Source of the Polling client system

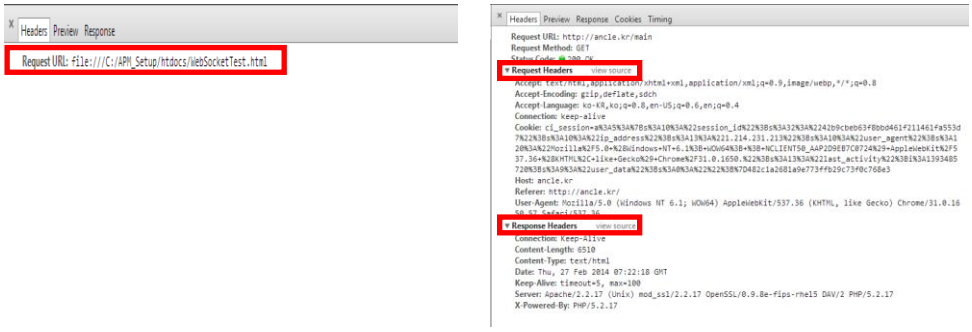
By comparing the results from the WebSocket and polling method client/server configurations, we analyzed the header information for the server in response to client requests.

4. Analysis Result

4.1. WebSocket and Polling Methods

We compared the overhead of the network implemented via the server and the client. As shown in Figure 5(a), the system is a WebSocket header data request and the response does not exist. On the other hand, as shown in Figure 5(b), in the polling method, whenever there

is a response to the HTTP request, the HTTP headers are passed, and the associated overhead occurs.



(a) Value of WebSocket header (b) Value of polling header

Figure 5. Comparison of the ActualOoverhead of WebSocket and Polling Method

In order to communicate multimedia data of one gigabyte, the polling method is used. The WebSocket system needs the capacity of the header data to be about 1,000 bytes; the header is not needed because of the low data capacity. It is assumed that when ten persons access the multimedia data of one gigabyte using the polling method, (that the header data capacity is 10,000 bytes, then when 100 people access the multimedia data of one gigabyte, that the header data capacity will be 100,000 bytes. On the other hand, with the WebSocket type, the additional capacity does not occur. These additional header data generate the overhead for a large number of users on the network.

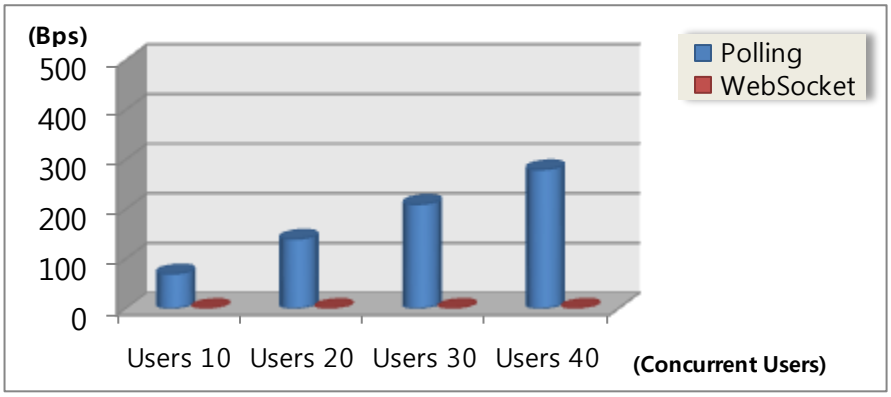


Figure 6. Comparison of the Overhead Generated by the Polling and WebSocket Method based on the Concurrent Number of Users

In addition, we also measured the transfer delay time and the overhead generated by increasing the number of concurrent connections. We compared the overhead generated by an increasing number of concurrent users. Figure 6 shows the results. As the concurrent number of users increase, the overhead of the polling method increases, while the overhead of WebSocket system was barely noticeable. In the polling method, starting from the client's request, the forward delay time, the time of 60 ms consuming, and the time while waiting re-

request, even if the response occurs after the response time from the server, all become overhead. However, using the WebSocket system, connections that occur after the first connection, resulted in the connection being maintained, with no additional latency. Figure 7 is a graph comparing the response times of the system using the polling and WebSocket methods. The x-axis represents the number of requests, and the y-axis represents the response time (ms).

