# VoIP

## Computer Networks

Maunil Vyas Deep Patel Shreyas Patel Mihir Gajjar

**School of Engineering and Applied Science**
**Ahmedabad University**

**Group - 15**

**Guidance By: Prof . Manish Chaturvedi**

6 December, 2016

# Outline

# Outline

# Abstract


**Implemented a VoIP system capable of :**

Simple voice calling as well as conference call facility among clients on LAN, WAN, Internet.....

# Abstract

**Implemented a VoIP system capable of :**

Simple voice calling as well as conference call facility among clients on LAN, WAN, Internet.....

- Used the SIP and RTP protocols.

# Abstract

**Implemented a VoIP system capable of :**

Simple voice calling as well as conference call facility among clients on LAN, WAN, Internet.....

- Used the SIP and RTP protocols.
- Made soft-phone in Python language using support of PJSIP/PJSUA library.

# Abstract

**Implemented a VoIP system capable of :**

Simple voice calling as well as conference call facility among clients on LAN, WAN, Internet.....

- Used the SIP and RTP protocols.
- Made soft-phone in Python language using support of PJSIP/PJSUA library.
- Used Asterisk as our SIP server.

# Outline

# VoIP

- Voice over Internet Protocol
- Transferring digitized voice over Internet in form of chunks/packets.

## Pros:

- Reduced bandwidth with using more efficient compression mechanism which maintains quality.
- Efficient bandwidth utilization (Bandwidth can be decreased or increased based on requirement)
- Cheaper service
- Service can be setup easily

# VoIP

- Voice over Internet Protocol
- Transferring digitized voice over Internet in form of chunks/packets.

**Pros:**

- Reduced bandwidth with using more efficient compression mechanism which maintains quality.
- Efficient bandwidth utilization (Bandwidth can be decreased or increased based on requirement)
- Cheaper service
- Service can be setup easily

# VoIP

- Voice over Internet Protocol
- Transferring digitized voice over Internet in form of chunks/packets.

## Pros:

- Reduced bandwidth with using more efficient compression mechanism which maintains quality.
- Efficient bandwidth utilization (Bandwidth can be decreased or increased based on requirement)
- Cheaper service
- Service can be setup easily

## Cons:

- Voice delivery is not guaranteed (Packet may be lost).
- Unpredictable Quality of Service (IP uses Best-effort-service)

# Outline

# Why SIP ?

- There are mainly two famous open source protocols are available for call initiation facility SIP and H.323.

| Functionality | H.323v3 | SIP | Choice |
|---|---|---|---|
| Backward Compatibility | Compatible | Not Compatible | H.323 |
| Fault Tolerance | Better | Good | H.323 |
| Complexity | More | Less | SIP |
| Protocol Encoding | Binary(ASN.1) | Text | SIP |
| Debugging | Hard | Simple | SIP |
| Code Reuse | Difficult | Easy | SIP |
| Ease of Customization | Difficult | Easy | SIP |
| Call Control(Hold, Transfer, Forward, Wait) | Yes | Yes | Equivalent |
| Call Setup Delay | 2-3 RTT | 2-3 RTT | Equivalent |
| Transport Type | TCP/UDP | TCP/UDP | Equivalent |
| Packet Loss Recovery | Good | Good | Equivalent |
| Media Transport | RTP | RTP | Equivalent |

Figure : Comparison of SIP and H.323

# SIP

## Our decision parameters for choosing SIP Over H.323

- Complexity.
- Code Reuse
- Call Control
- Call Setup Delay

- Session Initiation Protocol [RFC 2543]. .

# SIP

## Our decision parameters for choosing SIP Over H.323

- Complexity.
- Code Reuse
- Call Control
- Call Setup Delay

- Session Initiation Protocol [RFC 2543]. .
- Designed in 1996 by Mark Handley, Henning Schulzrinne, Eve Schooler and Jonathan Rosenberg.

