IMPLEMENTING CONTENT SHARING AND SESSION HAND-OFF BETWEEN WEB BROWSERS

An Integration of SIP Stack into Mozilla Firefox Web Browser

Michael O. Adeyeye and Neco Ventura
Department of Electrical Engineering, University of Cape Town, Private Bag X3, Rondebosch 7701, South Africa

Keywords: Web Browser, Session Mobility, User Agent Client Architecture.

Abstract: A new idea referred to as HTTP (Hypertext Transfer Protocol) Mobility is introduced into web browsing. It entails transferring existing web session between Web Browsers or User Agents. This HTTP mobility will be achieved by extending present-day Web Browsers to support Session Initiation Protocol (SIP). Both HTTP and SIP are application layer protocols in the OSI (Open System Interconnection) Layer Model. SIP can be used to establish, modify and terminate multimedia sessions or calls. It has been chosen because it has clearly defined session mobility types namely Third-party Call Control and Session Hand-off. This paper identifies the modifications that will be made to the present-day Web Browsers Architectures and exhaustively explains the implementation of HTTP Mobility with the aid of SIP Stack integration. The two services that can be provided between two user agents during HTTP Mobility are Content Sharing and Session Hand-off.

1 INTRODUCTION

Though HTTP can be made stateful (Kristol and Montulli, 2000), session continuity between User Agents has been achieved by varying approaches resulting in the need for standardization. Session Initiation Protocol (SIP) is a signalling protocol that has been adopted for the IP Multimedia Subsystem (IMS) by the 3rd Generation Partnership Project (3GPP). It has been extended to provide instant messaging and presence services through SIP Instant Messaging and Presence Leveraging Extensions (SIMPLE) (Campbell et al, 2002). The intention of this research is to implement HTTP Session Mobility based on SIP Mobility (Shacham, 2007). Mozilla Firefox web browser will be used to implement the client side of this research because it is FOSS, has rich documentations on the internet and a large community of contributors.

2 THE PROPOSED HTTP SESSION MOBILITY SERVICE

In this research, HTTP session mobility using SIP will be achieved by extending the capabilities of a User Agent Client and implementing a SIP Application Server which provides HTTP session transfer or content sharing without violating HTTP/1.1 security. This scheme is referred to as a Hybrid-based Architectural Scheme whereby the client and the proxy or application server are modified and improvised respectively. This service can be adapted to the 3G IMS because it also uses SIP for signaling and communication with the application server. The HTTP session transfer and content sharing will be achieved using SIP Session Hand-off and Third-party Call Control respectively. Excerpt of HTTP messages can be sent in the form of SIP instant messages.

3 DISCUSSIONS

3.1 The User Agent Client Architecture

Fig. 1 shows the modifications made to a User Agent Client Architecture. The bold rectangular boxes and the SIP stack represent parts of the architecture that will be extended and a new feature that will be integrated respectively. The Extension found in HTML Manager represents the User Interface (UI). When buttons on the UI are clicked,

O. Adeyeye M. and Ventura N. (2008).
IMPLEMENTING CONTENT SHARING AND SESSION HAND-OFF BETWEEN WEB BROWSERS - An Integration of SIP Stack into Mozilla Firefox Web Browser.
In *Proceedings of the Fourth International Conference on Web Information Systems and Technologies*, pages 278-282
DOI: 10.5220/0001525302780282

