Scientia
Interdisciplinary Studies in Computer Science
20(2): 119-128, July/December 2009

MobSIP: A SIP extension to support application layer handover in real-time multimedia communications with mobility requirements

Daniel G. Costa[1], Sergio Vianna Fialho[2]

[1]Department of Technology, State University of Feira de Santana
Av. Transnordestina, S/N, Novo Horizonte, 44036-900
Feira de Santana, Bahia, Brazil

[2]Department of Computing and Automation, Federal University of Rio Grande do Norte
Caixa Postal 1524, Campus Universitário, Lagoa Nova, 59072-970
Natal, Rio Grande do Norte, Brazil

danielgcosta@ecomp.uefs.br, fialho@pop-rn.rnp.br

Abstract
In real-time multimedia, the overall time in Internet communications must be low and constant, in order to keep real-time sense and received media quality. When mobility is a basic requirement, efficient and flexible solutions should be adopted, avoiding harming time sensitive applications. In order to support real-time multimedia communications with mobility requirements on Internet backbones, a novel SIP extension is proposed, adding direct support to handover procedures in SIP clients. The procedures of that new extension, MobSIP, are specified and implemented, allowing formal and experimental verifications.

KEY WORDS: SIP, application layer mobility, handover procedures, real-time multimedia.

Resumo
MobSIP: uma extensão SIP para suporte a handover em nível de aplicação em comunicações multimídia em tempo real com requisitos de mobilidade. Em comunicações multimídia em tempo real na Internet, o fator tempo deve ser baixo e constante, a fim de manter a noção de tempo real e a qualidade da mídia recebida. Quando a mobilidade é um requisito básico, soluções eficientes e flexíveis devem ser adotadas, evitando prejuízos às aplicações sensíveis ao tempo. A fim de suportar comunicações multimídia em tempo real com requisitos de mobilidade em *backbones* Internet, uma nova extensão SIP é proposta, adicionando suporte direto a procedimentos de *handover* em clientes SIP. Os procedimentos dessa nova extensão, MobSIP, são especificados e implementados, permitindo verificações formais e experimentais da solução.

PALAVRAS-CHAVE: SIP, mobilidade em nível de aplicação, procedimentos de *handover*, multimídia em tempo real.

1 Introduction

In modern networks, mobility requirements are demanding new solutions related with Internet evolution trends. When these requirements are associated with real-time multimedia communications, a complex environment is created. Mobile real-time multimedia applications are becoming common, due to wireless network spreading and new 3G/4G cellular technologies (Schiller, 2003). There are few mobility solutions that support that scope of communication, stimulating new specifications for this area.

Session Initiation Protocol (SIP) (Rosemberg and Schulzrinne, 2002) is an Internet application-layer protocol used to establish, control and tear down real-time multimedia communications between two or more users. Nowadays, SIP stays as the main solution for real-time multimedia communication in Internet environments. Together with Session Description Protocol (SDP) (Handley and Jacobson, 1998) and Real Time Protocol (RTP)/Real Time Control Protocol (RTCP) (Schulzrinne and Casner, 2003), SIP presents itself as a complete and flexible public architecture for real-time multimedia applications as videoconference, Voice over IP (VoIP), Voice over Demand (VoD), among others (Halsall, 2000).

Mobility on Internet backbones is formed by two main parts: user localization and terminal mobility. In the first one, we wish to know the address being used in a specific moment by a host, tough such address can't be predicted or previously known. To manage the user location (current IP address), SIP employs Registrars servers to keep track of mobile hosts while they move across different Internet networks. In the second part, a host that acquires a new IP address, received from a visited network, has to notify the destination endpoint about that new address in order to maintain current communication, with minimal loss of data and time. To support that service, SIP architecture employs proxy servers and redirect servers, which can add an extra delay to the overall communication, potentially harming time sensitive transmission. Other approach is to restart a communication, which can also impose additional overhead.

As an alternative to proxy and redirect servers and restart of communications, it is proposed herein a novel extension to standard SIP, called MobSIP. That extension adds a new message to standard Session Initiate Protocol, related with terminal mobility. Moreover, the procedures that should be adopted by communication endpoints are also specified, describing what should be done by a SIP client in order to support that new extension.

To attest the correctness of the proposed solution, the new extension was implemented in SIP clients. Doing so, the procedures specified for MobSIP could be verified in real communication scenarios, allowing comparisons with other solutions.

This paper is structured in the following way. Section 2 presents the concepts related with the SIP architecture. Section 3 completely describes the MobSIP extension. Section 4 brings implementation details and experimental verifications in communication environments composed by SIP clients that use MobSIP. At last, the conclusion and references are presented.

2 SIP architecture

Real-time multimedia transmission over Internet backbones can be supported by a hand of communication architectures. Among the possibilities, SIP architecture presents itself as a complete, flexible and robust solution for modern applications in this area.

Session Initiation Protocol is the core of this architecture, being used to create, close and manage real-time multimedia communications. Based on several messages, which can be used to request a service or to indicate an answer, SIP offers support for call signaling, data exchange and specific control among SIP clients or among such clients and SIP special servers. Table 1 presents the main request messages of that protocol.

Table 1. Main SIP request messages.