# SIP

## Our decision parameters for choosing SIP Over H.323

- Complexity.
- Code Reuse
- Call Control
- Call Setup Delay

- Session Initiation Protocol [RFC 2543]. .
- Designed in 1996 by Mark Handley, Henning Schulzrinne, Eve Schooler and Jonathan Rosenberg.
- It helps in setting up the session between calls.

# SIP

## Our decision parameters for choosing SIP Over H.323

- Complexity.
- Code Reuse
- Call Control
- Call Setup Delay

- Session Initiation Protocol [RFC 2543]. .
- Designed in 1996 by Mark Handley, Henning Schulzrinne, Eve Schooler and Jonathan Rosenberg.
- It helps in setting up the session between calls.
- It is text based protocol similar to SMTP and HTTP.

# SIP

## Our decision parameters for choosing SIP Over H.323

- Complexity.
- Code Reuse
- Call Control
- Call Setup Delay

- Session Initiation Protocol [RFC 2543]. .
- Designed in 1996 by Mark Handley, Henning Schulzrinne, Eve Schooler and Jonathan Rosenberg.
- It helps in setting up the session between calls.
- It is text based protocol similar to SMTP and HTTP.
- Uses port 5060 for non-encrypted traffic and port 5061 for encrypted traffic.

# Components of SIP

- User Agent .

# Components of SIP

- User Agent .
- Registrar Server (Registers User Agent).

# Components of SIP

- User Agent .
- Registrar Server (Registers User Agent).
- Location Server (Provides location of user to proxy servers).

# Components of SIP

- User Agent .
- Registrar Server (Registers User Agent).
- Location Server (Provides location of user to proxy servers).
- Redirect Server (Redirects the SIP requests to new address)

# Components of SIP

- User Agent .
- Registrar Server (Registers User Agent).
- Location Server (Provides location of user to proxy servers).
- Redirect Server (Redirects the SIP requests to new address)
- Proxy Server (Forwards SIP requests to other servers)

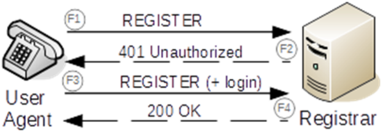


Figure : User Agent Registration Process

- User Agent .
- Registrar Server (Registers User Agent).
- Location Server (Provides location of user to proxy servers).
- Redirect Server (Redirects the SIP requests to new address)
- Proxy Server (Forwards SIP requests to other servers)

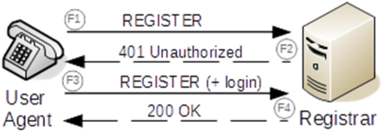


Figure : User Agent Registration Process

# SIP call placing mechanism

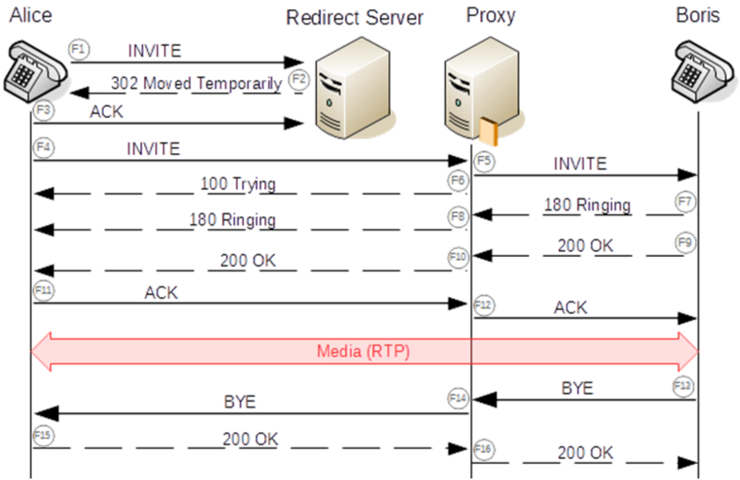


Figure : Call placing mechanism in SIP

# Outline

# RTP

- Real-time Transport Protocol [RFC 3550].

# RTP

- Real-time Transport Protocol [RFC 3550].
- First published in 1996.