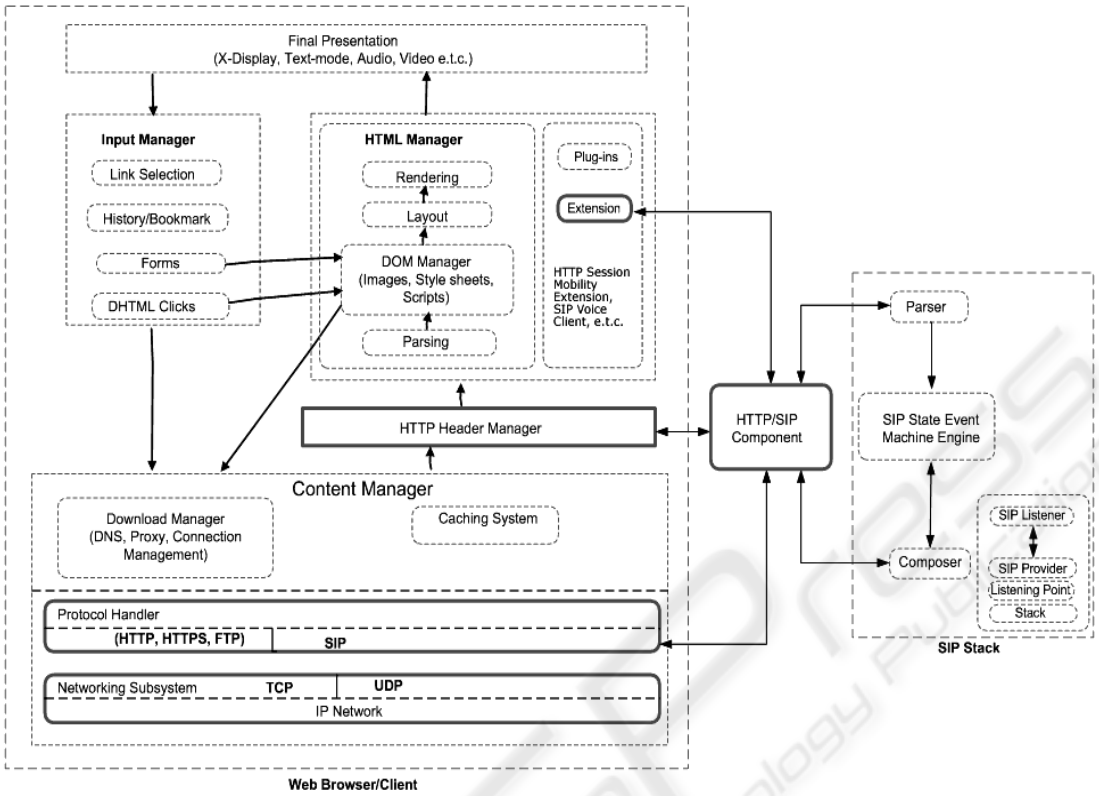

Figure 1: The User Agent Client Architecture.

they trigger events such as registering the client and sending excerpt of the HTTP request/response in form of SIP MESSAGE. It can be sent as a session mode Instant Messaging (IM) (Campbell et al, 2002) or a page mode IM. These processes take place inside the HTTP/SIP Component block. SIP transactions are handled by the SIP Stack and work seamlessly with the HTTP/SIP Component which helps in building the SIP messages. The HTTP Header Manager is required to interact with the HTTP/SIP Component by providing the HTTP Request/Response excerpts. The Protocol Handler implements new SIP protocol alongside with the existing protocols such as HTTP and HTTPS. In the same manner, the Networking Subsystem integrates User Datagram Protocol (UDP) which is required during the SIP transactions. The other blocks such as Input Manager remain unmodified.

3.2 The Mozilla Firefox Extension

Fig. 2 shows the UI of the extension which is currently under development. The two supported services are Content Sharing and Session Transfer. Content sharing refers to the process of sending same Universal Resource Locator (URL) to another User Agent. It does not require transferring session objects like cookies or hidden elements between User Agents. Session Transfer requires identifying which mechanism has been used to make HTTP stateful connection to a web server. It requires transferring session objects like cookies, URL and in some cases information in the entire HTTP Response. When Session Transfer is performed, the initiator loses stateful connection to the web server and he will be required to log in again should he want to continue.

Fig. 2 also shows the Preferences window where the User Agent can be used in an IMS context or with a SIP AS.