Message	Description
ACK	Used to confirm the reception of a message.
BYE	Used to quit a current communication.
CANCEL	Used to cancel an action.
INVITE	The first message of the SIP connection handshake.
MESSAGE	A generic message, with no specific function.
NOTIFY	Used to inform an event.
OPTIONS	Used to request information from SIP servers.
REGISTER	Message related to Registrar servers.
SUBSCRIBE	Used to register a user to receive some event from a special server.

The messages used to indicate an answer have the same structure of Hypertext Transfer Protocol (HTTP) (Fielding *et al.*, 1999) messages: numerical information creates a scope of answers, as, for example, a code between 200 and 299 representing success and between 400 and 499 indicating error.

To start a typical communication using SIP support, a three-way handshake has to be adopted before any data transmission. A SIP connection is established after sending and reception of specific messages. Figure 1 presents a typical successful three-way handshake specified for SIP. 180 RINGING is an optional message for the handshake, since it is used to indicate that the remote endpoint was notified about the intention of communication, but he/she has not accepted yet.

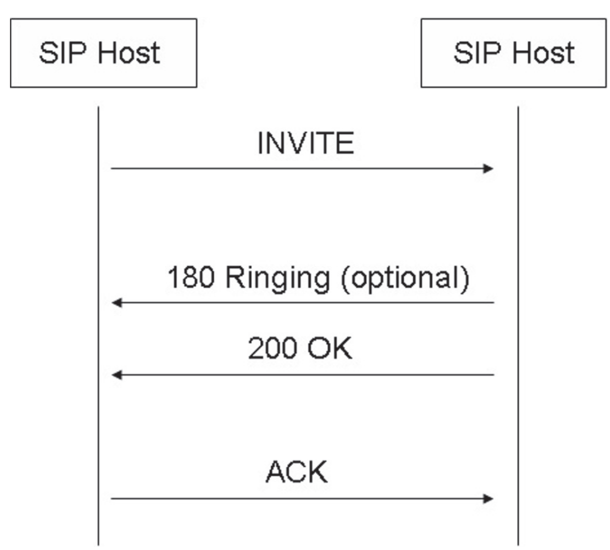

Figure 1. SIP handshake for communication establishment.

The transport service for SIP signaling can be provided by Transport Control Protocol (TCP) (DARPA, 1981) and User Datagram Protocol (UDP) (Postel, 1980).

SIP messages are supported by Session Description Protocol. Capability negotiation provided by SDP is an important service that has to be present in any real-time multimedia applications, although such service can come from other solution than SDP. As many codecs are currently available, for audio and video alike, and there is no way to predict what codec will be used, SDP allows communication endpoints to set codecs properly. SDP information is carried on the first and the second messages of SIP connection handshake, as presented in Figure 1.

Real-time multimedia data are not sent in SIP messages. Time sensitive data in Internet is encapsulated by RTP packets (Schulzrinne and Casner, 2003) instead. Timestamp information presented in RTP packet header is part of decoding and reproduction of received media in communication endpoint. While SIP controls the communication, RTP is used to send encoded audio and/or video, or even text information. Optionally, encoded real-time multimedia can be sent in a cryptographic stream. When such option is chosen, Secure Real-Time Protocol (SRTP) (Baugher *et al.*, 2004) should be used instead of RTP.

Many services are supported by SIP. One of them is user localization. Such service allows that the IP address being currently used by a host be known, even if that address just changed due to user mobility through different networks. SIP Registrars servers keep track of user movement across networks: when a host acquires a new IP, it sends a SIP message to its home Registrar informing that new address. The current IP can be discovered by asking the proper record at that SIP server. SIP User localization is a mobile communication service that can be used by any kind of application, although SIP mobility is specially designed for real-time multimedia data and chat with plain text.

3 MobSIP specification

Mobility in IP networks is requesting new tough solutions, due to the growing demand for improvement in applications and backbone structure. When time sensitive data makes part of that mobility demand, the resulting complexity encourages the specification of communication solutions adapted with these operation scopes. Real-time multimedia applications with mobility requirements are becoming common, due to the spreading of wireless network and new 3G/4G cellular technologies (Gast, 2005).

MobSIP is a solution for application layer handover, covering all traffic related with real-time multimedia communications (control messages and packets encapsulating encoded media). The following subsections describe the details of MobSIP extension.

3.1. Related work

Network mobility is not yet a complete and operational service widely available on public Internet backbones. Limitations in Internet structure, due to its original purpose, impose some restrictions for mobile communications (Clark *et al.*, 2005). So, many aspects of such communications have been treated in scientific papers in the last years, trying to improve the support for mobility in Internet.

Considering this paper, some works have a deeper impact. In Wedlung and Schulzrinne (1999), it is exposed SIP mobility based on proxy and redirect servers. Also, it presents the (re)send of INVITE messages as a way to keep current communication sessions, when a host movement is detected (and a new IP is acquired). A similar discussion is taken in Schulzrinne and Wedlung (2000), but focused on Internet application layer. In Dutta *et al.* (2004), SIP mobility is also discussed, but the main focus is how to treat eventual packet loss resulted from “slow” handovers.

