VoIP Cookbook:
Building your own Telecommunication Infrastructure

By
Onno W. Purbo
Anton Raharja

Edited By
Nurlina Noertam

Funded By
Internet Society Innovation Fund (ISIF)

One Destination Center
December 2011

Table of Contents

ABOUT THE AUTHORS 1
PREFACE 2
CHAPTER 1: VoIP Overview 3
- How VoIP Works for Dummies 3
- Where to Start? 4
- What Is Internet Telephony? 5

CHAPTER 2: Becoming a user 7
- PC to PC Internet Telephone Call 7
- Using softphone 11
  - Installing X-Lite 11
  - X-lite Configuration 15
  - Install Ekiga 19
  - Configuring Ekiga 19
  - Configuring Account in Ekiga 27
- Test your SIP Softphone 30

CHAPTER 3: VoIP Hardware for experienced Users 35
- Linksys PAP-2 Analog Telephone Adapter 36
- Linksys IP Phone SPA 941 41
- WiFi IPPhone 46
  - Linksys Wireless-G IP Phone 47
  - Hewlett Packard Ipaq 6395 56
    - Activating Ipaq 6395's Wireless Capability 56
    - Running SJPhone 58
    - SJPhone Features 64
    - Using SJPhone to place call through Ipaq 6395 65
  - Nokia 68
    - Nokia Wireless Configuration 69
    - SIP Server and Account Configuration in Nokia E61 73
    - Internet Telephone Configuration in Nokia 76
    - Registering to VoIP Softswitch 77
    - Calling using Internet Telephone in Nokia E61 80
- VoIP in ADSL Modem 82
  - ADSL Modem Configuration 83
  - VoIP Configuration in Linksys WAG54GP2 86

CHAPTER 4: Interconnectivity and Telephone Number Allocation 93
- Getting Free Washington State Telephone Number 94
- Free Internet Country: Country Code +882 97
- Introducing your country code to International VoIP network 104
- VoIP Rakyat's ENUM

.....106
Connecting to PSTN and Cellular Using VoIP Discount.....116
VoIP Cheap.....118
CHAPTER 5: Asterisk Softswitch.....120
Minimal Resource for Asterisk 121
Asterisk Installation.....121
Compile Asterisk.....122
Configuring Asterisk.....124
ENUM.CONF Configuration.....124
SIP.CONF Configuration 125
EXTENSIONS.CONF Configuration.....126
CHAPTER 6: Asterisk for Incoming and Outgoing calls 129
Defining SIP Channel in sip.conf 129
Asterisk as SIP Client.....129
Generic SIP configuration 131
DAHDI Usage For VoIP Cards.....141
DAHDI Architecture.....142
Kernel.....142
Tools.....142
DAHDI Sample installation.....143
DAHDI extensions.conf.....146
CHAPTER 7: Briker Softswitch.....148
Briker's Installation Process.....148
Briker's Console.....154
Briker's Web Configuration.....156
Zaptel Configuration.....159
SIP Trunk.....160
IAX2 Trunk.....163
H323 Trunk.....165
ZAP Trunk.....167
Outbound Routes.....168
Inbound Routes.....170
Interactive Voice Response.....171
Setup Recordings.....171
Ring Groups.....172
Pin Sets.....174
CHAPTER 8: OpenSIPS High Performance Softswitch.....175
Compile OpenSIPS.....175
Prepare User Database Server.....176
Use opensipsctl.....178
Some Routing Technique in OpenSIPS.....178
How to route to PSTN and Cellular.....179
How to route using Area Code for interconnected SIP Servers.....180

How to route ENUM Query in OpenSIPS....181
Test ENUM Query in OpenSIP....181
ENUM Routing Table in OpenSIPS configuration....182
CHAPTER 9: ENUM....184
Example of ENUM Service....184
Delegation Concept in ENUM....184
ENUM Implementation....186
BIND Installation....186
Setup BIND for ENUM Server....186
Test DNS for ENUM Query....188
ENUM Delegation in BIND....189
CHAPTER 10: Conference Server on Asterisk....191
Configuring Conference Room MeetMe....191
Configuring Dialplan for Conference....192
Activating Conference while Operating....193
CHAPTER 11: Trunk Peering in Asterisk....195
CHAPTER 12: NAT and Firewall....196
CHAPTER 13: Voicemail in Asterisk....198
CHAPTER 14: More on Asterisk's Dialplan....201
Pattern Extension....201
Attaching context....201
The Extension Pattern....202
Extension....203
Predefined Extension Names....203
Defining Extension....204
An interesting Extension Examples....206
Variable and Equation....208
Reloading....208
Forwarding to another Asterisk....208
CHAPTER 15: VoIP IP PBX Hardware....210
Linksys SPA9000....210
Linksys SPA9000 Configuration....211
Configuring VoIP on Linksys SPA9000....214
CHAPTER 16: Analog Telephone Adapter for connection to PSTN....219
Linksys SPA3000 Analog Telephone Adapter....220
Configure Linksys SPA3000....221
Linksys SPA3000 ATA Status....225
LevelOne VOI-2100 Analog Telephone Adapter....227
Linksys SPA400 with four FXOs....246
Using the SPA400 with Asterisk....246
Configure Asterisk to talk to Linksys SPA400....248
Connect PSTN using Linksys SPA9000 and Linksys SPA400....251
Configure Linksys SPA9000 to talk to Linksys SPA400....260

CHAPTER 17: OpenBTS....261
Open GSM Infrastructure....261
History....261
Field Test....261
Niue....262
GNURadio....262
Library Installation....263
WxWidget Installation....263
SWIG Installation....264
QWT Installation....264
GNURadio Installation....265
USRP Handling....265
USRP Verification....266
OpenBTS Installation....269
A Glimpse on OpenBTS Configuration....270
smqueue Configuration....272
Asterisk Configuration to work with OpenBTS....273
Automatic SIM Registration....274
OpenBTS Operation....275
CHAPTER 18: Peering Among Providers....276
Free SIP Proxy Servers....278
Becoming a Peer in SIP Network
....278
CHAPTER 19: Internet Telephony Bandwidth 280
Coding Decoding (CODEC)....280
Mean Opinion Score (MOS)....281
MOS and R Factor values for G.711, G.723, and G.729....283
Calculating The Required Bandwidth....284
Calculation for Call Center....287
VoIP Capacity Planning....289
CHAPTER 20: VoIP Evaluation....293
Evaluate VoIP Performance using VQManager....293
VQManager Installation 293
Some of the Important Scripts of VQManager....294
Activate VQManager Web Service....295
Changing the Monitored Interface....303
Inserting new Interface 303
Monitor VoIP Performance....304
Evaluate VoIP Performance using SIPp....313
Installation of SIPp....313
Installation of SIPp Webfrontend....313
Transaction Oriented Test using SIPp....314
Access to the SIPp Webfrontend....317

CHAPTER 21: VoIP Troubleshooting....328
CODEC and Vocoder....328
Preparing A VoIP Ready Network....329
Minimal requirement / configuration....329
Test prior to operation of the system....329
Some Useful References For VoIP Troubleshooting....330
References....331
VoIP Hardware....331
VoIP Softswitch....331
VoIP Client Software....331
Testing Software....331
APPENDIX A: Example of /etc/sip.conf....333
APPENDIX B: SIPp COMMANDS....343
APPENDIX C: File /usr/local/etc/opensips/cfg-test-uas.cfg....350

ABOUT THE AUTHORS

Onno W Purbo is a techie who wrote many practical ICT books. He has been a popular speaker at many seminars and conventions. He withdrew from his position as an Indonesian civil servant and retired as a lecturer at the Bandung Institute of Technology, becoming an ICT activist. He has written thousands of articles and papers and authored more than forty books on ICT and therefore received several awards, including a Sabatical Award from the International Development Research Center (IDRC), a Canadian Crown corporation aimed at helping developing countries use science and technology to find practical, long-term solutions. His profile is at http://opensource.telkomspeedy.com/wiki/index.php/Onno_W._Purbo

Anton Raharja is the founder of the largest community based SIP Softswitch VoIPRakyat in Indonesia. He is also the lead developer of Briker, an open source SIP softswitch appliance. Besides Briker, Anton actives in developing several open source applications, such as, Play SMS (SMS Gateway), PlayVoIP (the VoIPRakyat Engine), PlayBilling (Internet Cafe Billing System), WiFi Rakyat etc. He has served in many talk and seminars on VoIP and Open Source software. He is currently the Technical Director of PT. Jelajah Media Informatika, WAN-DKI, Jakarta and the CEO of PT. Infotech Media Nusantara, Jakarta. In 2008, he received a FOSS Award from the Indonesian Ministry of Information and Communication. His profile is at http://www.antonraharja.web.id/curriculum-vitae/

PREFACE

This book is aimed to provide a practical knowledge to setup a community based telephone network based over the Internet Infrastructure A.K.A. Internet Telephone or Voice over Internet Protocol (VoIP). Many real world example on equipment and application software setup and installations are provided.

