WebRTC enabled multimedia conferencing and collaboration solution

Adham Zeidan, Armin Lehmann, Ulrich Trick, Research Group for Telecommunication Networks,
University of Applied Sciences Frankfurt am Main, Germany
zeidan@e-technik.org

Abstract

WebRTC (Web Real-Time Communication) is an upcoming technology that enables web browsers with real-time communications capabilities such as audio, video and data communications using JavaScript APIs (Application Programming Interfaces). This article offers new possibilities in telecommunication presenting a solution for interoperation and migration between SIP-based systems and the emerging standards for WebRTC by an exemplary implementation and enhancement of an open source videoconferencing system.

1 Introduction

The WebRTC technology provides with its standards and capabilities a new vision and dimension of real-time communications services and applications through secure access over IP networks. It provides the ability of putting real-time communications capabilities such as audio, video and data communications into web browsers without the need of installing additional software or plug-ins. The standards of WebRTC are currently under joint development by the W3C (World Wide Web Consortium) and the IETF (Internet Engineering Task Force). The W3C is working on defining the APIs needed for the JavaScript web applications to interact with the RTC function, meanwhile the IETF is developing the protocols used by the browser RTC function to communicate to other communications endpoints. Ain addition security is an integral part within WebRTC. WebRTC 1.0 is specified in the W3C Working Draft (WD) webrtc-20120821 [1].

Several IEEE articles have been published recently describing the different challenges and possible alternatives to interoperate between standard SIP-based and WebRTC end systems [2;3;4]. This article demonstrates a solution for this convergence without the necessity of a media gateway. The presented solution furthermore enables the usage of SIP and WebRTC technologies in enterprise, social and educational networks and platforms.

The paper is structured as follows: The next chapter presents the general system architecture and protocols. Chapter 3 illustrates a detailed architecture and implementation of the suggested solution. In chapter 4 further applications and services are presented to show up the new possibilities for telecommunication regarding the convergence of WebRTC and SIP. Finally Chapter 5 is concluding the article and presents future prospects.

2 System architecture and protocols

In this chapter the scientific and theoretical background related to this research work is shortly described and explained.

2.1 Videoconferencing architecture

A videoconferencing system should consist at least of two components: focus and mixer (see Figure 1) [5]. The focus is a SIP user agent that is addressed by a conference URI, which identifies a conference. It represents a logical role that is responsible for controlling mixer and therefore the participant’s media streams.

The focus uses the conference media policy to determine the proper configuration of the mixer. Focus also provides a logical function, in which it acts as a notifier: accepting subscriptions to the conference state and notifying subscribers about changes to that state.

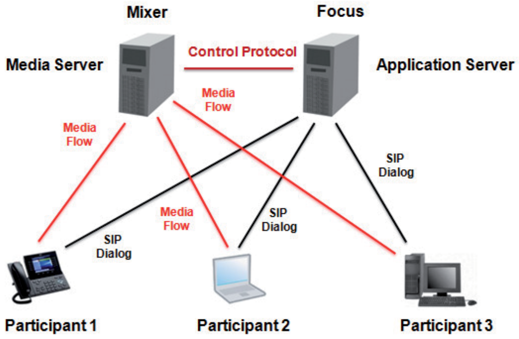

Figure 1 General overview of a conferencing system

On the other side, the mixer represents a system that receives RTP packets from one or more sources, and then changes the data format on demand, combines the packets in some type-specific manner and finally redistributes a new RTP packet to participants. The mixer makes timing adjustments among the streams and generates its own

ISBN 978-3-8007-3602-7

timing for the combined stream. This functionality is important to avoid time delay, since the timing among multiple participants will not be synchronized. Therefore all data packets originating from a mixer will be identified as having the mixer as their synchronization source [5; 6].

A conference can be represented by a unique URI that identifies the focus. Requests to the conference URI are routed to the focus for that specific conference. To join the conference, users usually send a SIP INVITE message to the conference URI.