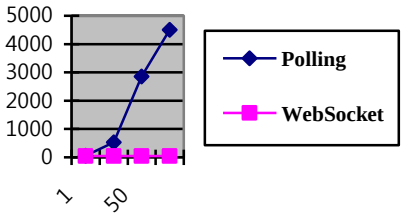


Figure 7. Comparison of the Response Time of the WebSocket and Polling Method

4.2. Cross-Browser

In this paper, a comparison was made between the overhead and response time of the polling and WebSocket methods. Currently, not all HTML5 functions work across browsers. For example, WebStorage is a new feature in HTML5 that provides a function similar to cookies, but allows more advanced data storage on the client. However, because different browsers store data in different repositories, data access cannot be performed where compatibility is an issue. This was subjected to cross-browser tests using the two methods, in order to solve this problem.

- Save each to each Browser
  If an inspection has not been performed to determine whether a cookie is present in each browser, use cookie as the recording method.

- Shared Flash Object
  Use the Shared Object for shared data storage, for client-side Flash presentation.

The next figure displays an information screen indicating that you are logged into Chrome for either of the previous methods, and is recognized by IE.

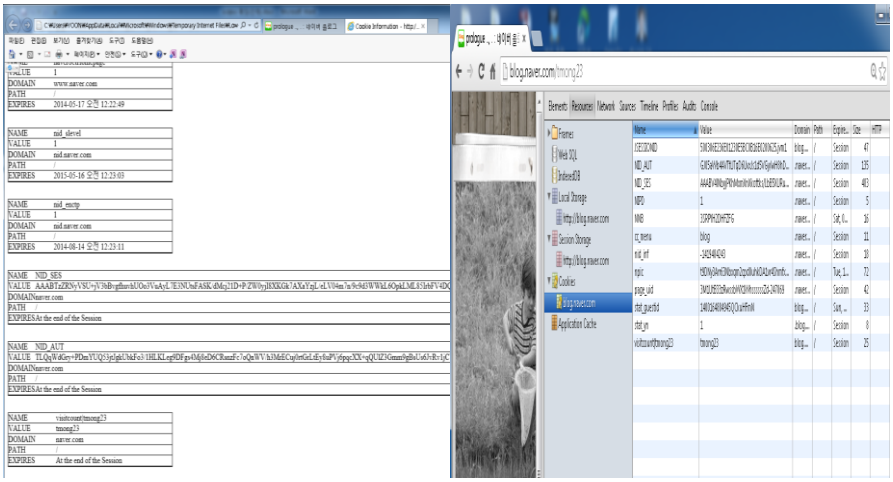


Figure 8. Cross-Browser compared with Two Browsers

5. Conclusion

Interest in the new Web Of Things (WoT) standard is rapidly increasing. Specific interest in this proposed standard, which connects large numbers of devices together using the web, will continue to increase, thus increasing interest in HTML5. In such a situation, with the advent of HTML5, we are able to provide better service, which is significantly faster than the existing Web services. HTML5 uses an open Web standard as a Web application platform and is a next-generation productivity improvement for Web development, which works together to create a better user experience.

HTML5 is a concept that includes a JavaScript API extension for CSS3, the web style sheet language. CSS3 is capable of providing a representation that is consistent with different browsers, in a convenient and effective manner. The JavaScript API allows the development of local applications, and can control the various resources and rich features. This includes WebStorage, WebWorker, WebSocket, and the GeolocationAPI.

The function of WebSocket technology is to improve performance. This reduces the network load when compared to conventional methods. The WebSocket technology is often used when real-time two-way data communication is required. In particular, the positive effects of using the WebSocket technology are more pronounced while using web applications based on real-time two-way data communication. On the other hand, not many browsers that support HTML5 support web sockets. Further, if security is enabled while using a web socket, it proves to be disadvantageous. Currently, technical research is actively promoted for using WebSocket technology to improve its performance. In addition, future research can be conducted to study cross-browser compatibility with WebSocket technology, and reduction of its security vulnerability can be researched.