# RTP

- Real-time Transport Protocol [RFC 3550].
- First published in 1996.
- Used to deliver audio and video between users.

# RTP

- Real-time Transport Protocol [RFC 3550].
- First published in 1996.
- Used to deliver audio and video between users.
- Major RTP implementations built on UDP as we need real time streaming.

# RTP

- Real-time Transport Protocol [RFC 3550].
- First published in 1996.
- Used to deliver audio and video between users.
- Major RTP implementations built on UDP as we need real time streaming.

## Sub Protocols

- RTP : Data Transfer.
- RTCP : To control transmission by providing feedback and synchronization

# RTP

- Real-time Transport Protocol [RFC 3550].
- First published in 1996.
- Used to deliver audio and video between users.
- Major RTP implementations built on UDP as we need real time streaming.

## Sub Protocols

- RTP : Data Transfer.
- RTCP : To control transmission by providing feedback and synchronization

## RTP Packet

| | | | | |
|---|---|---|---|---|
| Payload type | Sequence number | Timestamp | Synchronization source identifier | Miscellaneous fields |

# Outline

# SIP Setup

## Asterisk

- We have used Asterisk Server.
- The single package of asterisk has a functionality of Registrar, Proxy Server, Location Server and Redirect Server.

# SIP Setup

## Asterisk

- We have used Asterisk Server.
- The single package of asterisk has a functionality of Registrar, Proxy Server, Location Server and Redirect Server.

## Installation and Configuration in Ubuntu

- Write sudo apt-get install asterisk in console to install asterisk.
- Info related to Configuration: [3]

# Outline

# Soft-Phone

## Soft-Phone

- Built on Python using PJSUA/PJSIP library.

## Soft-Phone

- Built on Python using PJSUA/PJSIP library.

## How to use Soft Phone in our application?

- Enter your name and password.
- Enter m to make call, a to answer incoming call, h to hangup the call, and q to quit soft phone.
- After pressing m enter the name of user to whom you want to call.

