Contribution to Improving the Presence Base of VoIP Servers for Sending and Receiving Messages

Latyr Ndiaye(✉), Kéba Gueye, Samuel Ouya, and Gervais Mendy

LIRT Laboratory, Higher Polytechnic School, University of Dakar, Dakar, Senegal
{latyrndiaye, keba.gueye}@esp.sn, samuel.ouya@gmail.com, gervais.mendy@ucad.edu.sn

Abstract. The purpose of this article has been emphasized on the SIP Signaling Protocol in order to contribute to the improvement on sending and receiving message about the services provided by the VoIP servers. The current configuration of VoIP servers has allowed us to see that if a user connected by wifi to the VoIP server has disconnected involuntarily from the network without disconnecting his SIP client from the server, the server can not remove him from his base presence, where it stores all connected users, and cost the message that is sent to him is not stored and no longer he does not receive it.

To overcome this, we couple the Freeswitch intelligence used as a VoIP server coupled to a presence detection server and a MySQL database. This platform makes it possible to retrieve all messages with a non-connected recipient and store them in a database, then wait for their reconnection to send them the messages that concern them.

Keywords: Freeswitch · VoIP · SIP · MySQL · Wifi · Message

1 Introduction

Session Initiation Protocol (SIP) is an increasingly used protocol in the world of voice over IP. It is a signaling protocol used to open sessions in an IP environment, modify and close them. A session can simply be a telephone call (in reception and transmission) or a connection between several multimedia supports at the same time. The role of SIP is to open, modify, and release sessions. SIP, although widely used, must become even more mature. However, SIP-compatible VoIP products are becoming more numerous and more diverse. The choice of SIP in an IP environment comes from the fact that this protocol is extensible and easily integrates into different architectures. This protocol now has an important place in telecommunications. Specialists quickly became aware of its limitations and its need to interact with other protocols to be fully functional. The technological revolution noted in recent years in the telecommunications and networks sector allows researchers in the field to direct research in protocols such as the SIP protocol [1–3].

G. Mendy et al. (Eds.): AFRICOMM 2018, LNICST 275, pp. 56–66, 2019.
https://doi.org/10.1007/978-3-030-16042-5_6

This is why our research on the SIP protocol will allow us to detect flaws in it. These flaws noted compared to the connection by WIFI are a big problem for users and it is important to correct them. In this article, we will explain how this flaw was found and how we came up with a solution.

2 Motivation

A SIP study will allow us to understand the different methods, queries, and messages used. This will allow us to detect some flaws that this protocol is facing. This is why our article deals with the study of the SIP protocol and particularly with the MESSAGES events (sending and receiving messages). So we conducted experiments that allowed us to note some problems that the SIP protocol has. Different scenarios have been experimented and the results obtained have allowed us to see the problems that the SIP protocol faces. In the current SIP configuration, sending and receiving a message requires the direct connection of both clients to the SIP telephony server [5, 6].

2.1 IP-PBX

An IP-PBX is a professional telephone system designed to transmit voice or video over a data network and interact with the normal public switched telephone network (PSTN). The traditional PBX is based on the hardware circuit switch while the IP PBX is an IP telephony system that uses software switching [7, 8].

The system converges the voice and data backbone, simplifies network and service management, provides flexible/scalable solutions, and most importantly, many customizable service packages. There are many proprietary IP-PBX systems such as 3CX, Freeswitch, Kamailio (former OpenSER), IP-PBX Matrix, IP-PBX Magiclink, VoIP software, and so on [9]. However, it is still desirable to have an open source solution that will offer the freedom of customization if needed as well as keep the cost low.

2.2 SIP Signaling

Session Initiation Protocol (SIP) is a signaling protocol defined by the Internet Engineering Task Force (IETF) for establishing, releasing, and modifying multimedia sessions [4, 10]. SIP is used in VoIP PBXs and also in IMS as a signaling protocol for session control and service control. It therefore replaces both ISDN User Part (ISUP) and Intelligent Network Application Part (INAP) protocols in the world of telephony by providing multimedia capability [11]. SIP is the unifying protocol of VoIP architectures. Its use involves associated protocols, especially SDP, for session description, and RTP/RTCP, for real-time transport (and control) of multimedia data streams.

