A SIP-BASED WEB SESSION MIGRATION SERVICE

Michael Adeyeye, Neco Ventura

Department of Electrical Engineering, University of Cape Town, Private Bag X3, Rondebosch 7701, South Africa

David Humphrey

School of Computer Studies, Seneca College, Toronto, Canada

Keywords: Web Session Migration, SIP, HTTP, Web Browser Extension.

Abstract: Web session handoff is one of the ways of improving the web browsing experience; other ways include the use of bookmarks and web history synchronization between two PCs. This paper discusses the implementation and evaluation of a SIP-based web session migration service. A graphical tool, which is called Data Flow Diagram, is used to describe how the session migration service works. This work is compared with other existing web session migration approaches. In addition, the large scale deployment and limitations of the service are also discussed. Although all web sessions could not be migrated, the session mobility service worked in a Peer-to-Peer environment and offered SIP functionalities within web browsers. That is, a web browser can now act as an adaptive User Agent Client to surf the Internet and set-up multimedia sessions like a SIP client. In summary, it is a novel approach to web session migration in which SIP is used to transfer session data. It also borrowed SIP Mobility mechanisms to introduce new service, namely content sharing and session handoff, to the web browsing experience.

1 INTRODUCTION

There already exist client-based, server-based and proxy-based architectural schemes for web session handoff (Canfora et al, 2005; Hsieh et al, 2006; Song H. 2002). A hybrid-based architectural scheme is proposed and implemented in this research work. Existing web browsers architecture is modified by installing an extension, and an optional SIP proxy can be used for client (web browser) registration and session data encryption. An extension enables one to pass information and add functionality to existing interfaces. The extension developed in this work is referred to as "TransferHTTP Extension". The implementation leverages the Mozilla Framework and solely works on Mozilla Firefox version 2.0. Mozilla Products developed by Mozilla Corporation are Free Open Source Software (FOSS), though the Corporation has flexible licenses, which interested party could choose from, to develop commercial products. Session Initiation Protocol (SIP) is a signalling protocol for establishing, modifying and terminating multimedia sessions. Although Hypertext Transfer Protocol (HTTP) is a stateless protocol (Fielding et al, 1999) SIP can be used as a stateful or stateless protocol (Rosenberg et al, 2002). HTTP is a stateless protocol because it makes multiple and unique HTTP Requests for every component, such as pictures, of a web page. HTTP was not developed with the intention of migrating its session between a web server and a web browser to another web browser.

In this research work, a SIP stack is integrated into a web browser to transfer web session data and provide multimedia services. The SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE) was exploited. Web session data are transferred between web browsers using the SIP MESSAGE method in a pager mode. In previous work, we proposed a modification to web browsers architecture by the use of an extension (Adeyeye and Ventura, 2007). The new service, namely Content Sharing and Session handoff, is borrowed from Third Party Call Control and Session handoff in SIP Mobility, respectively. Session handoff is defined as the ability to move existing web session between two web browsers, and content sharing is defined as the ability to simultaneously view the same web page on two web browsers at the same time. Content sharing requires transferring only a web page

Universal Resource Locator (URL); for example, Alice refers Bob to visit the same news website that she is browsing at the same time. On the other hand, Session handoff requires the transfer of a web page's URL with its session data; for example, moving an email session between two PCs.

The rest of this paper is arranged as follows: Section 2 discusses related SIP projects. Section 3 discusses related Industry work. Section 4 describes the design of the TransferHTTP Extension. Section 5 and 6 discuss the implementation and results, respectively. The results entail HTTP session mobility test, possible deployment and commercialization. Section 7 discusses evaluation of the extension with other existing web session handoff approaches. Lastly, Section 8 presents conclusions for this paper.

