WebSocket Proxy System for Mobile Devices

Hirotaka Nakajima*
*Graduate School of Media and Governance
Keio University
Kanagawa, Japan
hiro@awa.sfc.keio.ac.jp

Masao Isshiki†
†Department of Home Electronics, Faculty of Creative Engineering
Kanagawa Institute of Technology
Kanagawa, Japan
masao@he.kanagawa-it.ac.jp

Yoshiyasu Takefuji‡
‡Faculty of Environment and Information Studies
Keio University
Kanagawa, Japan
takefuji@sfc.keio.ac.jp

***Abstract*—HTTP is absolute essential part of the Internet. However, in the large delayed network such as mobile data network, the performance of HTTP is influenced by "TCP acknowledgement delay". In this paper, HTTP over WebSocket proxy system is proposed to defeat the delay. In proposal, HTTP is used to achieve end-to-end communication. Each HTTP request and response is encapsulated into WebSocket packet at the client and server. In this paper, proposed system is implemented on Android smartphone and evaluated delay difference characteristics.**

Keywords—WebSocket; HTTP; proxy; TCP ack;

I. INTRODUCTION

Web browsing is the one of most used functionalities of the smartphones and the importance of the Web is accelerated with these demands. Moreover, by the rapid growth of the HTML5 standardization process and use-case, Web is recognized as the application runtime platform.

Despite the huge studies and improvements, the 3G mobile data networks are slow, delayed and lossy network compare to the wired network such as broadband connections.

In parallel with the mobile data network, HTTP is widely used as transfer protocol for the Web-related resources not only HTML but also images and movies. Generally HTTP communication takes place over TCP/IP connections. Due to TCP's acknowledgement mechanism, throughput of the TCP connection is determined by the delay of the underlay protocol; hence, throughput of the HTTP connection is determined by the delay of IP. Recent investigation has reported that the average resources of the existing Web site are 43.91 and the number of HTTP GET requests is 44.56[1]. According to HTTP protocol, the number of the maximum HTTP connection is restricted to 2. Several studies have reported that many of current Web browsers ignore this simultaneous connection rule and create 6-8 simultaneous connections to each host. Nevertheless taking the maximum HTTP connection number of the current browser implementations, it takes long period of time to load whole resources of the Web pages with the mobile data network. Ample studies have demonstrated the method to reduce the Web page loading time by 1) reducing the number of resource 2) reducing the size of resource 3) avoiding the HTTP maximum connection limit with multiple hostnames[2]. Since all these methods require the modification of the Web pages, applications or servers, it is difficult to deploy these methods largely. In contrast, few studies have focused on the way to transport resources.

WebSocket[3] is transport protocol standardized at the IETF that enables a single TCP connection to use multiplex message on the client and the server. WebSocket is aimed at helping the web browser and web server to communicate the bidirectional multiple messages simultaneously with less effect of delay and packet-loss rate. We can apply the WebSocket for transport protocol of HTTP. A key concern for applying WebSocket for the transport protocol of HTTP is no payload semantics on WebSocket. Since WebSocket protocol only defines the protocol header, generally we can send/receive any data with WebSocket.

In this paper we focused on HTTP over WebSocket proxy method. This is because though HTTP is slow under delayed network, WebSocket is faster; HTTP proxy is widely used in the corporate or campus networks: proxy model does not require special attention at the server side. We can apply widely apply this method.

II. HTTP OVER WEBSOCKET PROXY SYSTEM

HTTP over WebSocket Proxy System is proposed as a system that can transfer multiplex HTTP sessions without the effect of underlying delay.

As shown in Section I, in HTTP communication environment, the large delay of network causes drastically through put degradation. WebSocket is one of protocols that can avoid the "TCP acknowledgement delay". However because WebSocket is developed to enable both Web browser and server to transfer bidirectional-multiplexing messages, WebSocket does not have semantics of payload. That is WebSocket acts as same as TCP stream, WebSocket does not care the content of payload. Necessarily the semantics of data between client and server are needed. In this paper, to avoid this problem, a HTTP over WebSocket proxy system is proposed.

