SIMPLE (SIP for Instant Messaging and Presence Leveraging Extensions) Principals, concepts and Performance Evaluation

Imad MOURTAJI,
Computer and Communication Systems Laboratory, Abdelmalek Essaadi University
Tangier Morocco

Mohammed BOUHORMA,
Computer and Communication Systems Laboratory, Abdelmalek Essaadi University
Tangier Morocco

Abstract -- **The Session Initiation Protocol (SIP) is a signaling protocol for creating, modifying, and terminating sessions with one or more participants. These sessions include Internet telephone calls, multimedia distribution, and multimedia conferences. SIP is especially used for subscriptions and notifications of presence and Instant Messaging. Presence is defined as the willingness and ability of a user to communicate with other users on the network. Instant Messaging (IM) refers to the transfer of messages between users in real-time.**

In this paper, we will try to present an overview of SIP, and we will show how SIP is used to explore Presence and Instant Messaging services. An evaluation of performance will be presented to define differences between those services.

Key words: SIP, IM, Presence, Evaluation of performance.

I. INTRODUCTION

There are many applications of the Internet that require the creation and management of a session, where a session is considered an exchange of data between an association of participants. The implementation of these applications is complicated by the practices of participants: users may move between endpoints, they may be addressable by multiple names. Numerous protocols have been authored that carry various forms of real-time multimedia session data such as voice, video, or text messages. The Session Initiation Protocol (SIP) works in concert with these protocols by enabling Internet endpoints (called user agents) to discover one another and to agree on a characterization of a session they would like to share.

The IETF has produced many specifications related to Presence and Instant Messaging with the Session Initiation Protocol (SIP) [RFC3261]. Collectively, these specifications are known as SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE). These specifications cover topics ranging from protocols for subscription and publication to presence document formats to protocols for managing privacy preferences. The large number of specifications can make it hard to figure out exactly what SIMPLE is, what specifications cover it, what functionality it provides, and how these specifications relate to each other.

This paper is organized as follows. In the first section we'll give an overview of The Session Initiation Protocol (SIP). In the second section we'll present Presence and Instant Messaging and how SIP is used and in the last section we'll evaluate performances of Presence and Instant Messaging services. In the end, we'll achieve this paper with a conclusion, with future works and perspectives.

II. SIP OVERVIEW

SIP is an application-layer control protocol that handles the setup, modification, and tear-down of multimedia sessions. SIP is used in combination with other protocols to describe the session characteristics to potential session participants. SIP is based on a request and response transaction model similar to HTTP. Each transaction consists of a request that invokes a particular method or a function on the server and at least one response.

SIP is generally considered to be a client–server protocol. It has two classes of entities:

- **Client:** A client is an application program that sends a SIP request. The client can be a software program, such as Windows Messenger 5.0, or a hardware device, such as a SIP telephone.
- **Server:** A server generally responds to a request sent by a client. A server can be a software application, such as Live Communications Server 2003, or a hardware device.

SIP servers have different roles, such as:

- Proxy server: A SIP proxy works like an HTTP proxy server. When a client sends requests to the proxy, the proxy either handles them or forwards them to other servers.

- **Redirect server:** A SIP redirect server accepts a SIP request and conveys to the originating client the way to route the call.

- **Registrar server:** A SIP registrar server accepts registration requests and maps a client's address to a user's sign-in name, or SIP URI. Typically, a registrar is combined with a proxy or redirect server.

Session Establishment Example:

Figure 1 shows the SIP message exchange between two SIP-enabled devices. The two devices could be SIP phones, phone clients running on a laptop or PC (known as softclients), PDAs, or mobile phones. It is assumed that both devices are connected to an IP network such as the Internet and know each other's IP address.

