

IPS: A Lightweight Framework for Cross-Platform Multimedia Streaming Server

Chao-Hsien Hsieh
School of Software
Engineering, Qufu Normal
University, Qufu, China

Chih-Hong Ke
Department of Information
Management, Chang Jung
Christian University, Taiwan

Chiang Lee
Department of Computer
Science and Information
Engineering, National Cheng
Kung University, Taiwan

ABSTRACT

Multimedia presentation has been popular by used in the field of education, especially in online learning. The most common practice is to conduct a presentation by utilizing media player on web or native applications. However, this mode of operation belongs to a heavyweight practice which incurs poor performance. Considering smooth operation on all types of smartphone or PC platforms, a lightweight framework must be utilized. HTML5 and WebSocket provide good opportunities or technologies for lightweight implementation. Although both have been discussed in some papers, not any work in the literature is able to integrate these two technologies to handle multiple data types and process multimedia stream efficiently. In this study, the authors develop and implement a real-time synchronous Interactive Presentation System (IPS) which fulfills the above goal. The experimental result shows that the required bandwidth for video streaming in web applications which servicing 30 clients is around six times lower than that of native applications. CPU usage is five times lower. Memory size is ten times lower. Overall, this study provides a system framework for efficient online presentation of web applications.

Keywords

Multimedia, Streaming, WebSocket, HTML5, RTSP.

1. INTRODUCTION

In recent years, many studies have been conducted on the trend of using YouTube as a supplementary online learning material by teachers (Williamson, 2012). Traditional face-to-face lecture presentation is changing from text only to the inclusion of multimedia. This breaks down the learning barrier and provides a flexible learning platform. The use of multimedia courseware for integrating ESP (English for Specific Purposes) course depicts an excellent learning experience with the aid of multimedia (Tsai, 2012).

In such a multimedia rich environment, in order to play embedded video/audio of the presentation slide smoothly, attention must be given to the following two aspects of the operating platform. First, whether or not it is necessary to install additional components. The need to install additional components may result in expenditure costs, component compatibility, and even affect the execution performance, and so on. Second, whether or not cross-platform environment exists. The presentation must work efficiently on various platforms (smartphone, tablet, portable device, PC, NoteBook) or equipments (Windows, Android, iOS).

With regard to additional installations on the components, the more common practice nowadays is of two kinds: (1) transmission by the use of traditional desktop application; (2) transmission by embedding flash in a web page. If using the desktop application method, the content consumer must

first install the software to ensure proper function. Some software may require a sum of money, while others are free to use, for example, Windows Media Player, VLC media player ("VideoLan Organization," 2013), etc. As for plug-in flash player in a web page, the system will not work properly if not installed or if not supported by the installation. For example, iOS system does not support flash-related components, so multimedia information cannot be browsed in flash mode. As to the existence of cross-platform, if required to work successfully on multiple system/platform, the current common practice is to develop a variety of different systems/platforms applications. However, the cost of this approach is obviously quite high. At present, there are two major plug-ins, Adobe Flash and ActiveX, which are used in playing multimedia embedded in web browser. Nonetheless, there are security issues in these approaches. An essential disadvantage of both the Adobe Flash and the ActiveX applications is their inherent security defects. Since Flash player requires that both these software be installed, users can be traced by their behavior or unknowingly be a victim of cyber-attack (Watanabe, Cheng, Kansen, & Hisada, 2010). Users are concerned that the Internet security policies and procedures of ActiveX reflect "risk management" rather than "risk avoidance" (Bellovin et al., 2000). This means that ActiveX carries some risk issues, including security, etc.

The RTSP (Real-Time Streaming Protocol) system is a widely used application for audio or video streaming (Cortes & Luling, 1998). And RTSP server is always used in multimedia streaming system, which organizes many programs such as RTP/RTCP protocols, to process video stream. However, the RTSP server does not provide bidirectional communication even though it has HTTP/TCP capability. Audio/video stream is only broadcasted from server to client. The server side does not simultaneously get any text, audio, or video from the client.

In this study, an interactive presentation system (IPS) was developed based on a framework that can broadcast audio or video files in real-time. The framework utilizes the many advantages of HTML5 with WebSocket for streaming multimedia files. HTML5 with WebSocket server is much simpler than RTSP as a lightweight streaming server, and can be used on the web page directly in different platforms without extra embedded software. The system used in this study is similar to RTSP, capable of processing video, audio, and text multimedia stream. In addition, it can separate and synchronize sending and receiving of different data types (video, audio, and text according to personal preference). This feature is very important, especially when applied to distance education because lecturers present slides that contain text, picture, audio, and/or video through the Internet. Concurrently, the audience can type text

questions and download one or more online video, audio, or picture.