There are several videoconferencing models. Figure 1 illustrates one of the most commonly used models. In this model, two servers are involved. One of these servers is the Application Server, which is responsible for managing the SIP signalling, the membership of participants and the media policies. The Application Server in this model represents the focus seen by all participants in a conference. However, it does not provide any media support. Therefore a second server referred as Media Server or Mixing Server is required to perform the necessary media mixing functions. To connect the media streams of each user to the mixer, a media control is used by the focus in the Application Server to communicate with the Media Server [5].

2.2 SIP (Session Initiation Protocol)

The Session Initiation Protocol (SIP) is a session-layer signalling protocol used for initiation and control of Multimedia over IP sessions, where the term “IP sessions” represents connection oriented communication between participants in IP-based networks. The session may be VoIP (Voice over IP), video telephony, video conference, multimedia distribution and IMS (IP Multimedia Subsystem) applications [7]. The SIP protocol was standardized by the IETF and is mainly specified in RFC 3261 [8].

2.3 WS (WebSocket)

WebRTC does not provide signalling which is essential to establish a communication session; therefore the WebSocket technology is used.

The WebSocket protocol enables message exchange between clients and servers on top of a persistent TCP connection. It enables bidirectional communication between clients and servers in web-based applications. The protocol consists of two parts: a handshake followed by data transfer, layered over TCP and TLS (Transport Layer Security). The WebSocket protocol is specified in RFC 6455 [10].

The WebSocket protocol acts as a transport protocol for SIP. The term “SIP over WebSockets” defines and specifies SIP as a WebSocket sub-protocol to enable the usage of SIP in web-oriented deployments and applications. The WebSocket SIP sub-protocol is used to carry SIP requests and responses through a WebSocket connection, where modern web browsers include a WebSocket client stack complying with the WebSocket API as specified by the W3C [11].

3 Detailed system functionality and architecture

The implemented and developed solution with all its elements with respect to the architecture is illustrated in Figure 2.

The MCU (Multipoint Control Unit) videoconferencing system was developed and published by Medooze [12]. The open source version of the videoconferencing system provides several features and advantages, among others the easy integration with any SIP infrastructure, allowing the implementation of several SIP Call Servers like Kamailio or PBX (Private Branch Exchange) like Asterisk.

The solution supports custom layouts and continuous presence allowing the view of all participants on screen simultaneously. It provides an administration web interface for operating and managing conferences. In addition to that, the open source solution supports a various number of media codecs such as Opus, Speex, G.711 and G.722 audio codecs as well as H.263, H.264 and Google VP8 video codecs for HD (High Definition) video resolution.

3.1 System architecture and functions

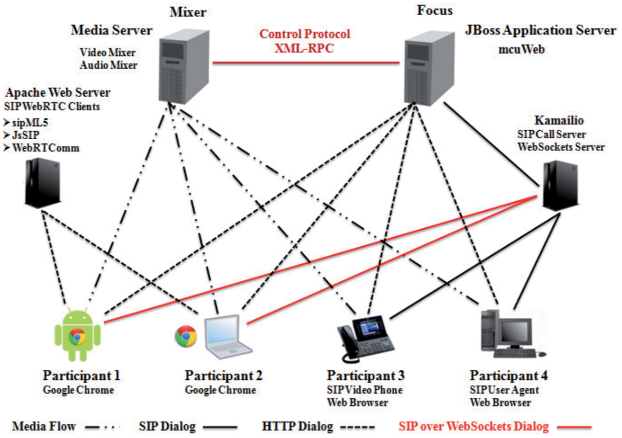

Figure 2 General overview of the implemented and developed solution

The open source videoconferencing system from Medooze consists of three important components (see Figure 2):

- Media Server, which represents an open source component that provides all media handling functionalities such as audio, video and text encoding and decoding, media mixing, web broadcasting and finally, recording and playback of files. The Media Server can be divided into the following components: open source encoder libraries, video and audio mixers, FFmpeg and an XML-RPC (Extensible Markup Language- Remote Procedure Call) server.
- mcuWeb application that represents an open source Java application based on JavaServer Pages and HTTP Servlets. This component is responsible for commanding the Media Server through the XML-RPC interface. It provides an administration web interface for managing and operating the service as well.