We would like to thank many friends at http://www.asterisk.org, http://www.opensips.org, http://www.voiprakyat.or.id, http://www.e164.org as well as many forum and mailing lists without whom it would be impossible for us to gain a lot of knowledge and ideas.

I would like to thank many of our comrades that managed to keep their spirit high in making a significant change in Indonesian telecommunication area. Some of them are Sumaryo, Donny BU, Basuki Suhardiman, Hariyanto Pribadi, M. Ichsan, Heru Nugroho, Michael Sunggiardi, and Judi Prasetyo; as well as many friends on the mailing lists.

Onno W. Purbo would like to thank the International Development Research Center (IDRC) http://www.idrc.ca to support his earlier work on VoIP. Especially to ICT4D group, specially, Richard Fuchs, Renald Lafond, Graham Todd, Josh Skinner, Steve Song, Nancy Smyth, Heloise Emdon, Mireille Leroux and Frank Tulus.

We would like to thank Information Society Innovation Fund – ISIF http://www.isif.asia, especially Sylvia Cadena and her team for supporting us in documenting our knowledge on community based Internet Telephony.

We hope this book will enable more community based telecommunication and telephone providers over the regional Internet. Furthermore, we hope it will enable a low cost access to telecommunication in the region.

Jakarta, February 2011
The Authors

CHAPTER 1: VoIP Overview

In many countries, specially, the developing countries, people seeks for low cost communication solutions. Today, Internet is becoming more accessible for many people and corporates in these countries. Having access to Internet, one may easily deploy Telephone network over the Internet infrastructure. It is known as Voice over Internet Protocol (VoIP) also known as Internet Telephony.

The cuurent VoIP technology is quite advanced. It is currently similar if not more advance and may replace the existing telephone technology even recognizing the +<country code> <area code> <phone number> format. Interestingly, most of the technology is open source and readily available on the Internet. Furthermore, our experience shows that the current Softswitch performance on a Xeon Server machine is fairly similar to medium size Telco switch. Thus, it would be beneficial for those who wish to implement telecommunication infrastructure to seek solution in VoIP technology.

This book is aimed to provide a practical knowledge to setup a community based telephone network based over the Internet Infrastructure A.K.A. Internet Telephone or Voice over Internet Protocol (VoIP). Many real world example on equipment and application software setup and installations are provided. It is hoped to enable more community based telecommunication and telephone providers over the regional Internet. In the end, it will enable a low cost access to telecommunication in the region.

How VoIP Works for Dummies.

A overly simplified figure on how VoIP network work is shown in Figure 1.1. The heart of VoIP network is the softswitch. It stores all information on the subscribers. In a simple view, a VoIP softswitch basically has a table mapping the phone number of the subscriber and the computer or IP address of the subscriber.

Everytime, a subscriber wants to make a call to another subscriber. The client equipment will ask the softswitch the destination address of the other subscriber. The destination address can be an IP address. Thus, the softswitch basically store in its table, the phone number of the subscriber and their IP address.

VoIP will be more fun, as we can use IP Phone instead of a computer as subscriber equipment. IP Phone looks similar to normal phone. However, it is much smaller than a computer. Thus, the client equipment may be run 24 hours without consuming too much electricity.

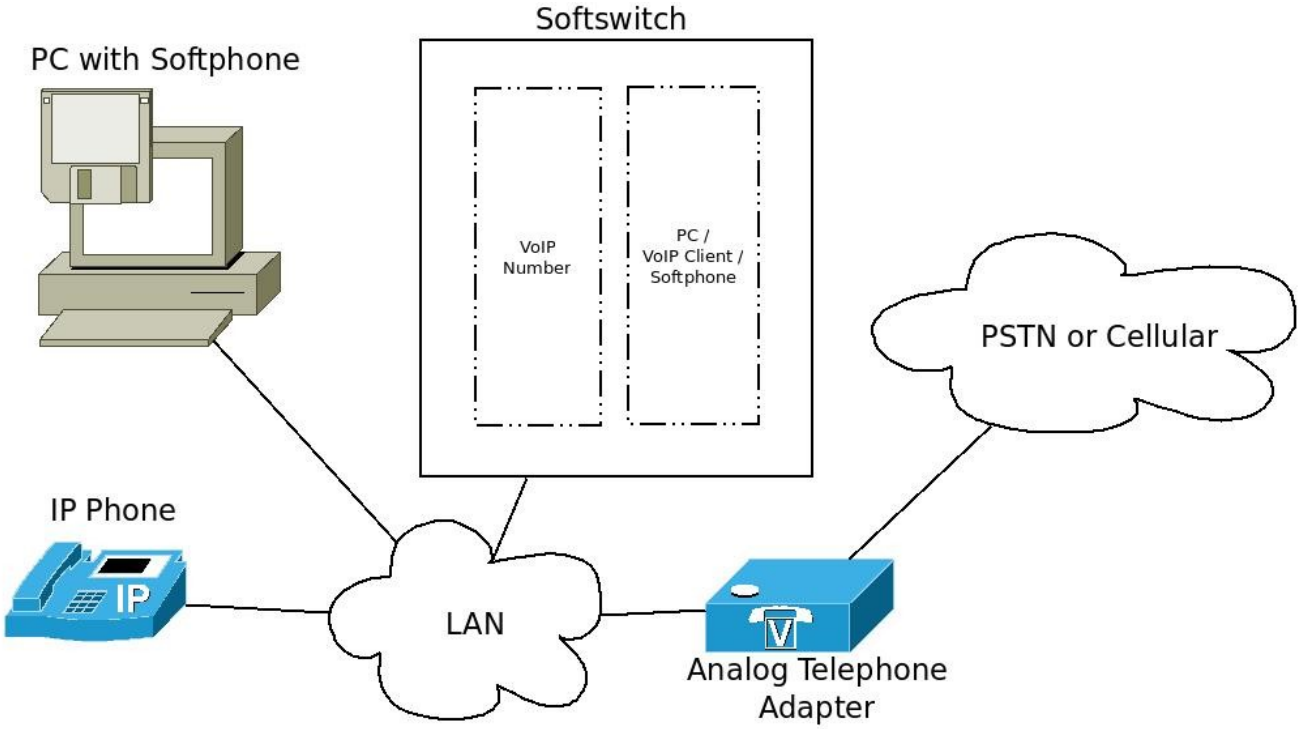


Figure 1.1 How VoIP Works.

For more advanced user, we may insert an an Analog Telephone Adapter (ATA) into the network. An ATA is another type of client equipment. It may act as gateway between VoIP network and legacy phone network. Thus, anyone on VoIP network may call to the old phone network.

Where to Start?

The book is designed to meet the need for

- Those who wish to try and to become a VoIP user only.
- Those who wish to explore on how to setup more advance VoIP user appliances.
- Those who wish to find VoIP corporate solutions.
- Those who wish to explore on setting up a homebrew softswitch.
- Advanced techies that wants to know in depth how to operate a Telco over Internet.

For VoIP newbie users, equiped with PC, sound card and access to the Internet, might want to read “Becoming a user” (CHAPTER 2) and little bit of “ Interconnectivity and Telephone Number Allocation” (CHAPTER 4).

For those who wish to explore VoIP appliances might interested in “ VoIP Hardware for experienced Users” (CHAPTER 3). Chapter 3 covers a lot of hardwares, including, IP Phone, Wifi Phone, Analog Telephone Adapter, ADSL Modem.