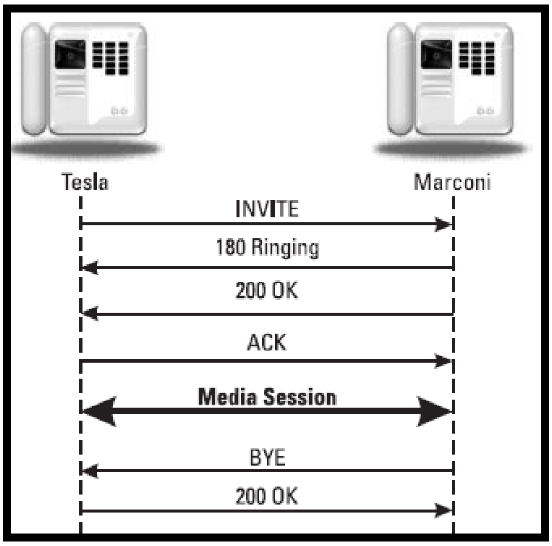

Figure 1 A simple SIP session establishment example

The process to establish a session starts with an INVITE message, which is sent from a calling user to a called user, inviting the called user to participate in a session. The caller might receive a number of interim responses before the called user accepts the call. For example, the caller might be informed that the called user is being alerted (the phone is ringing). When the called user answers the call, an OK response is generated and sent to the calling client. The calling client sends an ACK message after which media, such as voice, video, or text, is exchanged. After one of the users hangs up, a BYE message is generated and sent to the other client. The other client confirms that the session is over, after which the call ends.

The INVITE contains the details of the type of session or call that is requested. It could be a simple voice (audio) session, a multimedia session such as a videoconference, or a gaming session. The INVITE message contains the following fields:

INVITE sip:Marconi@radio.org SIP/2.0
Via: SIP/2.0/UDP lab.high-voltage.org:5060;branch=z9hG4bKfw19b
Max-Forwards: 70
To: G. Marconi <sip:Marconi@radio.org>
From: Nikola Tesla <sip:n.tesla@high-voltage.org>;tag=76341
Call-ID: j2qu348ek2328ws
CSeq: 1 INVITE
Subject: About That Power Outage...
Contact: <sip:n.tesla@lab.high-voltage.org>
Content-Type: application/sdp
Content-Length: 158
v=0
o=Tesla 2890844526 2890844526 IN IP4 lab.high-voltage.org
s=Phone Call
c=IN IP4 100.101.102.103
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000

Since SIP is a text-encoded protocol, this is actually what the SIP message would look like "on the wire" as a UDP datagram being transported over, for example, Ethernet. The fields listed in the INVITE message are called header fields. They have the form Header: Value CRLF.

The 180 Ringing is an example of a SIP response message. Responses are numerical and are classified by the first digit of the number. A 180 response is an informational class response, identified by the first digit being a 1. Informational responses are used to convey noncritical information about the progress of the call. Many SIP response codes were based on HTTP version 1.1 response codes with some extensions and additions. Anyone who has ever browsed the World Wide Web has likely received a "404 Not Found" response from a Web server when a requested page was not found. 404 Not Found is also a valid SIP client error class response in a request to an unknown user.

The 180 Ringing response has the following structure:

SIP/2.0 180 Ringing
Via: SIP/2.0/UDP lab.high-voltage.org:5060;branch=z9hG4bKfw19b ;received=100.101.102.103
To: G. Marconi <sip:marconi@radio.org>;tag=a53e42
From: Nikola Tesla <sip:n.tesla@high-voltage.org>;tag=76341
Call-ID: j2qu348ek2328ws
CSeq: 1 INVITE
Contact: <sip:marconi@tower.radio.org>
Content-length: 0

When the called party, Marconi, decides to accept the call (i.e., the phone is answered), a 200 OK response is sent. This response also indicates that the type of media session proposed by the caller is acceptable. The 200 OK is an example of a success class response. The 200 OK message body contains Marconi's media information:

SIP/2.0 200 OK
Via: SIP/2.0/UDP lab.high-voltage.org:5060;branch=z9hG4bKfw19b ;received=100.101.102.103
To: G. Marconi <sip:marconi@radio.org>;tag=a53e42
From: Nikola Tesla <sip:n.tesla@high-voltage.org>;tag=76341
Call-ID: j2qu348ek2328ws
CSeq: 1 INVITE
Contact: <sip:marconi@tower.radio.org>
Content-Type: application/sdp
Content-Length: 155
v=0
o=Marconi 2890844528 2890844528 IN IP4 tower.radio.org
s=Phone Call
c=IN IP4 200.201.202.203
t=0 0
m=audio 60000 RTP/AVP 0
a=rtpmap:0 PCMU/8000

III. PRESENCE AND INSTANT MESSAGING

Presence is the ability to sense the willingness of another user to communicate. Instant Messaging (IM) is a way of exchanging short text messages in near-real time. Presence is often used to determine when another user is available in order to start an instant message exchange. Often, messages are grouped together in a window and shown in sequential order, turning it into a conversation.

To address IM and interoperability, the IETF standardized two IM and presence protocols. One was a set of SIP extensions known as SIMPLE (SIP for Instant Messaging and Presence Leveraging Extensions) [3] and XMPP (Extensible Messaging and Presence Protocol) [4], which are based on the Jabber open source client. Today, both SIMPLE and XMPP are used to interconnect various closed IM systems. The Instant Messaging architecture is shown in figure 2 and its elements are in table 1. Presence architecture is shown in figure 3 and its elements are in table 2.