- SIP Application Server (Focus), which is responsible for the SIP signalling and running the mcuWeb application, where the open source version of the multi-videoconferencing system is using JBoss Application Server with SIP Servlets.

3.2 System implementation

Figure 3 illustrates the architecture overview of the developed and implemented videoconferencing system. According to Figure 3, the SIP Call Server Kamailio, which represents a SIP Proxy Server, SIP Registrar and Location Server, was implemented to the system to enable the registration and authentication of legacy SIP User Agents and for routing SIP messages (requests and responses) between these UAs and the Application Server.

As soon as Kamailio receives a SIP INVITE message sent from a SIP User Agent to join a videoconference, it forwards the request to the SIP Application Server. The SIP signalling is handled by the mcuWeb application. Finally, the mcuWeb commands the Media Server via the XML-RPC interface to establish the RTP media session with the SIP UA.

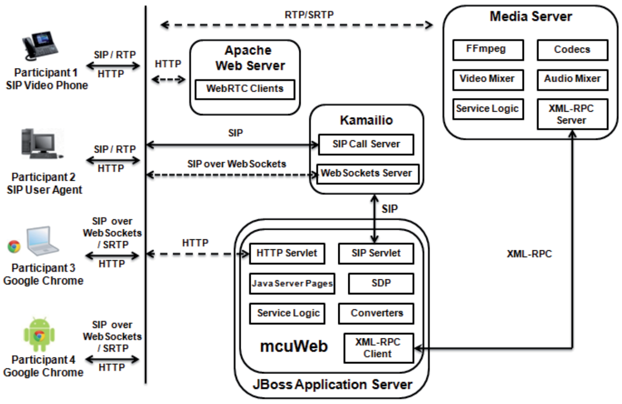

Figure 3 Architecture overview

Moreover, Kamailio was configured to support WebSockets by adding the new developed WebSocket module. This configuration allows Kamailio to act as WebSocket server that is capable of sending and receiving SIP over WebSockets messages. Finally, several open source SIP WebRTC clients were implemented to the system. All of these solutions represent SIP WebSocket clients relevant to RFC 6455 [10] and to draft-ietf-sipcore-sip-websocket [11]. The implemented WebRTC clients are written in JavaScript and provided by a locally implemented Apache web server (see Figure 3).

The WebSocket signalling procedure is illustrated in Figure 4. First, the participant has to download the WebRTC client from the Web server using Google Chrome Web browser. To register the WebRTC client on the Kamailio SIP Call Server, a TCP connection has to be first established between Kamailio and the WebRTC client followed by an opening WebSocket handshake to switch protocols from HTTP to WebSocket, where Kamailio and the WebRTC client define and specify SIP as a WebSocket sub-protocol. From now on and during this session, all SIP messages will be transferred over the WebSocket protocol. The main function of the WebSockets server (Kamailio) is represented in this scenario by translating SIP over WebSockets messages coming from WebRTC clients to SIP messages, and then forwarding them to the Application Server and vice versa.

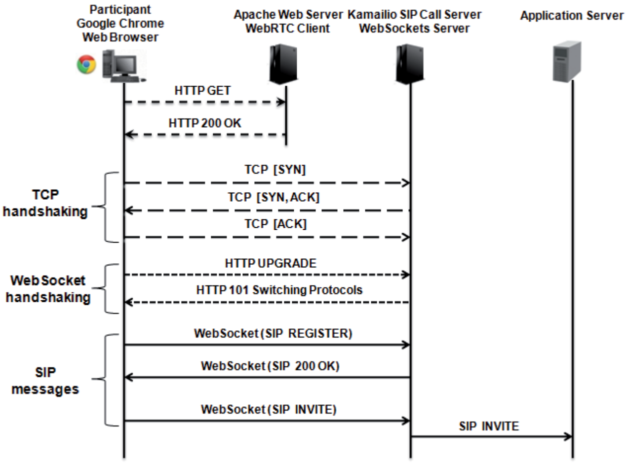

Figure 4 Signalling of SIP WebRTC clients

3.3 Development of additional services

The previous sections presented the architecture and implementation of the system providing videoconferences with a HD resolution. Furthermore, the open source solution was extended and modified to embed documents. This development enables participants to interact and deal with documents within a live videoconferencing session. The slide presentations development can be divided into the following components:

