SIP SIGNALING IN IP TELEPHONY

Ms. Deepali H. Shah
Instrumentation & Control Department,
L. D. College of Engineering,
Gujarat University, Navrangpura,
Ahmedabad-380015,Gujarat, India,
tvshah2000@yahoo.com

Dr. B. V. Buddhadev
Computer Department,
L. D. College of Engineering,
Gujarat University, Navrangpura,
Ahmedabad-380015, Gujarat, India
bvbld@yahoo.com

ABSTRACT

Internet telephony offers the opportunity to design a global multimedia communications system that may eventually replace the existing telephony infrastructure. SIP is developed for signalling Internet telephony services. This paper highlights role of SIP in IP telephony system as well as the design and implementation aspects of a telephony system based on SIP. Developing a fully functional system requires to set up a server based on Asterisk, connecting clients to that server with the help of soft phones and configuring the soft phones with the server. In our implementation, we have connected the clients to the server with the help of SIP protocols. Soft phones are used as VoIP clients which uses SIP as primary communication protocols to connect to Asterisk PBX Server. The introductory part of the paper contains brief description of IP Telephony. Detailed description of SIP is covered. In the later part of the paper, we discuss about the configurations required for the communication and finally design and implementation aspects of the telephony system.

***Keywords* -** VoIP, SIP, Dialplan, IP Phone, Soft phone, UA, UAC, proxy server

1. INTRODUCTION

Today most of the telephony is still made on the old Public Switched Telephone Network (PSTN). This means that a call reserves the connection between the two users and no one else can use this connection. When the call is disconnected the line is free for other users again.

The difference with Internet telephony, also known as Voice-over-IP (VoIP) or IP telephony (IPtel), is that the transport is made on an IP-network. Here it is possible to send packets between two or more parties without reserving the connection. This is done by digitizing the audio signals, recorded by using the microphone of the computer, encapsulating them into packets and then sending them across networks using the Internet protocols. The other side then decapsulates the packets and plays the original message through the speakers of the computer. Other media, such as video and shared applications, are also possible to include without any major changes.

The negative side with this new method is that it is difficult to guarantee any Quality of Service (QoS), as there is no way to guarantee when a packet arrives. This problem could be minimized by not using the public Internet and use private managed networks instead, where a certain QoS can be guaranteed.

The need for signalling functionality distinguishes Internet telephony from other Internet multimedia services such as broadcast and media-on-demand services. This signalling has to be able to create and manage calls. One of the problems in Internet telephony is locating participants for a phone call. Personal mobility, call delegation, availability, and desire to communicate make the process of signalling more complex. For this, it is possible to use the Session Initiation Protocol (SIP) that is part of the protocol stack that has emerged from Internet Engineering Task Force (IETF) [5] – the Internet Multimedia Conferencing Architecture. The SIP translates application-layer addresses, establishes and manages calls. Another protocol in use that emerged from the International Telecommunication Union (ITU) [6] is the H.323, which is similar to the SIP.

2. SESSION INITIATION PROTOCOL

The Session Initiation Protocol is actually an application layer protocol for creating, modifying and terminating sessions. These sessions can be multimedia conferences, Internet telephone calls and similar application consisting of one or more media types as audio, video, whiteboard etc. SIP invitations, used to create sessions, carry session descriptions, which allow participants to agree on a set of compatible media types. Participants can be a human user, a “robot” (e.g. media server) or a gateway to another network. These can communicate via multicast or via a mesh of unicast relations, or a combination of these.

The session can range from just a two-party phone call to a multi-user, multimedia conference or an interactive gaming session. SIP does not define the type of session, only its management. To do this, SIP performs four basic tasks [2]:

- Locating users, resolving their SIP address to an IP address

- Negotiating capabilities and features among all the session participants
- Changing the session parameters during the call
- Managing the setup and teardown of calls for all users in the session

SIP is built on a client-server model, using requests and responses that are similar to Internet applications. SIP usually uses User Datagram Protocol (UDP) as its transport protocol, but it can also use TCP. The default SIP port for either TCP or UDP is 5060. It uses the same address format as e-mail, with a unique user identifier and a domain identifier. A typical SIP address looks like one of the following [2]:

sip:1112223344@mycompany.com
sip:1112223344@10.1.1.1

This allows Domain Name System (DNS) to be used to locate users, and it also allows SIP to integrate easily with e-mail. SIP uses Multipurpose Internet Mail Extension (MIME) to describe the contents of its messages. Thus, SIP messages can contain information other than audio, such as graphics, billing data, authentication tokens, or video. Session Description Protocol (SDP) is used to exchange session capabilities and features.