Acknowledgement

This research was financially supported by the Ministry of Knowledge Economy(MKE) and Korea Institute for Advancement of Technology(KIAT) through the Research and Development for Regional Industry.

References

[1] C. Y. Lee, "The Next Generation of Web Standards HTML5", Journal of ASKO (**2012**).
[2] E. M. Lee, "The Impact of HTML5 on Web", Journal of KIISE, vol. 29, no. 55, (**2011**).
[3] H. H. Choi and G. H. Kim, "A Study on the Web Service Migration Based on Websocket", Journal of KIPS (**2011**).
[4] H. Y. Kim, D. H. Lee, D. Y. Park and E. H. Lee, "Model Definition for Type and Relation of HTML5 based Smart TV Application", Journal of KOSBE, (**2013**).
[5] Y. D. Kim and M. H. Kim and I. Y. M, "Web based File Transmission System using HTML5", Journal of KONI, (**2012**).
[6] S. H. Lee, W. K. Lim and J. H. Kim, "Issue of information security and the outlook of the HTML5 next generation Web standard", Journal of KSII, (**2012**).
[7] Y. K. Yang, "Performance Analysis of Asynchronous Data Communication using WebSocket of HTML5", Kangwon University (**2013**).
[8] H. H. Choi and G. H. Kim, "A Study on Service Migration in HTML5 based HTTP Streaming Environments", Journal of KMS, vol. 14 , no. 905, (**2011**).
[9] D. L. Lee, "SGML, HTML, XML technical and Prospects", Sorabol, vol. 19, no. 181, (**2001**).
[10] M. K. Sin, "Design and Implementation of Hotjava Multicast Protocol Handler for Real-time Distribution of HTML Documents", Conference on Communication Software, vol. 1, no. 131, (**1997**).
[11] K. H. An, T. K. Lee, K. S. Kim and M. P. Hong, "A Study on the Improvement Scheme of Real-time Booking System which is Based on the HTML5 WebSocket", Journal of KIISE, vol. 2013, (**2013**).
[12] I. S. Lee, S. R. Cheon, S. H. Sin and H. W. Kim, "A Study on the Real-Time Retrieval System by Polling I / O Method", Journal of IEEK, vol. 10, no. 38, (**1986**).
[13] S. J. Hong, "Performance Analysis of Polling Systems", Journal of KIIE, vol. 1996, no. 75, (**1996**).
[14] C. H. Park and H. Y. Jeong, "Design and Implementation of a Scheduling Secretary Agent System Based on HTTP Client/Server Mechanism", Journal of KIPS, vol. 7, no. 862, (**2000**).
[15] K. K. Lee and J. Y. Choi, "A Study on the Use of the Web Storage HTML5", Journal of KIISE, vol. 38, no. 430, (**2011**).

Authors

Park Jin Tae, M.S
Computer Science and Engineering,
Korea University of Technology and Education,
Gajeon-ri, Byeongcheon-myeon, Dongnam-gu, Cheonan-si,
Chungcheongnam-do 330-708, Korea
E-mail: wlsxo05@koreatech.ac.kr

Hwang Hyun Seo, M.S
Computer Science and Engineering ,
Korea University of Technology and Education,
Gajeon-ri, Byeongcheon-myeon, Dongnam-gu, Cheonan-si,
Chungcheongnam-do 330-708, Korea
E-mail: smilebear1@koreatech.ac.kr

Yun Jun Soo, B.S
Computer Science and Engineering ,
Korea University of Technology and Education,
Gajeon-ri, Byeongcheon-myeon, Dongnam-gu, Cheonan-si,
Chungcheongnam-do 330-708, Korea
E-mail: yuntn55@koreatech.ac.kr

Moon Il Young, Ph.D
Computer Science and Engineering,
Korea University of Technology and Education,
Gajeon-ri, Byeongcheon-myeon, Dongnam-gu, Cheonan-si,
Chungcheongnam-do 330-708, Korea
E-mail: iymoon@koreatech.ac.kr