The proposed HTTP over WebSocket proxy system works as follows. The proposed HTTP over WebSocket proxy system is client and server model. Every HTTP over TCP packet (over IP) is encapsulated into WebSocket over TCP (over IP) packet at a proxy that is running on client device. Encapsulated packets are transmitted between proxy client and proxy server. Since WebSocket does not have a number of maximum sessions within WebSocket session, proxy client can handle the HTTP request many as requested. As a result, the end user can transmit multiple HTTP request to the server without HTTP simultaneous connection limitation. Fig. 1 shows a sematic structure of the proposed HTTP over WebSocket proxy system.

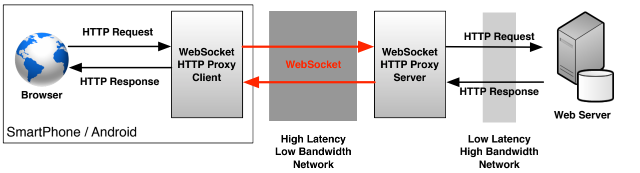

Fig. 1. A semantic structure of the proposed HTTP over WebSocket proxy system

First of all, a web browser transmits HTTP request header and payload to proxy client that is running on the same host of the web browser. When proxy client received the HTTP request header, proxy client encapsulates the received HTTP request into WebSocket. If the request has payload, proxy client transmit encapsulated HTTP header, then payload is transmitted continuously. When encapsulated HTTP request header and payload are received at proxy server, proxy server decapsulates the request header or payload. Then proxy server transmits the HTTP request data (both header and payload) to destination HTTP server. When proxy server receives HTTP result data that is returned from HTTP server, proxy server encapsulates HTTP result data (both header and payload) and transmits to proxy client. When encapsulated HTTP result data is received at proxy client, proxy client finally transmits the HTTP result data to the web browser.

III. EXPERIMENTAL RESULTS

To measure the HTTP throughput, we measured throughput to load 150 simple image files (file is 1517 bytes, format: Portable Network Graphics) from one web server.

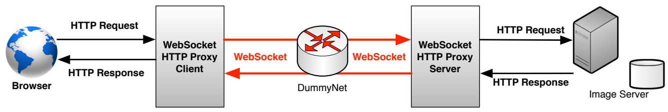

Fig. 2. A topology of the experimental network to measure the throughput

The effectiveness of the proposed system is evaluated with two experimental network topologies. In both cases, throughput to load 150 image files is measured with two kinds of HTTP resource transmission systems. First is the 6 simultaneous HTTP session, second is the single HTTP over WebSocket session.

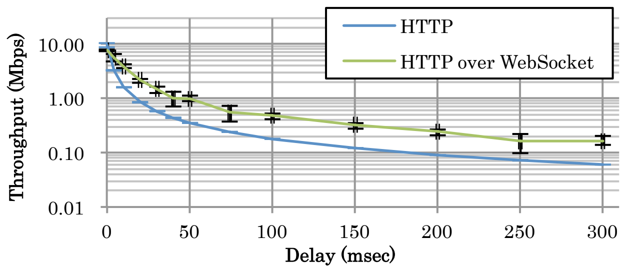

Fig. 3. Throughput versus delay, the value of delay is changed from 5 msec to 300 msec.

First, we evaluated with the topology that has dummynet[4] network simulator between proxy and client. Fig. 2 shows throught versus delay, value of the delay is changed from 5 msec to 300 msec. Throughput of HTTP over TCP decreases inversely with delay, and it reached 60kbps when the delay is 300 msec. Thus, HTTP is easy to be influenced from the delay among the network. On the other hand, in the proposed method, although decreases by the influence of the delay, throughput is not decreased easily less than HTTP. Throughput of HTTP over WebSocket saturates at 0 msec delay (actual delay is 0.049msec) because of overhead of encapsulation. As a result, the proposed system can achieve up to 169% faster than HTTP with 300 msec delay. (HTTP: 60kbps, HTTP over WebSocket: 162kbps)

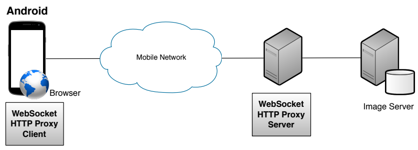

Fig. 4. A topology of the experimental network which has 3G mobile data network between proxy client and server.

