Real-Time Multimedia Session Splitting and Seamless Mobility in Session Initiation Protocol Environments

Thomas Stähle, Thorsten Kaczmarz, Jürgen Müller, and Michael Massoth
Department of Computer Sciences
Hochschule Darmstadt University of Applied Sciences
Darmstadt, Germany
Email: {thomas.staehle, thorsten.kaczmarz}@stud.h-da.de, {juergen.mueller, michael.massoth}@h-da.de

***Abstract*—Multimedia applications have became more usable in recent years, since mobile broadband internet access became available. Additionally, the Session Initiation Protocol turned out to be the standard signaling protocol for Next Generation Mobile Networks. There is an increasing demand on solutions for seamless mobile communication. Two of the most important services are Voice and Video over Internet Protocol. This paper addresses issues for transferring an established session to another device without interruption as well as splitting multimedia streams over different devices. Finally, a description about the ongoing implementation of one transfer and one split mechanism for mobile phones is introduced.**

***Keywords*-Session Initiation Protocol; Voice over Internet Protocol; Seamless Mobility; Next Generation Mobile Networks**

I. Introduction

In recent years the global evolution of the internet and its bandwidth has enabled the development of many multimedia applications such as Voice over Internet Protocol (VoIP) or video streaming.

The daily used services are provided by separated networks until now. One of the main concepts in future networking is to merge all these networks into a single one, a Next Generation Mobile Network (NGMN). It will be based on the Internet Protocol (IP) and provide gateways to ensure compatibility to legacy systems [1].

The networks themselves have altered. For example, techniques and bandwidth for accessing the internet increased. The Global System for Mobile Communication (GSM) provided a much lesser bandwidth than the Universal Mobile Telephony System (UMTS) using High Speed Packet Access (HSPA). Long Term Evolution (LTE) will provide even more bandwidth in near future. This evolution enables almost all real-time multimedia services that are currently known [2].

Today, people are using social networks (e.g., Facebook) to exchange messages, videos, and photos in real-time with their friends all over the world. It is most likely that the users want to use their services wherever they are, even while traveling. Improvements in power consumption, processing power, and memory make mobile devices capable of using real-time multimedia services with high performance [3].

An increasing amount of services are transferred from local computers to the internet or to mobile phones. Therefore, users can transport their files and services and access them wherever they are. Seamless mobility is a precondition for mobile networks [4]. All these developments underline the need of users to use their services and devices everywhere.

This paper is structured as follows: First, an overview about different types of mobility is given in Section II. Some use cases that apply on our perspective are introduced in Section III. The following sections describe transfer and split mechanisms. In Section VI, the ongoing implementation is described. Finally, a conclusion is drawn and future work is suggested in the concluding Section VII.

II. Types of Mobility

An increasing number of real-time multimedia services are provided for mobile devices, since broadband internet became available for them [5]. Therefore, a concept of mobility is required. This can be divided into four categories as follows [6].

A. Personal Mobility

Personal mobility allows users to initiate and receive calls from any device and location. Therefore, they need a mechanism to be reachable on multiple devices, which is provided by a Session Initiation Protocol (SIP) registrar. A user is addressed by his SIP URI. There are two types of SIP URIs: The temporary SIP URI which is mapped to the permanent SIP URI. A permanent SIP URI is similar to an email address (e.g., sip:alice@home.de) and is used for a location independent addressing. Temporary SIP URIs contains a IPv4 or IPv6 address in the host part of the URI (e.g., sip:alice@192.168.1.100) and are used to address a specific user agent directly. A stateful SIP proxy could support call forking, which makes it possible to send an call attempt to multiple devices in parallel [7].

B. Service Mobility

Service mobility means that every service is usable with the same data across different devices. Such data include address book entries, call log, or speed dial settings. The idea is to maintain the data only once and changes will be synchronized with all participating devices. SIP does not provide this kind of mobility by default. Berger et al. proposed a seamless mobility architecture, based on multimedia, device integration, events, location-awareness, privacy, and invisible users [8]. Shacham et al. introduced another architecture with features such as support of heterogeneous devices, location-based configuration, and limited configurations [9].

C. Terminal Mobility

With terminal mobility a terminal can move between different networks without any interruption of the ongoing session. The best known solution is the concept of Mobile IPv4 [10] and Mobile IPv6 [11]. They enable mobile user equipment users to switch networks while maintaining a permanent IP address.