Regarding SIP mobility, the present paper brings new contributions with the specification of a novel SIP extension to support application-layer handover with no support expected from SIP servers or even (re)establishment of communication sessions. Moreover, it is focused on terminal mobility instead of personal and service mobility, as in Wedlung and Schulzrinne (1999), benefiting real-time multimedia applications with a demand for fast handover. In fact, it is proposed an end-to-end handover service with some similarities with the transport-layer terminal mobility service specified in Xie and Stewart (2007), but in a real-time multimedia domain.

3.2. User Localization and handover

Internet mobility is composed by two main parts: user localization and terminal mobility (handover). Both of them must be treated properly in order to allow mobile communication on IP backbones.

Communications among IP hosts in a mobile context may be of two kinds: (a) the ones initiated from mobile hosts to non-mobile hosts, in wired or wireless networks, and (b) the communications targeted to mobile hosts, no matter the origins. In (a) there is no need of any user localization mechanism. When a host is wired, its address can be known previously by some means usually presented in IP networks, as Domain Name System (DNS) (Mockapetris, 1987). However, in (b) we wish to know the address being used in a specific moment by a host, though such address cannot be predicted or previously known.

To manage user location information, standard SIP uses Registrars servers to keep track of mobile hosts while they move across different Internet networks. In this approach, each host must register itself in its home Registrar, which associates the currently used IP address with a globally unique SIP address (in the form sip:user@domain). When a host acquires a new address from the network it moved to, that new information is registered in its home SIP Registrar.

To start a communication, a host has to know the SIP address of the destination endpoint (for example, sip:daniel@uefs.br). The domain part of this SIP address represents the home Registrar of the destination endpoint. Using DNS, for example, the IP address of that Registrar can be discovered. Now, all the host has to do is to query the Registrar for the user part of the SIP address. That query returns the current IP address of the terminal being used by the user who has this SIP address. It does not matter the terminal being used, since the user is located by his/her SIP address.

The second part of any network mobile solution is the terminal mobility. A mobile host (that has to be a wireless host too), from the power level of received signals broadcasted by access points (Gast, 2005), can identify a changing in the network it is attached to. This identification is managed by network link layer. Sometimes, changing in physical network may result in changing in logical network. In cases when the logical network is the same, there is no need to acquire a new IP address. IP routing uses a subnet mask to decide if packets must be routed to a pre-defined path or if packets must be delivered locally. So, in the same logical network, changing of wireless cell does not require a new IP address, though it can be done. If the mobile host goes to a different logical network, it has to set a new IP address, which indeed can be automatically received from some network service, as Dynamic Host Configuration Protocol (DHCP) (Droms, 1997). To identify logical network changing, it is expected some support from an upper layer, since link layer does not understand IP address concept. As soon as the host realizes it is in a different logical network, it can start the procedure to set/receive a new IP and subnet mask information. Additionally, hosts in Internet usually need to know the address of default gateway and DNS servers to send resolving queries.

Typically, when a mobile host receives a new IP address, the current communication is lost. To keep such communication, hosts have to indicate that new address to the remote endpoint, if any. There are protocols and architectures that provide handover support, as Stream Control Transmission Protocol (SCTP) (Stewart and Xie, 2007), Mobile IP version four (Perkins, 2002) and six (Johnson *et al.*, 2004) and Host Identity Protocol (HIP) (Moskowitz and Nikander, 2006; Ratola, 2004). All of them require special support from network communication devices or can potentially harm time sensitive applications or even prohibit multicast routing (Deering, 1989). Such handover support can also be expected from SIP redirect and proxy servers, in different ways, though an additional overhead will be imposed by packet redirection. Moreover, redirection of SIP messages still requires some mechanism to redirect real-time multimedia packets. Finally, reestablishment of communication sessions also imposes an additional overlay, as will be shown later.

3.3. NewIP Message

In order to support handover procedures without using proxy servers or restart approaches, a SIP extension is proposed herein. It is expected better performance using a non-server handover solution when compared with centralized communication based on packet redirection. That new extension, MobSIP, is to be used together with traditional SIP localization procedures supported by Registrar servers. Although it is still necessary to use servers to register current IP address, SIP standard localization procedures have a minimal impact in real-time multimedia communications, in a different way of redirection of real-time packets (Wedlung and Schulzrinne, 1999).

MobSIP specifies a new message to inform the destination endpoint about the new IP address that has to be used in the current communication session. That new message, named NewIP, is to be employed in a dynamic way, every time any of the communication endpoints acquires a new IP address. In SIP terminology, NewIP is classified as a request message.

As SIP is a text-based protocol, NewIP follows the basic format specified in (Resnick, 2001), as all SIP standard messages. So, that new message is formed by three distinct fields, following the restrictions and details of SIP request messages (Rosemberg and Schulzrinne, 2002). Table 2 presents the three parts that form the NewIP message.

Table 2. NewIP message structure.

Message field	Description
Start line	Identify the message. Example: NewIP sip:user@domain SIP/2.0.
Header	General information of the message. In current version of NewIP, the allowed headers are: From, To, Call-ID and CSeq.
Body	Contains the message payload. In NewIP message, the new address of the host will be written in this field just after a blank line, with no additional information.