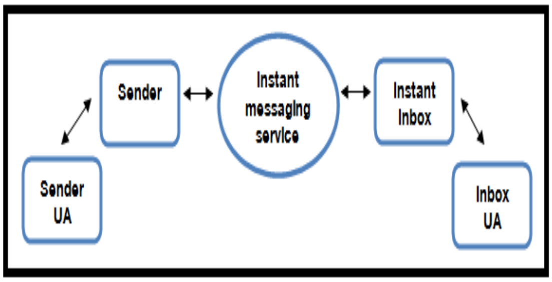

Figure 2 IM architecture

TABLE 1 INSTANT MESSAGING ELEMENTS

Instant Messaging Service	Protocol used to transport IM
Sender	Formats message for IM service
Instant inbox	Receives message from IM service
Sender user agent	User interface for gathering IM contents from user
Inbox user agent	User interface for rendering IM to user

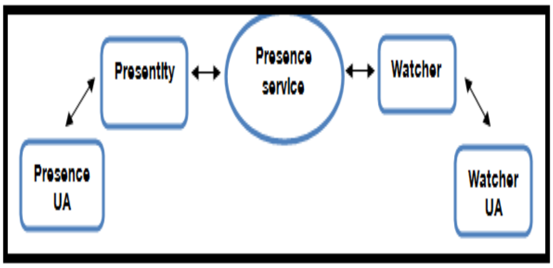

Figure 3 Presence Architecture

TABLE 2 PRESENCE ELEMENTS

Presence Service	Protocol used to transport presence information
Presentity	Publishes presence information to presence service
Presence user agent	User interface for gathering presence information about user
Watcher	Requests and receives presence information from presence service
Watcher User agent	Renders presence information received to the user

A. *Presence with SIMPLE*

The SIP events framework was defined in RFC3265 [5] which defined the SUBSRIBE and NOTIFY methods. SUBSCRIBE is used to establish a dialog and ongoing association between two UAs. In the Presence architecture of figure 4, the watcher send the SUBSCRIBE request to the presentity. If the subscription is authorized, the presentity will send NOTIFY wherever the state of the presentity changes, and at regular intervals. The basic call flow is shown in figure 4.

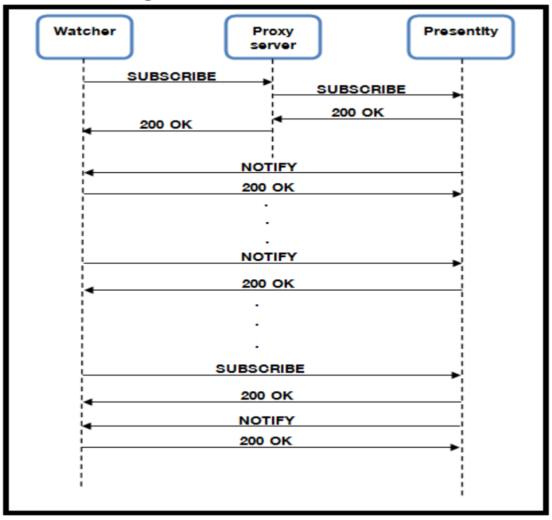

Figure 4 Example SUBSRIBE and NOTIFY call flow

An example SUBSCRIBE request is shown below:
SUBSCRIBE sip:ptolemy@rosettastone.org SIP/2.0
Via SIP/2.0/UDP proxy.elasticity.co.uk:5060
; branch=z9hG4bK34841123
Via SIP/2.0/UDP parlour.elasticity.co.uk:5060
; branch=z9hG4bKABDA ; received= 194.0.3.4
Max-Forwards: 69
To: sip:Ptolemy@rosettastone.org
From: Thomas Young <sip:tyoung@elasticity.co.uk>;
tag=1814
Call-ID: 452k59252058dkfj34924lk34
CSeq: 3412 SUBSCRIBE
Allow-Events: dialog
Contact: sip:tyoung@parlour.elasticity.co.uk
Event: dialog
Content-Length

The NOTIFY method is used by a user agent to convey information about the occurrence of a particular event. A NOTIFY is always sent within a dialog when a subscription exists between the subscriber and the notifer. An example NOTIFY request is shown here:
NOTIFY sip: UDP parlour.elasticity.co.uk SIP/2.0
Via SIP/2.0/UDP cartouche.rosettastone.org :5060
;branch=z9hG4bK3841323
Max-Frowards: 70
To: Thomas Young <sip:tyoung@elasticity.co.uk>;
tag=1814
Call-ID: 45k59252058dkfj349241k34
Cseq: 3 NOTIFY
Contact: sip:ptolemy@cartouche.rosettastone.org
Event: dialog
Subscription-State : active ; expires=180
Allow-Events : dialog
Content-Type : application/xml+dialog
Content-Length : ...
(XML Message Body not shown ...)