Another solution is mid-call mobility, where a moving device sends another INVITE request to the session partner after it obtains a new IP address [6]. This request is sent to inform the session partner about the new address.

D. Session Mobility

Session mobility makes it possible to transfer an ongoing session from one device to another without any interruption. The session can be transferred completely or only partial.

SIP has two ways for session mobility. The first approach is Third Party Call Control (3PCC) [12]. The second approach uses the REFER method to provide session mobility over multiple devices [13].

The restriction of these approaches is that an ongoing session is transferred completely. It is not supported to split a session into separate media streams such as audio and video [5].

The detailed possibilities for session mobility are described in more detail afterwards, since this paper focuses on it.

III. Use Case Description

Multiple scenarios are possible in the context of our work. First, let us assume that a representative user has a smart phone with camera in use. Additionally, the user has an SIP-enabled television and Hi-Fi components at home.

These are only the most obvious scenarios for the session transfer and session spit. In Figure 1 is a use case diagram that gives further illustration.

A. Coming Home

Alice comes home while she has a phone call with Bob. She still wants to talk a bit with him but does not want to use her mobile phone any longer. She could have many reasons such as a low battery on her mobile phone or that she does not feel comfortable with the mobile any longer. However, she comes home and transfers the call to her devices at home. The voice transmission output is transferred to the Hi-Fi components while the voice input is done by a headset.

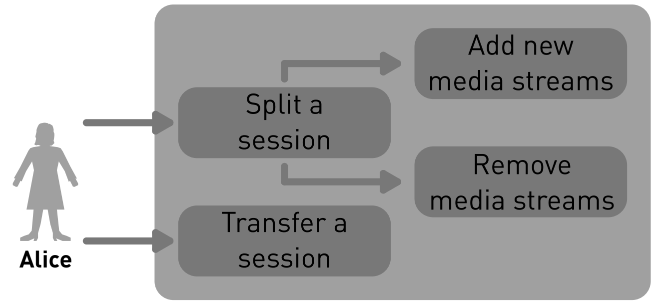

Figure 1. Use case diagram for session mobility.

B. Coming to a Smart Home

This scenario is very similar to the one above. She transfers the complete phone call to a central device at her home, instead of transferring the parts of it one by one. This central device then splits the audio streams immediately to pre-configured devices (e.g., the Hi-Fi components and the headset).

Additionally, it is possible to configure the central device in order to act differently, depending on the time of day. This could be that it transfers the complete session to the handset of her cordless phone during the night. Therefore, she can continue to talk with Bob without the risk waking up her husband.

C. Session Extension

In addition to the scenarios above could it be desirable of to enrich the current session. Bob maybe want to explain Alice some facts, she cannot understand them just by hearing it. Therefore, it is possible to add a video stream to the phone call, even if this is not supported with her mobile phone. This video stream is transmitted to the televisions in her home. Therefore, she can see what he explains.

IV. Session Transfer

Schulzrinne and Wedlund discussed two possibilities of transferring a session from one device to another [6].

In 3PCC the transfer-initiating device invites the new device and changes the current media streams, so that they are redirected to another device. The second way is the REFER request, which is sent from the transfer-initiating device to the session partner. The session partner establishes a new session to the referenced device after he receives the REFER request. He transfer the current session to the new device and quits his old connection to the originating device after a successful session setup.

The following sections are illustrated by a example scenario, which is performed between the two users Alice and Bob. Alice uses her mobile phone and is on her way home. Bob uses a video phone at work. The third device is Alice's

 ISBN: 978-1-61208-129-8

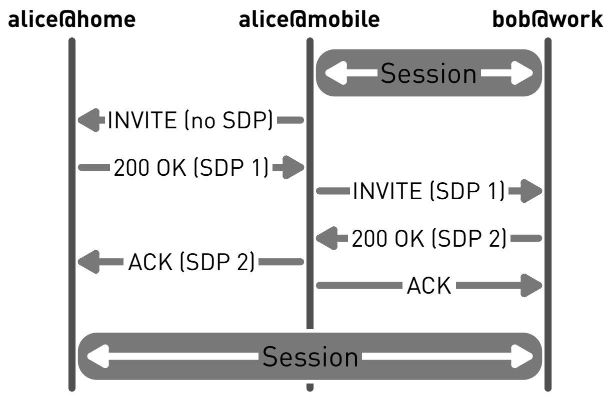

Figure 2. A session transfer using 3PCC.

video phone at home where the session should be transferred to.

