INTEGRATED MASTER (MSC) IN
ELECTRICAL AND COMPUTER ENGINEERING
NETWORKS AND INTERNET SERVICES

VoIP

Project

Paulo Rogério Pereira, SEPTEMBER 2019

1. Objective

This project has as objective developing an *Interactive Voice Response* (IVR) system based on Voice over IP (VoIP) and understand the protocols used.

The system should be based on an Asterisk PBX [1] and accept multiple user in voice calls and/or voice and video calls.

The SIP [2] signaling protocol should be used to manage voice calls. Sound and video transmission should be made over RTP/UDP [3].

A report should be written describing how to install the system and how the protocols work, based on time diagrams.

2. Architecture

The architecture required for this project is shown in figure 1. An Asterisk PBX (Private Branch eXchange) runs in a Linux machine, being available in the laboratory internal network. This PBX supports multiple protocols, allowing the registration of SIP phones such as the X-Lite [4] or phones such as the Cisco 7905G. For this project, you are recommended to use the laboratory PC under Windows to run X-Lite and either the Asterisk server in the nscotia server within the laboratory network or an Asterisk server in a Linux virtual machine.

Each telephone has an extension assigned, compatible with the PBX configuration. Although the SIP protocol supports alphanumeric extensions, to allow dialing in telephone numeric keyboards, it is preferred to use only numeric extensions.

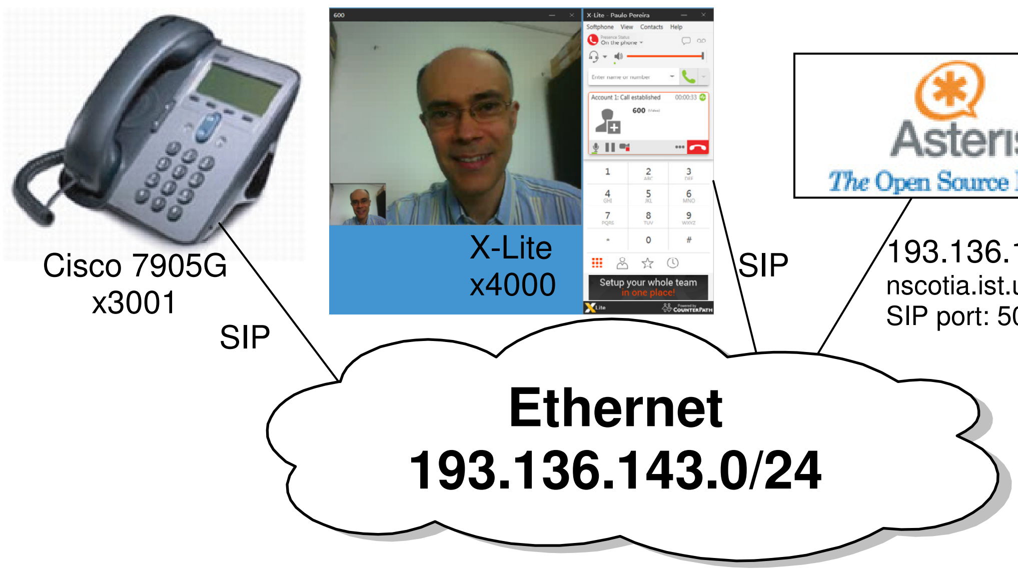


Figure 1: Network Architecture.

When run, the telephones register themselves in the PBX using the SIP protocol, allowing the acceptance of telephone calls from other phones connected to the same PBX, using the laboratory local network. The PBX operates as a SIP proxy server and SIP registrar. All the signaling traffic is exchanged through the PBX.

3. Specification

The PBX should allow users to establish phone calls with other users using extensions 300X or 400X.

The IVR application should work at the extension corresponding to the smallest student number from the students' numbers of the group members.

The menu system should support the following options:

1. Vote on your favorite football club.
2. Know the current voting results.
3. Play the Artillery Game.
4. Obtain the information regarding a Portuguese Zip Code.
9. Connect to the operator.
0. Hang up.

A call arriving to the menu system should get a ringing tone and be answered 3 seconds after it arrives.

After answering, the main menu should be reproduced in a cycle, repeating the instructions until a key is pressed. Video should not be used. You may use Portuguese, English or a mix of both. After each option, the user should be returned to the main menu, unless the call is hang up.

For the voting, it is enough to have 2 different voting options. In each telephone call, it should be possible to vote only once. The voting results should be global to all phone calls.

For the Artillery Game, the user is told that he has a cannon with range 2500 meters, the distance to the enemy target, which is a random number between 100 and 2500 meters and is given 3 shots to try to kill the enemy. For each shot, the uses chooses an angle between 1 and 89 degrees for his cannon. The distance reached by his shot is given by the following formula:

$$2500.\sin(2.\theta/(180/\pi))$$

where θ is the angle in degrees chosen by the user. The user is told by how much his shot went over the target or by how much his shot went short of the target. The shot is successful if it is within 45 meters of the target and the game ends with a victory. If the user does not destroy the enemy target with the third shot, then the enemy destroys him and the game ends with a defeat. After the end of the game, the winner is announced and the call should return to the main menu.