At the PABX level, SIP is used by network devices such as IMS (CSCF in particular) to control the call (establishment, modification, end) during any multimedia session. A session is established when two or more participants exchange data. SIP can handle multi-recipient, group or automatic calls. In any type of network, there are always four types of signaling:

Registration signalling: by it, a terminal registers in the network. It contains the procedures for downloading the profile and managing the location. It is performed by the SIP registration procedure (SIP REGISTER) (Fig. 1).

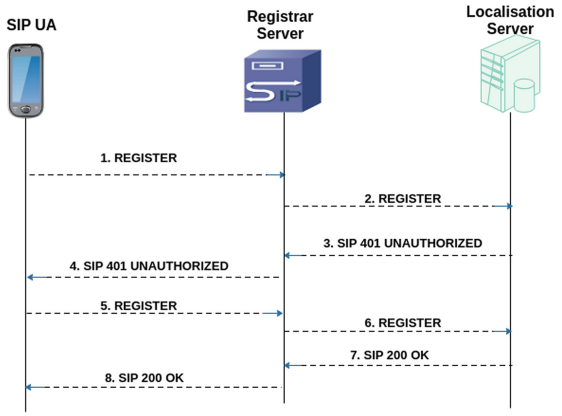

Fig. 1. REGISTER SIP registration procedure

Call signaling: it establishes an end-to-end association between the endpoints wishing to communicate. This type of signaling is characterized by reference exchange. This is done in IMS and PBXs through the SIP INVITE procedure (Fig. 2).

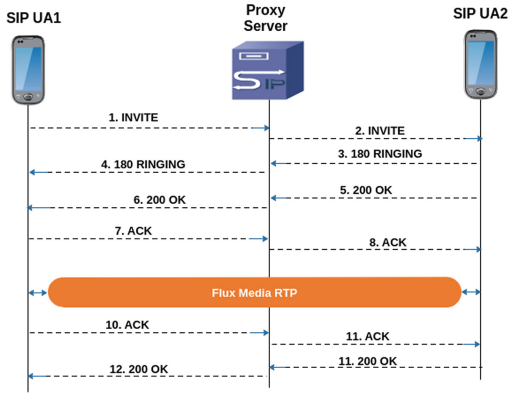

Fig. 2. INVITE session setup procedure

Connection signaling: this is the assignment of a support service to a call. Gradually we will reserve resources in the network according to the QoS required for the service. In SIP, this signaling is done through SDP headers that describe the traffic and

resources required. At the transport level, the RSVP (Resource Reservation Protocol) and DiffServ (Differentiated Services) mechanisms are used to ensure the quality of service in the IP network.

Intelligence signaling: it allows substitution treatment compared to normal call processing. In a similar way to intelligent networks of RI (INAP) or CAMEL type, services are executed by the equivalent of service platforms that are application servers (AS) [11].

3 Studies of SIP Protocol Vulnerabilities

In the current SIP configuration, sending and receiving a message requires the direct connection of both clients to the SIP server. The architecture in the following implements this configuration of a SIP server (Fig. 3).

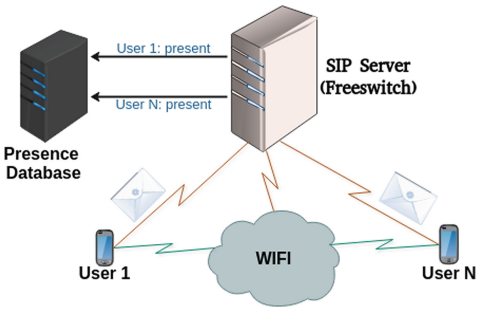

Fig. 3. Initial architecture of a SIP server

In this architecture, User 1 and User N are all connected to the SIP telephony server. It has a presence database where it will store all users connected. Once a user connects, the server detects its connection and records it at the presence database so it has information about its presence, that is, REGISTER events. When the connected User 1 sends a message to the User N, the server consults its presence database to obtain the information on the User N. If it sees that it is present at the presence base, it retransmits it. the message that is intended for him. Otherwise, if the User N is absent at the presence database, the message intended for him is lost. The following figures show illustrations of faults noted on users' attendance servers in VoIP PBXs.