The IPS is an interactive system, which is controlled by the speaker and received by approved users. Also, it can be online broadcasting from server to clients. The clients consist of both speaker and users. The advantages of IPS are as follows.

- Support for many browsers with context-aware application. The web application created in this paper supports different browsers and versions, for example, Internet Explorer (v9,v10), Google Chrome (v15,v16,v17), and Mozilla Firefox (v9,v10,v11) (Zhu, Zhang, Zhu, & Zheng, 2012).
- One time writing usable for all platforms during system development. The developers do not need to duplicate implementation in different platforms.
- Efficient static video stream. The video tag of HTML5 is excellent for static video files compared to Flash, ActiveX, and DirectX (Zhu et al., 2012).
- Capability to run live video streaming. Compared with long polling, WebSocket has better communication behavior with respect to its latency by not measuring server queue and HTTP request from client (Pimentel & Nickerson, 2012). RTSP/RTP occupies a network connection, but WebSocket does not. When the number of clients is increased, more network bandwidth is required. Also, because WebSocket's characteristic is scalable, bandwidth consumption will not increase rapidly even though client device grows in number.

The primary contributions of this work are the development and implementation of the IPS architecture. Its goal is to develop a novel approach in combining the features of WebSocket, HTML5, and Visual Basic Application (VBA) into the framework of IPS, thus allowing accessibility in a cross-platform environment. The system reduces CPU and memory usage in comparison to traditional video stream and allows simultaneous use of audio, video, or text. It provides convenient and seamless operation for server and client to upload and download slides, audio and video files, and personal questions. In addition, a real-time feedback mechanism calculates the number of online audiences and presents it through statistics charts. This system can run on any location and device.

The rest of the paper is organized as follows: Section 2 presents related works on multimedia streaming, HTML5, WebSocket, and RTSP. The environment is presented in Section 3, and later, in Section 4, the proposed system is discussed. In Sections 5 and 6, the authors present the prototype, namely, presentation slides demonstrated in web application and the performance evaluation of the system, respectively. Eventually, conclusions and future works are given in Section 7.

2. RELATED WORKS

Even though Davids et al. talked about web conference for presentation and voice/video/text/chat modes, they did not discuss the issue of embedded video/audio in slides (Davids, Johnston, Singh, Sinnreich, & Wimmreuter, 2011). Also, flash players are necessary for their work. Zhu et al. designed and implemented a new media player based on the HTML5 with WebSocket for real-time video surveillance (Zhu et al., 2012). Although they used these

two technologies in real-time video surveillance issues, they did not investigate the play and performance of cross-platform live video streaming. Their system did also not support online speaking. The system in this thesis combines the properties of HTML5 and WebSocket, and applies them on cross-platform streaming real-time multimedia playback. At the same time, the performance is compared to that of traditional video.

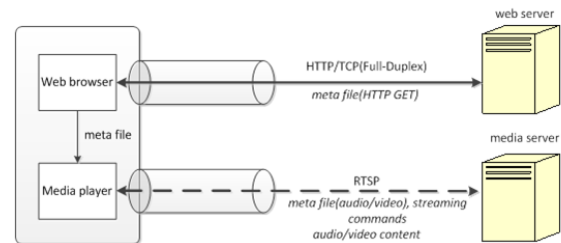


Fig 1: Video stream between HTTP/TCP and RTSP

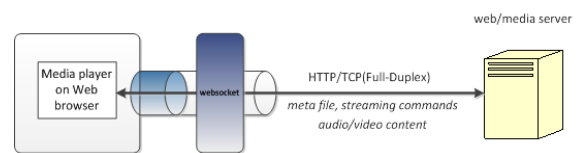


Fig 2: Video stream between HTTP/TCP and WebSocket

Figure 1 is the video stream structure when using traditional web and media server, where RTSP implies SETUP, PLAY, PAUSE, TEARDOWN, and media stream (Bang & Kim, 2012). This opens two ports for transmission: one is of the control command stream; the other is the content of the audio/video stream transfer. At the same time, HTML with RTSP belongs to a half-duplex network transmission mode. The following are the five characteristics of RTSP. First, it is an application-level protocol that goes through HTTP/TCP using syntax and operations. Then, RTSP can issue one-directional requests between servers and clients. Next, RTSP server uses SETUP, PLAY, PAUSE, TEARDOWN, and so on to maintain status. After that, RTSP messages can be copied out of system. Finally, in order to operate on various platforms, RTSP must be implemented for each platform (Zhu et al., 2012).