# Outline

# Images of working model

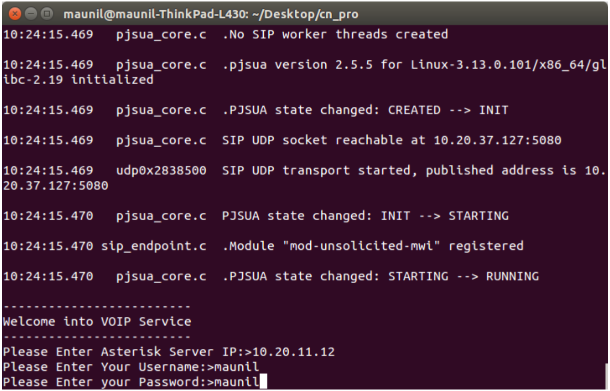


Figure : User 1 Registration

# Images of working model

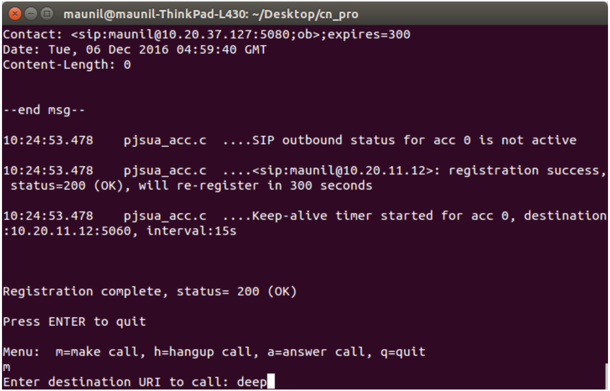


Figure : Making Call to deep

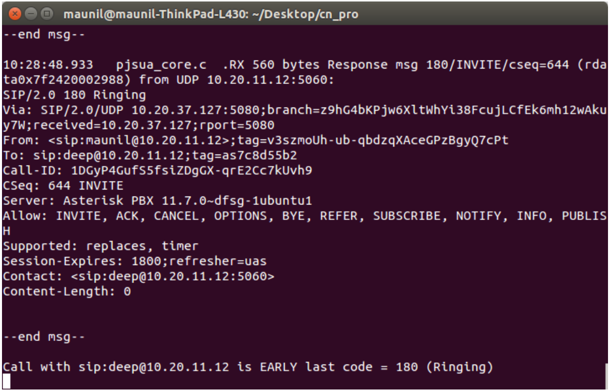


Figure : Ringing info at user1

# Images of working model

```
ramkabir@Ramkabir: ~/Documents/CN_PRJ
Expires: 300
Contact: <sip:deep@10.20.11.17:5080;ob>;expires=300
Date: Tue, 06 Dec 2016 05:02:49 GMT
Content-Length: 0


--end msg--

10:28:02.943    pjsua_acc.c  ....SIP outbound status for acc 0 is not active

10:28:02.943    pjsua_acc.c  ....<sip:deep@10.20.11.12>: registration success, s
tatus=200 (OK), will re-register in 300 seconds

10:28:02.943    pjsua_acc.c  ....Keep-alive timer started for acc 0, destination
:10.20.11.12:5060, interval:15s



Registration complete, status= 200 (OK)

Press ENTER to quit

Menu:  m=make call, h=hangup call, a=answer call, q=quit
10:28:48.948   pjsua_core.c  .RX 863 bytes Request msg INVITE/cseq=102 (rdata0x7
```

Figure : User 2 Registration

# Images of working model

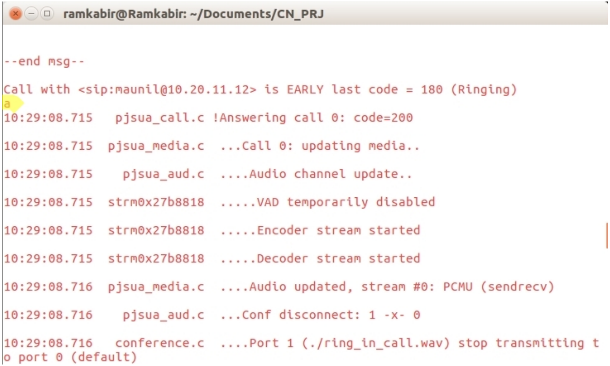


Figure : User 2 Answering

# Images of working model

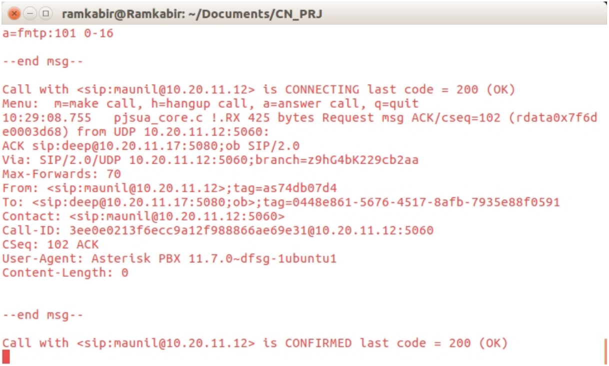


Figure : Call connection info at User 2

# Images of working model

```
ramkabir@Ramkabir: ~/Documents/CN_PRJ
tatus=200 (OK), will re-register in 300 seconds

10:32:58.045    pjsua_acc.c  ....Keep-alive timer started for acc 0, destination
:10.20.11.12:5060, interval:15s

h
10:35:10.771    pjsua_aud.c !Conf disconnect: 1 -x- 0

10:35:10.771   pjsua_call.c  Call 0 hanging up: code=603..

10:35:10.771   pjsua_core.c  ....TX 396 bytes Request msg BYE/cseq=2538 (tdta0x2
7bc940) to UDP 10.20.11.12:5060:
BYE sip:maunil@10.20.11.12:5060 SIP/2.0
Via: SIP/2.0/UDP 10.20.11.17:5080;rport;branch=z9hG4bKPje647864c-656f-4ce9-91ee-
cb00af3d3ead
Max-Forwards: 70
From: <sip:deep@10.20.11.17;ob>;tag=0448e861-5676-4517-8afb-7935e88f0591
To: <sip:maunil@10.20.11.12>;tag=as74db07d4
Call-ID: 3ee0e0213f6ecc9a12f988866ae69e31@10.20.11.12:5060
CSeq: 2538 BYE
User-Agent: pjsip python
Content-Length:  0
```