- File upload Servlet: This component is used to upload the document to the videoconferencing server
- Converters: Four converters that can be used to convert PowerPoint, Word, Excel, PDF documents and different types of images to PNG images
- XML-RPC methods: The XML-RPC interface was extended to enable the transfer of new parameters needed to add a document to a videoconferencing session and to change the slide number
- FFmpeg decoder and converter: This component is used to decode the PNG images and convert the RGB (Red Green Blue) color space to the YUV (Luminance Hue Chrominance) representation used in live video streaming applications
- File download Servlet: This component is used to download the document from the videoconferencing server
- HTML forms: Several HTML submit buttons and select elements that were programmed in the conference JavaServer Page to allow the interconnection between participants and the videoconferencing system to upload a document, to change the number of the streamed slide, to magnify or shrink a portion of a live streamed document, to download or delete a document from the videoconferencing server
- Zooming module: This component allows participants to magnify or shrink a portion of a streamed slide

- Service logic: The service logic of the mcuWeb application and the media server was extended to enable the developed components to work together and cooperate to provide the slide presentation service

A file upload request represents an HTTP request submitted using the POST method (see Figure 5). The upload request will be forwarded from the HTTP Servlet to the file upload Servlet, which can parse that request. As soon as the document has been completely uploaded to the server, the conference manager, which is in charge of handling the service logic, will be informed. In response, the conference manager will first examine the format of the uploaded document and according to that it orders the corresponding converter to convert the document to PNG images.

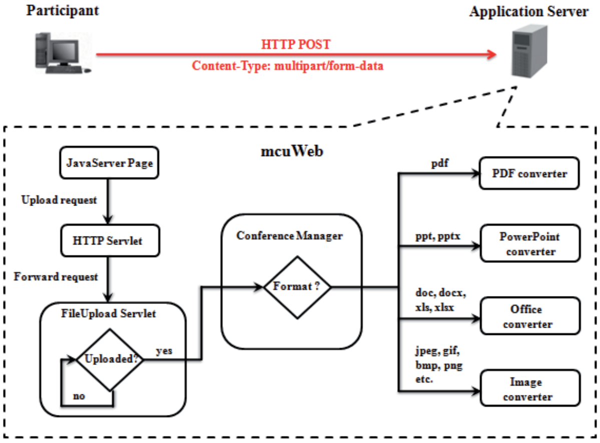

Figure 5 Uploading and converting documents

Moreover, the conference manager invokes all functions and methods responsible for creating all necessary parameters that are required to add the document to a videoconferencing session. These parameters will then be forwarded to the XML-RPC client, which will build the corresponding requests and send them to the media server through the XML-RPC interface. The XML-RPC server on the media server will extract the values of these parameters and forward them to the processing logic.

Finally, the processing logic will then order the Software FFmpeg to load the corresponding PNG image, decode it, convert it, stream it and send it to the video mixer to get mixed with other video streams.

Another possibility to change the number of the streamed slide can be accomplished using DTMF (Dual Tone Multi Frequency) signalling. Therefore, the SIP Servlet of the solution was extended to allow the transfer of DTMF signals over SIP INFO messages. When the Application Server receives a SIP INFO message, the SIP Servlet, which is in charge of handling SIP requests and responses, sends the value of the DTMF signal to the service logic. The service logic will then determine which slide has to be streamed and forwards this value to the media server through the XML-RPC interface. As a reaction, the processing logic on the media server will order the FFmpeg software to load and stream the corresponding PNG image (see Figure 6).