A. Third Party Call Control

This method needs a third party that controls the call until the transferred session ends, as the name of this method indicates [12]. The session transfer with 3PCC is depicted in Figure 2.

Alice maintains a session on her mobile phone with Bob. Alice went home and wants to transfer the session to her home device. Therefore, Alice sends an INVITE request to it without Session Description Protocol (SDP) information. Following, Alice receives a 200 OK on her mobile phone from her home phone with an attached SDP offer (i.e., SDP 1). These SDP offer contain information about the supported media streams and sets of media codecs. Then, Alice updates the current session by forwarding this received SDP data to Bob in a reINVITE. Bob answers with a 200 OK and sends his SDP answer (i.e., SDP 2) including a selected media stream and codec. Finally, Alice sends an ACK request from her mobile phone with Bob's SDP offer to her home phone. An ACK request to Bob completes the transfer. Now, Bob sends his media streams to Alice's home phone and receives the media streams from there.

While the media streams are sent between Bob's and Alice's home phones, Alice's mobile phone still handles the signaling. Alice's mobile phone acts as the controller in this scenario. This is a disadvantage. Another variant is to let the controlling be done by a central controller that never keeps any media streams and only manages the session.

B. REFER Method

The REFER method [13] is the second possibility to transfer a SIP session to another device. The transfer-triggering device sends his session partner the SIP URI of the new device, where the session should be transferred, in the REFER request. The anchor device can retire from the session after a successful transfer. Figure 3 illustrates the transfer of a session via REFER method.

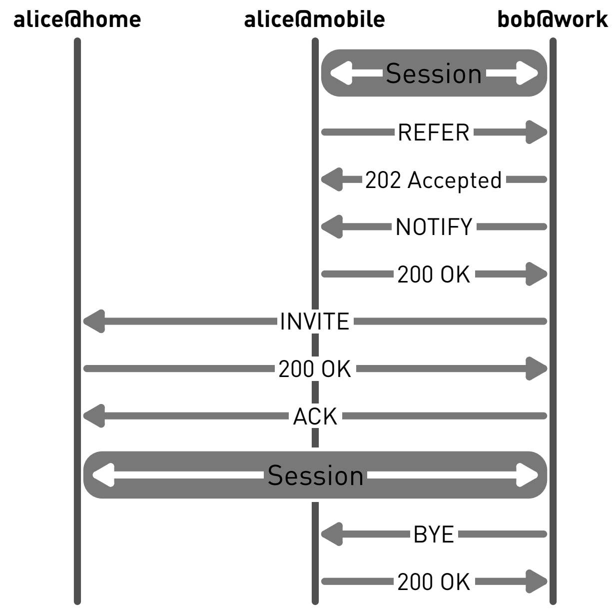

Figure 3. A session transfers using the REFER method.

Alice maintains a session on her mobile phone with Bob and wants to transfer the session to her home phone. Therefore, Alice sends a REFER request with the SIP URI of her home phone in the Refer-To header field to Bob. An exemplary REFER request with the Refer-To header field is presented in Listing 1. Bob establishes a new session with Alice's home phone via a new INVITE request if he accepts the REFER request. Bob quits the session with Alice's mobile phone by sending a BYE request, after the transfer is completed successfully

```
REFER sip:bob@work SIP/2.0
To: Bob <sip:bob@work>
From: Alice <sip:alice@mobile>;tag=193402342
Via: SIP/2.0/UDP proxy.mobile:5060
Call-ID: 0815@home
CSeq: 1 REFER
Refer-To: <sip:alice@home;method=INVITE>
Max-Forwards: 70
Content-Length: 0
```

Listing 1. An exemplary REFER request for a session transfer.

V. Session Split

Chen et al. proposed a mechanism to split a session over multiple devices, which uses the REFER request with the Mobility header field [14]. Another approach proposed from Shacham et al. groups all devices involved into a virtual device. Within this virtual device 3PCC is used to manage the session split [15].

In this section we assume that Alice uses an IP-based television with a camera instead of her internal phone.

 ISBN: 978-1-61208-129-8

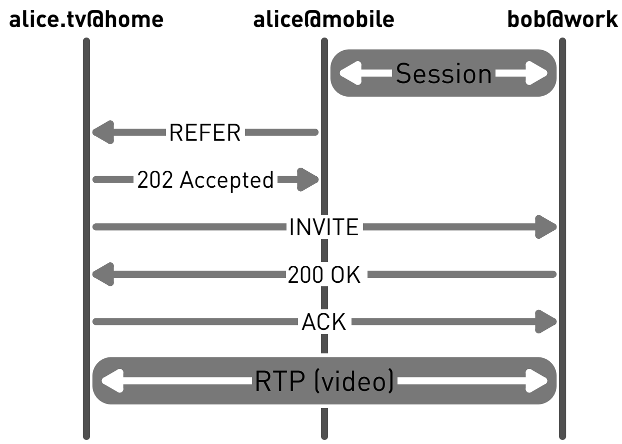

Figure 4. Split a session using SSIP.