B. *Instant Messaging with SIMPLE*

Instant Messaging with SIP was a very early SIP extension in RFC 3428 [6]. In addition to this simple transport, SIP extensions for “iscomposing” or “istyping” have been standardized. The MESSAGE method is used to transport instant messaging (IM) using SIP. IM usually consists of short messages exchanged in near-real time by participants engaged in a text conversation. All UAs that support the MESSAGE method must support plan/text format; they may also support other formats such as message/cpim or text/html. A MESSAGE request normally receives a 200 OK response to indicate that the message has been delivered to the final destination. An IM response should not be sent in the message body of a 200 OK, but rather a separate MESSAGE request sent to the original sender. An example of MESSAGE call flow is shown in figure 5.

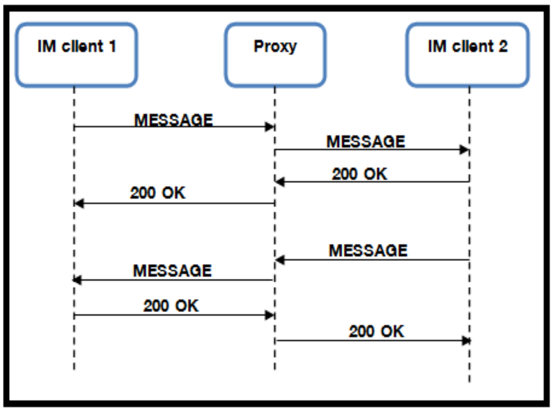

Figure 5 SIP MESSAGE call flow showing instant message transport

An example of MESSAGE request is shown here:
MESSAGE sip:editor@rcs.org SIP/2.0
Via SIP/2.0/UDP lab.mendeleev.org:5060
;branch=z9hG4bk3
Max-forwards: 70
To: editor@rcs.org
From: “ D. I. Mendeleev” dmitry@mendeleev.org
;tag=1865
Call-ID: 93847197172049343
CSeq: 5634 MESSAGE
Subject: First Row
Contact: <sip: dmitry@lab.mendeleev.org>
Content-Type: text/plain Content-Length: 7 H, He

IV. SIMULATIONS AND EVALUATION OF PERFORMANCES

Simulations scenarios were achieved using the network simulator NS2 [7]. The simulation area was 1000m by 1000m. The node number was between

10 and 50 nodes. The movement speed of nodes was between 0 and 18 m/s, and times of simulations were 180 seconds.

To define difference between utilization of Instant Messaging and Presence using VNSIP [8][9][10], we achieved many types of simulations, and we analyzed behaviors when node speeds and node numbers are increased.

A. Failure rates

The figure 6 shown here illustrates the failure rates of session setting according to nodes mobility for Instant Messaging and Presence services

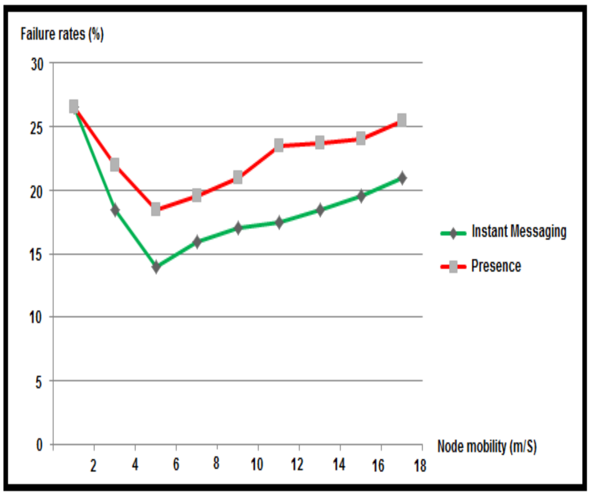

Figure 6 Failure rates by mobility of nodes for IM and Presence services

We observe good results of both services when mobility of nodes is not important. However the mobility is higher the failure rate increases. The figure 7 shown here illustrates the failure rates of session setting according to number of nodes for Instant Messaging and Presence services