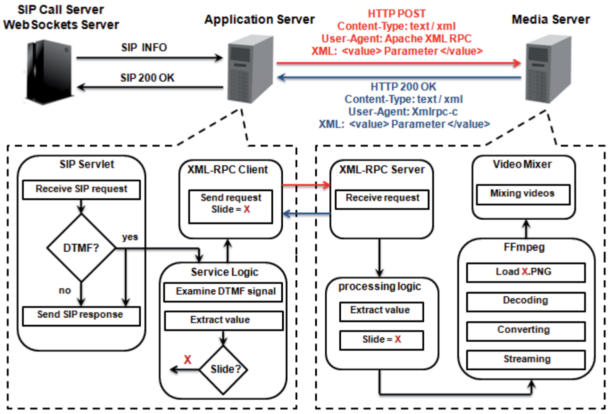

Figure 6 DTMF signalling over SIP INFO messages

3.4 Security

In the world of telecommunications, security plays a fundamental role that has to be taken into account and considerations. To protect the system from security threats, the signalling part of the service is encrypted using the TLS protocol. Therefore, the WebSocket Secure protocol (WSS) is used by WebRTC clients to transfer SIP messages. In addition to that, the media flow between the media server and SIP User Agents or SIP WebRTC clients is done using the Secure Real-Time Transport Protocol (SRTP), where the SDP Security Descriptions for Media Streams (SDES) is used for SRTP key negotiation and for association management. Finally the web management interface can be only accessed using the Hypertext Transfer Protocol Secure (HTTPS).

4 Applications and services

The developed and implemented solution provides users a wide range of applications and services. The following is an explanation and description.

4.1 Multiple videoconferencing sessions

The implemented system can handle multiple videoconferencing sessions. It supports up to 16 participants per videoconference. Video profiles allow users to set the resolution of the video, the bit rate and the sample rate. Ad-hoc templates allow the user to create videoconferences automatically with predefined parameters. When the first participant dials in and matches the ID parameter from a template, a videoconference is automatically created with the preconfigured parameters and the participant is joined to it. Additionally, the solution provides the ability of recording videoconferences. As soon as a videoconference is deleted, its contents (video and audio) will be saved in the format FLV (Flash Video).

4.2 VAD (Voice Activity Detection)

The solution supports the VAD technology, which is used to detect the presence or absence of audio. According to this implementation, the active speaker among participants will be shown in the defined video's position.

4.3 Live Web broadcasting

The media server supports live Web broadcasting as well as flash streaming using the Real-Time Messaging Protocol (RTMP). To enable the broadcasting functionality, a live streaming channel has to be created first on the Application Server using the mcuWeb application. Furthermore, a media live encoder is required to encode audio and video in real time to the media server. In this project the Adobe Flash Media Live Encoder 3.2 is used. Based on this seminars could be provided.

4.4 WebRTC support

The MCU has all functionalities required to support the WebRTC technology. It supports SRTP, ICE (Interactive Connectivity Establishment) and STUN (Session Traversal Utilities for NAT), AVPF (Audio-Visual Profile with Feedback) with multiplexing and feedback. It supports the necessary codecs as well, such as H.264 and VP8.

To make proper use of the WebRTC technology, the SIP Call Server Kamailio was implemented and configured to act as a WebSockets server. According to this implementation, Kamailio is able to receive SIP over WebSockets messages and then translating them into SIP messages. Moreover, several SIP WebRTC clients were installed and configured on a locally implemented Apache web server.

4.5 Live documents and slide presentations

The slide presentation development allows users to choose a document located on their PCs, upload it to the videoconferencing server and then add it to a live videoconferencing session, where the developed solution supports PDF files, Microsoft PowerPoint, Word and Excel documents as well as OpenOffice documents and several types of images, among others png, bitmap, jpeg, gif etc. In addition to that, it provides the ability to change the number of the streamed slide, to download the document from the server or to delete it.