3.1 Sending Message Between User: Both Connected (User 1000 and User 1001)

To do the tests, we used the CSipSimple SIP client (Fig. 4).

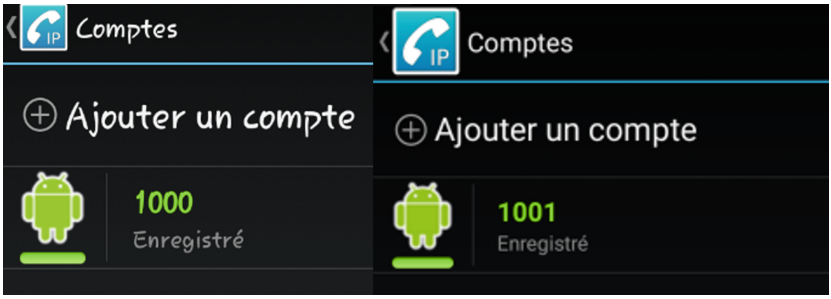

Fig. 4. Registration of user 1000 and 1001

The user 1000 sends a message to the connected user 1001 (Fig. 5).

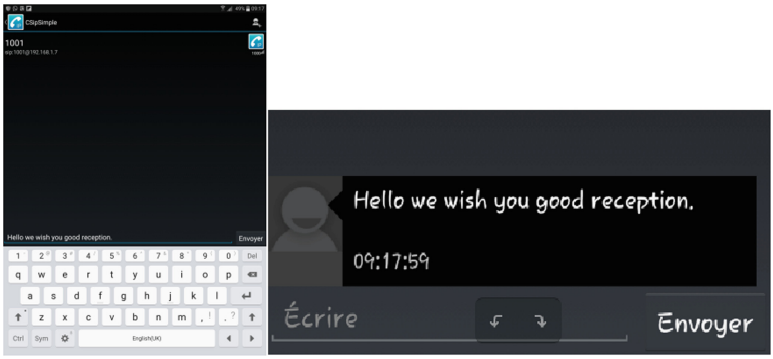

Fig. 5. Sending message from user 1000 to user 1001 and Receipt of the message by the user 1001

For this initial configuration of the SIP server, if all users are connected to the SIP server, the sent messages always arrive at their destination as we can see for the message sent by the user 1000 to the user 1001. This example is the case where both users are connected, we will now see the case where one of the users is not connected.

3.2 Sending Message Between User: One Connected (1000) and the Other Disconnected (1001)

The user 1000 sends a message to the user 1001 not connected (Figs. 6 and 7).

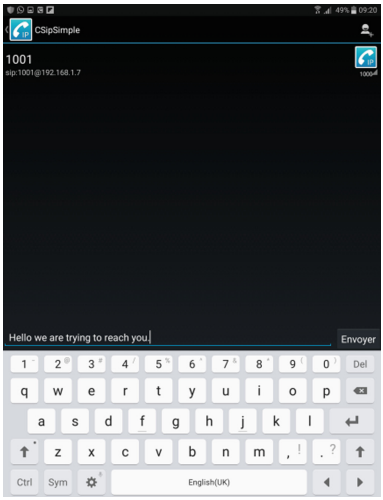

Fig. 6. Send message from the user 1000 to the user 1001 not connected

```
2018-06-13 09:21:50.876536 [ERR] sofia_presence.c:272 Chat proto [sip]
from [<sip:1000@192.168.1.7>;tag=vrsoUgywr48sO7etKy9cCi1.eul3r9OZ]
to [1001@192.168.1.7]
Hello we are trying to reach you.
Nobody to send to: Profile internal
```

Fig. 7. Informations from the server

After sending this message, we will consult our server and this thanks to its console, we observe the error messages.