As the front-end of HTML with RTSP always contains an embedded media player, DirectX, Flash, and ActiveX are the three popular media players being used in real-time video surveillance (Zhu et al., 2012). These media players not only receive meta files from web servers but also communicate with media servers through RTSP. RTSP can create and control audio and video media streams continuously between media server and clients (Liu, 2000). As seen in Figure 1, HTTP/TCP has meta file including text and graphics between web server and web browser. However, a multimedia file must be processed by the embedded media player, such as Flash and ActiveX, on web browser (Zhu et al., 2012). The client can use HTTP to retrieve the presentation description file. RTSP's purpose is to offer services on video or audio streaming similar to what HTTP does in text and graphics. There are two main differences between RTSP and HTTP. First, HTTP is a stateless protocol and RTSP server must retain "session states" that ensure the correlation of RTSP requests while

streaming video/audio. Second, HTTP is an asymmetric protocol which involves the client's requests and the server's response. Conversely, the media server and the client can issue requests through RTSP. For instance, the server can publish a request that is able to set playing back parameters in a stream.

This new structure of video stream combines HTTP/TCP and WebSocket. It is full-duplex and is shown in Figure 2. The operations and services of the web/media server are combined through HTTP/TCP and WebSocket (Pimentel & Nickerson, 2012). Meta files, streaming commands, and audio/video content can be transmitted through HTTP/TCP. The media player on web browser in Figure 2 has the same meaning as in Figure 1. The difference lies in the absence of plug-in software on the website. Media player works directly with its own web browser support for HTML5 tag.

The HTTP/TCP with WebSocket mode of operation for multimedia data is described as follows. A conventional method called "polling" is used in sending data from the server to the client. Whenever the browser asks the web server for the video data, the browser must connect with the server. The server then sends the data to the browser and closes the connection. This method of transporting video data increases both the burden of traffic on the network and the delay time tremendously. As a solution, HTML5 uses WebSocket protocol for bidirectional communication with a server. After being connected, the HTTP protocol is upgraded to the WebSocket protocol. Then the server directly sends data to the browser without the need to re-connect each time. Furthermore, the ports used in WebSocket are 80 and 443, which enable video data to pass through the firewall without any problem. (Pimentel & Nickerson, 2012; Zhu et al., 2012)

This approach has two advantages. First, in the comparison of video delivery protocols (Ma, Bartos, Bhatia, & Nair, 2011), two ports are required for the video stream of the system in Figure 2, while three ports are needed for Figure 1. The three ports in Figure 1 denote that HTTP will occupy a communication port for use of the web browser, whereas the other two ports, namely, RTP/RTSP and RTCP, will be used for streaming multimedia content and sending control command. This means that the case in Figure 1 occupies more resources than the case in Figure 2 when streaming video. Second, the hardware environment in Figure 2 incorporates web/media server into a machine. This lessens the cost of hardware and software in the system. The framework of the IPS in this study belongs to the structure in Figure 2.

3. ENVIRONMENT

The environment of this study is in four parts: (1) the difference of FTP, HTTP, and TCP/IP; (2) HTML5; (3) WebSocket and socket; and (4) Visual Basic for Applications.

3.1 The difference of FTP, HTTP, and TCP/IP

It is necessary to understand the differences of the specific names FTP, HTTP, and TCP/IP. FTP is used for transferring files between computers through a network. HTTP is a protocol used for transferring webpages. TCP/IP is the Internet protocol suite which transferred data follows by network definition. Both FTP and HTTP pass through TCP/IP.

The TCP/IP model is based on a four-layer model. The TCP/IP protocol suite consists of the application, transport, and Internet layers of this model. As shown in Figure 3, each layer of the TCP/IP model is equal to one or more layers of the seven-layer Open Systems Interconnection (OSI) model followed by the International Standards Organization (ISO). The software stack of RTP/RTSP is allocated in the top three layers corresponded by TCP/IP model. For instance, RTSP, RTP, and RTCP are implemented in RTP/RTSP layer, which increases disk space usage, CPU occupancy rate, and memory occupancy rate when RTSP server is streaming video or audio. In contrast, the WebSocket layer needs a lower programming function to accomplish video streaming.