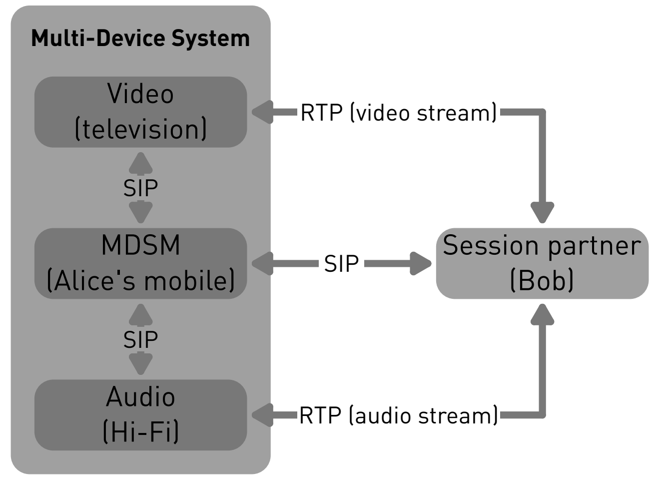

Figure 5. Schematic representation of a Multi-Device System.

A. Split Session over Multiple Devices

Chen et al. introduce an extension header field called Mobility [14]. The Mobility header field contains information about the current session and the media stream that should be split. Therefore, the header field contains the Call-ID of the session, an incoming split attempt belongs to. The header field is placed in REFER and INVITE requests. Figure 4 illustrates the session split with SSIP.

Alice maintains once more a multimedia session on her mobile phone with Bob. Alice now wants to split the video stream to her television. Therefore, she sends a REFER request to it. The Refer-To header field contains Bob's SIP URI and the Mobility header field contains the current Call-ID and the stream that should be transferred. In Listing 2 is an exemplary REFER request with Mobility header field. Then Alice's television sends an INVITE request to Bob in order to establish a video stream session. The Mobility header field in this INVITE request contains the Call-ID from the session between Alice's mobile phone and Bob. Therefore, Bob knows that this partial session belongs to his ongoing session with Alice's mobile phone [14].

```
To: Television <sip:alice.tv@home>
From: Alice <sip:alice@mobile>;tag=193402342
Via: SIP/2.0/UDP proxy.mobile:5060
Call-ID: 0816@home
CSeq: 1 REFER
Refer-To: <sip:bob@work;method=INVITE>
Mobility: 0815@home; media=video
Max-Forwards: 70
Content-Length: 0
```

Listing 2. An exemplary REFER request for a session split with SSIP.

B. Mobile Node Control

Shacham et al. combine several devices to a virtual device or Multi-Device System (MDS) [15]. One of these devices is the Multi-Device System Manager (MDSM). It is able to control the session and split certain media streams to other devices in the virtual device via 3PCC. This is called Mobile Node Control (MNC). Figure 5 shows the schematic design of a MDS with a MDSM, a television, and an audio device.

If Alice comes home while she is in a video call with Bob on her mobile phone, she can use her mobile phone as MDSM and split the video stream (e.g., to her television). Figure 6 illustrates the split mechanism.

The mobile phone acts as MDSN and sends an INVITE request without SDP information to the television in order to initiate the session split. She receives a SDP offer from the television in the 200 OK response (i.e., SDP 1). The MDSM sends a reINVITE to Bob with this SDP offer, which invites Bob to send the video stream to the television. Then, Alice's mobile phone receives a 200 OK response from Bob, which contains Bob's SDP answer (i.e., SDP 2). The MDSM only has to forward this SDP answer to the television to conclude

