International Journal of Computer Applications (0975 – 8887)
Volume 124 – No.4, August 2015

Implementing the VOIP Communication Principles using Raspberry Pi as Server

Preeti V. Murkute
ME-Electronics & Telecommunication Engineering (Digital Electronics)
Department of Electronics & Telecommunication Engineering
SSBT's College of Engineering and Technology, Bambhori, North Maharashtra University, Jalgaon

V.M. Deshmukh
Department of Electronics & Telecommunication Engineering
SSBT's College of Engineering and Technology, Bambhori
North Maharashtra University, Jalgaon

ABSTRACT
VoIP stands for Voice over IP (Internet Protocol), a variety of methods are there for establishing two-way multi-media communications over the Internet or other IP-based packet switched networks. VoIP has two goals: The first is reduction in telephone charges by sending the call as packets over the data lines or internet. The second goal is to provide flexible voice networks, like allowing multiple calls on same physical link. Applications of real-time VoIP communication have come into widespread use over the Internet. [2] VoIP in Embedded systems is one of the blind spots, where due to its over reliance on PC environment restricts its application. In this project we have reviewed the application of VoIP on embedded systems and also proposed a system to set its application using better components. The proposed hardware uses raspberry pi board with VoIP protocols. It adapts the SIP (Session Initialization Protocol), IAX/IAX2 (Inter Asterisk Protocol: Open source trucking protocol).The system sets a small intranet that can be used by any organization for communicating within itself at no data charges. The raspberry Pi is programmed at kernel level of Linux and thus works as a server. The system utilizes the freely available telephony software like Asterisk, FreePBX for the development of codes which makes the system cost effective.

General Terms
Session Initialization Protocol, Server-Client model, Asterisk, Raspberry Pi Board, Call Processing Language (CPL), Linux

Keywords
VoIP, Embedded Systems, SIP, Asterisk protocols, Kernel codes, Session Initialization Protocol, Server-Client model, Raspberry Pi Board

1. INTRODUCTION
A computer Network is a system of interconnected computers and the various computerized peripherals. The computer networks are classified upon various factors. The system works on the Linux kernel coding, the raspberry pi and the asterisk. The following paper follows the details of the proposed system. The system is cost effective, easily installed and can be used by any of the organization. The software used are open source and easily available. The transmission is duplex and clear. The open source software have their own library and their own set of instructions which are handy to use and can be modified according to our requirement.

2. THE SYSTEM ESSENTIALS

2.1 The Linux Kernel Module
Linux is fast and stable open source OS for PC's and workstations that features professional level internet services, extensive development tools, fully functional graphical user interface and a massive range of applications ranging from office suites to multimedia applications. [6] The kernel is the heart of Linux operating system. It interacts with the hardware. It is loaded into memory when system is booted. The Linux kernel is distributed under the GNU General Public License (GPL), the terms of which are set out by the free software foundation. A module is a piece of code that can be loaded or unloaded into the kernel according to the demand. The kernel performs various primitive operations on behalf of user programs. [6]

2.2 The Raspberry Pi
The raspberry pi is a small arm based PC which runs Linux from SD card. The raspberry pi that we are using is generation 1 model B. It consists of ARM 7 processor. The raspberry pi needs to be installed and configured like the other hardware. [7] The features of raspberry pi are as specified in the table.

Feature	Specification
CPU	700MHz ARM1176-JZFS
GPU	Broadcom VideoCore IV
Memory	256MB LPDDR2-800
Video	HDMI, composite
Audio	HDMI, stereo analog
USB	2 x USB2.0 (model B)
Storage	SD card
Networking	10/100 Ethernet
Power	5V micro USB

Table 1: The Raspberry pi specification

To flash the SD card is the main task of the project. The first task is to decide which Linux distribution is required to use with the raspberry pi. Each distribution has its advantages and disadvantages. The advantage is that SD card can be flashed again for any number at any given point. [7]

2.3 The Asterisk
Asterisk is a software implementation of a telephone private branch exchange (PBX); it allows attached telephones to make calls to one another, and to connect to other telephone

services, such as the public switched telephone network (PSTN) and Voice over Internet Protocol (VoIP) services. It is an Open Source GNU GPL licensed. The main advantage is its availability of ready to use modules. It also has the ability to create / edit modules. It can be programmed in native C/C++. [6]Asterisk Gateway Interface (AGI) available as API. It supports all standard VOIP protocols including SIP and H.323.It comprises of the Cross Platform / Native compilers to port / compile in embedded processors like ARM. This makes our work possible of incorporating this feature with our arm based raspberry pi. Another feature of asterisk is the Ease of implementation in Unix/Linux systems. It can be tailored to fit any no. of and variety of users.