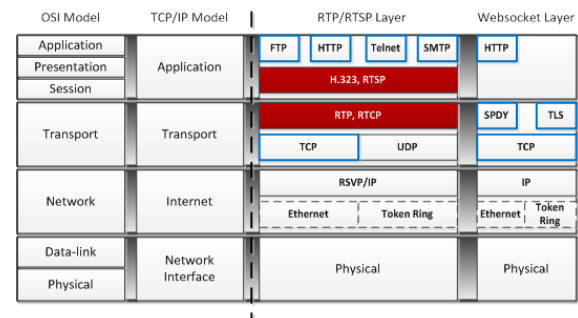


Fig 3: Comparison of network model with RTP/RTSP layer and WebSocket layer

Nowlan et al. presented the WebSocket layer (Nowlan, Tiwari, Iyengar, Aminy, & Fordy, 2012) as seen in the rightmost column of Figure 3. Also, the WebSockets over TCP notes for implementation of the WebSocket protocol go through the transport layer of TCP/IP model (Gezelter, 2011). Its two main functions are SPDY and TLS and are described as follows. First, SPDY's main purpose is the improvement of web pages performance. Second, Transport Layer Security (TLS) is the protocol used when visiting secure websites with URL beginning with https://.

3.2 HTML5

HTML5 (Hickson, 2011) is one of the fastest growing web-related technologies with the following advantages (Hoy, 2011):

- The new features/tags in HTML5 are native support for audio and video playback in the browser.
- In HTML5, there is no need to download and install both Adobe Flash and ActiveX.
- Another much-anticipated new feature of the HTML5 standard is the ability to build web-apps that behave more like local applications.
- The web page generated using HTML4 or earlier versions produces a screen offset on the mobile platform because of different monitor sizes. However, building a web page using HTML5 can display the same size monitor without any offset, making HTML5 suitable for cross-platform operations.
- HTML5 also provides support for client-side SQL databases and offline caching. This is particularly important for mobile users, since most have limited connectivity and bandwidth on their devices.

- Canvas element is another new feature of HTML5. Developers will be able to combine vector graphics, images, audio, and video within a predefined space on the page.
- HTML5's Geolocation features will be of particular interest to mobile developers and users.
- Another change being offered by HTML5 deals with refining the syntax of the language. Earlier versions of HTML combined presentational and semantic markup in one document. HTML5 deprecates presentational markup such as font color and size tags in favor of style sheets.
- HTML5 browsers can support local storage mechanism. The advantage of local storage is that it can be used to retain browsed data in the user's computer for subsequent use.

As Pascual et al. mentioned, the architecture of the web browser can be applied to any platform (iOS, Windows Phone, and PC, etc.) independently (Pascual Espada, Gonzalez Crespo, Sanjuan Martinez, Pelayo G-Bustelo, & Cueva Lovelle, 2012). Some elements in the architecture are implemented on the server side and have accessed web back-end APIs. They are partially dependent on the platform. The HTML5 for IPS can work for this situation of cross-platform. Thus, HTML5 not only works for this cross-platform situation, but also possesses better efficiency to play video than Adobe Flash with app in mobile devices (Jobs, 2010; Watanabe et al., 2010).

3.3 WebSocket and socket

WebSocket (Hickson, 2013) belongs to full-duplex communications, which is a bi-directional web technology, over a TCP connection. WebSocket protocol and API were standardized by the W3C. Both web browser and web server can allocate the design and implementation of WebSocket. Then, different clients or server applications can use it. WebSocket protocol not only provides more interactive opportunities for a browser and web site to transmit real-time multimedia content but also optimizes the bandwidth in order to release the client's unused connection. Accordingly, it can directly send live content to the browser without the client's request. Also, the messages are permitted to return as the connection is kept open. As long as web applications on the server support WebSocket, the WebSocket and web applications can be combined into a real-time server similar to RTSP with web applications.

WebSocket can initialize a connection through HTTP. It also arranges web sockets protocol. If the server supports to accept more TCP sockets, more communication streams can be created. Some framing and HTTP-compatible handshake for WebSocket belongs to a normal socket. The web server forwards the HTTP-compatible handshake because it allows WebSocket to connect to the same port on the running web server. When a connection is first set up, the web server can record this status without accessing the loop. The number of browser clients is not limited by WebSocket. So, WebSocket can consist of both a non-browser and browser client with a server at the same time.

However, Socket and WebSocket have the following differences. First, socket runs over TCP/IP, however, WebSocket belongs to HTTP over TCP. Second,

WebSocket does not connect to the server permanently. In contrast, socket must connect to the server permanently.

4. PROPOSED SYSTEM

The first sub-section describes the overall design concept of the IPS system, as well as the meanings of the symbols used. The second describes the component functions of the server side. The third sub-section describes the client side's audience function. The fourth describes the client side's lecturer function. The fifth describes the systems processing flow and methodology.