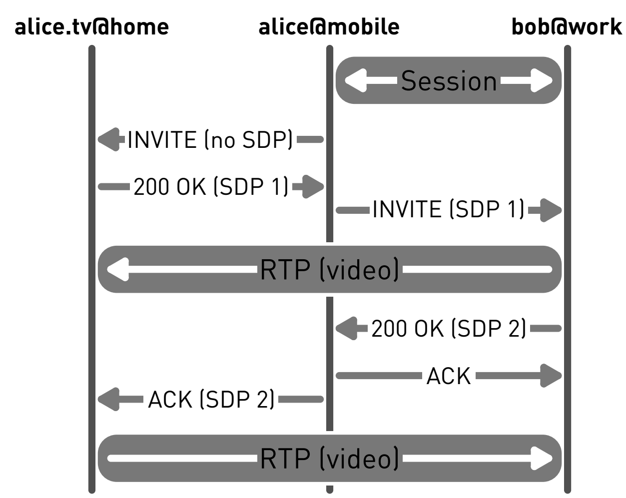

Figure 6. Split a session using a MDS in MNC mode [15].

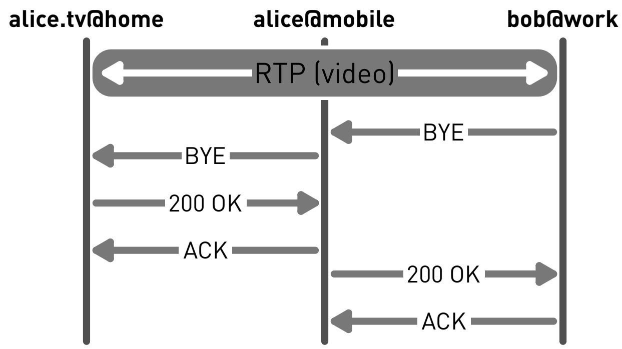

Figure 7. Tearing down a partial session using MDS in MNC mode.

the session split.

Alice only has to change the SDP information again if she wants to combine previously separated session again. Then, Bob sends the video stream to Alice's MDSM, which manages the session tear down with the television.

Alice has to forward an incoming BYE request from Bob to all devices in the MDS that are part of the current session, as depicted in Figure 7. Alternatively, she has to send a own BYE request if she decides to end the session.

VI. Implementation

This section introduces our ongoing implementation process for session mobility on mobile phones.

A. Method Decision

As described in Section II, session mobility include the session transfer and session split process. We chosed one of each aspects for the implementation.

Section IV explained two methods for transferring a SIP session (i.e., 3PCC and the REFER method). The disadvantage of 3PCC is that a central controller is always required. This is sometimes not desired (e.g., the battery of the initiating device runs out of energy) [14]. The REFER method lacks the ability to split up a session and transfer only certain media streams to other devices. The decision is made on the REFER method, because the session should be transferred completely from one device to another without any restrictions. Furthermore, the REFER method is already implemented in the most SIP stacks.

Section V explained two methods to split a SIP session (i.e., SSIP and MNC). The drawback of SSIP is that every user agent needs to understand the new Mobility header. According to this, every SIP user agent needs to be modified. The advantage of MNC is that only the user agent that initiates the split has to be modified. All other user agents involved can be regular SIP user agents. The MNC method has been chosen, because only the device that has to be capable of spliting and handle the partial sessions, have to be modified. In our opinion only few devices need such capabilities.

B. Testbed

To implement these methods the mobile phone operating system Android [16] was selected, because its application framework is well documented and easy to learn. The selected user agent is the open source application sipdroid [17]. It is released under the GNU General Public License (GPL) and uses the mjSip SIP stack [18]. mjSip is based on the SIP standard [7] and is released under the GNU GPL, as well. The REFER method is also still implemented in mjSip. Therefore, we can focus on the actual implementation of the selected methods.

sipdroid already has rudimentary preparations for the session transfer with the REFER method implemented, but there were bugs, which prevented the transfer from being successful. Furthermore, sipdroid has no proper multi session handling.

Unfortunately, the implementation process is not finished until this publication. Therefore, we are not able to present any evaluation results for now.

C. Security Considerations

Session mobility contains powerful procedures that are attracting abuse. An attacker could want to transfer a session to another destination of his or her use or interest. Another opportunity is that it could be used to hijack a session. The same for the session split functionality. An attacker could use this function to observe an ongoing session.

Therefore, it is important, if not necessary, to secure these procedures. This could be done by an authentication of the participating users (e.g., with Secure Multipurpose Internet Mail Extensions (S/MIME) [19]) [14]. Additionally, all security mechanisms that prevent an observation of the session, or even the session setup, could be applied to ensure higher security.

VII. Conclusion and Future Work

Mobility will be a very important application feature in the near future. This paper showed that SIP provides mechanisms to support significant session mobility by design. Nevertheless, SIP does not provide a way to split a session over multiple devices.