Figure 8 shows the messages at the log level of our server and that the message sent to the user 1001 has not arrived at the destination. We note that this message sent without connection of the User N has not arrived at its destination.

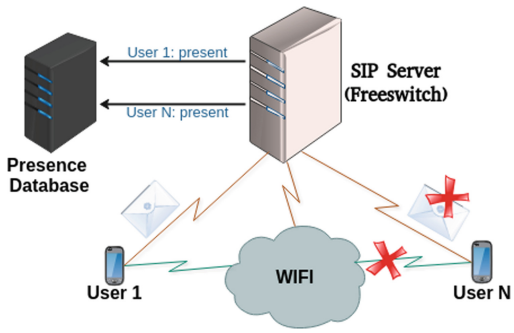

Fig. 8. Architecture on the noted connection problem

It is found that another problem arises, if the user disables its WIFI indicator, the server does not have disconnection information to be able to properly disconnect the user from the presence database. From the cost, he continues to see it as this one was still connected to the server. The figure below implements the enumerated phenomenon.

The User N decides to disable his WIFI light and as we see the server continues to register as present at its base of presence. The message sent by User 1 to User N did not arrive at destination. This is simply because the server still has User N registration information in its presence database, so it tries to send the message when it is not actually connected. From the cost this message will not be received by the User N.

The following figures present the different steps that led to this study. In this figure, both users are connected to the telephony server. They are present at the level of the presence base (Fig. 9).

	call id	sip user	sip host	presence hosts	contact	st
1	QWN-Q4PPA755	1000	192.168.1.7	192.168.1.7,192	"" <sip:1000@1	Re
2	c7gHjeB2EQsXO	1001	192.168.1.7	192.168.1.7,192	"" <sip:1001@1	Re

Fig. 9. The presence base of the telephony server

The experiment implemented is that if the user 1001 decides to disable only are WIFI LED. If we update our presence base, we continue to have its presence at the level of it. The user 1000 will send him a message. We notice at the level of the logs of our SIP server that the sent message does not reach its destination. The following figure shows the message sent by the user 1000 to the user 1001 (Fig. 10).

```
2018-06-13 09:42:32.355696 [INFO] switch_cpp.cpp:1365 *******   Evenement Message    *********
2018-06-13 09:42:32.355696 [INFO] switch_cpp.cpp:1365 ipv4 : 192.168.1.7
2018-06-13 09:42:32.355696 [INFO] switch_cpp.cpp:1365 sender : 1000
2018-06-13 09:42:32.355696 [INFO] switch_cpp.cpp:1365 recipient : 1001
2018-06-13 09:42:32.355696 [INFO] switch_cpp.cpp:1365 message : Hello you're there?
```

Fig. 10. Message sent to 1001 which disables its WIFI

We notice that the message arrives at the server but the user will never receive it. To overcome this, we have proposed a solution that is described in detail with the architecture that has been implemented in the next section. This leads us to do a thorough study of SIP to provide a solution to this kind of problem.

4 Implementation

To implement this solution, we have installed a SIP freeswitch server that we have exploited by sending requests and thanks to its ESL module (Event Socket Library), we manage to listen on MESSAGES type events. This listening of events allows us to detect the various faults noted in the previous section.

To solve the flaws, we developed a script in python and adapted to SIP servers as freeswitch and asterisk, we also set up a MySQL database that will be used to store messages whose owners are not registered at the SIP server. In this script, we listen to the MESSAGES events of freeswitch so that the flaws we noted relate to message reception problems in case of bad connection. This script listen infinitely loop on these events, it allows us to recover any event that arrives on our telephony server in order to exploit it, either we send it to the recipient or it is to store in a MySQL database.

This database also allows us to have the traces of the different messages exchanged. As soon as the recipient who was absent reconnects, our script receives the event reconnection message, it checks in the database if the user in question has no messages store in absence. If this is the case then the script retrieves the message (s) from the database and reformats it to send it to its recipient. If this is not the case, the script notifies in the SIP server console that the user does not have a missed message. The proposed architecture is given in the following figure (Fig. 11).