Those who are more interested in corporate solutions might be interested in “ VoIP IP PBX Hardware” (CHAPTER 15) and “Analog Telephone Adapter for connection to PSTN” (CHAPTER 16). Any materials on “ VoIP Hardware for experienced Users” (CHAPTER 3) would also help.

For those who wish to setup a homebrew VoIP softswitch, it is beneficial to read “ Briker Softswitch” (CHAPTER 7) and with little effort to read “ Asterisk Softswitch” (CHAPTER 5) and “ OpenSIPS High Performance Softswitch” (CHAPTER 8). For advance homebrewer a topic on “ENUM” (CHAPTER 9) might be of interest to set the system to recognize +<country code> <area code> <subscriber number> numbering format as used in Telco network.

The rest of the topics, such as, VoIP Bandwidth, conference server, detailed on dial plan, trunking, peering, evaluation of VoIP performance, VoIP troubleshooting are aimed for more advanced users that really wants to fine tune the Infrastructure.

What Is Internet Telephony?

In a simple definition, Voice over Internet Protocol (VoIP) or internet telephony is a telephone network over the internet (TCP/IP) network. Thus, you could use VoIP anywhere as long as you are connected to the internet.

There are two main internet telephony technology, i.e., H.323 and Session Initiation Protocol (SIP) that are frequently used. The former is an older standard developed by the International Telecommunication Union (ITU), the leading United Nations agency for information and communication technology issues. The latter, SIP, is a more advanced technology developed by the Internet Engineering Task Force (IETF), a large international community concerned with the internet architecture and its development. In short, it is sufficient for you to know that these protocols are the main engine of VoIP communication. This book will concentrate on SIP technology as it is currently the main engine behind many advanced VoIP deployment on the Internet.

How good the quality of VoIP communication depends on the type of Codec employed in a given communication. Short for Coding-Decoding, Codec is a process of turning analog signal to digital signal vice versa, allowing audio and video to be sent over the computer network. In such process, codec minimize the use of bandwidth for transferring the signal data while ensuring that the voice received remains clear. A variety of codecs have been developed.

Despite that VoIP communication can be provided for free, you still need to meet some basic requirements. They include the required equipments and software. At the very least, you need an IP-based network using TCP/IP and a computer with sound cards, headsets, microphone speaker and have the computer be connected to a network or the Internet. Softphone, the software required for VoIP communication, is provided for free.

If you have more money to spend, you can buy VoIP-ready equipments that can be operated with no need for configuration or very minimal configuration. In addition, you can avoid the hassle of turning on your computer each time you want to communicate through VoIP. At the minimum, you can buy an IP-Phone, a phone that can be plugged into LAN network. Some of these IP Phones have WiFi capability, allowing you to use the phone when connected to a hotspot network. There are many devices enabling VoIP communication, some of which may or may not need configurations.

If you're building a much more complicated network, you can implement IP PBX or Internet Telephony Gateway also known as Analog Telephony Adapeter (ATA), a medium between internet telephony network and conventional phone network.

CHAPTER 2: Becoming a user

Now that you know what VoIP is, you may want to learn how to communicate using VoIP technology. For practical reason, in this early stage of learning, we will use a SIP provider called VoIP Rakyat http://www.voiprakyat.or.id., in order to help you gradually understand how to use VoIP. The service is not favorable for users outside Indonesia, as VoIP Rakyat's server is physically located in Indonesia, thereby making the VoIP audio quality good for those who live in Indonesia but not so for users outside Indonesia. However, the knowledge gain from VoIP Rakyat experiences may be used for any available SIP providers in your country.

PC to PC Internet Telephone Call

This part will explain the simplest VoIP call technique using a computer to call another computer or VoIP network. All you need is a computer with a sound card, headset, microphone and internet connectivity. How much these equipments will cost depend on the specification you use. But since VoIP for personal use does not require sophisticated equipment, these equipments will not cost much. Of these requirements, bandwidth is perhaps the most important, as it determines how good the voice quality of your VoIP communication.

For this PC-to-PC communication, you need to register with a SIP provider. The one we provided as an example in this book is http://www.voiprakyat.or.id. Then you will learn how to install a softphone in your computer—the software required for VoIP communication—how to configure the softphone to register yourself using the SIP account you have created and how to use the softphone.