Figure : User 2 Hangup

# Images of working model

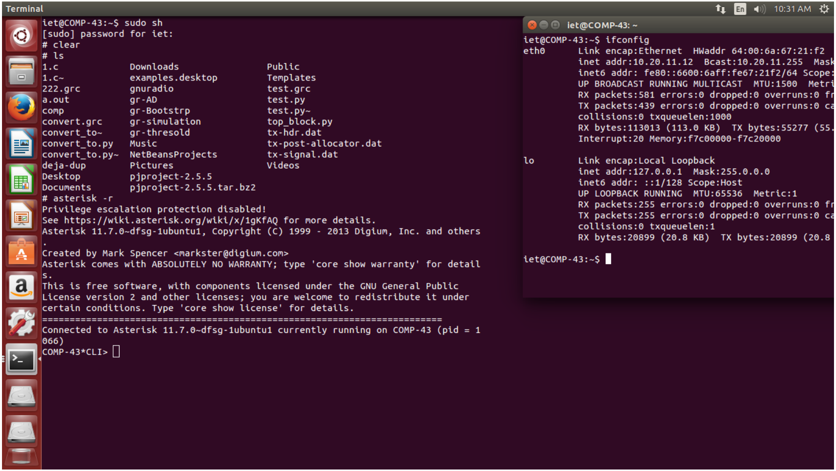


Figure : Asterisk Server

# Outline

# Wire shark Analysis

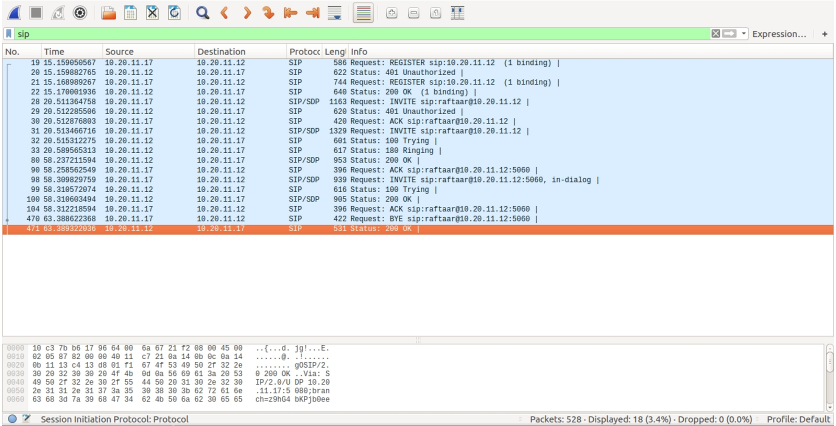


Figure : Call Initiation Using SIP

# Wire shark Analysis

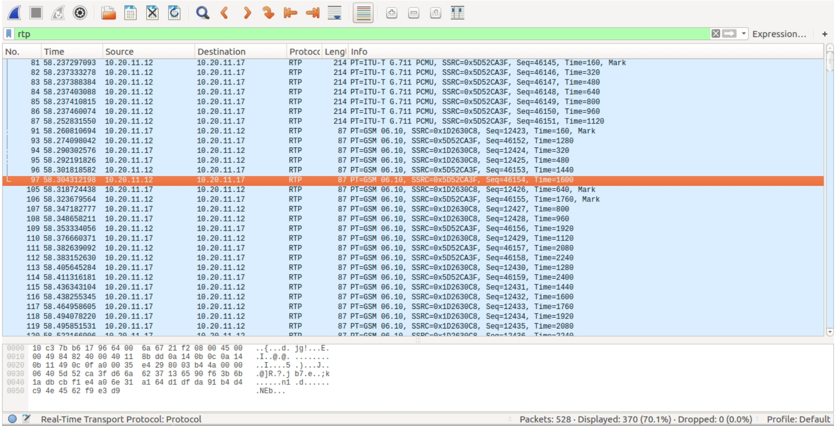


Figure : RTP

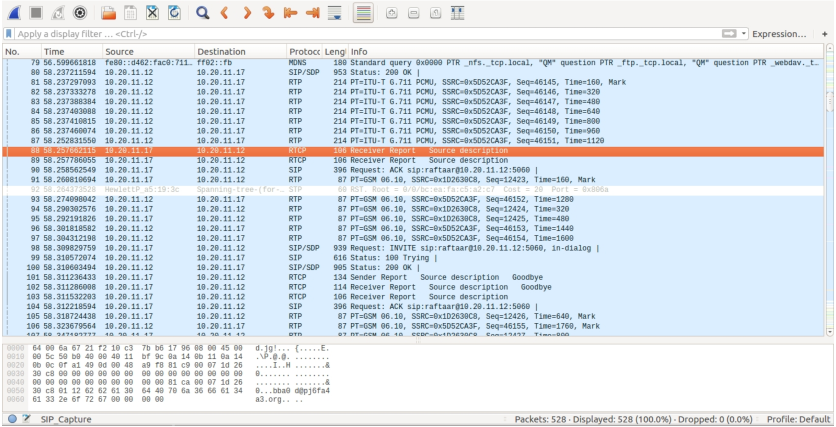
Apply a display filter ... <Ctrl-/>
Expression...
No. Time Source Destination Protocol Length Info
79 56.599661818 fe80::d462:fac0:711… ff02::fb MDNS 180 Standard query 0x0000 PTR _nfs._tcp.local, "QM" question PTR _ftp._tcp.local, "QM" question PTR _webdav._t…
80 58.237211594 10.20.11.12 10.20.11.17 SIP/SDP 953 Status: 200 OK |
81 58.237297093 10.20.11.12 10.20.11.17 RTP 214 PT=ITU-T G.711 PCMU, SSRC=0x5D52CA3F, Seq=46145, Time=160, Mark
82 58.237333278 10.20.11.12 10.20.11.17 RTP 214 PT=ITU-T G.711 PCMU, SSRC=0x5D52CA3F, Seq=46146, Time=320
83 58.237388384 10.20.11.12 10.20.11.17 RTP 214 PT=ITU-T G.711 PCMU, SSRC=0x5D52CA3F, Seq=46147, Time=480
84 58.237403088 10.20.11.12 10.20.11.17 RTP 214 PT=ITU-T G.711 PCMU, SSRC=0x5D52CA3F, Seq=46148, Time=640
85 58.237410815 10.20.11.12 10.20.11.17 RTP 214 PT=ITU-T G.711 PCMU, SSRC=0x5D52CA3F, Seq=46149, Time=800
86 58.237460074 10.20.11.12 10.20.11.17 RTP 214 PT=ITU-T G.711 PCMU, SSRC=0x5D52CA3F, Seq=46150, Time=960
87 58.252831550 10.20.11.12 10.20.11.17 RTP 214 PT=ITU-T G.711 PCMU, SSRC=0x5D52CA3F, Seq=46151, Time=1120
88 58.257662115 10.20.11.17 10.20.11.12 RTCP 106 Receiver Report Source description