Most of the configuration of asterisk requires three main packages. They are: the main asterisk program, the telephony drivers and the PRI library. Different packages and the source code need to be obtained according to our requirement. The packages that are downloaded are archives containing the source code and so need to be extracted before compilation. [6]

2.4 The Code Extensions

```
[from-pstn]
exten =>s,1,Verbose(1, Caller ${CALLERID(all)}
same=>n,Answer()
same=>n,Wait(3)
same=>n,Playback(welcome)
same=>n,GoSub(post,ss,1)
same=>n,Playback(thanks)
 same=>n,Hangup()
[post]
include => default_cont    ////   connection of two nodes
exten => ss,1,Background(gud/post)
same=>n,WaitExten(5)
exten => 1,1,Set(CDR(aa)=711)
same=>n,Return()
exten => 2,1,Set(CDR(aa)=712)
 same=>n,Return()
[default_cont]
exten => i,1,Playback(invalid)
same=> n,Hangup()
exten => t,1,Playback(timeout)
 same=> n,Hangup()
exten => h,1,DeadAGI(convert_recordings.sh)
same=>n,DeadAGI(my_uploader.php,${var1},${CALLERID
(num)},${CDR(var2)},${CDR(aa)})
```

Script to update DB

```
#!/usr/bin/php -q
<?php
        declare(ticks = 1);
 pcntl_signal(SIGHUP,  "sig_handler");
require('/var/lib/asterisk/agi-bin/phpagi.php');
$agi = new AGI();
$uniqueid = $agi->request['agi_uniqueid'];
$callerid = $agi->request['agi_callerid'];
 $channel  = $agi->request['agi_channel'];
 $agi->verbose("uniqueid: ".$uniqueid ."\n",3);
  $agi->hangup($channel);  exit(0);
function sig_handler($signo)
{
GLOBAL $uniqueid,$callerid //query asterisk database
  $conn1 = mysql_connect("localhost", "root", "asterisk.me");
  mysql_select_db("asteriskcdrdb", $conn1);
}
```

From the first code [from pstn], indicates a code for publically calling the server.as per the requirement of our application we included the [post] section. ‘ss’ is the name given to our server and ‘1’ is its value. As we have created only two members for the database in our code we give their extensions and define their set numbers ‘711’ and ‘712’. The other addition to the coding section is the script to update the database. Here we defined the function ‘sig_handler’, which we have connected to our asterisk library extension using the mysql files. The agi section of code shows that the connection can be established using two ways: using the IP address or using the names given 711/712. The coding part created huge lot of errors while it was done using python. Thus we have used the kernel codes and have patched most of the coding part.

3. THE CALL FLOW PROCESSES

The Session Initiation Protocol (SIP for short) is a Voice over IP protocol designed by the Internet Engineering Task Force. MMUSIC which stands for Multi-party Multimedia Session Control created the SIP protocol. This protocol is intended primarily for creating, modifying and terminating sessions with one or more users. The sessions are generally VoIP telephone calls or conferences. [5]

SIP Characteristics: Unlike H.323, SIP is a text-based protocol. HTP version 1.1 is the base of the format of SIP request and response. There are basically three protocols that are used by the end-users to communicate:

Session Description Protocol (SDP): It is used to exchange information about audio/video channels. The SDP is also the product of the MMUSIC group.

Real Time Protocol (RTP): It is used to send the real-time streams of audio or video across the network. The messages between the two end points are transmitted as transactions. A transaction consists of a request and the related response or responses. The messages are given a transaction ID for its transaction and the ones those belong to same transaction have the same IDIn SIP it is called as CSeq. Every transaction that is performed should have a unique CSeq number. The ACK (Acknowledge) is a single exception which uses the same CSeq number as that of the transaction for which it sends acknowledge. Two modes are possible when the communication is through TCP: For all the transaction same TCP channel is utilized or for each transaction a new connection is established.

SIP Entities: With a logical distinction the SIP network contain a number of entities. In practice some of the entities are combined together in a SIP server called as the Proxy/Registrar server. Some of the entities are:

User Agent: A User Agent is any hardware or software SIP telephone. The User Agent performs two roles, the client role wherein it sends request and the server role where it receives requests and responses.