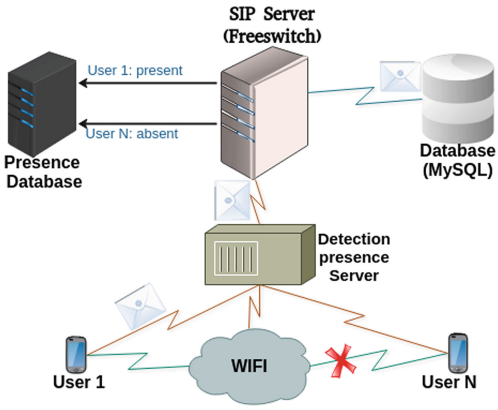

Fig. 11. Architecture of the proposed solution

5 Results and Discussions

Following the problem identified earlier, we will propose a solution that will allow us to overcome this. The architecture below shows how the solutions have been implemented and a description of this solution will be provided. In this architecture, there is the presence of another server that is called here "presence detection server". This server works as follows:

It sends a ping every five seconds to users connected to the SIP server.

- If the User N, who was previously connected to the server, decides to disable his WIFI LED, automatically the presence detection server by sending him a Ping gets as UNREGISTERED answer.
- He then contacts the SIP server to notify him of the disconnection of the User in question.

The SIP server consults its presence base, which always sees the User as logged in, and informs him of the disconnection of the latter by asking him to update the new information. User information is therefore updated, and is therefore absent from the presence database. If a message is sent to this User N, then the presence detection server retrieves the message, retransmits it to the SIP server and it will simply store the message in the MySQL database while waiting for the User to reconnect. N. The following figure shows that both user 1000 and 1001 are well connected to the SIP server and a ping is sent to them at every moment to see their presence detected (Fig. 12).

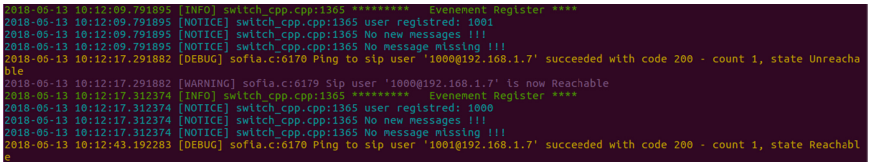

Fig. 12. User 1000 and 1001 connection

We note for our two users the messages succeeded Ping sent. Now if the user 1000 disables its WIFI simply, our Ping allows us to know if it is well connected or not. The following figure shows the return messages after the ping sent to the user 1000 which disables his WIFI (Fig. 13).

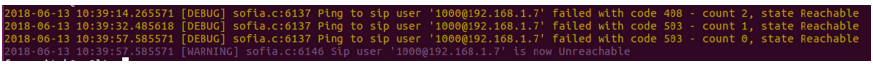

Fig. 13. Return message after ping to 1000

The user 1000 is removed from the server's presence database. This allows us to be able to know if a user is connected or not in order to be able to take measurements in relation to the messages he receives. If the user 1001 sends a message to the user 1000, the message is retrieved and stored in the database. MySQL data. The following figure shows the message sent to the user 1000 (Fig. 14).

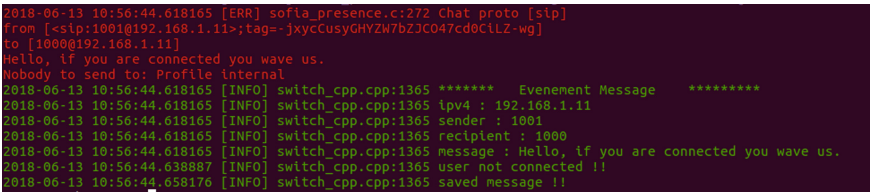

Fig. 14. Message sent to the user 1000 after our modification.

We note that the message is well saved in the database to wait for the good connection of the user 1000. The following Fig. 16 shows the reconnection of the user 1000, the message that was stored is sent to him (Fig. 15).