2 RELATED SIP PROJECTS

Projects that take advantage of SIP extensibility include the Akogrimo Project (The Akogrimo Project, 2008). It involves embedding web service data in a Session Description Protocol (SDP). In addition, in an expired IETF Internet draft (Wu and Schulzrinne, 2001), two approaches are identified for transferring URL between two web browsers. The first approach is by sending the URL via a SIP MESSAGE method. This research work is similar to this first approach. Furthermore, in this research work, the session data attributed to the URL can also be transferred. The second approach identified by Wu and Schulzrinne is by using SIP NOTIFY method. This approach is described as a way of achieving conference model of web browsing whereby a browser could be notified when a web page on another browser has changed. The shortcoming of the project is that it is discontinued neither are the approaches standardized. However, the approaches show different ways of achieving web share using SIP MESSAGE and SIP NOTIFY methods.

Another project that exploits the SIP extensibility and very similar to this research work was carried out by (Munkongpitakkun et al, 2007). In the project, a SIP stack is also integrated into a web browser, and a SIP MESSAGE method is used to transfer session data as well. Although it is very similar to this research work, the project, like other related academic work, only addresses session handoff. In terms of the signalling, two web browsers have to establish a call session using a SIP INVITE method before a session handoff can take place. This approach introduces unnecessary overheads in the signalling because it is not stated if users need to be involved in a multimedia session, such as voice call, before session handoff can take place. Although the work is also an attempt to solve the session handoff problem, the implementation details are limited and no software is available to confirm their findings.

3 RELATED INDUSTRY WORK

Google Browser Sync is an extension for the Mozilla Firefox, which makes it possible for users to synchronize their currently opened tabs and windows across different PCs, where the extension must have been installed (The Google Browser Sync, 2008). Mozilla Weave aims to create a repository where users' bookmarks, history, privacy settings and preferences can be stored and retrieved as they move across PCs (The Mozilla Weave, 2008). Extensible Messaging and Presence Protocol (XMPP), formerly known as Jabber Protocol, is also widely used to achieve interaction between User Agent Clients (UACs) (Saint-Andre, Ed. P., 2004). It is commonly used for instant messaging and developing online games between two or more UACs.

These industry solutions, such as Google Browser Sync, Mozilla Weave and XMPP, are distinctively different from this research work, though all of them tend to improve the web browsing experience. Although Google Browser Sync transfers session data between PCs used by the same person, it suffers a triangular delay. That is, session data are first pushed to the Google web server and then pulled by the extension at another end after user's identification. Similarly, Mozilla Weave also suffers the same triangular delay found in Google Browser Sync, though it gives users more control over their data and personal information than Google Browser Sync (The Google Browser Sync, 2008).

In addition, both Mozilla Weave and Google Browser Sync are based on conventional HTTP, which does not provide support for Peer-to-Peer (P2P) interaction, pure asynchronous events and multimedia sessions. XMPP, however, could be used in place of SIP to move session data, but it is not capable of providing multimedia services that SIP offers.

4   DESIGN OF THE TransferHTTP EXTENSION

A Data Flow Diagram (DFD) is a graphical tool in software engineering used for modeling information-processing systems, whole organizations, business planning and strategic planning (Yourdon, E., 1989). Figures 1 and 2 provide the highest DFD levels (Level 0) of two major functionalities in the extension, namely web session transfer and web session receipt. Figures 3 and 4 present Level 1 for the same functionalities.
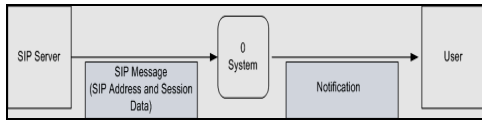

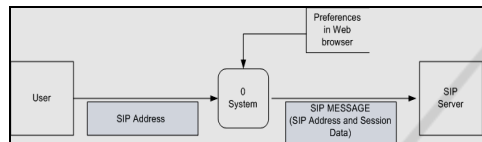Figure 1: Level 0 DFD for web session transfer.

