Advances in Computational Sciences and Technology
ISSN 0973-6107 Volume 10, Number 8 (2017) pp. 2507-2517

http://www.ripublication.com

Call recording solution using SIP

Shrikant Subhash Warghade
PG Scholar, VIT, Vellore, India.
Intern at ShoreTel,

Karthikeyan B
Associate Professor, VIT, Vellore, India.

Suraj Rodrigues
Manager Software Engineer, ShoreTel, India.

Nagaraj Hebbar
Senior Tech Lead Software Engineer, ShoreTel, India.

Abstract

Voice over Internet Protocol (VoIP) has launched 20 years ago. After its launching, VoIP has become one of the most popular and powerful technologies of the 20th and 21st century [9]. With millions of users from business phones to social networking apps, VoIP has become an underlying technology that has power the way we people connect to each other. Developing at 6% compound annual growth rate (CAGR), VoIP is expected to have a total market of $82.7 billion by 2017 [9]. On top of this, as indicated by the site Telecom Reseller, VoIP is one of the top performing businesses of this decade, alongside biotechnology and e-commerce. Whereas the wired communication is one of the worst.

In this paper call recording solution is developed by using the latest technology such as SIP. This solution is scalable and can be implemented by telecom service provider companies which works on VoIP.

Keywords: VoIP(Voice over Internet Protocol), Session Initiation Protocol(SIP), SIP recording (SIPREC), Real time Transport Protocol(RTP), Communication Session(CS)

I. INTRODUCTION

In some countries call recording is mandatory for voice analytic, security reasons and many more. Some countries record all the calls and keep them verifying for security reasons. For a telecommunication service provider company who have customers in such countries as well having its customers like call centers, call recording solution is very important for them. The call recording provides some benefits which are as follows [12] :-

1. Quality Assurance :- When maintaining a business or a call center, couple of things are more important than performance and quality with regards to speaking with clients [12]. Call recording permits you to log and survey calls to hear how your employees are interfacing with clients, guaranteeing they are giving the correct data in a most effective way.

2. Enriched Training :- If the organization have an excellent performing team, then their call recording can be used to train the new hire or low performing employees [12]. This can help to improve the feedback sessions which ultimately provide more targeted and focused feedback to employee.

3. Dispute Protection :- It has been seen that in any business problems arise with customers or clients at some point [12].These dispute happen with small miscommunication and later turn into much serious issue. Call recording provides a medium to your employee to back and review calls, to check what was said.

Along with this there are many other benefits like improved employee productivity, customer understanding, regulatory compliance, increased customer satisfaction, etc [12]. So in this project as a proof of concept call recording solution is developed with the latest technology using SIP.

II. LITERATURE SURVEY

In this project Session Initiation Protocol(SIP) is used. Other protocols like Session Description Protocol(SDP), User Datagram Protocol (UDP) [13], Transmission Control Protocol(TCP), Real-time Transport Protocol(RTP) are also used. SIPP which is a free open source tool for SIP protocol, X-Lite [1], PhonerLite [5] are the softphones are used. Wireshark which is a free open source packet analyzer is also used for capturing the packets [16]. Inter-process Communication like socket based communication for TCP, UDP is studied [7]. SIPREC Client can be used to record the media sessions to the recording devices [3] [8]. A Session Border Controller (SBC) which is a dedicated software application or hardware device governs the manner in which phone calls are initiated, carried out and terminated on a VoIP [10] is also studied.

2.1 VoIP Technology

VoIP is a technology that permits you to convey voice and multimedia (like images, videos) content over the internet. It is one of the least expensive approach to communicate at anytime, anyplace with the internet's accessibility. VoIP has following advantages [14] :-

- Flexibility
- Low cost
- Video conferencing
- Portability
- No extra cables

In a VoIP call the end point can be a smartphone, laptop or any device which can receive and transmits the multimedia content over the internet. The following figure gives an idea about the VoIP call.

