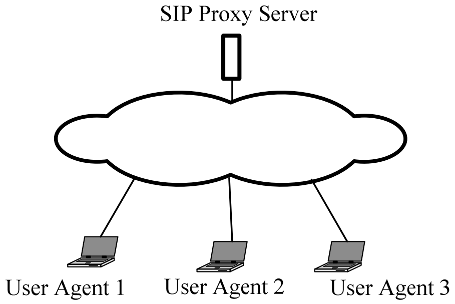
“Voice Call Establishment and Conference Call over IP, a Hands-on SIP-Based Experiment”

Project Objective

The objective of this project is to learn the implementation of Voice over IP (VoIP) using SIP connections and a conference calling through a hands-on experiment over Ad-Hoc networks. The learning process is carried out by utilizing a server and a few VoIP users as clients. The server uses Asterisk software.

Hardware Setup and Tool Downloads

A. **Hardware Setup.** Use three machines or virtual machines for **Part I** of the project: one as the SIP proxy server and two VMs as user agents or VoIP clients. Use four VMs for **Part II** of the project by including an additional VoIP client as shown below.



B. SIP Proxy Server Preparation

- **Install Linux OS.** Install Linux (Ubuntu) on the laptop that you want it acts as SIP server (Linux platform is necessary).
  Note: There is a possibility that Server won’t work properly on a particular Linux machine, so install Linux on each machine as a backup.

- **Download Asterisk Software on Server Laptop.** Asterisk software is used for telephone systems which allow attached telephones to make calls to one another. One of the important features of Asterisk nowadays is its support of VoIP protocols (such as SIP). Do the following steps:
  a) Go to link http://downloads.asterisk.org/pub/telephony/asterisk/old-releases/ and download asterisk-1.6.1.6.tar.gz. (Reference: http://www.asteriskguru.com/tutorials/)

  b) Extract the software from the downloaded compressed file,
  c) We have to install some files required to make the Asterisk server work. These files are called the “dependencies” of the server and they are found by doing:
  ➢ sudo apt-get update [Press Enter Button]
  ➢ sudo apt-get -y install swig g++ automake1.9 libtool python-dev libcppunit-dev sdcc libusb-dev libasound2-dev libsdl1.2-dev python-wxgtk2.8 subversion guile-1.8-dev libqt4-dev ccache python-

```
opengl libgsl0-dev python-cheetah python-lxml libqwt5-qt4-dev libqwtplot3d-qt4-dev qt4-dev-tools
fftw3-dev doxygen python-numpy [Press Enter Button]
```

- **Install the Server.** The following command gets the application (extracted server file) and then installs it:

  ```
  sudo apt-get install asterisk   [Press Enter Button]
  ```

a) At this point, we are in the root directory and we have to go to the asterisk server directory. We can enter the asterisk server directory by doing:

```
cd /etc/asterisk   [Press Enter Button]
```

b) In the asterisk directory, there are two files, which are of our interest. The "sip.conf" file and the "extensions.conf". To enter the "sip.conf" file, do:

```
sudo gedit sip.conf     [Press Enter Button]
```

and a new "sip.conf" file will open. To add anything in this, scroll down to the end of the page. Similarly, to enter the "extensions.conf" file, do:

```
sudo gedit extensions.conf  [Press Enter Button]
```

and a new "extensions.conf" will open. To add anything in this, scroll down to the end of this file.

- **Configuration of sip.conf file.** In the sip.conf file, we can configure everything related to the SIP protocol such as adding new users. Example:

```
[general]
port=5060;Port  to bind to (SIP is 5060)
 bindaddr=x.x.x.x; x.x.x. x = Asterisk server IP address
 allow = ulaw ; Allow all codecs
```

The sip.conf file starts with a [general] section with the default configuration for every user and peer (providers). These default values can be overwritten in the particular configuration of each user or peer, which will be shown below. SIP servers use port 5060 UDP. "bindaddr" is the address of the server. This address can be found by writing the "ifconfig" command on the terminal window. "allow" tells us that which codecs are allowed.

Now we create two new users: At the end of the sip.conf file, we add the following lines to add two SIP users, which will be communicating with one another through the sip server:

```
[2000]
username=2000
type=friend
secret=password
host=dynamic
context=from-sip

[2010]
username=2010
type=friend
secret=password
host=dynamic
context=from-sip

[2020]
username=2020
type=friend
```

```
secret=password
host=dynamic
context=from-sip
```

'secret' can be anything of your choice. (Note: a space after password may also be a part of it so press enter after you finish password.)

[2000] and [2010] are the contexts, which will be defined in "extensions.conf" file. Username is the name that is displayed on the soft phone when you are connected to the server. "user" type is used to authenticate incoming calls, "peer" for out coming calls and "friend" for both. In our example we have a "friend" extension to make a call. "Secret" is the password used to authenticate. In this case, we have used "password". We have put "host=dynamic" meaning that the telephone will be able to connect from any IP address. We can limit this user to access with only one IP address or a domain name. If we put "host=static" it would not be necessary that the user will register itself with the password provided in "secret". Finally "context=from-sip" shows the context where the instructions for this extensions will be executed in "extensions.conf" file.