Registrar: The server receives the Register messages from the clients. The registered messages include the location information i.e the IP address of the client. The task of the Registrar is to keep the location information of each registered user in the database. [5]

Proxy: The proxy server performs the task to receive the requests and forward them to the target point. Most people prefer to the messages through proxy server instead of the direct route because: 1.to enforce the call policy. 2. To access the location database maintained by the registrar this is also called as the location service. [12]

Gateway: The server that receives the SIP calls and translates it into another telecommunication network is called as a Gateway.

For proposed project Proxy, Registrar and Gateway are one and the same. We have used RaspberryPi board and programmed it to be an all in one gateway/proxy/registrar to make things simple and efficient.

1. *Registration:*

Considering the flow of a typical SIP call which includes the IP phones and the asterisk server. As per our system the client is the X-lite, a SIP softphone. Registering it at the asterisk PBX. The Asterisk's IP address is 10.18.1.7, while the client is at 10.18.1.6 and we registered the telephone number 711.

For this registration the SIP telephone needs to send a REGISTER request.

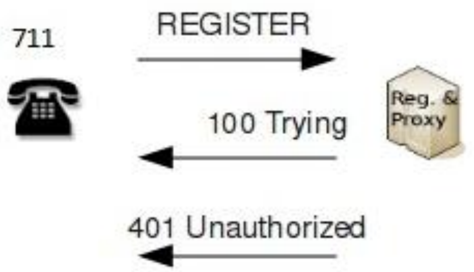


Figure 1: SIP registration, phase 1

The Register server on receiving the request sends a provisional acknowledgement response "100 trying". Upon receiving the acknowledgment the client comes to know that his request has reached and is being processed thus does not make another one. The processing of request involves the authentication test, if found that the request is not authenticated a response of "401 Unauthorized" is sent to the client. So the user now requires sending another Register request along with its authentication. [6]

The next step in the process is sending of authentication packets by the X-lite soft IP phone to the server. The registrar server will again first respond with "100 Trying" provisional acknowledgement and then compare. If this time the there's no error, a response of "200 ok" is sent back and the end point is stored in the database. The database is usually shared between the registrar and the proxy server so that the proxy can use it to connect calls. The figure below shows the message exchange:

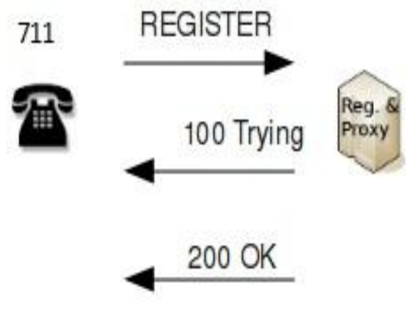


Figure 2: SIP registration, phase 2

Call Flow: While a response of "100 Trying" is sent to the client simultaneously a INVITE request is forwarded to the target telephone by the SIP proxy server. As the proxy server has access to the location database it knows about the IP addresses of all registered telephones. Steps are shown in Figure A below

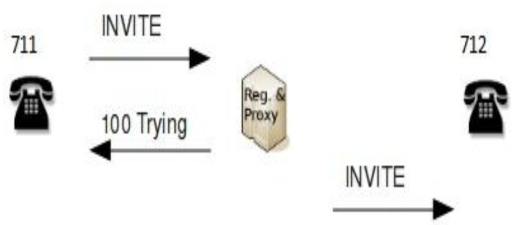


Figure A

The telephone 712 starts ringing and sends the response "180 Ringing" to the proxy server. The proxy will forward the response to the telephone 711. [6]

The called user picks up the phone and his/her telephone sends the response "200 OK". The response body contains an SDP message so that the caller knows where to send his RTP stream. The proxy server forwards the response to the caller.

The caller (telephone 711) confirms the receipt of "200 OK" with the ACK message. The proxy server forwards the ACK to the telephone 712. At this point, the call has been established and both parties start sending their RTP streams. Above mentioned steps are shown in Figure B.

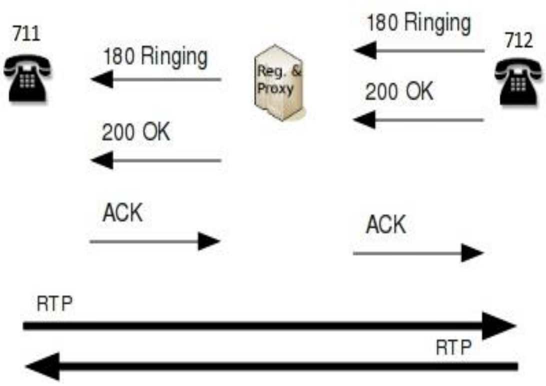


Figure B

When one of the users hangs up, his/her telephone sends the request BYE and the SIP proxy forwards the message to the other party. The other party responds to the BYE request with "200 OK" (again, the proxy server forwards the response to the other side). Both parties stop sending RTP data and the call is over. The events described above are shown in Figure C.