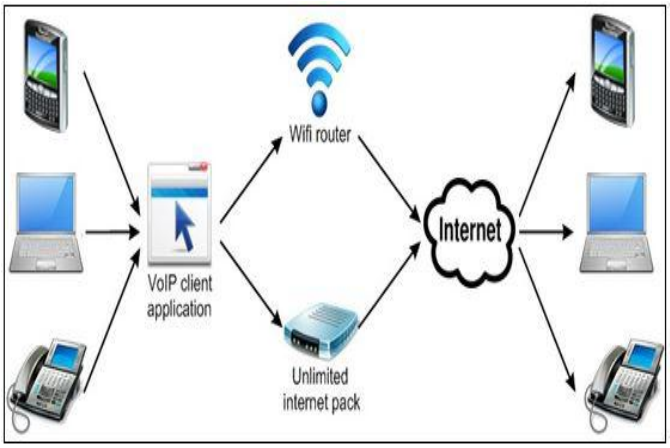

Fig.1. VoIP Call

2.2 Session Initiation Protocol-Overview

SIP is a protocol which is used to create, modify and terminate a media session over the internet protocol [11] [14]. The media session is the call between the two endpoints. Smartphones, laptop, computer or any device which can transmit and receive the media over the internet can be an endpoint. Internet Engineering Task

Force(IEFT) defined the SIP protocol. SIP is an application layer protocol. The details of it are documented in RFC 3261. SIP has a client-server architecture. It includes the use of URI and URL from the Hypertext Transfer Protocol (HTTP) and the text encoding scheme and header style from Simple Mail Transfer Protocol(SMTP). SIP takes the help of SDP for the session description and RTP for the delivery of the audio and video files over the IP network. SIP supports both unicast (two party) and multicast (multiparty) sessions. There are other SIP applications also which are instant messaging, online game, video conferencing and streaming multimedia distribution.

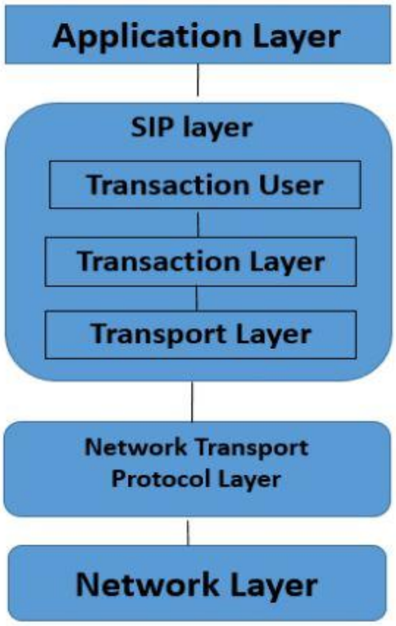

Fig.2. SIP location in general scheme

2.3 Real-time Transport Protocol

For the data with real-time characteristics, the RTP provides an end-to-end delivery services to it. The delivery services are timestamping, payload type identification, sequence numbering and delivery monitoring [2]. The RTP is a network protocol used for the delivery of audio and video over IP networks. The communication and entertainment which involves the streaming data such as audio conferencing, video conferencing, television services and many more extensively used the RTP protocol. RTP neither provide any guarantee for timely delivery nor it provide quality-of-service guarantee. For that it depends on the lower layer services. It doesn't bother about underlying services whether it is sending the packets in sequence or not. The sequence number included in the RTP helps receiver to arrange the packets in the sequence. RTP is always used in conjunction with RTCP which is RTP control protocol. RTP takes care of the delivery of media where as RTCP monitors about transmission statistics and Quality-of-Service.

III. PROPOSED SYSTEM ARCHITECTURE

The figure below shows the system architecture of cloud call recording solution. The main components of the system are phone (softphone for prototyping purpose), Session Border Controller, SIPREC client which will be taking care of recording media sessions.