For option 4, the information can be retrieved from the page:
http://www.ctt.pt/feapl_2/app/open/postalCodeSearch/postalCodeSearch.jspx

An Asterisk Gateway Interface (AGI) script should be used to get the web page in real-time, process it and store the text in an Asterisk variable. Check the AGI chapter in [7][8][9]. Then, the text to speech (TTS) for Asterisk using Google Translate interface [10] should be used to provide the information to the user through voice. The information about a Portuguese Zip Code can be retrieved by sending an HTTP POST request to address:
http://www.ctt.pt/feapl_2/app/open/postalCodeSearch/postalCodeSearch.jspx
sending a string with the zip code digits in the message request body. For the 1049-001 IST zip code, the string to be sent in the POST would be:
cp4=1049&cp3=001&method%3AsearchPC2=Procurar

and the string to be passed to the TTS interface would be:
INSTITUTO SUPERIOR TÉCNICO. Avenida Rovisco Pais, 1. Lisboa. 1049-001 LISBOA.

The operator should be extension 4000, corresponding to another telephone, to which the call should be redirected when option 9 is chosen in the menu. Music should be played to the user while he waits for the operator to pick up the call. If the operator does not answer the call within 10 seconds, the user should be informed that the operator is not available. In either case, the call should return to the main menu in the end.

4. SIP

The SIP (Session Initiation Protocol) [2] protocol is a signaling protocol that allows managing sessions between a set of participants. The sessions can be telephone calls, text messages, multimedia contents distribution, or multimedia conferences.

The SIP protocol uses a syntax similar to the HTTP protocol. The request messages starts with a line identifying the type of message (REGISTER, INVITE, BYE, etc, method), the SIP domain and protocol version. Then, the message continues with header lines, a blank line and an optional message body.

Responses to request messages have a response code and a textual description, for instance "100 Trying" answer or "200 Ok", followed by header lines, as for request messages. Answers 1xx are informative. Answers 2xx are success. Any other answer should be considered unsuccessful.

Some messages, such as the INVITE message and the corresponding answer messages, after the header, use the SDP (Session Description Protocol) [5], providing information about the voice/video flows that should be established.

5. RTP

The RTP (Real-time Transport Protocol) protocol [3] is appropriate for real time data transfers, such as audio and video, offering sequence number association mechanisms, coding format identification and temporal information. RTP leaves the processing of this information to the application, by not providing by itself any delivery, sequence or quality of service (QoS) guarantee mechanisms. Note that RTP can be used over any transport protocol, being normally used over UDP.

6. Asterisk

For the Asterisk PBX to work, it should be configured through a set of files in the `/etc/asterisk` directory, from which the following stand out:

extensions.conf	definition of extension number contexts and their association to telephones;
sip.conf	definition of SIP users and their association to an extension from an extension context;

A more detailed description on how to configure Asterisk can be found in [7][8][9]. Annex B of [7] has a detailed description of the applications that may be used in the Asterisk scripts. It is possible to get a similar description with the command "`core show application <application name>`" on the Asterisk command line interface.

To use video, it is necessary to include in the sip.conf file the following option [11]:

```
videosupport=yes
```

The sound files used by Asterisk are by default in directory /usr/share/asterisk/sounds

Note that the Asterisk PBX runs as user "asterisk", so read permission should be given to all users for the sound/video files that Asterisk should have to read (and to all directories until the directory where the files are); and write permission to all users should be given to the directory where Asterisk should write the message recordings or other files. It is recommended that global variables (e.g. PATH) are used to define the directories where the students' files are, to ease the project installation process.

Asterisk has preconfigured: extension 1000 for the welcome menu, extension 600 for echo test, extension 1235 for direct access to leaving a message in Voicemail and extension 8500 for accessing VoiceMail. There is a mailbox created with username "1234" and password "4242". Voice (and video, if available) messages left in the voicemail are stored in directory /var/spool/asterisk/voicemail/default/1234, which can be used to generate menus with voice. Current Asterisk versions do not reproduce stored videos correctly, so you should use menus only with voice.

If Asterisk is running as a *daemon*, a console may be created with the command (run as sudo, except on the nscotia server):

```
asterisk –vvvvr
```

where the number of "v"s corresponds to the amount of output for debugging (verbosity) during Asterisk operation.

Some interesting commands at the Asterisk console are:

```
help
sip show users
```

```
sip show peers
sip show channels
sip reload
sip set debug
dialplan reload
dialplan show globals
core set debug 4
core show application Background
channel request hangup all
core show applications like music
core show applications describing file
core show function LEN
quit
core stop now
```

7. Implementation Details

It is recommended that you implement your project in the following phases:

1. installation of the Asterisk PBX and a softphone (X-Lite), if not working in the laboratory. The necessary packages are: asterisk, and the corresponding dependencies. The following packages are optional: asterisk-mp3, mpg123, sox, libsox-fmt-all, wireshark.
2. creation of the main menu with only menu options 9 and 0.
3. study of messages exchanged.
4. writing of the report.
5. implementation of the remaining menu options.