3.2.1 The Implementation and its Toolkits

These are customized User Agents in which additional functionalities are provided such as integrating SIP stack into them. It is required that the excerpts of the HTTP request/response header are sent in an XML format. There will be predefined tags that wrap up information such as URL (Universal Resource Locator), session ID or cookies

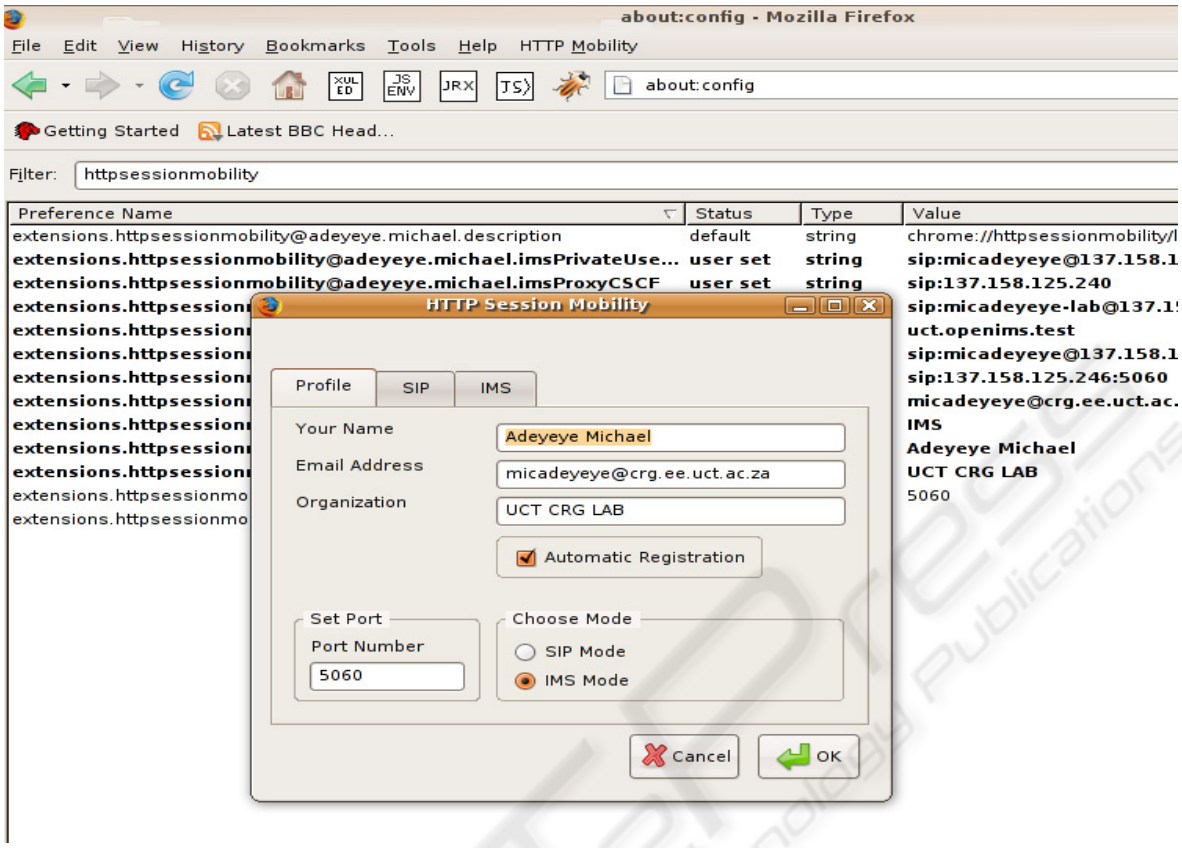

Figure 2: The Mozilla Firefox Extension.

information per request. The UAC is an extension to existing web browsers. It will query the XML, extract the relevant information and insert it into the web browser where appropriate.