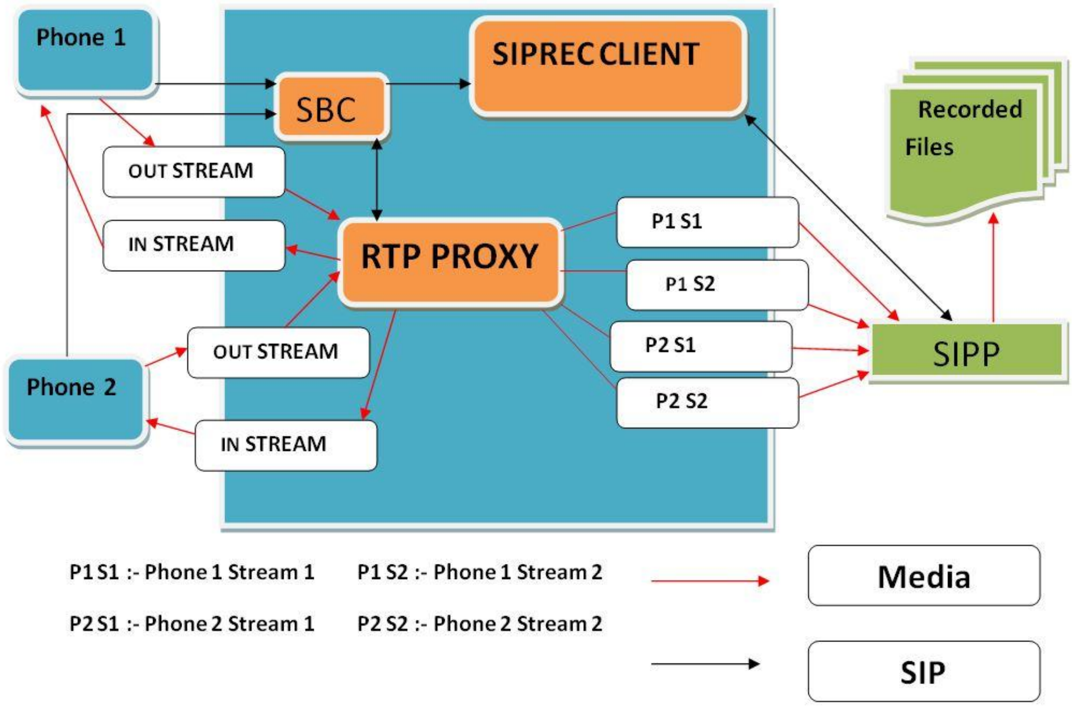

Fig.3. Block diagram

3. METHODOLOGY

3.1 Demonstrate a softphone connect to SIPP server :-

The implementation of the project basically starts with a small task of establishing a call between two softphone. It looks like the configuration is complicated of the softphone, but once you complete that it is very easy to use the softphone. The next task is to demonstrate the softphone connect to SIPP server. The softphone is configured with the IP address of the SIPP server and the port number. The softphone will have its identity number also. At the SIPP server, there is a provision that we can create a scenario using its IP address and keeping the server in listen mode. Now from softphone you can dial any number and the call will land up at the server end. At the server end, the screen will get update for INVITE, 100, 180, ACK, BYE, 200. The number of times one make a call, it will be updated for all the number of calls landing at the server end successfully.

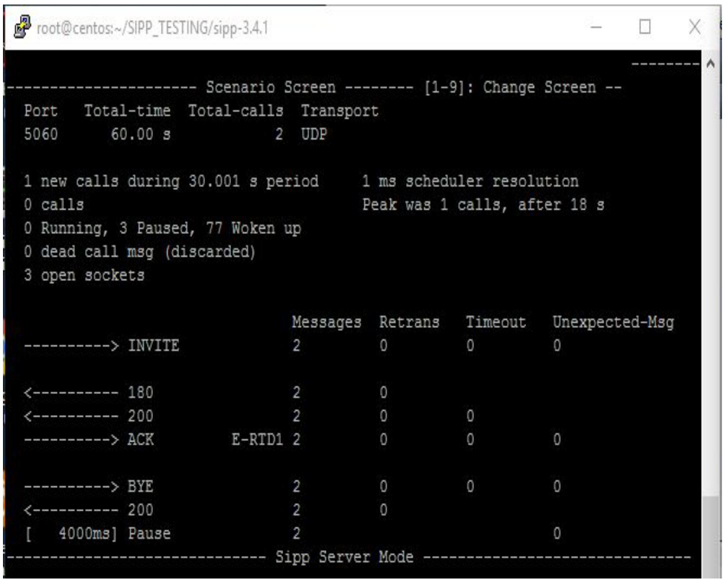

Fig.4. Server end: Demonstrate a softphone connect to SIPP

The RTP-echo command in command line with IP address will enable to produce the echo. Whatever RTP stream received by the server will be reverted and that can be listen. The RTP-echo will be checking for the enabling of the media socket, once the media socket is enabled. It will pass the port number where it will be listening the RTP stream and at the same time all the received RTP stream will be forwarded to the destination port. This destination port is nothing but the port where we are able to get back the echo. This has given the inspiration for the recording the media session.