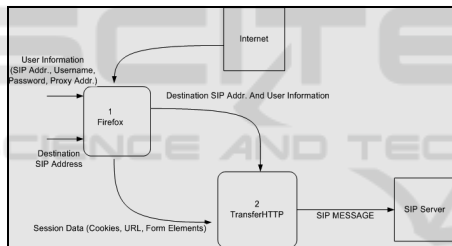Figure 2: Level 0 DFD for web session receipt.

Figure 3: Level 1 DFD for web session transfer.

In Figure 1, a user enters the destination SIP address, and the SIP client's credentials and proxy address are pulled from the Preferences of the browser into the system. Thereafter, the system pushes the session data with the destination SIP address to the SIP proxy server. In Figure 2, the web session receipt, the SIP proxy server forwards the received information to the destination system, and the system sends a notification to the user.
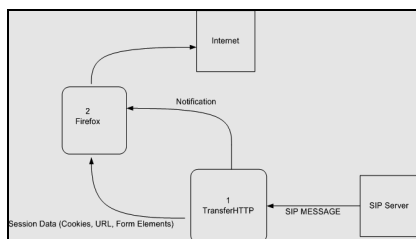

Figure 4: Level 1 DFD for web session receipt.

Figures 3 and 4 provide more detailed diagrams. In Figure 3, the two arrows indicate that the SIP client credentials and the destination SIP address are entered by the user. During a web session transfer, the login details, also referred to as the user information, are sent with the destination SIP address to the TransferHTTP extension.

Simultaneously, the session data with the URL are sent to the extension. An arrow from the Internet to the browser (Firefox) shows that the browser is currently accessing the Internet. TransferHTTP processes this information and sends it out as a SIP MESSAGE to the appropriate SIP proxy server. In Figure 4, the SIP MESSAGE is forwarded to TransferHTTP of the destination browser. First, TransferHTTP sends a notification to the browser. If the notification is accepted, the session data will be forwarded to the browser, and the resource will be pulled from the Internet.

5   IMPLEMENTATION

Until now, SIP had only been integrated into application servers. For example, IBM Websphere Application Server Version 6.1 was built on SIP Servlet 1.0 specification. The SIP Servlet 1.0 specification was standardized through Java Specification Request (JSR) 116 (Burckart E., 2008).

In this implementation, a small footprint SIP stack was used. Its file size was 2.2MB, while a typical web browser installer could be 9MB. After a successful integration of the SIP stack, the web browser was subjected to performance tests to determine if its memory consumption, upload speed and download speed changed when the SIP stack was running as a background service. The browser's performance was not hindered, and the results were discussed in a previous paper.
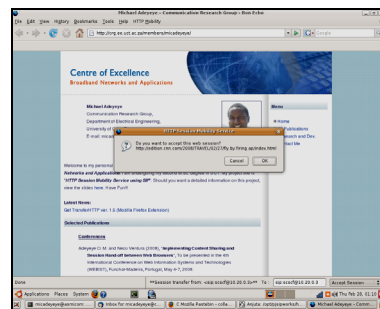

Figure 5: The TransferHTTP Web Browser Extension.

The SIP stack was wrapped as a shared library, and a new Cross Platform Component Object Model

(XPCOM) was written to interact with it. XPCOM makes it possible to write language-agnostic components thereby separating an implementation from its interface (Boswell et al, 2002). This approach provided a new layer of abstraction to the web browser in order to integrate the new protocol, SIP.

6 RESULTS AND DISCUSSION

Figure 5 shows the installed extension in a web browser. A new menu "HTTP Mobility" is introduced at the menu bar. A sub-menu "Preferences" can be found under the "HTTP Mobility" menu when clicked. In the Preferences, though not shown here in order to save space, a user sets among other things, his SIP address, SIP proxy address, username and password. Figure 5 shows a typical web session transfer notification at the status bar. Next to this notification message is a text field that accepts the destination UAC's SIP address. A drop-down menu with options - Register Client, Make a Call, Content Sharing, Transfer Session, Accept Session and De-register Client - exists next to the text field. This implementation also provides a voice interaction between two UACs (web browsers) when the option "Make a Call" is chosen after the UAC must have registered to a SIP proxy server. The following sub-sections discuss the HTTP session mobility test, possible deployment and commercialization of this new service.