89 58.257786055 10.20.11.17 10.20.11.12 RTCP 106 Receiver Report Source description
90 58.258562549 10.20.11.17 10.20.11.12 SIP 396 Request: ACK sip:raftaar@10.20.11.12:5060 |
91 58.260810694 10.20.11.17 10.20.11.12 RTP 87 PT=GSM 06.10, SSRC=0x1D2630C8, Seq=12423, Time=160, Mark
92 58.264373528 HewlettP_a5:19:3c Spanning-tree-(for-… STP 60 RST. Root = 0/0/bc:ea:fa:c5:a2:c7 Cost = 20 Port = 0x806a
93 58.274098042 10.20.11.12 10.20.11.17 RTP 87 PT=GSM 06.10, SSRC=0x5D52CA3F, Seq=46152, Time=1280
94 58.290302576 10.20.11.17 10.20.11.12 RTP 87 PT=GSM 06.10, SSRC=0x1D2630C8, Seq=12424, Time=320
95 58.292191826 10.20.11.17 10.20.11.12 RTP 87 PT=GSM 06.10, SSRC=0x1D2630C8, Seq=12425, Time=480
96 58.301818582 10.20.11.12 10.20.11.17 RTP 87 PT=GSM 06.10, SSRC=0x5D52CA3F, Seq=46153, Time=1440
97 58.304312198 10.20.11.12 10.20.11.17 RTP 87 PT=GSM 06.10, SSRC=0x5D52CA3F, Seq=46154, Time=1600
98 58.309829759 10.20.11.17 10.20.11.12 SIP/SDP 939 Request: INVITE sip:raftaar@10.20.11.12:5060, in-dialog |
99 58.310572074 10.20.11.12 10.20.11.17 SIP 616 Status: 100 Trying |
100 58.310603494 10.20.11.12 10.20.11.17 SIP/SDP 905 Status: 200 OK |
101 58.311236433 10.20.11.17 10.20.11.12 RTCP 134 Sender Report Source description Goodbye
102 58.311286008 10.20.11.17 10.20.11.12 RTCP 114 Receiver Report Source description Goodbye
103 58.311532203 10.20.11.17 10.20.11.12 RTCP 106 Receiver Report Source description
104 58.312218594 10.20.11.17 10.20.11.12 SIP 396 Request: ACK sip:raftaar@10.20.11.12:5060 |
105 58.318724438 10.20.11.17 10.20.11.12 RTP 87 PT=GSM 06.10, SSRC=0x1D2630C8, Seq=12426, Time=640, Mark