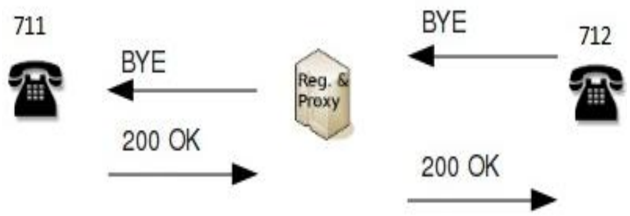


Figure C

4. SYSTEM SETUP

The project setup consist of the raspberry pi as server, two laptops or PCs, one router for LAN connections, patch cables, SD card loaded with Linux Operating system. One of the laptop is given an IP address of 10.8.1.10 and the other one is 10.18.1.11. The caller id of first laptop is 711 and that of second one is 712. Both the caller Id's are stored in the database. The raspberry pi which serves as a server has the IP address 10.18.1.6. The first thing required to do is installing the softphone X-lite softphone software in the laptops. The software requires Microsoft office visual c++ 2012 installed in it as it supports the visual basic programmed application. After the installation of the softphone software, the account setting of the clients is done. The domain, userid and proxy address is to be given. The following image shows the same.

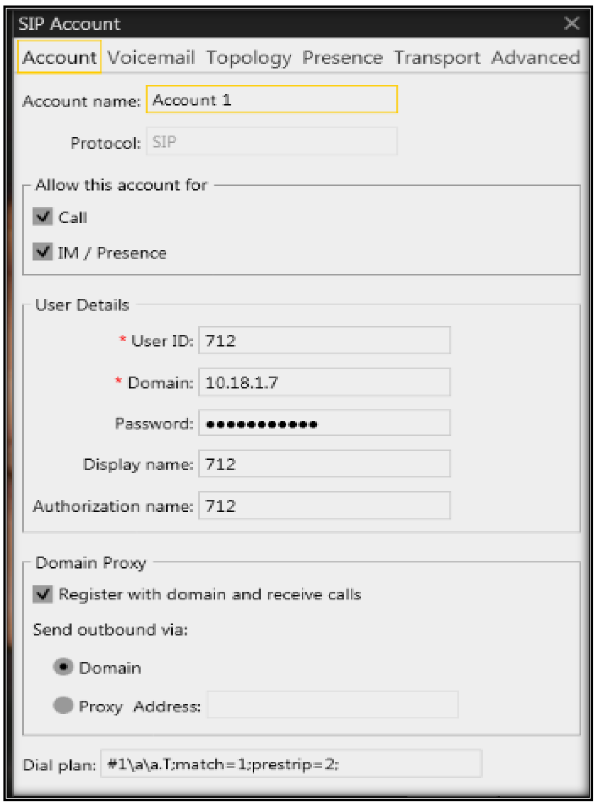
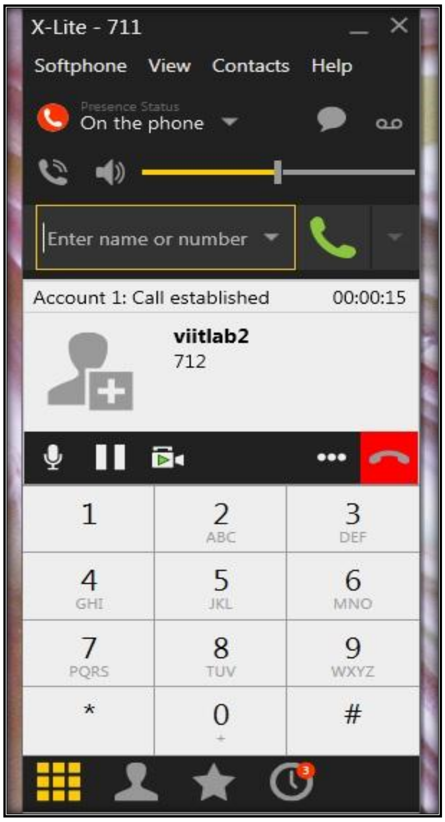