Moreover, the solution was extended to enable the transfer of DTMF signals over SIP INFO messages, which allows SIP User Agents and SIP WebRTC clients that support this functionality to change the number of the streamed slide or to magnify/shrink a streamed slide according to the transferred DTMF signal.

4.6 Virtual whiteboard

The development of the virtual whiteboard allows the collaboration between participants to draw text and images on live streamed slides on videoconferencing sessions. Participants can set the number of the boards, the font colour, the font size and the position of entered text and images. In addition to that, this development enables participants to change the number of the streamed whiteboard slide, to clear a specific slide, to save the whiteboard in the PDF format and finally, to download the saved whiteboard from the server.

Furthermore, the open source project SVG-edit, which represents a web-based, SVG (Scalable Vector Graphics) drawing editor that is written in JavaScript and works in any modern browser [13], was implemented to the videoconferencing system. This implementation enables participants to draw graphics on the live streamed virtual whiteboard using any web browser (see Figure 7). According to this development, SVG objects will be transferred as payload in the XML format using the HTTP protocol. The HTTP Servlet will then forward the payload to the processing logic, which is able to manipulate and generate SVG documents. Finally FFmpeg will be informed to load and stream the corresponding slide after it has been edited by the processing logic. Moreover, the system is being further improved to allow the transfer of SVG objects between WebRTC clients and the Application Server using WebSocket channels.

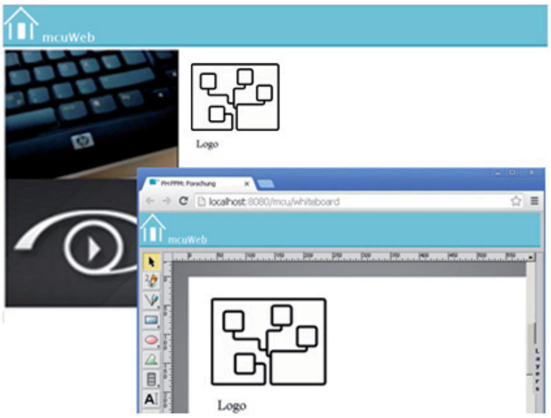

Figure 7 Virtual whiteboard

4.7 Zooming

The developed zoom functionality provides participants with the ability to magnify or shrink a portion of a live streamed document or whiteboard slide. This development was accomplished using the image processing software ImageMagick incorporation with the software FFmpeg (see Figure 8).

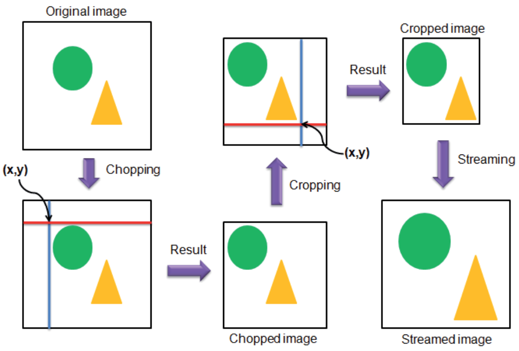

Figure 8 Zooming live streamed slides

According to Figure 8, the programming logic of the zoom service is represented by concentrating the selected portion of an image to the centre using cropping and chopping operations, whose coordinates are specified by the participants.

5 Conclusion and future perspectives

The implemented, developed and enhanced open source solution represents a voice and multimedia over IP application that can be used for videoconferencing, live web broadcasting and live slide presentations (see Figure 2). The videoconferencing system supports most common audio and video codecs. According to Figure 2, legacy SIP User Agents and modern SIP WebRTC clients can be used to join the same videoconferencing session.

Legacy SIP User Agents include both software and hardware (SIP phone or SIP video phone) based clients. On the other side, SIP WebRTC clients are provided by a web server and can be used from any device (PC, Smartphone, Tablet, etc.) that has the latest version of Google Chrome (version 30).

The slide presentations development enables participants to add documents to live videoconferencing sessions.