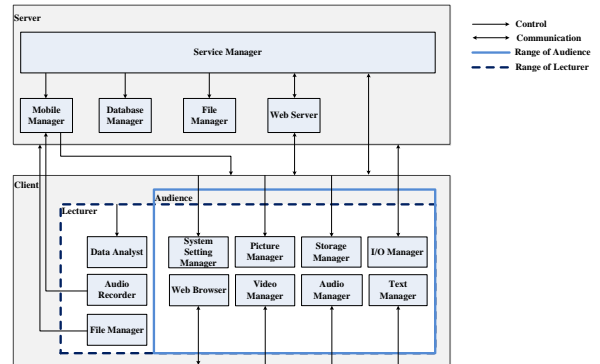


Fig 4: Architecture of IPS

4.1 System Design

The architecture of the system is based on client/server as shown in Figure 4. In Figure 4, there are two parts, server and client. If $A \rightarrow B$, where both A and B are components, it means that component A controls component B. $A \leftrightarrow B$ represents component A communicating with component B bi-directionally. In the portion for client, audience functions are surrounded by a boldfaced line. Also, a boldfaced dashed line encircles all functions for lecturer. Obviously, the seven managers and a web browser are the common functions for lecturer and audience. Data analyst, audio recorder, and file manager are particularly used for lecturer only.

This system is designed based on some considerations. First, clients have two identities, lecturer and audience. Each identity has its duty. The lecturer controls the presentation procedure, and the audience browses the presentation in real time and optionally downloads video, audio, picture, or slide notes. Also, they can upload their own question to the lecturer. Second, the server can manage broadcasting work, command collection and dispatch, and resource handling. Lastly, this system belongs to web applications used in client/server architecture because the advantages of code security for web server and system efficiency for web browser are applicable to this work. Also, it is suitable for multi-platforms with different clients.

4.2 Server for central management

As depicted in Figure 4, service manager controls the transaction of the file manager, database manager, web server, and mobile manager on the client side. The file manager locates physical heterogeneous files, which are matched through index stored in the database manager. The database manager manages data access with different tables, for example, picture, video, audio, text, and so on. The mobile manager stores recorded audio from the lecturer's mobile device. The web server is the system engine and communicates with the client and the service

manager directly. Then, broadcasting with simultaneous operation of the slide is sent to clients.

4.3 Client for audience

In Figure 4, client for audience (CA) contains seven components: system setting manager, picture manager, storage manager, I/O manager, video manager, audio manager, and text manager. First, the system setting manager provides users who are permitted to access the web page and authorized by specific usability of functions. Second, the picture manager manages pictures that are converted from the presentation slide after executing a disassemble program. The CA decides the necessity to download or not. Third, the storage manager includes disk and memory storage. Memory storage means that local storage is used for temporary data in client side. The partial web page data being used in common is stored in the memory of the client side, for instance, current transmitting status, login data, and the date and time of sending and receiving. Fourth, the I/O manager controls and displays monitor between client and server. Fifth, as in the picture manager, the video manager focuses on video files that are managed after the slides have been disassembled. Sixth, the audio manager has similar functions with the last point for audio files. Both the video and audio managers allow users to optionally download video and audio files. Finally, the text manager focuses on text management. It is available for users to upload personal questions to lecturer, to download notes of each slide from the lecturer, and to download notes written by the user during playing of the presentation slide. CA has a transaction GUI component, the web browser, which transacts data on the web page. The web browser provides a GUI for communication between the client and the server.

4.4 Client for lecturer

In addition to the eight components of the audience, there are three extra functions for the lecturer in Figure 4; they are the file manager, the audio recorder, and the data analyst. The file manager assists the lecturer in uploading his/her PowerPoint file to the web page directly. The function of the audio recorder is to record the lecturer's voice and then forward it in real time to the mobile manager through a mobile device. The feedback function is based on the data analyst including the status of user download/upload and accumulation of joining lecture.

The web browser for this system is not limited by the operating system. It is important for the lecturer web page to communicate with the server because the presentation slide is displayed on it. The function of the web browser is similar to CA that communicates between client and server through GUI. Because of HTML5 technology, which is capable of crossing platforms, the lecturer can control the contents on the monitor so that they will be the same as that shown on the monitor of the audience at the same time. Video and audio files embedded in the original PowerPoint file are allocated to the same sequential location after the PowerPoint slide is automatically disassembled by a kernel program. Two kinds of text boxes are available for the lecturer to manage. One text box collects audience questions, which are inputted during presentation. Another text box is for notes for the slides written by the lecturer in the original presentation. In order to provide a better understanding for the lecturer, IPS can summarize audience habits for reference. Statistical charts show the quantity of audience downloads for video, audio, slides,

slide notes, or personal notice, and also the amount of personal questions uploaded for each presentation.