6.1 HTTP Session Mobility Test

Figure 6 shows that the web session data vary among websites. This web browser extension was tested on some prominent websites. The intentions were to gather information on the size of data transferred during a session handoff and find out if session handoff could fail on some websites. The compositions of session data can be URL, cookies and hidden input elements. These session data are sent between web browsers in an XML format (Adeyeye and Ventura, 2008).

The packet size of the SIP MESSAGE body was measured when transferring each website's homepage before and after signing in. This packet size refers to the content-length of the SIP MESSAGE header, which is always zero when there is no data in the SIP MESSAGE body. The sampled websites already had some cookies in the cookies list of the web browser before signing in. Results obtained from analyzing the XML data showed that

Yahoo and Hotmail had almost the same amount of cookies. After signing in, more cookies were generated in Gmail and Hotmail than in Yahoo. In Figure 6, Yahoo session data size skyrocketed because it contained many input elements, especially button type, at the user's homepage. Hotmail had more cookies than Gmail and its user's homepage had very few input elements. Facebook, before and after signing in, had more cookies than its hidden input elements. Session handoff was also successfully tested on websites that used SSL to encrypted submitted data. Examples were the department's private web mail service (at https://crgmail.ee.uct.ac.za) and an eCommerce website (at https://www.firstinlandonline.net). Each website had a session data size less than 500B.

Session handoff on some websites, however, was partially successful or not successful. Although session data were successfully transferred, the exact web page with it current interaction, such as a web chat session, could not be continued on some websites. An example was Meebo, which is based on AJAX. AJAX enables websites to use XMLHttpRequest to asynchronously post and retrieve data via HTTP without changing a URL. Some Mash-ups, websites that present information gathered from other websites to users, also failed. Examples of these Mash-ups were http://www.alertnet.org, http://www.3dgeomaps.com and http://www.rockstarapps.com.

This failure can also be attributed to their ability to update web page's contents without changing its URL, during a HTTP POST or GET request. Other websites that achieved partially successful session handoff were those made up of FRAME/IFRAME HTML tags. In such cases, a column or row of the FRAME-based/IFRAME-based website redirected a user to a log-in page. The reason is that only cookies of the parent FRAME/IFRAME were transferred when session handoff was tested on these websites. As a result, a prompt or log-in request appeared in a column or row of the FRAME-based /IFRAME-based website.
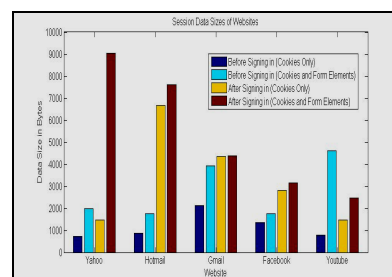

Figure 6: Session Data Size (Cookies and Hidden Input Elements) on prominent Websites.

An example was this URL: http://www.ngportal.com/Certdumps/parent_sites.php. The web page had a frame pointing to http://www.actualtests.com/default.asp?show=exams. Both websites, NGPORTAL and ACTUALTESTS, require logging in, though the log-in processes could be done at the parent website, NGPORTAL. During a session handoff, only cookies of NGPORTAL were transferred thereby resulting in a log-in request into ACTUALTESTS at the destination web browser.

In summary, web browsers are based on Same Origin Policy (SOP) in which cookies are selectively sent to a web server based on its domain or sub-domain (Wang et al, 2007). RFC 2965 (Kristol and Montulli, 2000) provides information on how cookies should be selectively sent to a web server and what modifications to cookies are not encouraged from end users. This implementation makes no change to cookies. It also deletes cookies from the source UAC during session handoff to prevent two or more UACs from having the same session data. In addition, the current web browser tab is also closed during a session handoff. This measure ensures all hidden input elements are cleared and will require logging in again to access the newly generated cookies and hidden input elements.