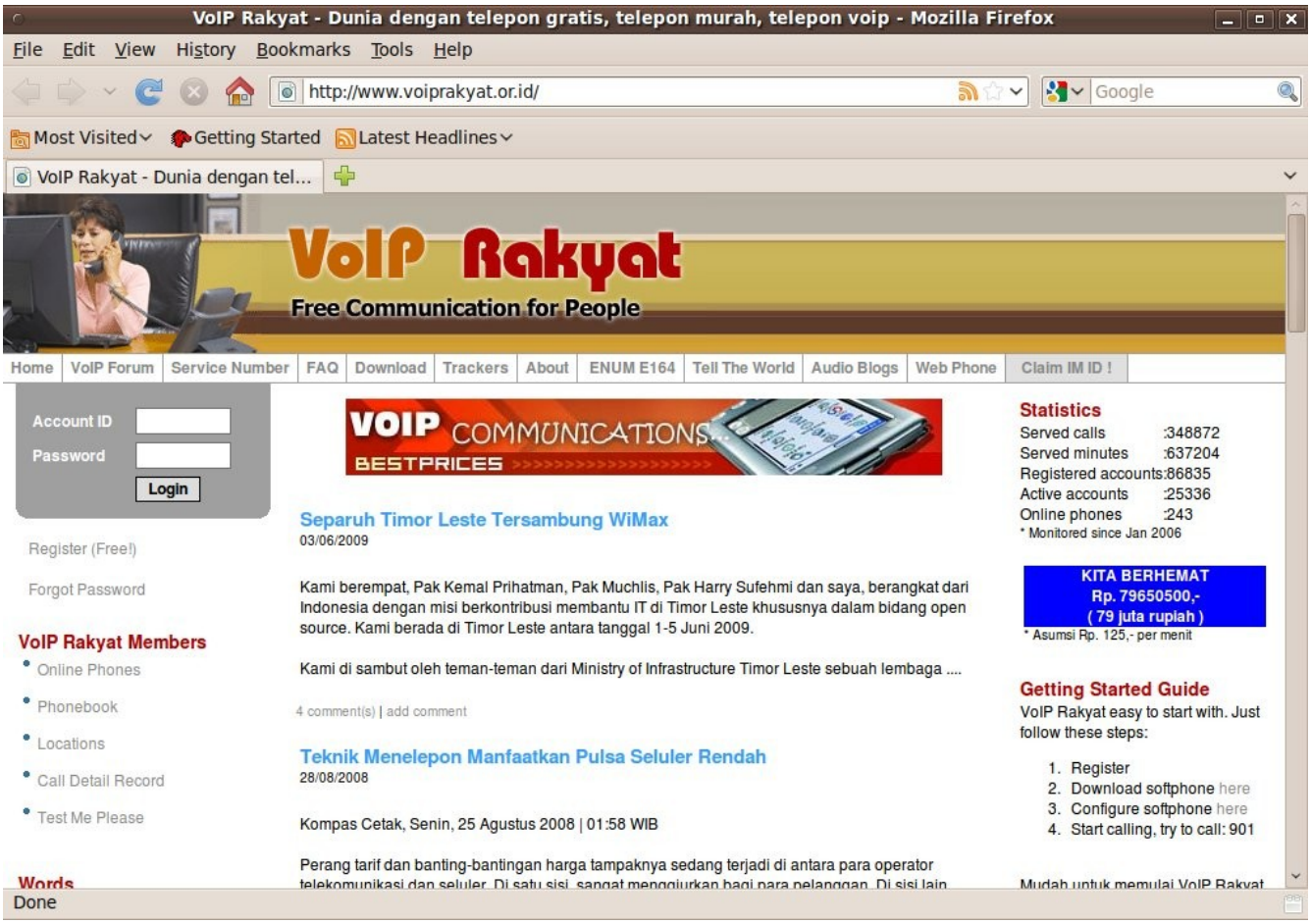


Figure 2.1 a free SIP provider called VoIP Rakyat

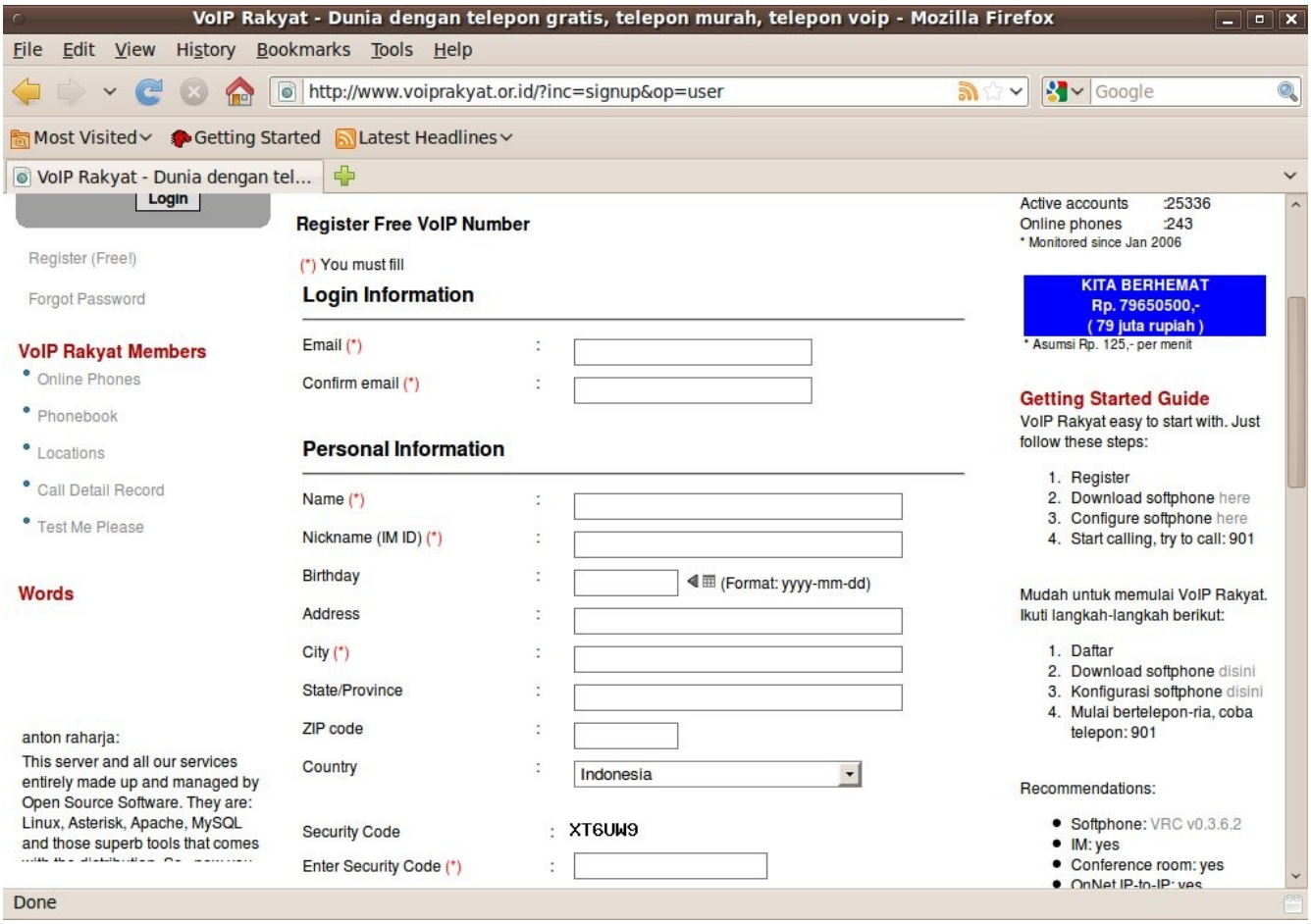


Figure 2.2 Registration Process in VoIP Rakyat

Click Register (Free) in order to obtain a free VoIP Rakyat number. With Register (Free) clicked, there are some information you have to fill in. These include your e-mail address, name, address, city and country. Nick Name field is provided for Jabber (chatting) account. At the end of registration process, we need to enter the provided Security Code.

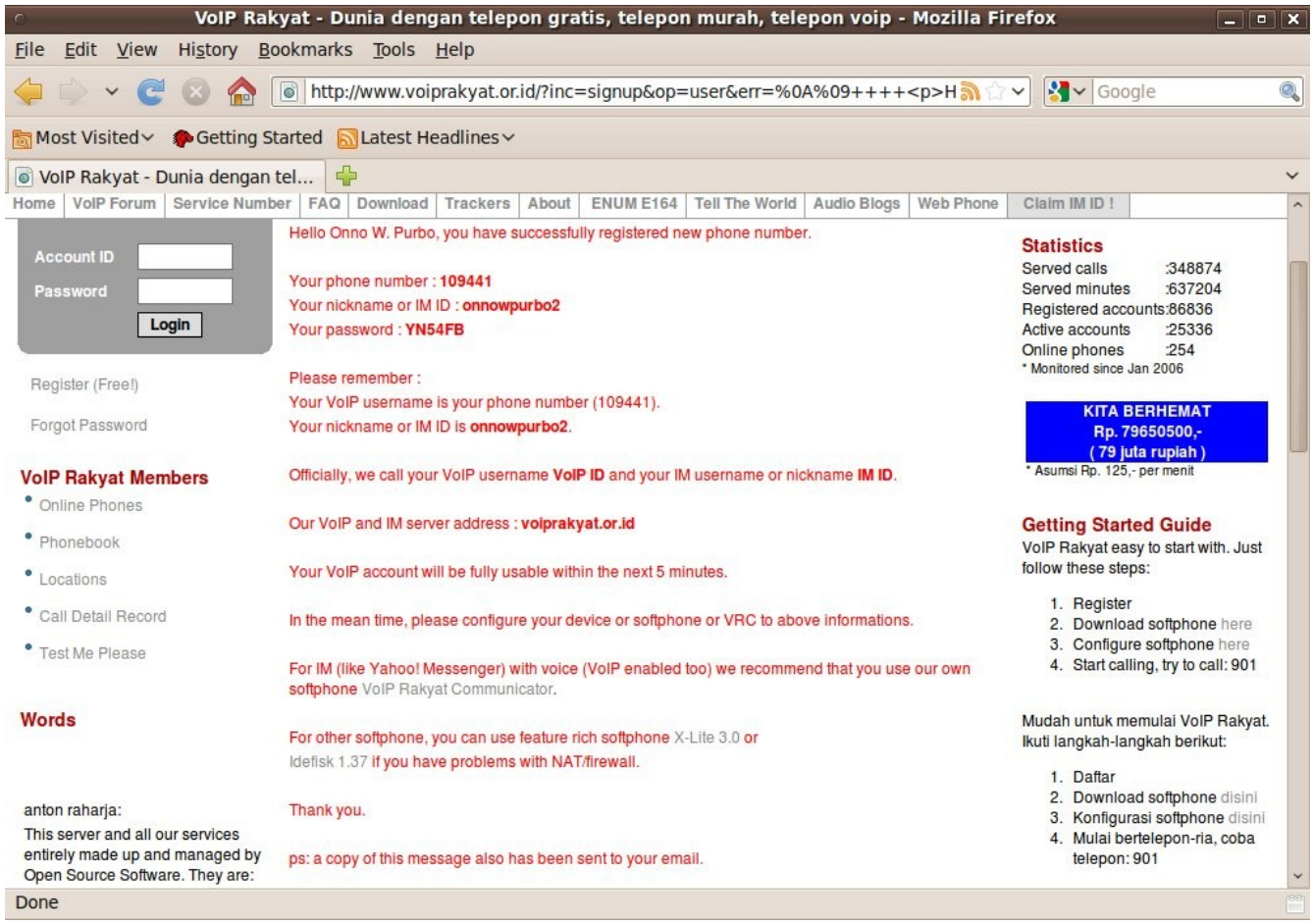


Figure 2.3 Successful Registation Process

After all information is filled in correctly, VoIP Rakyat provides us with a VoIP number, the password, Nick name required to allow us to make a call and chatting through VoIP Rakyat network. Please note that the server name is voiprakyat.or.id.