SIP version in NewIP messages was defined as 2.0, in the same way it is done in all SIP standard messages (Rosemberg and Schulzrinne, 2002). Since MobSIP is only a small extension for SIP, there is no reason to propose a change in version number, as the type of the message can be easily identified reading the first line.

Figure 2 presents an example of a NewIP message. In that example, the message informs the new IP address for the communication: 10.30.0.17.

```
NewIP sip:daniel@uefs.br SIP/2.0
From: sip:sergio@ufrn.br
To: sip:daniel@uefs.br
Call-ID: bsd34bsd71f@10.30.0.25
CSeq: 1 NewIP

10.30.0.17
```

Figure 2. A typical NewIP message.

Call-ID and CSeq fields are used to identify logical relationship among messages, providing a soft security mechanism against non authentic NewIP senders. This behavior is identical to standard SIP messages processing.

Figure 3 presents a typical usage of NewIP messages, which have to be sent right after the reception of a new IP address due to movement across wireless networks. The 200 OK response message is used to confirm the correct reception of NewIP.

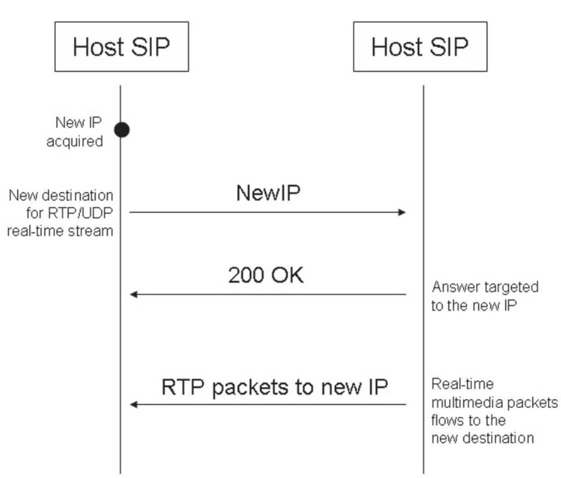

Figure 3. Using NewIP message.

In order to avoid extra delay in real-time multimedia communications that are also mobile, hosts must start to send packets to the new destination as soon as it receives a NewIP message from the remote endpoint, even tough a 200 OK message is not sent yet.

As one can note, the new IP address can be discovered from the source address field in IP datagrams which encapsulates SIP messages (in fact, UDP or TCP encapsulates SIP messages, and that complete structure goes inside IP datagrams). However, this approach is not used here, since the NewIP message would have to be sent anyway to indicate the handover. Additionally, textual information about the new IP address makes easier the identification and monitoring of MobSIP communications.

3.4. MobSIP operation

MobSIP is a mobility solution based only in an application layer protocol. Doing so, it is expected no additional overhead from network devices, as routers and switches. The final solution provides a flexible and efficient way to support real-time multimedia communications with mobility requirements on Internet backbones, as well as any communication that uses SIP for signaling control.

In order to ensure the correct operation of the proposed SIP extension, attesting its definitions are free of deadlocks and misunderstandings, the formal specification language SDL (Specification and Definition Language) and its extension SDL/GR (SDL Graphical Representation) (SDL, 2009) were employed. With the specification in SDL, syntactic and semantic verification could be performed.

A SIP terminal was defined as three SDL blocks: SIP Standard Control, SIP User Localization and SIP NewIP. The MobSIP solution is formed by SIP User Localization and SIP NewIP blocks. The first one is already defined for SIP, so there is no need for additional specification. The new element is SIP NewIP block, which specifies a new message to be used for handover support in application layer and the procedures related with this service.

Figure 4 brings the SDL specification for MobSIP overall operation, composed by SIP User Localization and SIP NewIP SDL blocks. Table 3 describes the signals and states presented in that specification diagram.

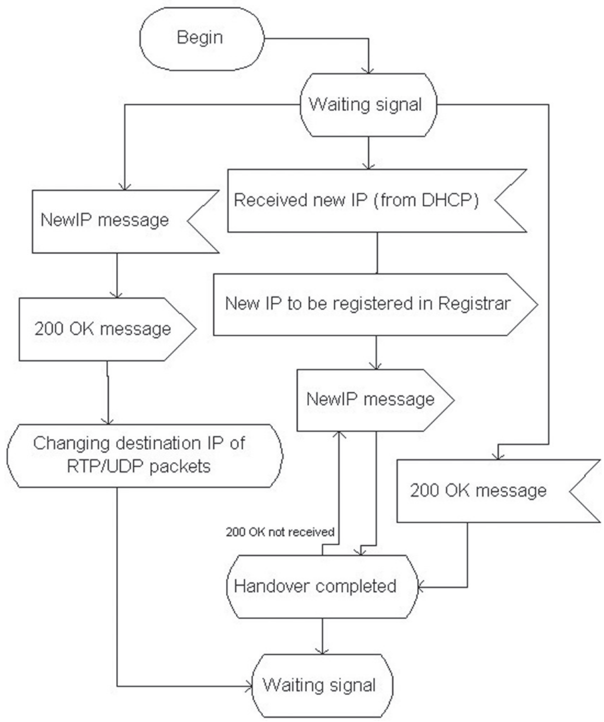

Figure 4. SDL specification of MobSIP procedures.