3.2 Demonstrate SIPP server handling RTP stream and saving it to a file :-

The stream of RTP packets will be transmitted from client to server. The call between the two user agents is nothing but the stream of RTP packets. This is also called as a communication session. The condition will be checked when a call is placed. If the media socket is generated i.e. if it holds the value greater than zero and rtp echo is enabled, it will create a thread which will have function pointer pointing to the rtp echo thread with argument to that function pointer as the media socket. This is the socket which will be bound with the port number. The stream of RTP packets will received and transmitted through this port. Basically RTP packets will have header as well as the payload. The first task is to save this entire packet into the file.

Fig.5. RTP packet writing into a file

3.3 RTP parsing and storing payload into a file :-

The RTP packet will have header along with the payload. The payload is data which we are looking for recording the communication session. So let's understand the RTP packet structure.

Bit offset[a]	0–1	2	3	4–7	8	9–15	16–31
0	Version	P	X	CC	M	PT	Sequence number
32	Timestamp						
64	SSRC identifier						
96	CSRC identifiers ...						
96+32×CC	Profile-specific extension header ID						Extension header length
128+32×CC	Extension header ...						

Fig.6. RTP packet Structure

The RTP packet will include a header whose minimum size is of 12 bytes. In some cases extension header may be present [15]. After the header the RTP payload will be present. This payload is required to save into the file. The header fields explanation is as follows:-

Version :- It will contain the information of the version of the protocol. It will be of 2 bits in size.

P(Padding) :- It is of 1 bit in size. It is used to indicate the extra padding bytes if present at the end of the RTP packets. Basically padding is required to fill the block of certain size in order to meet up with an encryption algorithm. The important thing is that the last byte of the padding will have the information about the number of bytes that has been as a padding.

X(Extension) :- This is of 1 bit size having an information about the presence of extension header.

CC(CSRC count) :- It will contain the information about the number of CSRC identifier which will be followed by the fixed header. It is of 4 bits in size.

M(Marker) :- It is of 1 bit. It is used at application level. On the off chance that it is set, it implies that the present information has some unique pertinence for the application.

PT(Payload Type) :- It is of 7 bit in size. It demonstrates the arrangement of the payload and decides its elucidation by the application. This is indicated by a RTP profile.

SSRC :- It is of 32 bit in size. Synchronization source identifier remarkably recognizes the well spring of a stream. The synchronization sources inside the same RTP session will be one of a kind.

CSRC :- This is also 32 bit in size. Contributing source IDs specify contributing sources to a stream which has been created from different sources.

3.4 RTP parsing :-

The task is to take care of each RTP packet, removing its header and saving the RTP payload into the file. Defining the different variables and storing the values of the RTP packet fields in them is the task. So considering a single RTP packet, need to check the padding. If padding is set, then need to check the last byte value in the RTP packet. This byte will have the value of number of bytes of padding which is added with that packet. So checking for each and every packet and saving it into the file by removing the padding if it is present.

3.5 Handling multiple concurrent RTP stream :-

While parsing the RTP stream and writing into a file a new challenge come up. In a case of multiple calls, all the communication sessions were written into a single file. So the task is to handle the multiple concurrent RTP stream which will make the solution scalable too. The basic idea to overcome this problem is to create the socket with every incoming call i.e. when an SIP INVITE request receives, assigning the port number using some programming logic, binding it with the socket. After binding it with socket, passing the RTP stream to that socket where it will receive as well as send it back in case of echo. At the same time it should parse the RTP stream with individual call and dump into the file for separate call. Once it receives the BYE request for a particular call it will release that port number. So this port number can be made free and can be used again. In this way handling of multiple concurrent RTP stream has been done successfully.

IV. RESULT AND CONCLUSION

The communication session between server and client is recorded and saved with To, From, Call-ID as a file name successfully. For the prototyping the softphone was used. The following figure shows the recording of the communication session.