With the account provided, all we have to do is to transform our computer into telephone handset so that to can be used to call over the internet telephony network.

Using softphone

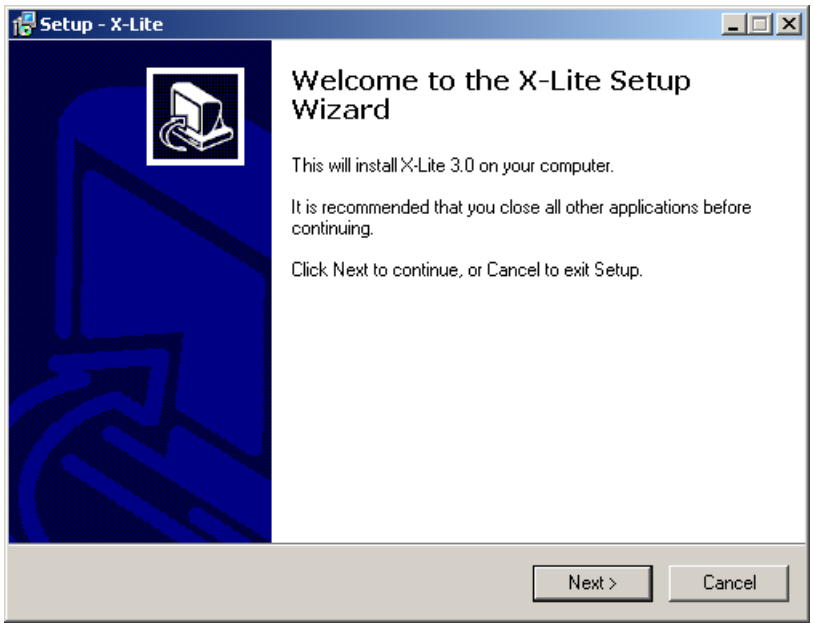
Select the right softphone for your computer. Most of these softphones can be downloaded from VoIP Rakyat http://voiprakyat.or.id/download/, or you can find each of them from its website.

- Cubix - http://www.virbiage.com/cubix.php
- Idefisk - http://www.asteriskguru.com/idefisk/free/
- SJPhone - http://www.sjlabs.com/sjp.html
- X-lite - http://www.xten.com/index.php?menu=download
- Ekiga - http://ekiga.org

You need only one of softphones, depending on whichever works or suitable for you;

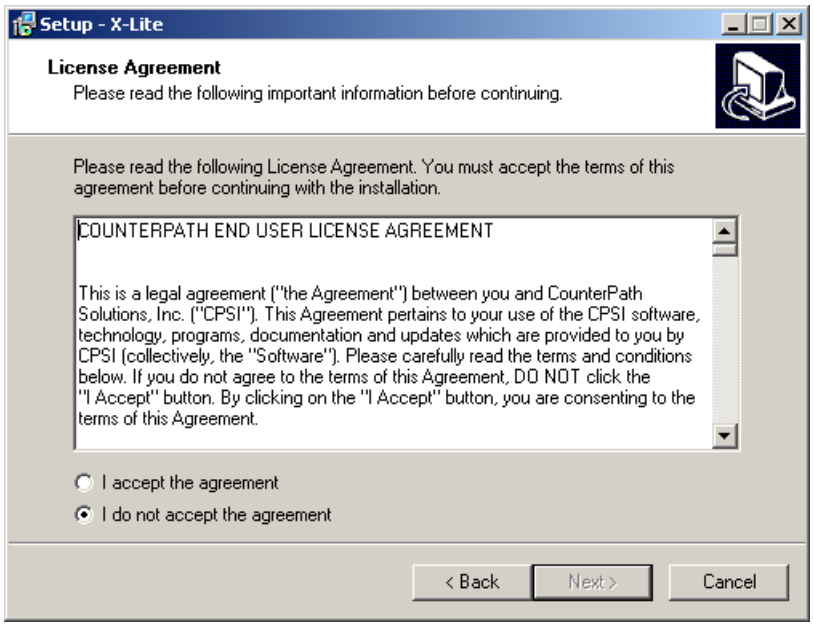
Installing X-Lite

Figure 2.4: X-Lite Welcoming Installation Window



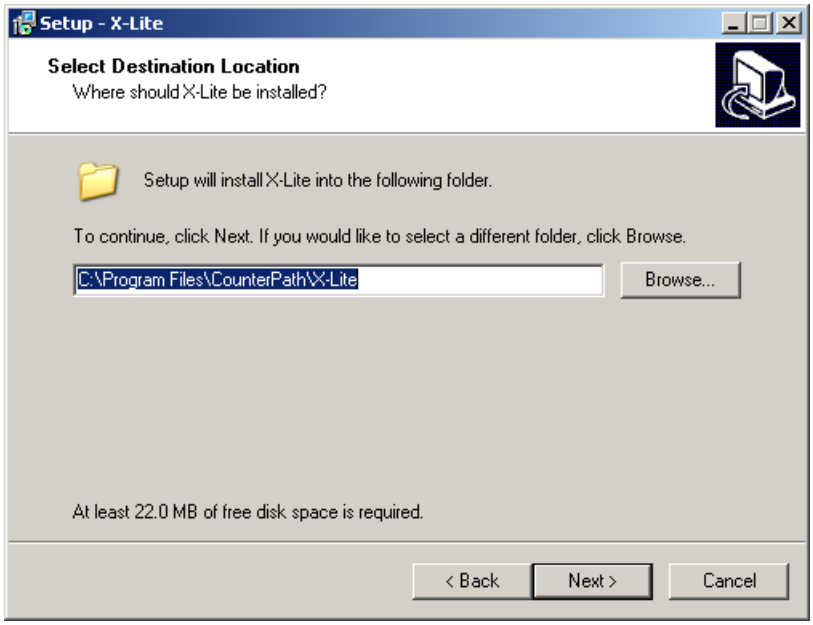
Once x-lite installer program is run, we will be directed to a Welcoming Dialog Properties. Click on "Next" to proceed to the next step of the installation process.

Figure 2.5: Counterpath End User License Agreement



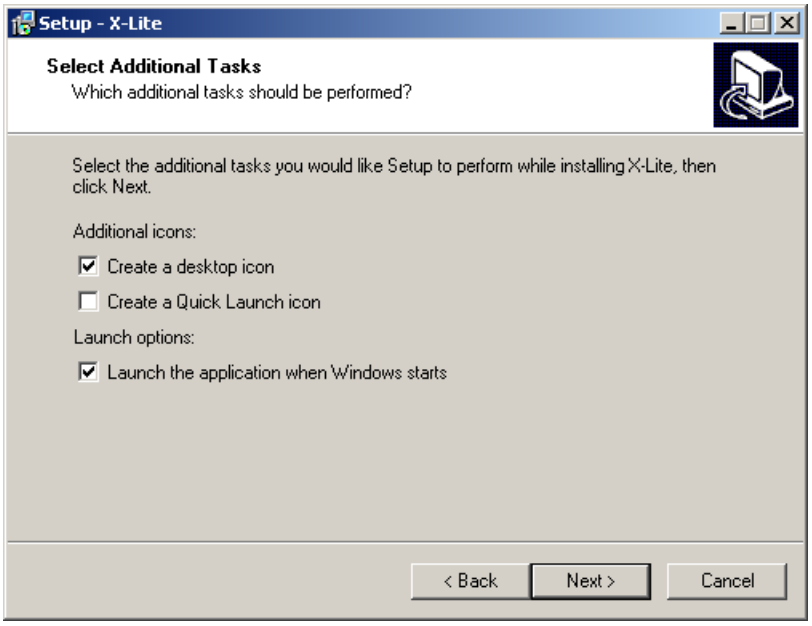
What appears next is the licensing agreement between x-lite creator and you being the user. This ensures that x-lite will not be liable for the poor VoIP voice quality produced by x-lite. Just like earlier, click on the “I accept the agreement” button and click Next.