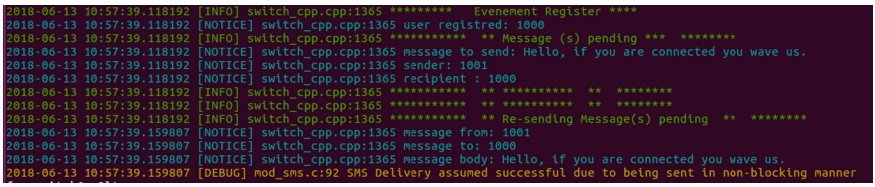

Fig. 15. Message sent after reconnecting the user 1000

One faith the user in question reconnects, he receives the message (s) which were stored in the database and which had for recipient him.

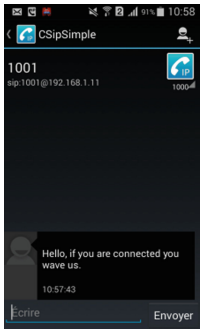

Fig. 16. Message sent after reconnecting the user 1000

6 Conclusion

The global study of the SIP protocol concerning the sending and the receiving of the SIP messages made it possible to note the problems of reception of the message sent in case the recipient is not connected. We found that this message is lost and never reaches its destination in this case. This resulted in the solution we propose in this article. A solution we tested with the SIP server freeswitch but it remains adapted to other SIP servers as an asterisk. So today we can say that this solution can greatly contribute to the improvement of SIP message sending and reception conditions and therefore to a perfect reliability of SIP MESSAGES events, which is for us an advantage in a context where we tend towards IP at all with the arrival of the fourth generation (4G) and the prospects of a fifth generation (5G).

References

1. Abbasi, T., Prasad, S., Seddigh, N., Lambadaris, I.: A comparative study of the SIP and IAX VoIP protocols. In: Canadian Conference on Electrical and Computer Engineering, Saskatoon, Sask, pp. 179–183 (2005). https://doi.org/10.1109/ccece.2005.1556904
2. Saliha, M., Amina, Z., Chafia, Y.: A new authentication and key agreement protocol for SIP in IMS. In: 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA), Marrakech, pp. 1–7 (2015). https://doi.org/10.1109/aiccsa.2015.7507136
3. Yang, L., Lei, K.: Combining ICE and SIP protocol for NAT traversal in new classification standard. In: 2016 5th International Conference on Computer Science and Network Technology (ICCSNT), Changchun, pp. 576–580 (2016). https://doi.org/10.1109/iccsnt.2016.8070224
4. Rosenberg, J., et al.: SIP: Session Initiation Protocol, IETF RFC3261, June 2002
5. Yu, L.: Improving query for P2P SIP VoIP. In: 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, Liverpool, pp. 1735–1740 (2012). https://doi.org/10.1109/trustcom.2012.183
6. Hosseinpour, M., Seno, S.A.H., Moghaddam, M.H.Y., Roshkhari, H.K.: Modeling SIP normal traffic to detect and prevent SIP-VoIP flooding attacks using fuzzy logic. In: 2016 6th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, pp. 274–279 (2016). https://doi.org/10.1109/iccke.2016.7802152
7. Hersent, O., Gurle, D., Petit, J.P.: IP Telephony Packet-Based Multimedia Communication Systems. Wiley, Hoboken (2005)
8. Johnston, A.B.: SIP: Understanding the Session Initiation Protocol. Artech House, Boston (2007)
9. List_of_SIP_software#Proprietary_license (n.d.). Wikipedia. http://wikipedia.org/wiki/List_of_SIP_software#Proprietary_license
10. Ansari, A.M., Nehal, M.F., Qadeer, M.A.: SIP-based interactive voice response system using FreeSwitch EPBX. In: 2013 Tenth International Conference on Wireless and Optical Communications Networks (WOCN), Bhopal, pp. 1–5 (2013). https://doi.org/10.1109/wocn.2013.6616224
11. Imène ELLOUMI: Management of mobility between access networks and Quality of Service in an NGN/IMS approach. Thèse doctorat: University of Carthage (Tunisie) (2012)