3. SIP FUNCTIONAL COMPONENTS

SIP endpoints are called user agents (UA) and can be various devices, including IP phones, cell phones, PDAs, Cisco routers, or computers running a SIP-based application. UAs can act as either clients or servers. The user agent client (UAC) is the device that is initiating a call, and the user agent server (UAS) is the device that is receiving the call. The SIP protocol defines several other functional components. These functional entities can be implemented as separate devices, or the same device can perform multiple functions [1].

A. ***Proxy server***

This server can perform call routing, authentication, authorization, address resolution, and loop detection. A UA sends its call setup messages through a proxy server. The proxy server can forward the messages if it knows where the called party is located, or it can query other servers to find that information.

B. ***Redirect server***

UAs and proxy servers can contact a redirect server to find the location of an endpoint. This is particularly useful in a network that has mobile users whose location changes.

C. ***Registrar server***

UAs register their location with a registrar server, which places that information into a location database.

D. ***Location server***

This server maintains the location database for registered UAs.

E. ***Back-to-back user agent (B2BUA)***

This server acts as a UA server and client at the same time. It terminates the signalling from the calling UA and then initiates signalling to the called UA.

F. ***Presence server***

This server gathers presence information from Presentities and subscription information from Watchers, and sends status notifications.

All these functions work together to accomplish the goal of establishing and managing a session between two UAs.

4. SIP CALL FLOW

Basic SIP session setup involves a SIP UA client sending a request to the SIP URL of the called endpoint (UAS), inviting it to a session. If the UAC knows the IP address of the UAS, it can send the request. Otherwise, the UAC sends the request to a proxy or redirect server to locate the user. That server might forward the request to other servers until the user is located. After the SIP address is resolved to an IP address, the request is sent to the UAS. If the user takes the call, capabilities are negotiated and the call commences. If the user does not take the call, it can be forwarded to voice mail or another number. Various scenarios are as following [1]:

- Call flow between two SIP Gateways
- Call flow using Proxy Server
- Call Flow Using Multiple Servers

5. SIP PROS AND CONS

A. ***Pros***

- SIP works independently of the type of session, or the media used, giving it flexibility.
- It is an open standard, allowing multivendor support and integration. Applications can be written to customize SIP uses.
- SIP messages are clear text, making troubleshooting easier.
- SIP can accommodate multiple users with differing capabilities.

B. ***Cons***

- Processing text messages puts a higher load on gateways. The router must translate that text into a language that the router can understand.
- SIP is a fairly new protocol, so fewer people understand it than the older protocols.
- SIP features are still being developed, and many vendors have proprietary implementations of the protocol.

6. SIP CONFIGURATION

In our experimental setup, we have connected two soft phones.

or communication, users are to be created. In sip.conf file, we can configure everything related with the SIP protocol; add new sip users or define sip providers. The sip.conf is configured as shown below [3], [4].

```
[1111]
type = friend
username = 1111
secret = 1234
disallow = all
allow = alaw
allow = ulaw
dtmfmode = rfc2833
qualify = 1000
host = dynamic
context = test
```

context=test specifies the location of the instructions used to control what the phone is allowed to do, and what to do with incoming calls for this extension. The context name configured in sip.conf matches the name of the context in extensions.conf, which contains the instructions.

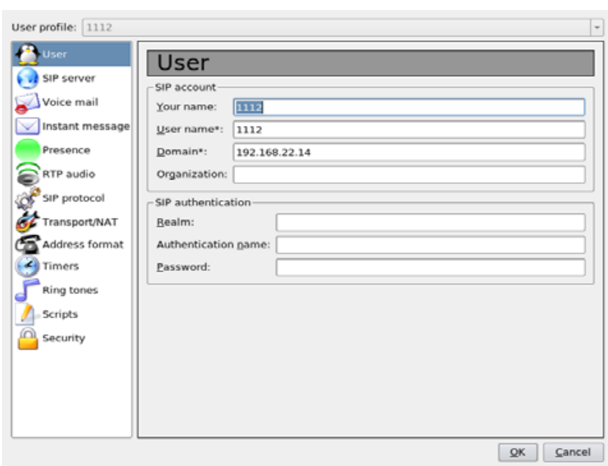
A. Dial plan Configuration

The configuration file "extensions.conf" contains the "dial plan" of Asterisk. It controls strategies for handling and routing of incoming and outgoing calls.

The corresponding dial plan should be written in extensions.conf.

exten => extension no, priority, application (argl, arg2 …)

```
[test]
exten => _[0-9].,1,Dial(Sip/${EXTEN})
```



B. Client Configuration

We've chosen to use twinkle client.

Fig. 1 User profile

Twinkle is a softphone for voice over IP and instant messaging communications using the SIP protocol. We can use it for direct softphone to softphone communication or in a network using a SIP proxy to route the calls and messages.

To use Twinkle, we have created a user profile. It contains the data for SIP account, like username, password, SIP proxy address and more. Multiple user profiles for different accounts can be created.