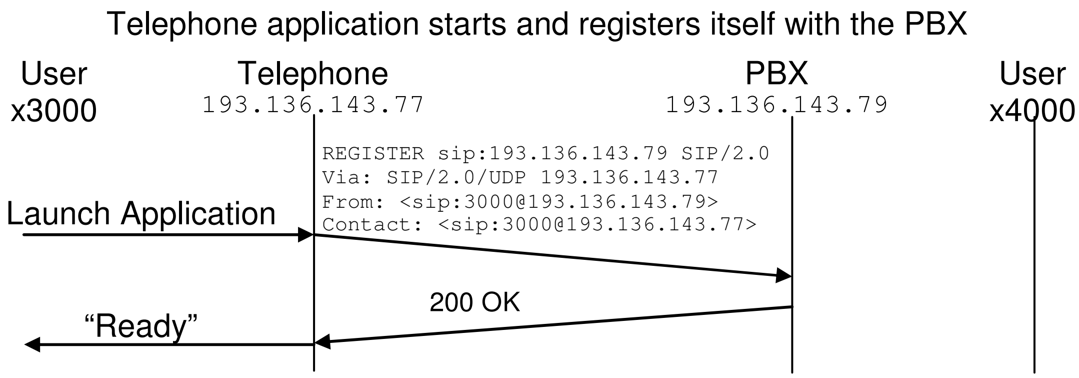
Note that the objective of the project is not producing nice recordings, but understanding the communication protocols and how an IVR system may be implemented. Anything that is not explicitly requested, should be implemented in the simplest possible way.

All SIP users should use the "demo" context and have a password configured. The example "sip.conf" file may be used as a starting point.

It is recommended to use the Wireshark [12] program to analyze the SIP, SDP and RTP traffic.

The report to be delivered should include:

- instructions on how to install the project, meaning: how the project should be installed from the ZIP file delivered, having the Asterisk and softphones already installed;
- illustration of the protocol stacks used, both for signaling and for sound and video stream transfer;
- a description of the operation of the protocols based on temporal diagrams, which should include the user, the telephone(s) and PBX, as well as the most relevant message parameters for the different situations of the project operation. An example (to be completed) is:



- the time diagrams should cover at least the following situations:
  - SIP user authentication;
  - phone application starts; user calls the menu system and the call is answered; user puts the call on hold; user resumes the call on hold; user disconnects the call; user chooses menu option "0"; user calls the menu system and cancels the call before it is answered; phone application ends;
  - if the user chooses a menu option, observe and describe the difference in the traffic when you change the X-Lite configuration in "Preferences" – "Calls" – "DTMF" from "Send via RFC 2833" (by default) to "Send via INFO";
  - the users chooses the connect to the operator option and the call is answered; for this case, observe and describe the differences in the traffic when a line "directmedia=no" is added to the "sip.conf" file for the users involved in the call; present new time diagrams for this case.

8. Project delivery

The project should be delivered through the Fenix project delivery system, in a ZIP file, containing the following files:

- Asterisk configuration files modified;

- sound files necessary;

- report in PDF format describing how to install the project, and with a description of the operation of the protocols used, supported in message time diagrams;

- any other files considered necessary to compile or run the applications.

The ZIP file should have a name as follows:

```
RSI-VoIP-<student number>-<student number>-<student number>.zip
```

Where "<student number>" should be replaced by the group students' numbers.

The deadline for delivery is:

22/12/2019, 23:59h, Delivery of the complete version of the project.

9. References

[1] Asterisk, the Open Source PBX. http://www.asterisk.org/
[2] SIP: Session Initiation Protocol. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler. IETF RFC 3261, June 2002.
[3] RTP: A Transport Protocol for Real-Time Applications. H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. IETF RFC 3550. July 2003.
[4] CounterPath X-Lite Softphone. https://www.counterpath.com/x-lite-download/
[5] SDP: Session Description Protocol. M. Handley, V. Jacobson. IETF RFC 4566, July 2006.
[6] RTP Profile for Audio and Video Conferences with Minimal Control. H. Schulzrinne, S. Casner. IETF RFC 3551. July 2003.
[7] Asterisk, The Future of Telephony, 2nd Edition. Jim Van Meggelen, Leif Madsen, and Jared Smith. O'Reilly, 2007. ISBN: 978-0-596-51048-0. http://cdn.oreillystatic.com/books/9780596510480.pdf
[8] Asterisk: The Definitive Guide, 3rd Edition. Leif Madsen, Jim Van Meggelen, Russell Bryant. O'Reilly, 2011. ISBN: 978-0-596-51734-2. http://asteriskdocs.org/

[9] Asterisk: The Definitive Guide, 4th Edition, Russell Bryant, Leif Madsen, and Jim Van Meggelen. O'Reilly, 2013. ISBN: 978-1-449-33242-6.
[10] Text to speech for Asterisk using Google Translate. http://zaf.github.io/asterisk-googletts/
[11] Asterisk video http://www.voip-info.org/wiki/view/Asterisk+video
[12] Wireshark Network Protocol Analyzer. http://www.wireshark.org/