Some mechanisms are discussed to provide application layer mobility support for NGMNs in this paper. Therefore, different solutions were presented to show how a session can be transferred to other devices or split over multiple devices. Every solution has its benefits and drawbacks.

The most appropriate solutions were chosen for an implementation. The programming of the session transfer is already completed, while the implementation of the session split is still ongoing. We expect the completion by the end of February 2011.

Furthermore, security considerations will be taken into account after a successful implementations. A suitable solution to ensure the authenticity of the transfer- or split-initiating party should be specified. We will also start a first evaluation of the performance and reliability of the implemented sipdroid extension.

Acknowledgment

We would like to acknowledge Ronald Moore for his helpful comments.

References

[1] F. L. C. Ong, X. Liang, P. Pillai, P. M. L. Chan, G. Koltsidas, F. N. Pavlidou, E. Ferro, A. Gotta, H. S. Cruickshank, S. Iyengar, G. Fairhurst, and V. Mancuso, "Fusion of digital television, broadband internet and mobile communications – part i: Enabling technologies," *International Journal of Satellite Communications and Networking*, vol. 25, no. 4, pp. 363–407, Jul./Aug. 2007.

[2] "Next generation mobile life," BITKOM / KPMG, Berlin, Germany, 2008.

[3] "Gartner says worldwide mobile device sales grew 13.8 percent in second quarter of 2010, but competition drove prices down," Press Release, Gartner, Aug. 2010.

[4] N. Banerjee, A. Acharya, and S. K. Das, "Seamless sip-based mobility for multimedia applications," *IEEE Network*, vol. 20, no. 2, pp. 6–13, Mar.-Apr. 2006.

[5] M.-X. Chen and F.-J. Wang, "Session mobility of sip over multiple devices," in *Fourth International Conference on Testbeds and Research Infastructures for the Development of Netwoks & Communities and Workshops (TridentCom 2008), Innsbruck, Austria – March 17 - 20, 2008*. New York, NY, USA: ACM, 2008, pp. 23:1–23:9.

[6] H. Schulzrinne and E. Wedlund, "Application-layer mobility using sip," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 4, no. 3, pp. 47–57, Jul. 2004.

[7] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "Sip: Session initiation protocol," RFC 3261, IETF, Jun. 2002.

[8] S. Berger, H. Schulzrinne, S. Sidiroglou, and X. Wu, "Ubiquitous computing using sip," in *Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video (NOSSDAV 2003), Monterey, CA, USA, June 1-3, 2003*. New York, NY, USA: ACM, 2003, pp. 82–89.

[9] R. Shacham, H. Schulzrinne, W. Kellerer, and S. Thakolsri, "An architecture for location-based service mobility using the sip event model," in *The Second International Conference on Mobile Systems, Applications, and Services (Mobisys 2004), Boston, Massachusetts, USA, Hyatt Harborside, June 6-9, 2004*. New York, NY, USA: ACM, 2004.

[10] B. Patil, P. Roberts, and C. E. Perkins, "Ip mobility support for ipv4," RFC 3344, IETF, Aug. 2002.

[11] D. B. Johnson, C. E. Perkins, and J. Arkko, "Mobility support in ipv6," RFC 3775, IETF, Jun. 2004.

[12] J. Rosenberg, J. Peterson, H. Schulzrinne, and G. Camarillo, "Best current practices for third party call control (3pcc) in the session initiation protocol (sip)," RFC 3725, IETF, Apr. 2004.

[13] R. J. Sparks, "The session initiation protocol (sip) refer method," RFC 3515, IETF, Apr. 2003.

[14] M.-X. Chen, C.-J. Peng, and R.-H. Hwang, "Ssip: Split a sip session over multiple devices," *Computer Standards & Interfaces*, vol. 29, no. 5, pp. 531–545, Jul. 2007.

[15] R. Shacham, H. Schulzrinne, S. Thakolsri, and W. Kellerer, "Session initiation protocol (sip) session mobility," RFC 5631, IETF, Oct. 2009.

[16] "Android.com," http://www.android.com/ 05.02.2011.

[17] "sipdroid: Free sip/voip client for android," http://www.sipdroid.org/ 05.02.2011.

[18] "Mjsip," http://www.mjsip.org/ 05.02.2011.

[19] B. Ramsdell, "Secure/multipurpose internet mail extensions (s/mime) version 3.1 message specification," RFC 3851, IETF, Jul. 2004.