6.2 Deployment and Commercialization

This HTTP mobility service is an application layer solution that could migrate web sessions of websites, SSL-based websites inclusive. While this implementation successfully runs in a P2P environment, it is strongly recommended that it is used in a client-server environment. To provide data integrity and confidentiality, it is advised that the SIP server should implement TLS in order to provide HTTP Digest Authentication and encryption of data during UACs interaction. An alternative is the use of UACs that support Secure Multi-purpose Internet Mail Extension (S/MIME). S/MIME is a feature required in a SIP stack that might be used in the implementation, though it is not currently found in most of the available SIP stacks. It offers end-to-end data encryption, while TLS-supported proxy servers offer a hop-by-hop data encryption (Camarillo and Mugiel, 2006).

In Figure 6, the data sizes in bytes represent the number of characters and could be used as a yardstick for charging users. These data sizes account for the cookies and the hidden input elements. It was observed that the session data size varies among web pages in the same website. An example was Youtube. Although not indicated in Figure 6, its session data size, after signing in, at the homepage was 2kB, and at a randomly selected page, it was 15kB. The session data size in this case also depends on the number of HTML Form elements in the webpage. Considering the varying session data sizes, a charging function could be integrated into the SIP server, and a web interface could be provided for users to monitor their usage.

7 EVALUATION OF TransferHTTP EXTENSION

Table 1 shows the comparison of TransferHTTP with other existing web session migration approaches. These approaches are discussed here because each is based on one of these architectural schemes – client, proxy and server – and were carried out in the academic environment unlike those works discussed in sections 2 and 3, which are more of Industry products. Although (Canfora et al, 2005) and (Hsieh et al, 2006) are proxy-based architectural schemes, only (Hsieh et al, 2006) is used in this evaluation. (Hsieh et al, 2006) is chosen because it has more functionalities, such as support for optional client program, than (Canfora et al, 2005).

The works compared with TransferHTTP are Browser State Preservation and Migration (BSPM) (Song H., 2002) and Stateful Session Handoff for mobile WWW (Hsieh et al, 2006). BSPM is a client-based architectural scheme, while mobile WWW (that is, Stateful Session Handoff for mobile WWW) is a proxy-based architectural scheme. Our implementation is based on a hybrid-based architectural scheme.

Both BSMP and mobile WWW provide web session handoff between two web browsers. This research work (TransferHTTP) also provides web session handoff. Session handoff in the three works requires that all cookies are sent to the web browser, though these cookies are sent using different mechanisms. In addition to functionalities in TransferHTTP, it also offers content sharing, which entails one web browser referring another web browser to have access to the same web resource.

These works however have more differences than similarities. Regarding modifications made to web browsers architecture, BSPM is a client-based architectural scheme that requires modifying the architecture of a web browser. TransferHTTP, which is a hybrid-based architectural scheme, also requires

Table 1: Comparison of TransferHTTP with other existing web session migration approaches.

Item	Approaches		
	BSPM	**Mobile WWW**	**TransferHTTP**
Modification to user devices	Substantially modified browser	No but an optional small client program can be installed.	A small client program is required
Session registration/tracking	Supports user authentication	By accessing UAP's web pages or by the client program	Unnecessary but supports client authentication
Tracking session information	Unnecessary	By UAP; Unable to track unsubmitted form fields when client program is not installed.	Unnecessary
Handoff action	Source saves to BSPM repository and then Target retrieves from it.	Target retrieves from UAP; UAP retrieves unsubmitted form fields from source via small client program.	Source sends session data, unsubmitted form fields inclusive, to Target via an optional SIP proxy (or UAP).
History and Cookie handoff	History is directly loaded into browser. All cookies are sent to browser.	History is sent as a list of web pages. UAP sends necessary cookies to browser during browsing and tracking step.	History is unnecessary. All cookies are sent to browser.
Web pages handoff	Content is directly loaded into browser.	UAP redirects browser to open URLs. With small client program, UAP sends URLs to client program and then browser opens the URLs.	Browser interacts with client program and opens URL.
Summary	A client-based scheme. Needs a repository that stores session data. A client is modified. No user-client interaction. Introduced session handoff.	A proxy-based scheme. A proxy is required to perform the session transfer process. An optional client program can be used to support user-client interaction. Introduced session handoff.	A hybrid-based scheme. Introduced SIP into the client and can use an optional SIP proxy. Supports user-client interaction. Introduced voice call, session handoff and content sharing.