Figure 2.6: Determine the location where the software will be installed



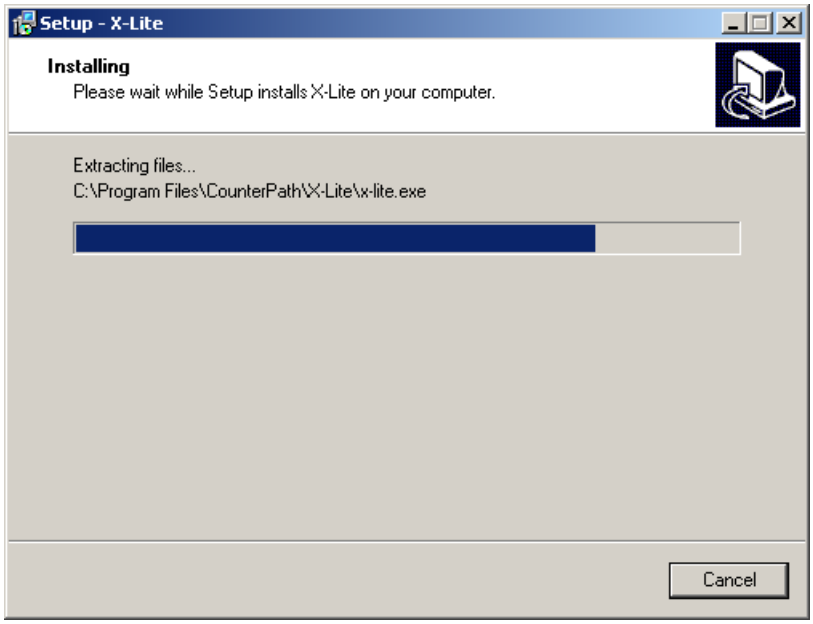
Next X-lite will ask where the program will be installed. The default folder is C:\ProgramFiles\CounterPath\X-Lite, as shown in Figure 2.6. You can change the folder if you want.

Whichever folder you choose, click Next to continue the installation process.



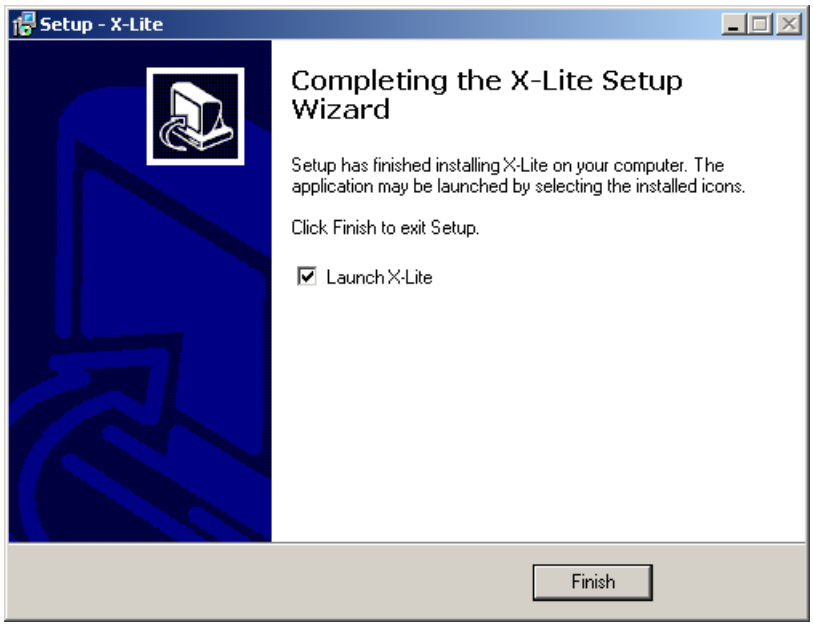
For quicker and easier way of using x-lite, you can add x-lite as a desktop icon or even set it to activate when Windows starts (See Figure 2.7). Click "Next" to proceed.

Figure 2.8: The dialog Windows indicating that installation is in progress



X-lite then extracts all files required for the program (See Figure 2.8).

Figure 2.9: The Windows showing that the installation process is completed



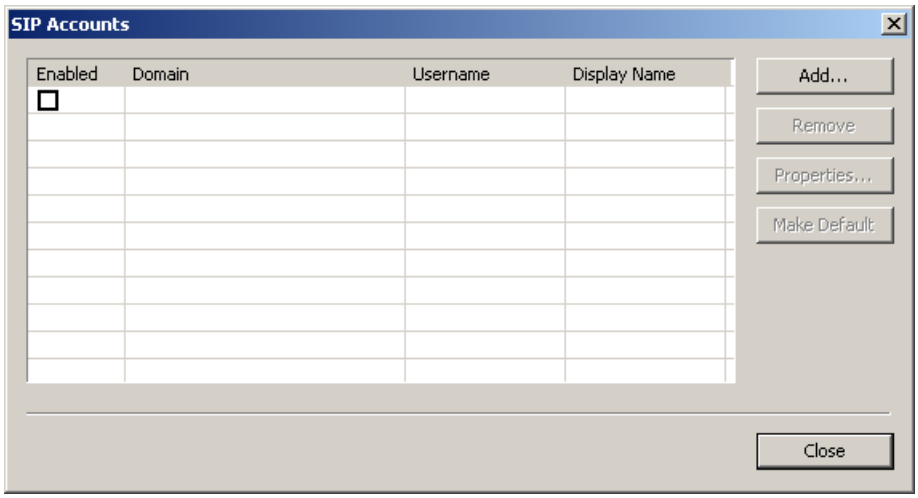
Once the installation process is completed, you can directly run X-Lite by checking the Launch X-Lite box and clicking Finish button.

X-lite Configuration

Although X-Lite could run the moment you completed the installation process, it does not mean you can use it immediately. You still have to configure the softphone. Its configuration menu can be opened by right-clicking on X-lite. X-Lite 3.0 has two lines that can be operated simultaneously. This implies we can establish two concurrent calls, each to different destination.

Figure 2.18 X-Lite appears just like an ordinary phone

Figure 2.14: Enter the SIP account you have created in X-Lite Configuration Dialog Window



In X-Lite 3.0 configuration, you can enter the SIP account(s) given by your provider. However, the free software version of X-Lite 3.0 seems to limit the number of SIP accounts only one account. The previous version, X-Lite 2.0, allows 10 SIP accounts to be stored and used. Click “Add” to enter the information of the SIP account you have just created in VoIP Rakyat or of any other SIP accounts.

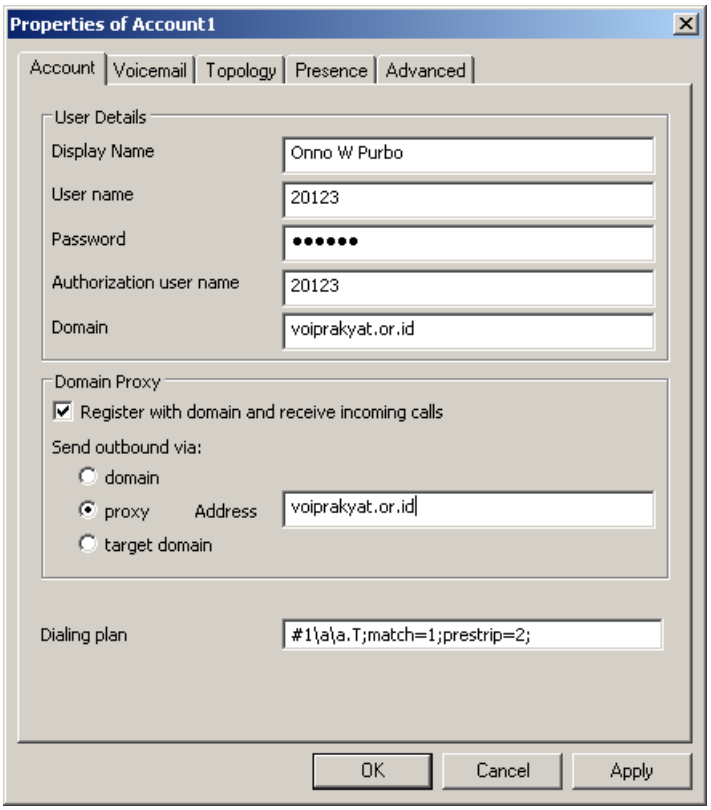


Figure 2.15: With the Add button clicked, you can see the Properties of the SIP account. What appears first is the Account tab.