Second, we evaluated the throughput with the topology which client is connected to 3G mobile data network shown on Fig. 4. Galaxy Nexus with 3G service by NTT DoCoMo is used for the evaluation. Fig. 5 shows throughput with 3G mobile data network. As shown, the proposed system achieves 137% faster than HTTP with real deployment environment at the average performance. (HTTP: 242kbps, HTTP over WebSocket: 1,358kbps)

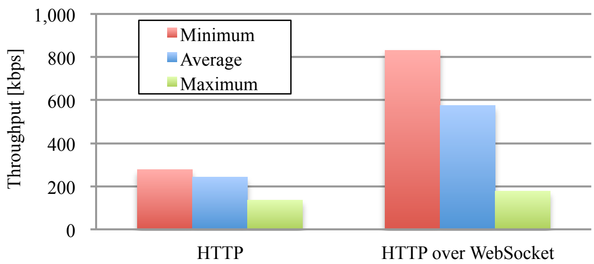

Fig. 5. Throughput with HTTP and HTTP over WebSocket via 3G mobile data network.

IV. RELATED WORK

Over the last decade, multiple research efforts have been conducted to speed up Web resource transportation. In this paper, we focused on Mobile network as a large-delayed and lossy network. Similarly, large-delayed and lossy network such as Satellites Link and Narrowband network have been focused.

M. Necker et al. proposed HTTP Proxy[5] that uses HTTP/1.1 pipelining[6] to avoid TCP acknowledgement delay. Similar to WebSocket, HTTP pipelining allows multiplexing streams within single HTTP sessions. HTTP pipelining features a sequential operation that causes head-of-line problem if the server takes time to process. By contrast, our proposed system allows non-sequential transfer that avoids head-of-line problem.

P. Davern et al. proposed HTTP Performance Enhancing Proxy (PEP)[7] that has similar architecture which has proxy server and client, to proposed system. In general with HTTP proxy, HTML document is parsed at the User-Agent, namely Web Browser and additional HTTP requests that fetch resources requested by parsed HTML document. By contrast on the HTTP PEP, proxy server parses HTML document and pre-fetches nested resources. This method would be effective with static web pages. Unfortunately most of today's web pages are dynamic, as it's impossible to pre-fetch additional resources that are required by Javascript or CSS. Our solution is more general as it enables any resources that are requested by User-Agent through proxy client.

V. CONCLUSION

HTTP is absolute essential part of the Internet. However, in the large delayed network environment such as mobile data network, the performance of HTTP and TCP, dependent protocol of HTTP, is influenced by delay among the network. In this paper, the HTTP over WebSocket proxy system is proposed. In proposed system, HTTP is used in order to achieve end-to-end communications. And WebSocket played a role to defeat TCP acknowledgement delay and simultaneous connection limit of HTTP. Each HTTP request and response is encapsulated into WebSocket packet at the proxy client, which located on the same host of the web browser, and decapsulated at the proxy server, which located on the edge of the mobile network. As a result, it was able to improve the throughput 169% faster in the emulated environment and 137% faster in the real mobile data network. Future work is to implement this system on the browser to avoid the overhead of the proxy and encapsulation.

REFERENCES

[1] S. Ramachandran. (2013). *Web metrics: Size and number of resources*. Available: https://developers.google.com/speed/articles/web-metrics

[2] S. Souders, "High-performance web sites," *Communications of the ACM,* vol. 51, pp. 36-41, 12/01/2008 2008.

[3] I. Fette and A. Melnikov, "The WebSocket Protocol," December, 2011 2011.

[4] L. Rizzo, "Dummynet: a simple approach to the evaluation of network protocols," *ACM SIGCOMM Computer Communication Review,* vol. 27, pp. 31-41, 01/01/1997 1997.

[5] M. Necker, U. o. S. Institute of Communication Networks and Computer Engineering, Pfaffenwaldring 47, D-70569 Germany, M. Scharf, and A. Weber, "Performance of TCP and HTTP Proxies in UMTS Networks," in *Wireless Conference 2005 - Next Generation Wireless and Mobile Communications and Services (European Wireless), 11th European*, 2005, pp. 1-7.

[6] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach*, et al.*, "RFC 2616-Hypertext Transfer Protocol," *The Internet Engineering Task Force,* 1999.

[7] P. Davern, U. C. C. Dept. of Comput. Sci., Cork, Ireland, N. Nashid, A. Zahran, and C. J. Sreenan, "HTTP Acceleration over High Latency Links," in *New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on*, 2011, pp. 1-5.