modifying the architecture of a web browser. On the contrary, mobile WWW is a proxy-based architectural scheme that does not require modifying the architecture of a web browser. It however supports an optional client program that modifies the architecture of the web browser when the client program is installed.

BSPM lacks user-client interaction, while mobile WWW provides user-client interaction when its optional client program is installed. TransferHTTP, like mobile WWW, also provides user-client interaction. User-client interaction offers a user the ability to continue a task, such as filling a form, rather than starting afresh at another end after a session handoff.

In terms of session registration/tracking, BSMP offers only user authentication, which is used to register a session. Mobile WWW however offers session tracking in addition to user authentication. Although TransferHTTP supports user or client authentication, it does not offer session tracking. History handoff is a feature found in BSPM and mobile WWW but not in TransferHTTP. TransferHTTP assumes a degree of privacy in its service. Since content sharing involves someone referring another person to the same web resource, it is unnecessary and considered to be an invasion of privacy to include the history of the source web browser. Session tracking is not provided in TransferHTTP, since a proxy is required to achieve it.

TransferHTTP could work in a P2P environment without a SIP proxy server. In addition, it introduces a new Protocol, SIP, into a web browser that makes it possible to set up voice call with the destination web browser. This offers a new way of collaboration in the web-browsing context. Lastly, unlike other approaches, it does not break the HTTP security rules. It was ensured during the implementation that no two web browsers would have the same session data, most notably cookies and hidden input elements.

8 CONCLUSIONS

This implementation (The TransferHTTP Extension, 2008) is referred to as a loosely-coupled approach because the SIP stack was not integrated into the core of the web browser; rather an abstraction was provided for the web browser and SIP stack to interact. Modifications made to the browser architecture were described in [Adeyeye and Ventura, 2007; Adeyeye and Ventura, 2008). This

implementation helps improve collaboration and mobility among the web users and encourages adaptive UACs. In this case, the web browser can also be used as a SIP client to make voice calls and be extended to perform other functionalities.

The packet size of the SIP stack was extended to 16kB in order to send session data at once. The data could be sent in chunks and merged at the destination UAC in order to propagate most SIP servers, should it be used over a large network. This is a custom implementation and could be deployed in a P2P environment. SIP is an extensible signaling protocol, and this implementation leverages its instant messaging capability to provide web session mobility. In addition, the implementation offers voice interaction alongside the web browser capabilities.

Regarding possible service commercialization, a flat rate could be charged periodically regardless of the varying session data sizes. Another approach could be to render it as a Value Added Service (VAS) to customers. Other areas that could be explored include implementing a policy control to block unwanted web session transfer request. Such restriction could be based on a domain name or a SIP address. While a session-based cookie expires in a short time of inactivity, a persistent cookie can provide access to a website over a long period of time. A session management mechanism could also be integrated so that a web session transfer request could be held for a long time without expiring when the destination SIP address can not be reached or a session-based cookie that expires in a short time is used by a web server.

In terms of specific contributions of this research work, while session handoff has been widely explored, content sharing is a new service in the web-browsing context that could encourage collaboration or community interaction between the Internet users. This implementation has provided a fast and efficient way of referring someone else to the same web page currently viewed by the referrer rather than the slow way of copying, pasting and sending the URL in a chat session or an email. In addition, web browsers now have unique SIP addresses to interact with one another like PCs, which have unique Media Access Control (MAC) or IP addresses.