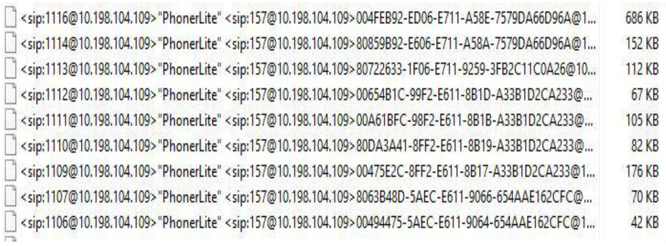

Fig.7. Recording of a communication session

The size of the recorded communication session in KBs, shows that the file has the content which is nothing but the RTP payload. These files are played by using the digital audio editor Audacity. These files are raw files and played in audacity.

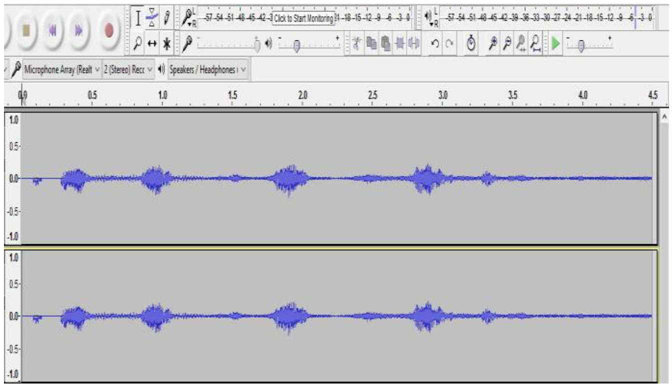

Fig.8. Combined scenario of RTP stream from Wireshark and raw file

V. FUTURE SCOPE

In future there is a scope for this project to work up on different codecs. With the current call recording solution, we are able to deal with codec G.711 [6]. So in future this solution should be compatible with all the codecs like G.729, G.723.1, etc. It will make this call recording solution to perform even more better. Session Border Controller can be integrated with this solution.

REFERENCES

[1] Counterpath Corporation. X-Lite for windows User Guide. http://www.counterpath.com/assets/product_documents/XLite_4_Windows_User_Guide_R6.pdf

[2] R. Frederick V. Jacobson H. Schulzrinne, S. Casner. RTP: A Transport Protocol for Real-Time Applications. https://tools.ietf.org/html/rfc3550, July,2003.

[3] L. Portman Ed. A. Hutton R. Jain K. Rehor, Ed. Use Cases and Requirements for SIP-Based Media Recording (SIPREC). https://tools.ietf.org/html/rfc6341, August,2011.

[4] Ed. C. Eckel A. Johnston A. Hutton L. Portman, H. Lum. Session Recording Protocol. https://tools.ietf.org/html/rfc7866#page-9, May,2016.

[5] Dennis Niels. PhonerLite. http://phonerlite.software.informer.com/.

[6] Tutorials Point. SIP-Codecs. https://www.tutorialspoint.com/session initiation protocol/session initiation

[7] Tutorials Point. What is Socket?. https://www.tutorialspoint.com/unix sockets/what is socket.htm.

[8] Andrew Prokop. Call recording with SIPREC.

https://andrewjprokop.wordpress.com/2014/02/26/call-recording-with-siprec/,

February 26,2014.

[9] Daniel Reed. What will the future of VoIP will be?.

https://www.sinch.com/opinion/future-of-voip/, February 1,2015.

[10] Lync MVP Steven van Houttum. What is a Session Border Controller (SBC) and Do I Need It?. https://blogs.technet.microsoft.com/nexthop/2013/07/08/what-is-a-session-border-controller-sbc-and-do-i-need-it/, July 8,2013.

[11] SIPP Tool Developer Team. SIPP reference documentation.

http://sipp.sourceforge.net/doc/reference.html, April 21,2014.

[12] Davisaa telephone systems inc. The importance of call recording in the work place. http://davissa.com/blog/benefitts-of-call-recording/, December 17,2015.

[13] Tutorialspoint. DCN-User Datagram Protocol.

https://www.tutorialspoint.com/data communication computer network/user datagram protocol.htm.

[14] Tutorialspoint. Session Initiation Protocol-Introduction.

https://www.tutorialspoint.com/session initiation protocol/session initiation protocol introduction.htm.

[15] Wikipedia. Real-time Transport Protocol. https://en.wikipedia.org/wiki/Real-time Transport Protocol.

[16] Wikipedia. Wireshark. https://en.wikipedia.org/wiki/Wireshark.