When a NewIP message is received, the destination IP address of all outgoing RTP packets is changed to the current address of the remote endpoint. On the other hand, when a host changes its current address, receiving a new one, it has to register such address at its home Registrar. As soon as possible, a NewIP message has to be sent to the remote endpoint of the communication, in order to indicate the new destination address of every RTP packet originated from the remote host. Each reception of a NewIP message has to be confirmed by a SIP standard 200 OK message.

The Session Initiation Protocol is specified to operate over TCP or UDP transport PDU (Protocol Data Unit) (DARPA, 1981; Postel, 1980; Rosemberg and Schulzrinne, 2002). As IP addresses of any communication can change due to host mobility, MobSIP specifies that only UDP can be used to support SIP, since TCP transport protocol is connection oriented. UDP is already the usual transport protocol for real-time multimedia communications (Schulzrinne and Casner, 2003).

After the formal specification is completed, syntactic and semantic verification were performed, attesting the correctness of the specification. At this point, we can ensure that MobSIP is a consistent specification, with no deadlocks and ambiguities.

Table 3. SDL signals and states of the specification.

Element	Description
200 OK message	Used to confirm the reception of a NewIP message. Reception of a 200 OK message from a remote host. The handover procedure is completed. Sending of a 200 OK message. The destination IP address of every IP datagram encapsulating RTP/UDP as payload is changed, reflecting the new address indicated in a received NewIP message.
Begin	Represents the initial state of a MobSIP terminal, when it is power on.
Changing destination IP of RTP/UDP packets	When handover is completed, real-time multimedia streams must be redirected to the new address of the remote endpoint.
Handover completed	This state indicates that handover procedures specified by MobSIP were executed properly.
New IP to be registered in Registrar	This signal represents a standard SIP message used to register the current IP address in a Registrar server. The address to be registered is received from a visited wireless network.
NewIP message	A signal that represents a NewIP message. Reception of a NewIP message from a remote host. A 200 OK message must be sent as an answer. Sending of a NewIP message. The terminal waits for a 200 OK message from the remote endpoint.
Received new IP (from DHCP)	When a host moves to a new wireless network and acquires a new IP address (probably from DHCP), it is indicated by this SDL signal (in practical means, it could be an event for the application).
Waiting signal	The terminal is ready to start a new communication. Such communication can be initiated by this terminal or by the remote endpoint.

4 MobSIP experimental verification

In order to verify the practical operation of the proposed mobility solution, MobSIP was implemented in a typical SIP multimedia communication terminal.

To implement MobSIP, the Jain-SIP Java API (Jain-SIP, 2009) was employed. This API allows the implementation of SIP terminals following the standard procedures presented in RFC 3261 (Rosemberg and Schulzrinne, 2002).

The SIP terminals with MobSIP extension were implemented with an additional software module able to send, to receive and to process NewIP messages. In order to avoid significant changes in Jain-SIP structure, NewIP messages were implemented using the generic type SIP MESSAGE (Table 1). This type of request message makes part of the SIP standard set of messages, but no functionality is expected from it. Doing so, the experimental operation of MobSIP could be tested in an easy but efficient way.

To measure MobSIP performance, it was necessary to specify a communication scope to be regarded. Since MobSIP is an application layer solution, completed independent from any network device, it was not compared in first moment with network layer mobile architectures, as MIPv4 (Perkins, 2002). Additionally, as MobSIP is not connection oriented, it was not also compared with connection oriented approaches, as Mobile SCTP (Xie and Stewart, 2007). In fact, MobSIP performance was evaluated referring to SIP standard solutions. For MobSIP verification, SIP proxies and restart of communication sessions (Re-INVITE) were considered.

A wireless communication environment was created, composed by four distinct IP logical networks with low load, as Figure 5 presents. All wireless cells are composed by IEEE 802.11b Access Points (Gast, 2005), interconnected by a IEEE 802.3u Fast Ethernet link (Spurgeon, 2000). For all experiments, host h1 and host h2 establish a SIP communication session and, some time later, host h1 moves from network n2 to network n3, using one of the available handover strategies (all based on UDP). The acquisition of a new IP address from a visited network was simulated manually, since the using of a specific network service, as, for example, DHCP, would have no advantage or even impact in the intended experimental verification. The same is valid for SIP Registrar servers, utilized in user localization procedures.

A SIP application acting as UAC (User Agent Client) and UAS (User Agent Server) (Rosemberg and Schulzrinne, 2002) was installed in both h1 and h2. When the communication is established, h2 initiates a RTP audio stream created from a WAV song lasting approximately 2 minutes, and directed to h1. Such procedures intend to simulate a real-time multimedia communication. For all experiments, the audio stream has to reach h1 in its new IP address, transparently.