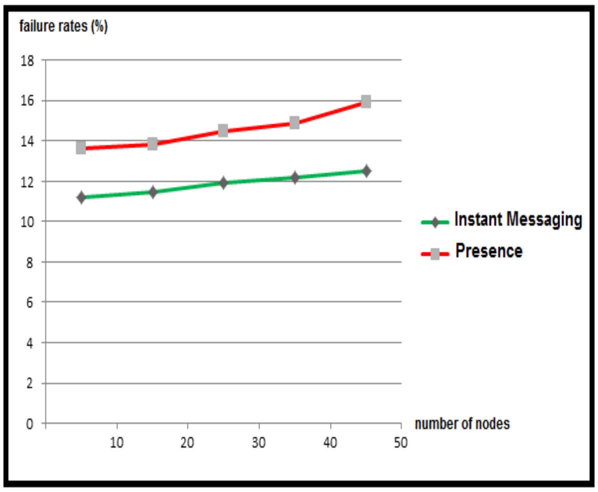

Figure 7 Failure rates by number of nodes for IM and Presence services

In this case, both services perform good results as we considered that nodes are immobile. When the number of nodes increases, the failure rate slightly increases also.

B. Bandwidth consumption

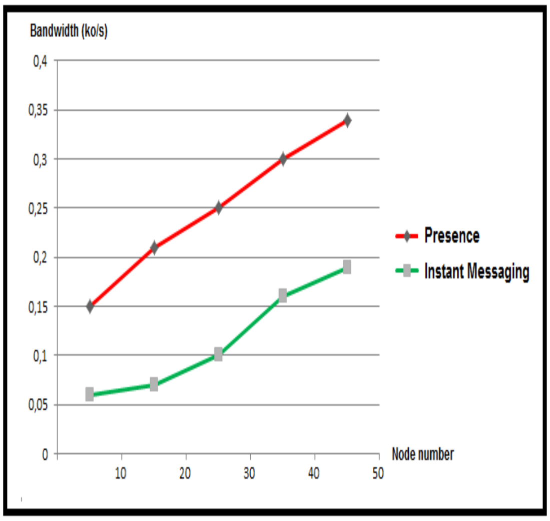

Figure 8 Bandwidth consumption by number of nodes for IM and Presence services

IM service presents better performances the Presence service in term of bandwidth consumption (see figure 8). We can explain this behavior that presence service uses two types of SIP request (SUBSCRIBE and NOTIFY), while IM service uses only MESSAGE request.

V. CONCLUSION

In this paper we have given an overview of SIP protocol, we have given an example of session establishment. Then we have shown how presence and instant messaging services can be implemented with SIP using a number of extensions. An Evaluation of performance was achieved to determine differences between Presence and IM regarding bandwidth consumption and failure rates.

REFERENCES

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handeley et E. Schooler, “SIP: Session Initiation Protocol”, RFC 3261, June 2002.

[2] Rosenberg, J., “SIMPLE made Simple: An Overview of the IETF Specifications for Instant Messaging and Presence Using the Session Initiation Protocol (SIP),” draft-ietf-simple-simple- 05 (work in progress), March 2009.

[3] Saint-Andre, P., (ed.), “Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence,” RFC 3921, October 2004.
[4] Roach, A., “Session Initiation Protocol (SIP)-Specific Event Notification,” RFC 3265, June 2002.

[5] Campbell, B., et al., “Session Initiation Protocol (SIP) Extension for Instant Messaging,” RFC 3428, December 2002.

[6] “The network simulator”, available at http://www.isi.edu/nsnam/ns

[7] J. Rosenberg and H. Schulzrinne, “SIP: Locating SIP Servers”, IETF RFC 3263, June 2002.

[8] I. Mourtaji, M Bouhorma, M. Benahmed, A Bouhdir “A new technique for adapting SIP protocol to Ad hoc Networks: VNSIP (Virtual Network for SIP) Illustration and Evaluation of performance” IJCNCS Volume 1, Issue 1, June 2013.

[9] I. Mourtaji, M Bouhorma, M. Benahmed, A Bouhdir “Performance Enhancement of VNSIP approach, using MCAC algorithm”
IJCNCS Volume 1, Issue 13, August 2013.

[10] I. Mourtaji, M Bouhorma, University Communication at Intelligent Ad Hoc Network using SIP services. IJCSI International Journal of Computer Science Issues, Vol. 11, Issue 1, No 2, January 2014