5. PROTOTYPE: IPS IN WEB APPLICATION

This paper implements a framework that is mentioned in this section. This framework is available for different web applications using HTML5 and WebSocket for different purposes. This application features not only interactive presentation but also compatibility in different platforms. In order to have a prototype application, IPS is implemented as simple interaction and possesses cross-platform ability. Audiences can download and browse texts, pictures, videos, or listen to audios without operating system restriction. Also, the lecturer can control the web page synchronously and interactively.

This system supports the different typeface of languages. Here, traditional Chinese and English languages are shown Figure 5(a), Figure 5(b), and Figure 6.

Figure 5(a) shows the lecturer's window, whereas Figure 5(b) the audience's window. In Figure 5(b), when the audience clicks "Ask a question?", a small dialogue window will pop up in the center of the screen. After input, received questions from audiences are displayed on the lower right side of the lecturer window, as shown in Figure 5(a). These questions are also exhibited simultaneously on the lower right side of the audience window as shown in Figure 5(b). Furthermore, located on the interface of Figure 5(b) are other functions such as download slide, audio, video, note of lecturer, personal question, or personal writing note. Text parts, note of lecturer, personal question, and personal writing note are handled by the WebSocket function. At the lower left side of Figure 5(b) is a text window that allows the audience to write personal notes at any time.



Fig 5(a): Real-time displaying text

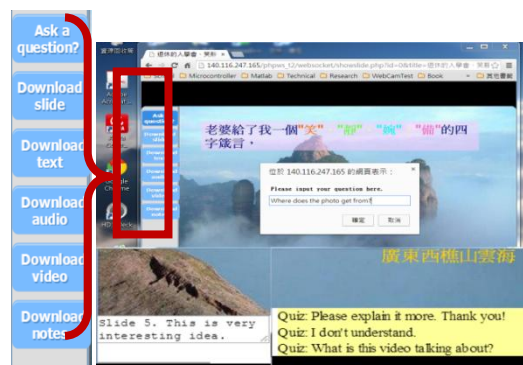


Fig 5(b): Real-time uploading text

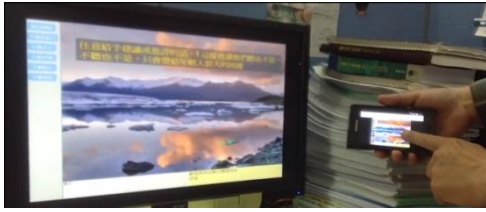


Fig 6: Synchronous cross-platform

Figure 6 represents a cross-platform situation where whatever is available for desktop PC is also offered synchronously to mobile.

6. PROTOTYPE EVALUATION

Before IPS performance evaluation, the compatible software must be made clear to the operation environment. Currently, many web browsers support HTML5. They are also capable of working with WebSocket API. These web browsers include version 9, 10, and 11 of Internet Explorer, version 15, 16, 17, and 18 of Google Chrome, and version 9, 10, and 11 of Mozilla Firefox.

Table 1. Hardware Configuration

Name	Description
CPU	4-core Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz
Memory	8GB
Hard Disk	500GB
OS	Window 7
Network Card	Gigabit Ethernet

In this section, the streaming video runs in a single machine for both web application (Web browser) and native application with different grouping client machines for each run. The client machines can be a smart phone, a tablet PC, or a desktop PC, as long as they can connect to the Internet through a web browser. The hardware configuration is shown in Table 1.

The video formats H.264, Ogg, or MP4, whose resolution is 640X360 in different web browser, are played in an unlimited loop. In order to make a consistent testbed, this paper uses MP4 to play on HTML5 in the client side. The tool for playing RTSP is VideoLAN Client (VLC) ("VideoLan Organization," 2013). This means that the web application used is HTML5 with WebSocket protocol, and native application is VLC including RTSP.

Many recording tools used in network performance are available. One study tested WebSocket usage for real-time applications (Chen & Xu, 2011). To implement and evaluate performance of WebSocket in a LAN Ethernet network, Wireshark software has been used in capturing and analyzing the size of IP packets traveling on the network (Chen & Xu, 2011). The bandwidth testbed uses Wireshark software. A 20-Mb testbed video file is used in an infinite loop during each test. This study conducts an experiment on three metrics: CPU usage, memory size, network bandwidth. The performance of CPU usage and consumption of memory size in the server are measured for RTSP and WebSocket protocol. Lastly, the transmission files can be used in various data types, for instance, video, audio, text, and so on. Obviously, this evaluation

experiment focuses on the network bandwidth of streaming video files because it costs more than streaming other files in the media server.