Figure 3: Account settings of client

After the account settings of X-lite softphone, change the LAN settings according to our server and router. To perform these settings:

- Open network and sharing Centre
- Right click on the LAN or Ethernet Connection status
- The patch cables are connected to from both the laptop to the router and from the router to the server.
- The router and server individually require the power connections
- The SD card is placed in the raspberry pi board.
- Now Click on Properties
- Network Connection Properties Window will open
- Double click on Internet Protocol Version 4 (TCP/IPv4)
- An IP address window will appear
- In this window select the Use of following IP address radio button and give the IP address of our server and gateway.
- After this change the X-lite softphone will show an enable status
- Now both the clients are connected to each other and can communicate
- A bidirectional communication takes place and by dialling 712 from first client phone call is placed to the other client.
- The call has no data charges and no delay. It is a free of cost call

Resulting X-lite soft phone Image:



5. CONCLUSION

After reviewing the VoIP basics and studying its features, our objective of developing a system which is cost effective and which utilizes the VoIP communication in embedded system have been fulfilled. The session initialization protocol (SIP) forms the channel of transmission in the system. Looking at the number of options available today for making use of SIP in projects/products one needs to take into account its feasibility of implementation. The open source revolution is less touched in telecommunication industry so it is our small attempt to make the use of these in embedded system as well. On the other hand using tools like Asterisk can help implementation lot faster at the same time reducing processing overheads. Thus a system is developed by using open source free ware software and it is a cost effective system that uses the basic VoIP as communication support and runs on small embedded hardware as its server.

Future Scope: The Java servlets are available with lots of documentation and an additional processor. By applying these changes to the designed system we can use it for higher applications. After these changes the developed system could be used for implementing VoIP in industrial and commercial applications.

6. ACKNOWLEDGMENTS

The satisfaction that accompanies the successful completion of any task would be incomplete without mentioning the names of people whose never-ending cooperation made it possible and whose constant guidance and encouragement crown all efforts with success. I am grateful to my project guide Prof. V. M. Deshmukh for the guidance, inspiration and constructive suggestions that helped me in this work. I am very thankful to my Parents Dr.V.D.Murkute and Mrs. Jaya V. Murkute and my brother Vaibhav Murkute for their whole hearted support during my work.

7. REFERENCES

[1] Jian Wang, Wenbo Zhai, “An Application of VoIP Communication on Embedded System” 2010 IEEE International Conference on Computer Application and System Modeling (ICCASM 2010),

[2] Zachary D. Lund, “A VoIP Implementation on an Embedded Platform,” IEEE Marquette University ,2009

[3] Samrat Ganguly, Sudeept Bhatnagar , "VOIP: wireless, P2P and New Enterprise Voice Over IP," England; Wiley, 2008.

[4] Divya G.S, Dr.P.C.Srikanth, “Embedded VoIP Communication System with Graphical User Interface Features” 4th ICCCNT 2013 July 4-6, 2013, Tiruchengode, India .

[5] Jan Janak, SIP Introduction, Copyright © 2003 FhG FOKUS

[6] Mark spencer, “Asterisk”, the future of IP telephony

[7] Uben Upton, Gareth halfacree, “Raspberry Pi User Guide”, Wiley Publication

[8] Eli Shoval, “Signal Processing Technologies in Voice over IP “,AudioCodes Ltd,2006

[9] Alex Rober, “FreePBX Powerful Telephony Solutions”, Aug 2009, pub. Ref.1200809

[10] Ankita Bhondge, Aditi Bhatkar, Sapna Fender, Sonali Thakre, Megha Goel ,” Interactive Voice Response System by Using Asterisk”, International Journal of Innovative Research in Computer and Communication Engineering(An ISO 3297: 2007 Certified Organization)Vol. 3, Issue 3, March 2015

[11] Daniele Rizzetto et al, “A voice over IP service Architecture for integrated Communication”, IEEE Networks, May/June, 1999, pg. 34-40

[12] J. Rosenberg/H. Schulzrinne/G. Camarillo/A. Johnston/J. Peterson/R. Sparks/M., “SIP: Session Initialization Protocol”, Network Working Group

[13] Bill Douskalis, “IP Telephony – The Integration of Robust VoIP Services”, Prentice Hall PTR 2000.

[14] Ur Goode, “Voice over Internet Protocol (VoIP)”, Proceedings of the IEEE, Vol.90, No.9, September 2002, pg. 1495-1517