- **Configuration of extensions.conf file.** The "extension.conf" file is made up of contexts between [ ] brackets. It is written in the following format:

  ```
  exten => extension, priority, command (parameters)
  ```

  "extension" is the caller number, "priority" is the order in which the commands will be executed under a given context, "command" is the thing to do, such as dial, hang-up, etc.

- Asterisk server is run by a command : sudo asterisk -r

C. Client Preparation (Windows OS platform)

Download and Install X-LITE 4.8.4: Download X-LITE soft phones from
(http://www.counterpath.com/x-lite.html) and install it on the two laptops that act as client computers.

Experiment

NOTE: Use a direct LAN cable connection connecting to WAN if the Ad Hoc network is not possible.

Part I.

This part requires two SIP clients. To capture the snapshot of the activity of the experiments use WireShark. Initially, connect only the server and one of the clients, then see and record the stack overflow in the Wireshark (features, packets involved, ports connected, and any notable processes)

Part I.1. Establish and Analyze a Successful Call Between 2 SIP Clients. In this part of the experiment you will demonstrate how the two SIP clients are configured on the two laptops and how they register with the Server and also how a call will be made between them via server. For this, some changes have to be made in the "sip.conf" file and "extensions.conf" file of the asterisk server.

a) Since we have to establish a call between two SIP clients, sip.conf is configured as done in Step 5

b) Next we have to configure extensions.conf file. To enter this file, we have to use the following command: sudo gedit extensions.conf. Then we have to set extensions for the two SIP clients which we have made in sip.conf file. The extensions are set as follows:

```
[from-sip]
```

```
exten => 2000, 1, Dial(SIP/2000,20)
exten => 2000, 2, Hangup

exten => 2010, 1, Dial(SIP/2010,20)
exten => 2010, 2, Hangup
```

Notice, 2000 or 2010 are the numbers assigned to the two SIP clients. Number 1 has a priority over 2 such that 1 is executed first and then 2. Dial refers to dialing mode and hangup means the phone is now hanged. (SIP/2000,20) means that we are using SIP protocol for calling 2000 and 20 refers to the time in seconds. Similarly, (SIP/2010,20) means SIP protocol is used to call the phone with number 2010 for 20 seconds.

c) **Configuration of X-LITE Soft Phone**. Right click on softphone and choose SIP account settings. Then click add and configure the parameters as displayed below:

Account Name	Always “PBX”
Protocol	Read only. Always specifies “SIP”
Display name	Name by which your phone is displayed. Choose 2000 & 2010 for your 2 SIP phones
User name	Enter a unique numeric value. A good example would be 2000 for the first phone and 2010 for the second. Each phone should have a unique user name, although you can have more than one SIP phone with the same credentials. But enter according to what we have configured in our sip.conf file.
Password	Passwords should be set according to we have configured in our sip.conf file. In our case it’s “password” for both SIP clients.
Domain	IP address our server.
Domain Proxy	Under this check on the box register with domain and receive incoming calls
Set outbound via	Check the domain box.

You should now be able to boot up your SIP phones and call each other. The numbers you would dial are whatever values you keyed in to the User Name fields. So in our example SIP phone 2000 can dial 2010 and SIP phone 2010 can dial 2000.

d) After configuring everything, use the “reload” command so that the changes made can take effect.

Part II.

Following experiments need three SIP clients. The third client can be included as user 2020 in **sip.conf** file from Part I.1, as follows:

```
[2020]
username=2020
type=friend
secret=password
host=dynamic
context=from-sip
```

The **extensions.conf** file should be added with user 2020 information to extensions.conf file of Part I.1.

```
exten => 2020, 1, Dial(SIP/2020,20)
exten => 2020, 2, Hangup
```

These settings are applicable to all experiments in this part.

Part II.1. Call on Hold. Assume user 2020 tries to call user 2000, when the call between user 2000 and user 2010 is already established. When user 2020 tries to call user 2000, it keeps user 2010 on hold and accepts the call from user 2020. And after completing the call with user 2020, it resumes a call with user 2010.

Part II.2. Call Conferencing._In this scenario, user 2020 tries to call user 2000, when the call between user 2000 and user 2010 is already established. When user 2020 tries to call user 2000, user 2000 keeps user 2010 on hold and accepts the call from user 2020. Then, user 2000 invites user 2010 again to join the pre-established call. As a result, a call conference is established.

Project Grading, Due, Report, and Assistance

a) For each part of the Experiment, observe and record the results. Express your observations and show snapshots of your results. Also show snapshots of all the connections activities to between the SIP clients via SIP Proxy server (Everything from Registering the SIP client to the termination of a call). Use Wireshark to capture these connection activities. The capture should show various SIP packets exchanged between the clients through server using timing diagrams feature of Wireshark.

b) Show and explain the logical flow chart shown in the Wireshark menu. Watch the SIP packet in the Wireshark and write a short summary on different fields in the packet of SIP.

c) For each part of the experiment, attach the snap shot of the MAC address of the server, and IP addresses of clients.

d) Describe the problems (if you faced any) and how you managed them while performing this project.