In addition to greater flexibility, lower IT costs and increased opportunities for innovation, open source solutions can provide and deliver real business value, simple license management, and efficient integrated management. Since the videoconferencing system consists of several open source projects, it offers developers a wide range of possibilities to add, implement and develop additional services and technologies. The following is a list of suggested improvements and extensions:

- DTMF signalling is done in this project using SIP INFO messages. Another way to transfer DTMF signals can be realised using the RTP protocol as specified in RFC 4733 [15].
- The media server contains a text mixer based on the real-time text codec T.140. This mixer is still under development and can be used to provide text chat functionality in a videoconference.
- Implementing additional WebRTC applications such as screen sharing, text chat and file sharing.
- Implementation and development of MSRP (Message Session Relay Protocol), which can be used within a SIP session to transfer files and Instant Messages.
- The SRTP key negotiation is done in this research work using SDES. Nevertheless, the negotiation could be done using the Datagram Transport Layer Security (DTLS) to authenticate the media flow.

This solution can be integrated in enterprise, social and educational networks and platforms, allowing users to invite each other to join live videoconferencing sessions, to share and stream documents, to create and download virtual whiteboards using both SIP and WebRTC technologies from any device that supports the SIP protocol.

6 Literature

[1] Bergkvist, Adam; Burnett, Daniel C.; Jennings, Cullen; Narayanan, Anant: WebRTC 1.0: Real-time Communication Between Browsers, W3C Working Draft, August 2012

[2] Singh, Kundan; Krishnaswamy, Venkatesh: A Case for SIP in JavaScript, IEEE Communications Magazine, April 2013

[3] Castaldi, Tobia; Miniero, Lorenzo; Pietro Romano Simon: On the Seamless Interaction between WebRTC Browsers and SIP-Based Conferencing Systems, IEEE Communications Magazine, April 2013

[4] Johnston, Alan; Yoakum, John; Singh Kundan: Taking on WebRTC in an Enterprise, IEEE Communications Magazine, April 2013

[5] Rosenberg, J.: A Framework for Conferencing with the Session Initiation Protocol (SIP), RFC 4353, IETF, February 2006

[6] Schulzrinne, H.; Casner, S.; Frederick, R.; Jacobson, V.: RTP: A Trans-port Protocol for Real-Time Applications, RFC 3550, IETF, July 2003

[7] Trick, Ulrich; Weber, Frank: SIP, TCP/IP und Telekommunikationsnetze: Next Generation Networks und VoIP, ISBN-10: 3486590006, ISBN-13: 978-3486590005, Oldenbourg Wissenschaftsverlag, Juli 2009

[8] Rosenberg, J.; Schulzrinne, H.; Camarillo, G.; Johnston, A.; Peterson, J.; Sparks, R.; Handley, M.; Schooler, E.: Session Initiation Protocol, RFC 3261, IETF, June 2002

[9] Syed, A. Ahson; Mohammad, Ilyas: Services, Technologies, and Security of Session Initiation Protocol, ISBN-13: 978-1-4200-6603-6, ISBN-10: 1-4200-6603-X, 2009

[10] Fette, I.; Melnikov, A.: The WebSocket Protocol, RFC 6455, IETF, December 2011

[11] Baz Castillo, I; Millan Villegas, J; Pascual, V: The WebSocket Protocol as a Transport for the Session Initiation Protocol (SIP), draft-ietf-sipcore-sip-websocket-09, IETF, June 2013

[12] MCU Media Server | Free software downloads at SourceForge.net. 2013. [ONLINE] Available at: http://sourceforge.net/projects/mcumediaserver/. [Accessed 08 July 2013]

[13] SVG-edit: A complete vector graphics editor in the browser (in JavaScript), Google Project Hosting. 2013. [ONLINE] Available at: http://code.google.com/p/svg-edit/. [Accessed 08 July 2013]

[14] ImageMagick: Convert, Edit, Or Compose Bitmap Images. 2013. [ONLINE] Available at: http://www.imagemagick.org/script/index.php. [Accessed 08 July 2013]

[15] Schulzrinne, H.; Taylor, T.: RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals, RFC 4733, IETF, December 2006