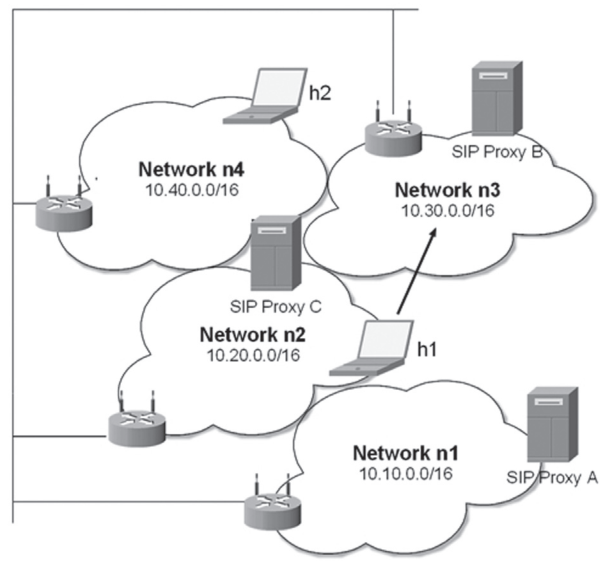

Figure 5. Experimental environment.

Five distinct experiments were executed, each of them regarding a specific solution: MobSIP, Redirection through Proxy A, Redirection through Proxy B, Redirection through Proxy C and Re-INVITE. For each of them, three different attempts were performed, trying to create a better scope for measurement.

In the first round of tests, it was checked the overall time for the handover procedure selected. When a new IP is acquired, the timestamp that represents such event is logged by the application. The difference between such mark and the timestamp for the reception of the first packet targeted to that new address indicates the time (delay) of the handover. Depending on the solution adopted, that time can be higher than the RTT (Round Trip Time) (Naylor and Opderbeck, 1974) between the two communication endpoints, as, for example, in some configurations of SIP Proxy servers, as redirected packets can take a longer path than packets directed forwarded/routed to its destination.

In fact, handover delay could be measured counting the time between the reception of a new IP address from the visited network and the exact moment that this information was available for the remote host and it could indeed start to send real-time multimedia packets to that new address. However, as tests considered third-part elements, as SIP proxies, such approach would not attest the real impact of the handover for the communication as a whole, since every packet sent after the handover will follow the new path created.

Table 4 presents the time between the acquisition of a new IP address and the reception of the first packet in that new location, from the remote endpoint. That first package can be a 200 OK message, sent in reply of a NewIP or an INVITE, or even a RTP/UDP real-time packet. As the experimental environment was composed by wireless cells with low load, in a controlled laboratory, the time measured closely reflects messages delivered with no error and without retransmission by timeout (in the case no reply is received). For the three attempts, variations in the measured delay are a result of many factors, as system resources and wireless signal propagation.

Table 4. Delay for the reception of the first packet after handover.

	Attempt 1	Attempt 2	Attempt 3
MobSIP	6.6 ms	6.5 ms	6.3 ms
Using Proxy A	9.8 ms	9.9 ms	10.3 ms
Using Proxy B	5.3 ms	5.7 ms	5.8 ms
Using Proxy C	8.9 ms	9.6 ms	9.5 ms
Re-INVITE	6.8 ms	6.9 ms	6.6 ms

The first experiment used the MobSIP specification. The following three experiments employed one of the three SIP proxies available. SIP communications can use proxy servers as a third element to redirect SIP messages and, together with RTP special elements (Schulzrinne and Casner, 2003), redirect also real-time multimedia data (Wedlung and Schulzrinne, 1999; Schulzrinne and Wedlung, 2000). The final experiment employed restart of the SIP communication session, but with no proxy.

For any of the five experiments, a series of three handover attempts were performed. For each test, it is desired to know the time between the acquisition of a new IP from the visited network and the reception of the first packet containing a SIP message or a real-time RTP/UDP packet from the remote endpoint. For MobSIP, this time is equivalent to the RTT between the communication peers, plus the time to send the NewIP message, to receive and process it and to send back the first packet to the new location of the remote endpoint (a 200 OK message or a RTP/UDP packet).

As one can see in Table 4, such time is very close to the measured delay of Re-INVITE approach, with a small difference in favor of MobSIP. In fact, restarting allows the communication to be reset regarding the new IP address received from the visited network (for example, delivered by a DHCP server). However, this approach demands reestablishment of buffers and system resources, and the final delay depends on how the application and the Operation System will treat this solicitation.

For communications using proxies, the overall delay is a function of the distance between the proxy and the hosts, and between the proxy and the access point it is "attached" to. When h1 moves, it sends a specific SIP message to the proxy, telling about the new location where packets must now be redirected. In Table 4, one can see that, for Proxy B, the delay measured was lower than the other four experiments. This happened because packets already sent to Proxy B, but that had not yet got there, would be forwarded by a proxy that knows the current location of h1. For the tests, it resulted in a better performance.

After the handover is completed, a new path is created for packets from h2 to h1. Even when a "close" SIP proxy is chosen, as happened with the experiment that used Proxy B, the delay caused by redirections results in a worse performance when compared with MobSIP experiment, since, after a while, packets will be considered from the origins.

For the second round of tests, it was desired to measure the path created by the handover solution, regarding 10 seconds, 20 seconds and 30 seconds after the handover was completed. To do so, the SIP application used in the experiments was implemented with an "accessibility" test: A SIP message of type MESSAGE, with no content, should be replied by another SIP message of same type. The difference between the timestamp registered in the moment of sending of this SIP message and the exact time of reception of the correspondent message from the remote endpoint indicated the RTT of the path. This way, every element that could process a SIP message would be considered for RTT computing, including proxies.