106 58.323679564 10.20.11.12 10.20.11.17 RTP 87 PT=GSM 06.10, SSRC=0x5D52CA3F, Seq=46155, Time=1760, Mark
0000 64 00 6a 67 21 f2 10 c3 7b b6 17 96 08 00 45 00 d.jg!... {.....E.
0010 00 5c 50 b0 40 00 40 11 bf 9c 0a 14 0b 11 0a 14 .\P.@.@. ........
0020 0b 0c 0f a1 49 0d 00 48 a9 f8 81 c9 00 07 1d 26 ....I..H .......&
0030 30 c8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0....... ........
0040 00 00 00 00 00 00 00 00 00 00 81 ca 00 07 1d 26 ........ .......&
0050 30 c8 01 12 62 62 61 30 64 40 70 6a 36 66 61 34 0...bba0 d@pj6fa4
0060 61 33 2e 6f 72 67 00 00 00 00 a3.org.. ..
SIP_Capture
Packets: 528 · Displayed: 528 (100.0%) · Dropped: 0 (0.0%)
Profile: Default

Figure : RTCP

# Outline

# Summary

- Successfully Implemented VoIP.
- Future Goals:
  - Android Application with better voice quality.
  - Video Calling.

# References I


Jim Kurose, Keith W. Ross,
*Computer Networking: A Top-Down Approach*.
PEARSON, November 2000.

WIKIPEDIA.
For Images.

Dalgic I, Fang H. Comparison of H. 323 and SIP for IP Telephony Signaling. InPhotonics East'99 1999 Nov 22 (pp. 106-122). International Society for Optics and Photonics.
For Asterisk Configuration.

Papageorgiou, P., 2001. A comparison of H. 323 vs SIP. University of Maryland.
For Comparison of H.323 and SIP

# References II

*http : //most − VoIP.readthedocs.io/en/latest/asteriskdocs/asteriskconfiguration.html*

For Asterisk Configuration.

http://www.pjsip.org/python/pjsua.html.
PJSUA Python library Support