In the account tab, you have to fill in your username, authorization user name, which is the phone number given by the provider, the password obtained from VoIP Rakyat or any other SIP providers; the proxy address, which is voiprakyat.or.id, the address of VoIP Rakyat. Other information you also have to fill are domain, which is voiprakyat.or.id, and Display name, any name you want to enter. This functions as a Caller ID in a telecom network.

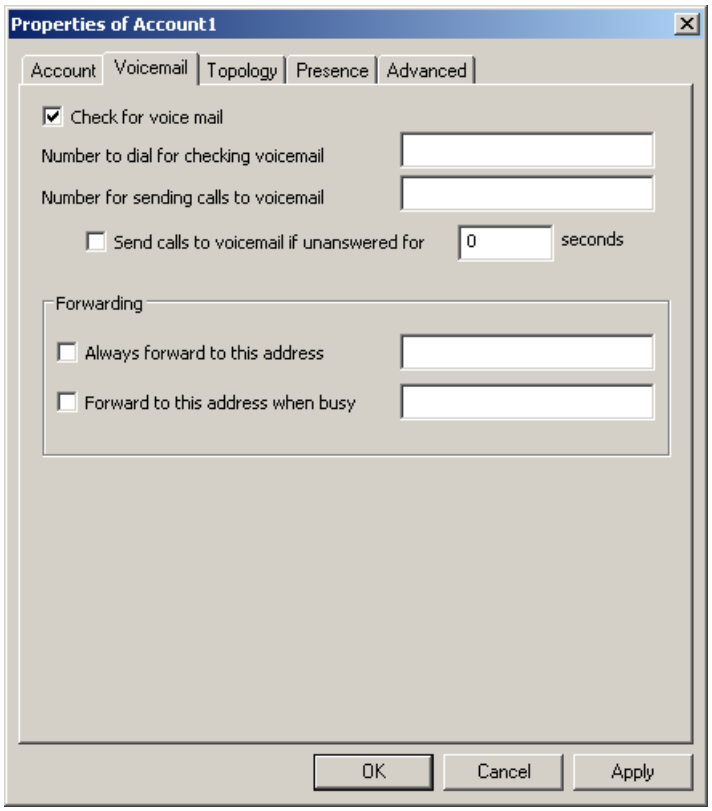


Figure 2.16: The Voicemail tab is used to set how you would like SJPhone manages your voicemail

In the Voicemail tab, we can determine to where we have to dial in order to listen to our Voicemail. For VoIP Rakyat, the number is 904. Enter this number into "Number to dial for checking voicemail". If you use othr provider, use the number provided by the provider instead.

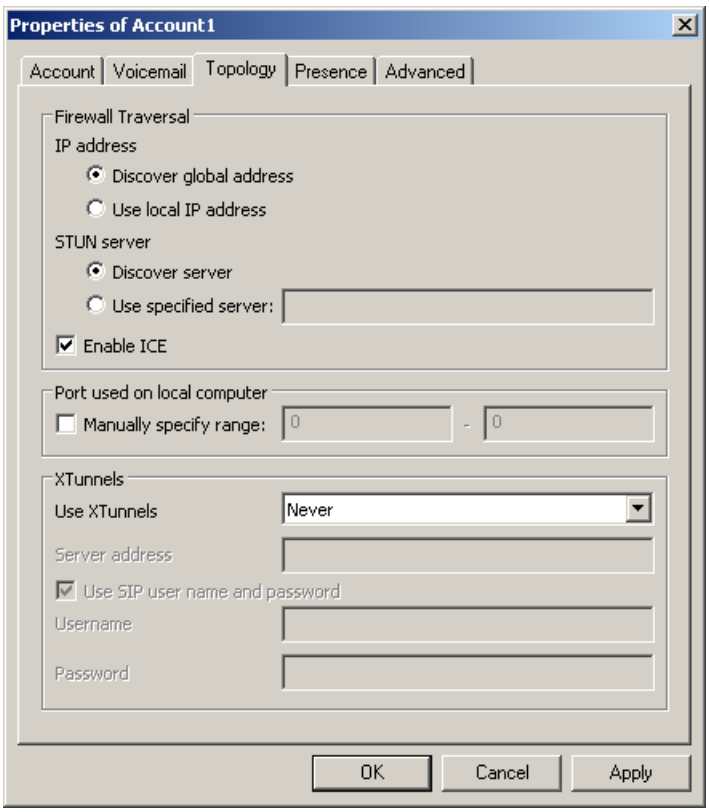


Figure 2.16: Set the parameters under the Topology tab to determine how SJPhone works with NAT/Firewall

In the Topology tab, you can activate X-lite's ability to penetrate Firewall/NAT, to identify the public IP address that is used and so on. You can also use the default settings that will automatically know the public IP address that we use. However, NAT may still be problematic, as not all configuration can be traversed by signaling protocol and media used a SIP provider.

For Presence and Advanced tabs, use the default values. Some parameters you can change are the time intervals used to periodically register our account to the SIP server. This ensures that the SIP account remains registered. After all configurations are completed, click "Ok" to activate the configurations.

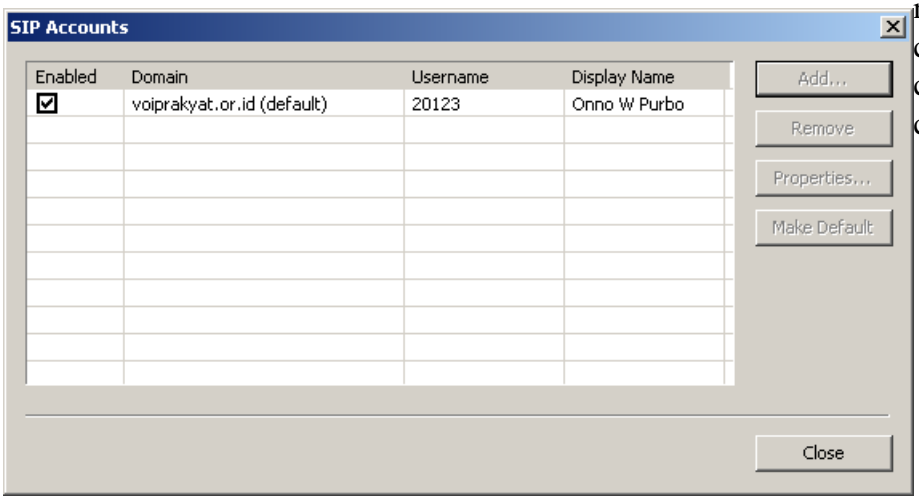


Figure 2.17: With the box under "Enabled" column ticked, you can now use your SIP account

Once everything is properly configured, the SIP account you have just configured will become available. Tick the box under the column “Enabled” to activate the account. Then click “Close” to close the SIP account menu.

With the configuration completed, you can now start using X-Lite. If the registration process is successful, you will get a message stating “Log in” on the softphone screen or otherwise the message states “Registration Error” and you have to check whether you have properly configured the softphone.

To place a call, click on the numbers already available or click the numbers on the keypad.

Install Ekiga

Ekiga (formely known as GnomeMeeting) is an open source SoftPhone, Video Conferencing and Instant Messenger application over the Internet. It supports HD sound quality and video up to DVD size and quality. Ekiga is interoperable with many other standard compliant softwares, hardwares and service providers as it uses both the major telephony standards (SIP and H.323).

To Install Ekiga in Ubuntu,

```
sudo apt-get install ekiga
```

Configuring Ekiga

Principally, We need to do

Ekiga -> Edit -> Accounts -> Add a SIP Account

The needed informartion would be

Name	: VoIP number
Registrar	: SIP Server
User	: VoIP number
Authentication User	: VoIP number
Password	: password VoIP

In the early start of Ekiga, we need to set several parameters. We may cancel the early configuration

process and do it later through Configuration Assistant menu from

Ekiga -> Edit -> Configuration Assistant

The detailed process of Configuration Assistance is as follows,

Figure 2.18 Welcome Banner

A welcome banner is shown from the Configuration Assistant menu. Press “Forward” button o move

forward the configuration process.