The proof of concept will be based on Mozilla related technologies. The needed software is classified as Free and Open Source Software (FSOSS) and Mozilla Corporation™ has a large community of developers and contributors that can help in the implementation.

Some of the toolkits are Cross-Platform Component Object Model (XPCOM), Cross-Platform Interface Definition Language (XPIDL) and Cross-Platform Front End (XPFE). Mozilla web browser Firefox has over 600 components which were writing as XPCOM and seamlessly integrated into it. XPCOM makes it possible to write language-agnostic components thereby separating the implementation from the interface (David, 2002). Some of the components required in this implementation are the Preferences Manager (Components.classes [‘mozilla.org/preferences-service;1’]), the Authentication Component (Components.classes [‘@mozilla.org/loginManager;1’]) and the Cookies Component (Components.classes [‘Component @mozilla.org/cookieService;1’]). Fig. 2 shows the User Interface of the extension after installing on Mozilla web browser Firefox. There are numerous Application Program Interfaces (APIs) to help integrate SIP stack into an application. Examples are Osip, Exosip and reSIProcate. Recently, a Mozilla based SIP Client called Zap (Zap, 2007) and Mozilla thunderbird extension for telephony services called Cockatoo (Cockatoo, 2007) integrated a SIP stack. This project will leverage on these existing solutions in order to prove the concept.

3.2.2 The Implementation Plan

Fig. 3 shows the flowchart of how this solution will be implemented. Where a user has successfully registered with the SIP AS and makes a request, typically a session hand-off, the mechanism used in making HTTP stateful has to be detected. There are many ways of making HTTP stateful. These include the use of session IDs or cookies, use of HTML hidden fields and URL encoding (Hal, 2002). The UAC and the SIP AS will have the intelligence to

detect what mechanism is used to make HTTP stateful. This feature will help in determining what excerpt of the HTTP header request needs to be transferred. For example, cookies or session ID and its associated URL can be encapsulated and sent between two UACs. Since web applications can implement more than one of these mechanisms, it will be ideal to search for all relevant information such as cookies, HTML hidden fields and transfer all with the entire URL via SIP MESSAGE.

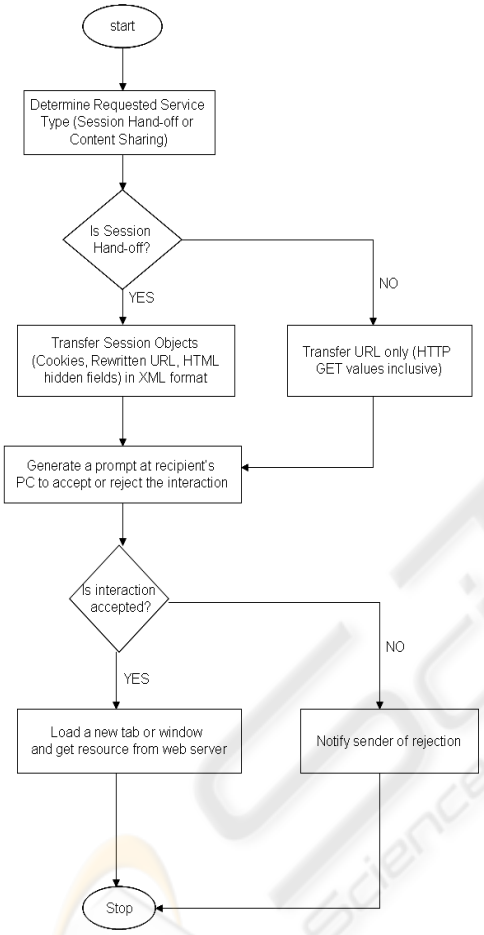

Figure 3: Flowchart of Content Sharing and Session Hand-off.