REFERENCES

Canfora, G., Santo, G. Di, Venturi, G., Zimeo, E. and Zito, M.V. (2005) "Proxy-based Handoff of Web Sessions for User Mobility," Proceedings of the Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous '05).

Ming-Deng Hsieh, Tsan-Pin Wang, Ching-Sung Tsai and Chien-Chao Tseng (2006) "Stateful session handoff for mobile WWW," Information Sciences, Elsevier Science Press, volume 176, pp. 1241-1265.

Song, H. (2002) "Browser Session Preservation and Migration," In Poster Session of WWW 2002, Hawaii, USA, May 7-11, 2002, pp. 2.

Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and Berners-Lee, T. (1999) "Hypertext Transfer Protocol – HTTP/1.1," IETF RFC 2616, June 1999.

Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and Schooler, E. "SIP: Session Initiation Protocol," IETF RFC 3261, June 2002.

Adeyeye, M. and Ventura, N. (2007) "Extending Web Browsers Architectures to support HTTP Session Mobility," Proceedings of CoNEXT '07, New York, U.S.A, Dec. 10-13, 2007.

The Akogrimo Project (2008), http://www.mobilegrids.org, June 19, 2008.

Xiaotao, Wu and Schulzrinne, H. (2001) "Use SIP MESSAGE method for shared web browsing," http://www3.tools.ietf.org/id/draft-wu-sipping-webshare-00.txt, November 14, 2001.

Munkongpitakkun, W., Kamolphiwong, S. and Sae-Wong, S. (2007) "Enhanced Web Session Mobility based on SIP," Proceedings of the 4th International Conference on Mobile Technology, Applications and Systems (Mobility 2007), Singapore, September 10-12, 2007, pp. 346-350.

Google Browser Sync (2008), http://www.google.com/tools/firefox/browsersync/, June 10, 2008

A prototype of Mozilla Weave (2008), http://labs.mozilla.com/2007/12/introducing-weave/, June 10, 2008.

Saint-Andre, Ed. P. (2004) "Extensible Messaging and Presence Protcol (XMPP): Core," IETF RFC 3920, October 2004.

Yourdon, E. (1989) Modern Structure Analysis. Prentice Hall, 1989.

Burckart, E.(2008) "Session Initiation Protocol in WebSphere Application Server V6.1," [Online], Available: http://www.ibm.com/developerworks/websphere/techjournal/0606_burckart/0606_burckart.html [May 16, 2008].

Boswell, D., King, B., Oeschger, I., Collins, P. and Murphy, E. (2002) Creating Applications with Mozilla, O'Reilly Press, USA, First Edition, 2002, pp 1-8.

Adeyeye, M. and Ventura, N. (2008), "Performance and Possible Deployment of HTTP Session Mobility Service using SIP," Proceedings of the 11th South African Telecommunication Networks and

Applications Conference (SATNAC), Durban, September 7-10, 2008.

Wang Helen J., Fan X., Howell, J. and Jackson, C. (2007) "Protection and Communication Abstractions for Web Browsers in MashupOS," Proceedings of the SOSP' 07, Stevenson, Washington, USA, October 14-17, 2007.

Kristol, D., Montulli L. (2000) "HTTP State Management Mechanism," IETF RFC 2965, October 2000.

Camarillo G., Garcia-Martin, M. (2006) The 3G IP Multimedia Subsystem, Wiley Press, England, Second Edition, 2006, pp. 97, 213-229.

The TransferHTTP Extension (2008) http://transferhttp.mozdev.org, September 5, 2008.

Adeyeye, M., Ventura, N. (2008) "Implementing Content Sharing and Session handoff between Web Browsers," Proceedings of the 4th International Conference on Web Information Systems and Technologies, Funchal, Madeira, Portugal, May. 4-7, 2008.