Some interesting experiments are discussed below. First, the task manager of Microsoft OS retrieves an average CPU value. In order to understand both video and audio cases, summary of the CPU's running percentage is shown in Figure 7. VLC(video) represents the transmission video file through VLC with RTSP. Also, the audio file through VLC with RTSP is shown as VLC(audio). Similarly, HTML5(video) and HTML5(audio) represent HTML5, where video and audio files are conveyed through WebSocket. The average value is taken from three maximum and minimum values during video or audio stream of the investigation. The formula is shown as (1).

$$\sum_{i=1}^n \frac{Max(CPU\%) + Min(CPU\%)}{2} \quad n = 1, 2, 3 \quad (1)$$

From Figure 7, when number N is equal to 1, HTML5(video) is almost two and VLC(video) is nearby ten. The difference between VLC(video) and HTML5(video) with video stream is at least five times in CPU percentage. The CPU percentage can be gradually increased based on the number of client. Also, HTML5(audio) is closer to two and VLC(audio) is approximately five. The audio stream of VLC is twice the amount of the audio stream of HTML5 in CPU percentage.

Second, this system shows a difference in memory size (GB), which is accounted for by the difference before and after video stream. Only video files, which are more costly than audio files due to memory size, are discussed. In Figure 8, the difference between VLC and HTML5 is ten times the average. This means that VLC uses a lot of memories because of RTSP for video stream in this framework.

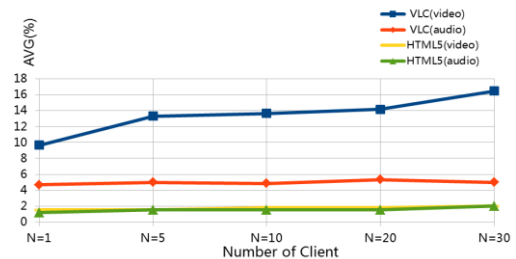


Fig 7: The usage of CPU

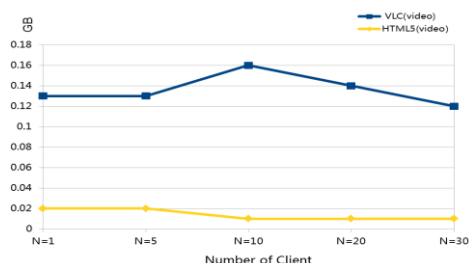


Fig 8: The consumption of memory size

Lastly, Wireshark (packet/s) tool keeps tabs on the bandwidth of the media server. It can retrieve a maximum value during video stream or audio stream. The experimental results are presented in Figure 9 that value of the VLC video stream is six times that of HTML5.

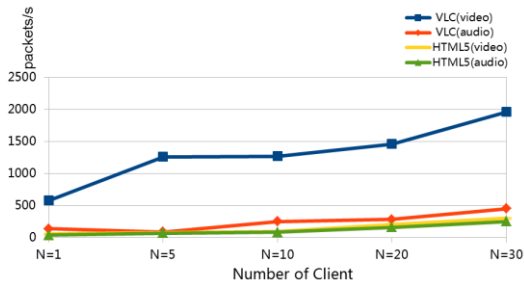


Fig 9: Bandwidth requirement

The bandwidth of each client is an average amount. In Figure 10, it shows the status of different groups that there is a ten times difference in the value of the video stream between HTML5 and VLC.

According to the above experiment, video files using HTML5 in this framework is extremely efficient, especially in a media server for multiple clients. Also, after the completion of the experiment, this work presents a better performing web application for the framework of multimedia streaming server.

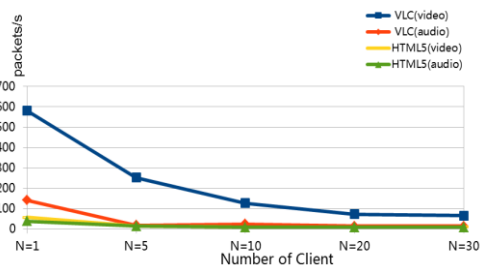


Fig 10: The average amount of bandwidth for each group clients

7. CONCLUSIONS

This web application for multimedia server has a better framework that provides high performance task. In the results, after being compared with native application, the system provides some important improvements, such as CPU usage, consumption of memory size, and the required amount of bandwidth.

First, comparing video stream with HTML5 and VLC, CPU usage by HTML5 with WebSocket is five times lower than VLC including RTSP. It means that a server can process many tasks at the same time. Second, the memory size requirement for HTML5 in video file is ten times lower than VLC. With this feature, a multimedia server has enough buffers to store transaction data. Lastly, from the experimental result, the video stream bandwidth of HTML5 is six times lower than VLC on average. Therefore, this system can exactly forecast the bandwidth followed by different requirements.