Fig. 4 shows the XML document format as message carrier. A native code in JavaScript or C++ will be written to extract necessary information from the message carrier and pass to the appropriate XPCOM in the browser. The built-in XML parser in Mozilla Firefox Web Browser can also be used to extract the necessary information from the message carrier. In addition to detecting what mechanism is used, another open problem is how session transfer can be achieved when accessing a secured website that implements security measures like SSL. Though SSL works between the Application Layer and the Transport Layer of the OSI Layer Model and helps encrypt data before transmission, there would be need to use a protocol such as TLS to encrypt data transferred between two UACs whenever it is discovered that SSL is used on a website whose content is about to be shared or handed-off. SIP supports most of the security features available in HTTP such as TLS, SMIME and SSL. Logically, it can be assumed that there will not be a problem since the extension works at the Application Layer where raw HTTP Header Request and Response are formed and transferred.

```
<xml version = "1.0">
<http-session>
<encodedURL>http://www.adinterax.com/request
.php</encodedURL>
<cookies>
<cookie_name>adxf</cookie_name>
<cookie_content>80047@1@799.98629@817.80072@
1@804.98706@7@817.8258@</cookie_value>
<cookie_domain>.adinterax.com</cookie_domain
>
<cookie_path>/</cookie_path>
<cookie_sendFor>Any type of
connection</cookie_sendFor>
<cookie_expires>30/12/15
17:01:16</cookie_expires>
</cookies>
<hiddenElement>
<element_name>categories</element_name>
<element_value>Microsoft</element_value>
<element_size>15</element_size>
<hiddenElement>
</http-session>
```

Figure 4: Message Carrier in XML format.

4 CONCLUSIONS

In this paper, only the client-side implementation strategies have been discussed. Also effort has been made to reduce assumptions that could make the paper vague and doubtful of its implementation. It is expected that many built-in interfaces/components of Mozilla Firefox Browser will be used. These include nsIPromptservice interface which will be used in creating prompt messages for users, nsIDOMDocument interface which will be used in tracing HTML Hidden fields and XBL (XML Binding Language) which will be used in automatically launching new tabs when an acceptance of Content Sharing or Session Hand-off request is made by clicking OK button of a prompt message.

How Content-Sharing and Session Hand-off are mapped to Third Party Call Control and Session Hand-off SIP mobility types respectively has been discussed in another paper owing to restriction in the number of pages of this paper. This is a research work in-progress which aims at developing an extension with a SIP stack for a web browser and a SIP Application Server in order to achieve Content Sharing and Session Hand-off. At present, effort is on integrating a SIP stack into the User Agent or Web Browser (Mozilla Firefox). Both of them are two autonomous applications and their integration requires an API that provides another abstraction of the SIP Stack so that the web browser can interact with it. The SIP Stack already provides a layer of abstraction by hiding the daunting task of composing SIP messages from scratch but making it possible to generate the SIP messages through simple procedural or object-oriented programming.

REFERENCES

D. Kristol, L. Montulli, 2000. “HTTP State Management Mechanism,” IETF RFC 2109.

B. Campbell, J. Rosenberg, H. Schulzrinne, C. Huitema, D. Gurle, 2002. “Session Initiation Protocol (SIP) Extension for Instant Messaging,” IETF RFC 3428.

R. Shacham, H. Schulzrinne, S. Thakolsri, W. Kellerer, 2007. “Session Initiation Protocol (SIP) Session Mobility,” Internet-Draft.

B. Campbell, R. Mahy, C. Jennings, 2007. “The Message Sessions Relay Protocol (MSRP),” Internet-Draft.

David Boswell, Brian King, Ian Oeschger, Pete Collins, Eric Murphy, 2002. Creating Applications with Mozilla, O’Reilly Press, USA, First Edition.

Mozilla based Sip Client “Zap”, 2007. http:// www.croczilla.com/zap.

Mozilla based Telephony Extension “Cockatoo”, 2007. http://cockatoo.mozdev.org.

Hal Berghel, 2002. “Hijacking the Web,” Communications of the ACM, Vol. 45 No. 4.