Table 5 presents the results of the second round of tests. The analysis of that table shows the similarity of MobSIP and Re-INVITE experiments, but also indicates the delay resulted by the using of SIP proxies, since messages do not flow necessarily through the shorter way.

Table 5. RTT for the new path between h1 and h2, after handover.

	10s	20s	30s
MobSIP	6.3 ms	6.7 ms	6.6 ms
Using Proxy A	11.1 ms	10.7 ms	11.4 ms
Using Proxy B	8.0 ms	8.3 ms	8.1 ms
Using Proxy C	10.3 ms	10.1 ms	10.8 ms
Re-INVITE	6.7 ms	7.0 ms	6.9 ms

After the execution of the experiments, it was noted that time for handover in MobSIP was lower than the other two approaches, but very close to Re-INVITE approach. However, some aspects should be regarded. First of all, MobSIP handover procedures employs only two messages (Figure 3), while restarting SIP communications demands at least three messages (Figure 1). Moreover, time for restarting/reallocating system resources must also be considered, and such time counts against the restart of the communication.

An additional point of Re-INVITE approach is the size of SIP messages in a new three-way handshake, when compared with NewIP/200 OK MobSIP handshake. To reestablish a communication, INVITE and 200 OK messages should also encapsulate a SDP message, while NewIP only carries the new IP address of the mobile host. In our experiments, INVITE messages for restart of the communication sized 572 bytes, while NewIP messages sized only 441 bytes. It has to be noted that, for our experiments, SDP payload in Re-INVITE specified only two codecs, and that size can grow depending on the information described by SDP. In the same way, 200 OK messages sent in reply of an INVITE sized 580 bytes, while 200 OK sent for NewIP sized only 412 bytes. Moreover, Re-INVITE requires a final ACK, which in our experiments sized 410 bytes. The difference, 709 bytes (almost 83% greater), can potentially impact real-time multimedia communications, tough such verification will be left for future works.

SIP terminals with MobSIP support were implemented using Jain SIP programming library. The movement of wireless host h1 to a different logical network allowed the verification of MobSIP operations: NewIP messages were properly emulated in SIP messages of type MESSAGE, with no harm to the overall solution.

The practical verification of MobSIP was a second validation of the proposed solution. Future works will regard deeper comparison of efficiency among different mobility architectures and protocols, in others Internet logical and conceptual layers. Nevertheless, the application layer mobility support provided by MobSIP put that solution one step ahead when compared with most of the network-dependent mobile architectures. As verified by tests, it also presents itself as a better solution than standard SIP mobility support, for terminal mobility.

As wireless links have a potential higher error rate than wired links, it is recommended the using of multimedia codecs with packet loss tolerance, as iLBC (Andersen *et al*., 2004). This is an additional recommendation that will be considered in future implementations of SIP terminals with MobSIP extension.

5 Conclusion

The last years have seen the increase of real-time multimedia applications in Internet. As wireless networks are getting common, those applications tend to become mobile. Supporting this specific but growing group is a big challenge addressed by MobSIP.

SIP is currently the main protocol for controlling multimedia sessions, especially for videoconference and VoIP applications. Real-time communications over Internet backbones that use SIP have a potential advantage when compared with other signaling protocols. In such context, MobSIP presents itself as a good solution for real-time multimedia communications with mobility requirements: its not-centered end-to-end nature brings a potential better performance when compared with traditional solutions in this area. Further practical verification and simulations of MobSIP will be important works in this way.

This paper presents not a final stage. New specifications of MobSIP will regard multipoint communication and quality of service guaranties. Moreover, security will guide future specification of MobSIP, as handover procedures supported by NewIP messages could be forged by a malicious host. P2P communication will be also regarded, using new standard specifications as P2P SIP (Bryan *et al.*, 2008).

At last, new practical verifications will be focused on deeper measurement of efficiency, mainly in handover procedures. Time sensitive data transmission requires minimal delay in reception, even when IP addresses are dynamically changed.

References

ANDERSEN, S.; DURIC, A.; ASTROM, H. 2004. RFC 3951. Internet Low Bit Rate Codec (iLBC). Available at: http://www.ietf.org/rfc/rfc3951.txt. Access on: 23/08/2009.

BAUGHER, M.; MCGREW, D.; NASLUND, M.; CARRARA, E.; NORMAN, K. 2004. RFC 3711. The Secure Real-Time Transport Protocol (SRTP). Available at: http://www.ietf.org/rfc/rfc3711.txt. Access on: 21/08/2009.

BRYAN, D.; MATTHEW, P.; SHIM, E. 2008. Internet Draft. Concepts and Terminology for Peer to Peer SIP. Available at: http://tools.ietf.org/html/draft-ietf-p2psip-concepts-02. Access on: 23/08/2009.

CLARK, D.D.; WROCLAWSKI, J.; SOLLINS, K.R.; BRADEN, R. 2005. Tussle in Cyberspace: Defining Tomorrow's Internet. *IEEE/ACM transactions on networking*, **13**(3):462-475.

DARPA. 1981. RFC 793. Transmission Control Protocol. Available at: http://www.ietf.org/rfc/rfc793.txt. Access on: 22/08/2009.

DEERING, S. 1989. RFC 1112. Host Extensions for IP Multicasting. Available at: http://www.ietf.org/rfc/rfc1112.txt. Access on: 22/08/2009.