The network settings are finalized in the SIP server [4]. SIP and RTP (Real-time transport Protocol) ports used for inbound and outbound calls are specified in system settings.

We configured SIP phones and registered them with the server. Once the phones are registered, we can make calls. Fig. 2 shows the call established between users 1111 and 1112.

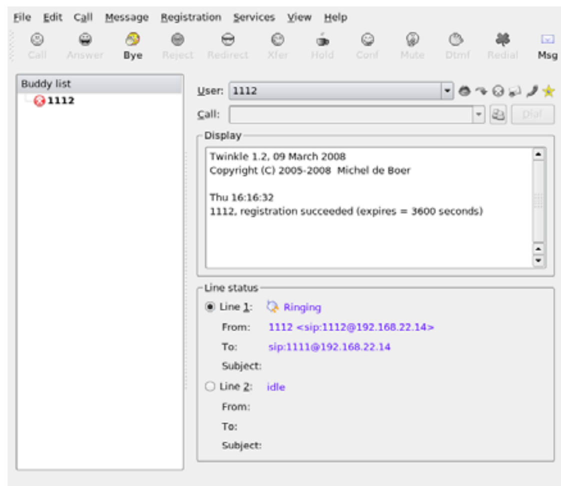


Fig. 2 Call established

By entering "SIP show peers" in the Asterisk CLI gives a list of all the configured and registered SIP phones.

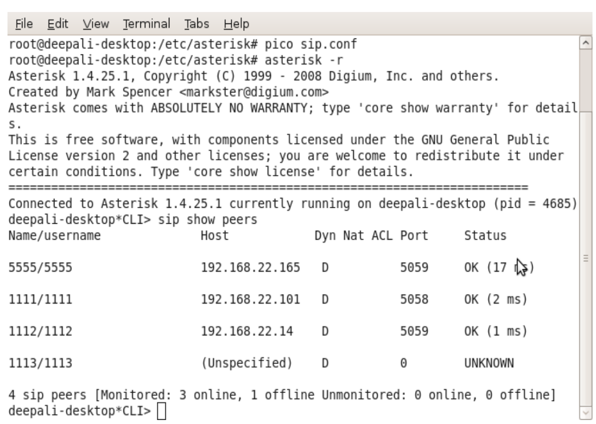


Fig. 3 peer hosts

6. CONCLUSION

SIP integrates many diverse applications, devices, and communications processes to deliver innovative multidimensional capabilities and features. This comes along with potential cost effectiveness in hardware and communication options. Many companies have started foraying into the field of communication to tap the advantages of SIP. Due to these advantages SIP is bound to become the powerful and most common communication choice in the future. We expect that design and implementation presented in this paper will be a valuable developing guide for similar kind of operations. VoIP is not only cost effective but also provides us various features which we generally don't get with the conventional circuit switched based PBX. Moreover, the system has provision for unlimited expansion. Since the system runs on a secure operating system like LINUX, it's much less prone to viruses, worms and hackers.

REFERENCES

[1] Sip: Protocol overview. http://www.radvision.com

[2] Internet Telephony: Architecture and Protocols an IETF Perspective, Henning Schulzrinne Jonathan Rosenberg

[3] Jim Van Meggelen, Leif Madsen, and Jared Smith, *Asterisk™: The Future of Telephony*, First Edition",2005

[4] Asterisk the Open Source PBX, http://www.asterisk.org

[5] International Engineering Task Force, IETF. http//:www.ietf.org

[6] International Telecommunications Union, ITU. http//:www.itu.int

BIOGRAPHY

Deepali Shah (tvshah2000@yahoo.com) received her Bachelor of Engineering (Instrumentation & control engg.) degree in 1997 from Gujarat University. She worked as project engineer in Solmac pvt. Ltd. for 7 months. She worked as lecturer in Computer engineering department from 2000 to 2004 in L. D. College of Engineering, Ahmedabad, Gujarat. At present, she is working as lecturer in Instrumentation and control engineering in L. D. College of Engineering. Her area of interests is embedded system, IP telephony, VLSI, control system, microprocessor, digital systems.

B. V. Buddhadev (bvbld@yahoo.com) received his Bachelor of Engineering (Electronics and communication engg.) degree from Gujarat University in 1980. He received MTECH(Computer and Science) from IIT, Delhi in 1995. His industrial experience is of 5 years. He is working in the L. D. College of Engg., Ahmedabd, Gujarat since 24 years. At present, he is professor and head of PG course in Computer Department of the same college. His area of interests is distributed computing system, advanced web technology, object oriented and software engg. With math practices, J2EE, java, web services, .net technology & sevice oriented architecture, distributed database. Under his guidance, 4 phDs are completed and at present, he is guiding to 4 Ph.D students. He is member of AICTE counselling, NBA & State purchasing committee.