8. ACKNOWLEDGMENTS

Our thanks to experts were supported by NSC under the Grant NSC 100-2221-E-006-250-MY3.

9. REFERENCES

[1] Bang, J., & Kim, D. (2012). Efficient RTSP-based multiple buffering and packet transmission methods for delivering OMA PoC Box service. *Computer Networks*, 56(15), 3468-3478. doi: 10.1016/j.comnet.2012.07.006

[2] Bellovin, S. M., Cohen, C., Havrilla, J., Hernan, S., King, B., Lanza, J., . . . Lowenthal, R. A. (2000). Results of the Security in ActiveX Workshop. Pittsburgh: CERT Coordination Center, Carnegie Mellon University.

[3] Chen, B., & Xu, Z. (2011). A Framework for Browser-based Multiplayer Online Games using WebGL and WebSocket. Paper presented at the 2011 International Conference on Multimedia Technology (ICMT).

[4] Cortes, F., & Luling, R. (1998). A parallel continuous media server complying to the RTSP protocol. In D. Hutchison & R. Schafer (Eds.), *Multimedia Applications, Services and Techniques - Ecmast'98* (Vol. 1425, pp. 97-110).

[5] Davids, C., Johnston, A., Singh, K., Sinnreich, H., & Wimmreuter, W. (2011). SIP APIs for Voice and Video Communications on the Web. Paper presented at the Proceedings of the 5th International Conference on Principles, Systems and Applications of IP Telecommunications.

[6] Gezelter, R. (2011). The WebSocket Protocol – Past Travails Are To Be Avoided. from <http://www.rlgsc.com/blog/ruminations/websocket-re-discovered-travails.html>

[7] Hickson, I. (2011). HTML 5 W3C working draft. from <http://www.w3.org/TR/html5/>

[8] Hickson, I. (2013). The WebSocket API. from <http://dev.w3.org/html5/websockets/>

[9] Hoy, M. B. (2011). HTML5: A New Standard for the Web. *Medical Reference Services Quarterly*, 30(1), 50-55. doi: 10.1080/02763869.2011.540212

[10] Jobs, S. (2010, April, 2010). Thoughts on Flash. from <http://www.apple.com/hotnews/thoughts-on-flash/>

[11] Liu, C. (2000). Multimedia Over IP: RSVP, RTP, RTCP, RTSP. from http://www-kiv.zcu.cz/~ledvina/vyuka/PDS/PDS-tut/Multimedia/ip_multimedia.pdf

[12] Ma, K. J., Bartos, R., Bhatia, S., & Nair, R. (2011). Mobile Video Delivery with HTTP. *Ieee Communications Magazine*, 49(4), 166-175.

[13] Nowlan, M. F., Tiwari, N., Iyengar, J., Aminy, S. O., & Fordy, B. (2012). Fitting square pegs through round pipes: unordered delivery wire-compatible with TCP and TLS. Paper presented at the Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation.

[14] Pascual Espada, J., Gonzalez Crespo, R., Sanjuan Martinez, O., Pelayo G-Bustelo, B. C., & Cueva Lovelle, J. M. (2012). Extensible architecture for context-aware mobile web applications. *Expert Systems with Applications*, 39(10), 9686-9694. doi: 10.1016/j.eswa.2012.02.151

[15] Pimentel, V., & Nickerson, B. G. (2012). Communicating and Displaying Real-Time Data with WebSocket. *Internet Computing, IEEE*, 16(4), 45 - 53. doi: 10.1109/MIC.2012.64

[16] Tsai, S.-C. (2012). Integration of Multimedia Courseware into ESP Instruction for Technological Purposes in Higher Technical Education. *Educational Technology & Society*, 15(2), 50-61.

- [17] VideoLan Organization. (2013). from <http://www.videolan.org/index.html>
- [18] Watanabe, T., Cheng, Z., Kansan, M., & Hisada, M. (2010). A New Security Testing Method for Detecting Flash Vulnerabilities by Generating Test Patterns. 469-474. doi: 10.1109/NBiS.2010.28
- [19] Williamson, A. M. (2012). Teacher Preparation Trends: Using YouTube to assess and supplement online learning. eLearn, 2012.
- [20] Zhu, G., Zhang, F., Zhu, W., & Zheng, Y. (2012). HTML5 Based Media Player for Real-Time Video Surveillance. Paper presented at the 2012 5th International Congress on Image and Signal Processing.