DROMS, R. 1997. RFC 2131. Dynamic Host Configuration Protocol. Available at: http://www.ietf.org/rfc/rfc2131.txt. Access on: 22/08/2009.

DUTTA, A.; MADHANI, S.; CHEN, W.; ALTINTAS, O.; SCHULZRINNE, H. 2004. Fast-handoff Schemes for Application Layer Mobility Management. *In:* IEEE INTERNATIONAL SYMPOSIUM ON PERSONAL, INDOOR AND MOBILE RADIO COMMUNICATIONS, 15, Barcelona, 2004. *Proceedings...* Barcelona, PIMRC 2004 p. 1527-1532.

FIELDING, R.; GETTYS, J.; MOGUL, J.; FRYSTYK, H.; MASINTER, L.; LEACH, P.; BERNERS-LEE, T. 1999. RFC 2616. Hypertext Transfer Protocol – HTTP/1.1. Available at: http://www.ietf.org/rfc/rfc2616.txt. Access on: 21/08/2009.

GAST, M.S. 2005. *802.11 Wireless Networks: The Definitive Guide*. 2nd ed., Sebastopol, O'Reilly, 632 p.

HALSALL, F. 2000. *Multimedia Communications: Applications, Networks, Protocols and Standards*. 1st ed., London, Addison Wesley, 1034 p.

HANDLEY, M.; JACOBSON, V. 1998. RFC 2327. Session Description Protocol. Available at: http://www.ietf.org/rfc/rfc2327.txt. Access on: 23/08/2009.

JAIN-SIP. 2009. Java API for SIP Signaling. Available at: https://jain-sip.dev.java.net/. Access on: 11/08/2009.

JOHNSON, D.; PERKINS, C.; ARKKO, J. 2004. RFC 3775. Mobility Support in IPv6" Available at: http://www.ietf.org/rfc/rfc3775.txt. Access on: 23/08/2009.

MOCKAPETRIS, P. 1987. RFC 1034. Domain names - concepts and facilities. Available at: http://www.ietf.org/rfc/rfc1034.txt. Access on: 25/08/2009.

MOSKOWITZ, R.; NIKANDER, P. 2006. RFC 4423. Host Identity Protocol (HIP) Architecture. Available at: http://www.ietf.org/rfc/rfc4423.txt. Access on: 21/08/2009.

NAYLOR, W.; OPDERBECK, H. 1974. RFC 619. Mean Round-Trip Times in the ARPANET. Available at http://www.ietf.org/rfc/rfc619.txt. Access on: 14/08/2009.

PERKINS, C. 2002. RFC 3344. IP Mobility Support for IPv4. Available at: http://www.ietf.org/rfc/rfc3344.txt. Access on: 22/08/2009.

POSTEL, J. 1980. RFC 768. User Datagram Protocol. Available at: http://www.ietf.org/rfc/rfc768.txt. Access on: 23/08/2009.

RATOLA, M. 2004. Which Layer for Mobility? Comparing Mobile IPv6, HIP and SCTP. *In*: *HUT T-110.551 SEMINAR ON INTERNETWORKING.* Available at: http://www.tml.tkk.fi/Studies/T-110.551/2004/papers/Ratola.pdf. Access on: 20/08/2009.

RESNICK, P. 2001. RFC 2822. Internet Message Format. Available at: http://www.ietf.org/rfc/rfc2822.txt. Access on: 24/08/2009

ROSEMBERG, J.; SCHULZRINNE, R. 2002. RFC 3261. Session Initiation Protocol. Available at: http://www.ietf.org/rfc/rfc3261.txt. Access on: 23/08/2009.

SCHILLER, J. 2003. *Mobile Communications*. 2nd ed., London, Addison Wesley, 492 p.

SCHULZRINNE, H.; CASNER, S. 2003. RFC 3550. RTP: A Transport Protocol for Real-Time Applications. Available at: http://www.ietf.org/rfc/rfc3550.txt. Access on: 26/08/2009.

SCHULZRINNE, H.; WEDLUNG, E. 2000. Application-layer Mobility Using SIP. *ACM SIGMOBILE Mobile Computing and Communications Review,* **4**(3):47-57.

SPURGEON, C. 2000. *Ethernet: The Definitive Guide*. Sebastopol, O'Reilly, 520 p.

SDL. 2009. SDL Forum Society. Available at http://www.sdl-forum.org. Access on: 15/08/2009

STEWART, R.; XIE, Q. 2007. RFC 4960. Stream Control Transmission Protocol. Available at: http://www.ietf.org/rfc/rfc4960.txt. Access on: 22/08/2009.

XIE, Q.; STEWART, R. 2007. RFC 5061. Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration. Available at: http://www.ietf.org/rfc/rfc5061.txt. Access on: 21/08/2009

WEDLUNG, E.; SCHULZRINNE, H. 1999. Mobility Support Using SIP. *In:* ACM/IEEE INTERNATIONAL CONFERENCE ON WIRELESS AND MOBILE MULTIMEDIA, WOWMOM, 2, Seattle, 1999. *Proceedings...* Seattle, p. 76-82.

Submitted on September 8, 2009.
Accepted on November 9, 2009.