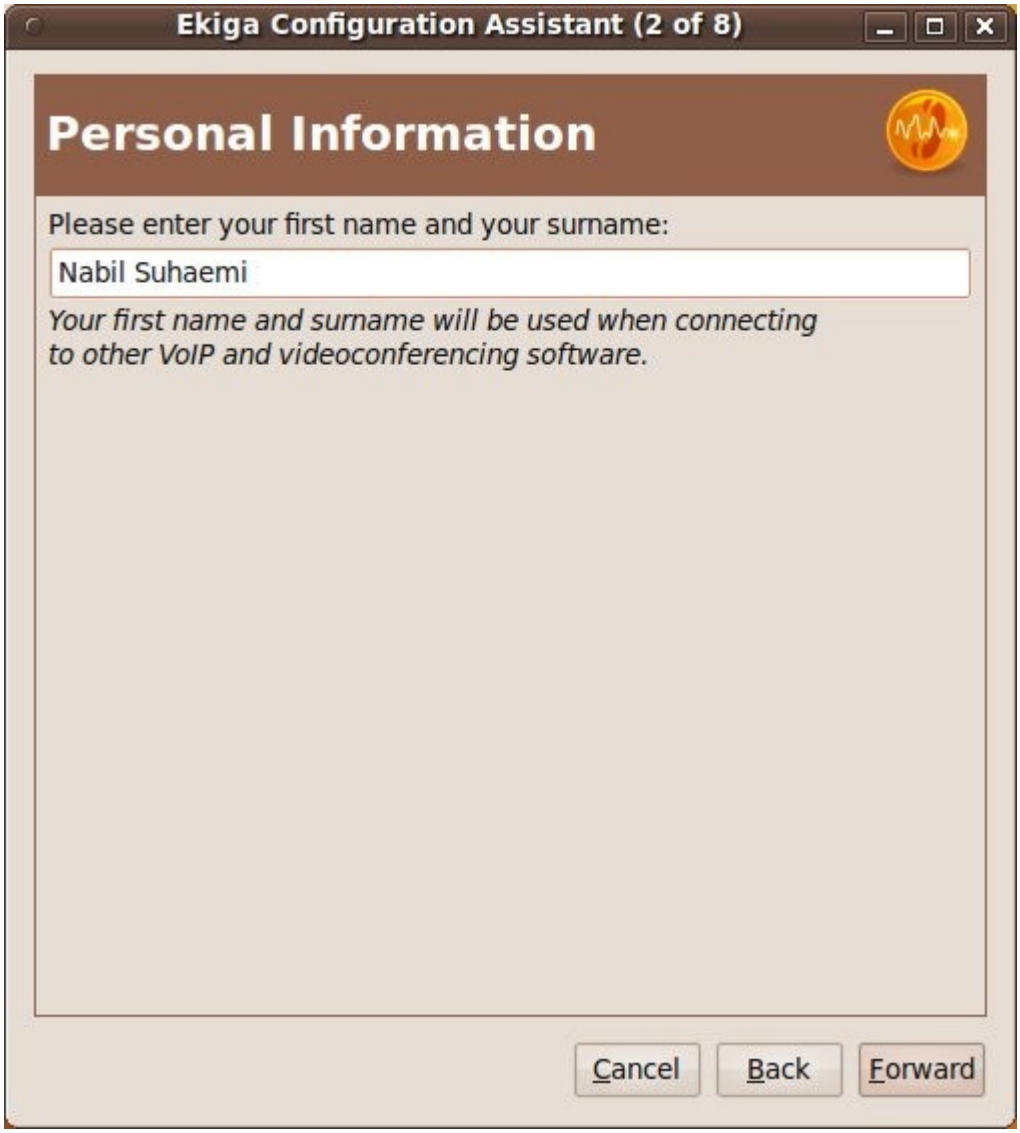


Figure 2.19 Enter Full Name.

The first step, we need to enter our full name into Ekiga. Then press “Forward” button.

Figure 2.10 VoIP Account

The next menu, we can submit our Account at Ekiga.net. Ekiga.net may offer an account to make callout call from VoIP. If we don't have any account at Ekiga.net, we may press “Forward” to continue.

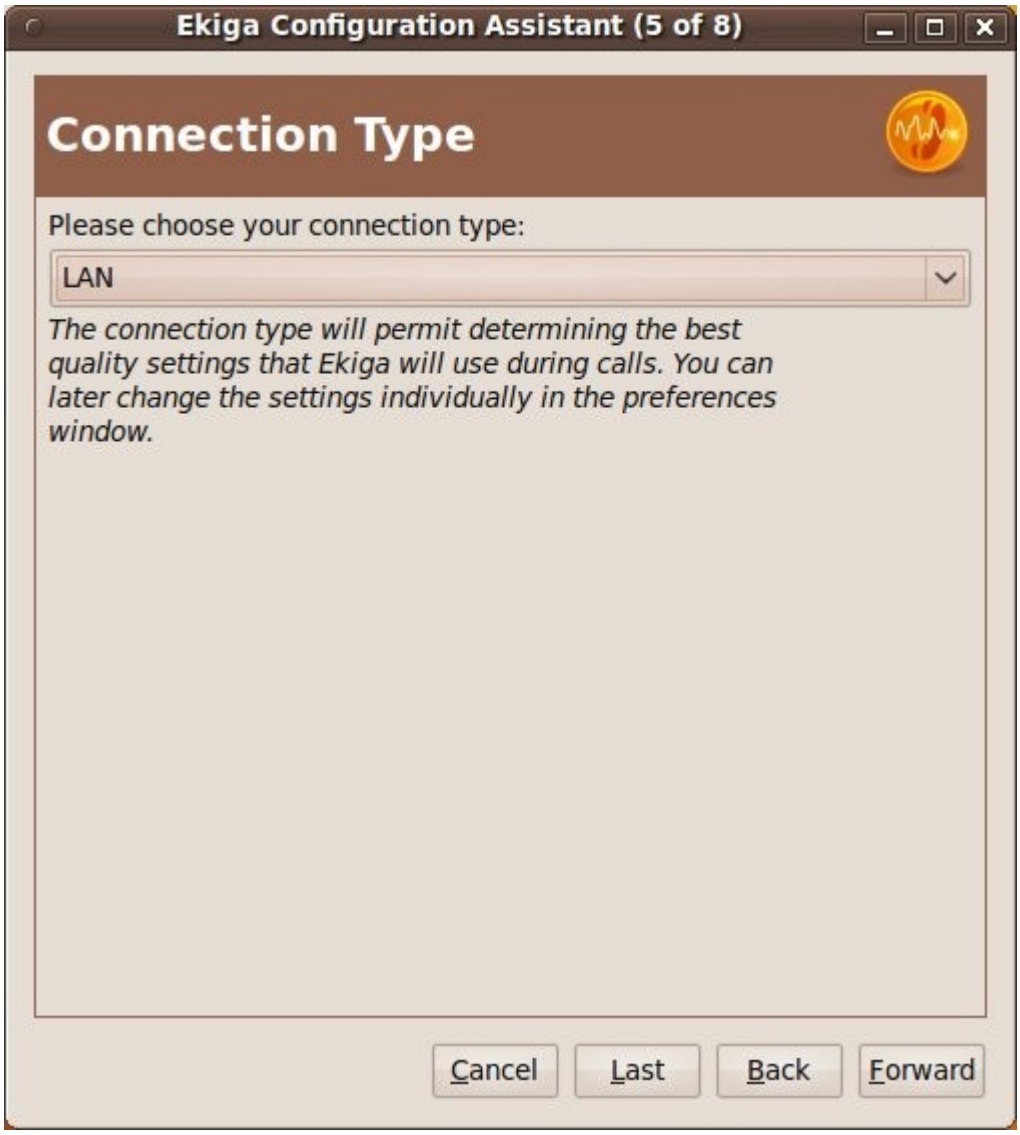


Figure 2.11 Type of Network.

Next, we need to set the type of network. This will affect the CODEC used to compress the audio. For a good performance in LAN environment, please select LAN. Press “Forward” to continue.

Figure 2.12 Type of Sound Card

Next, we need to set the type of sound card to be used in VoIP. Ekiga is fairly smart to detect the available sound card. We hardly need to choose or change the Ekiga's selected sound card. Next , we need to press “Forward” button.

Figure 2.13 Type of Video Card

Next, we can select the type of video device if one is connected. Ekiga is smart enough to detect any video device on the system. To continue, press “Forward”.

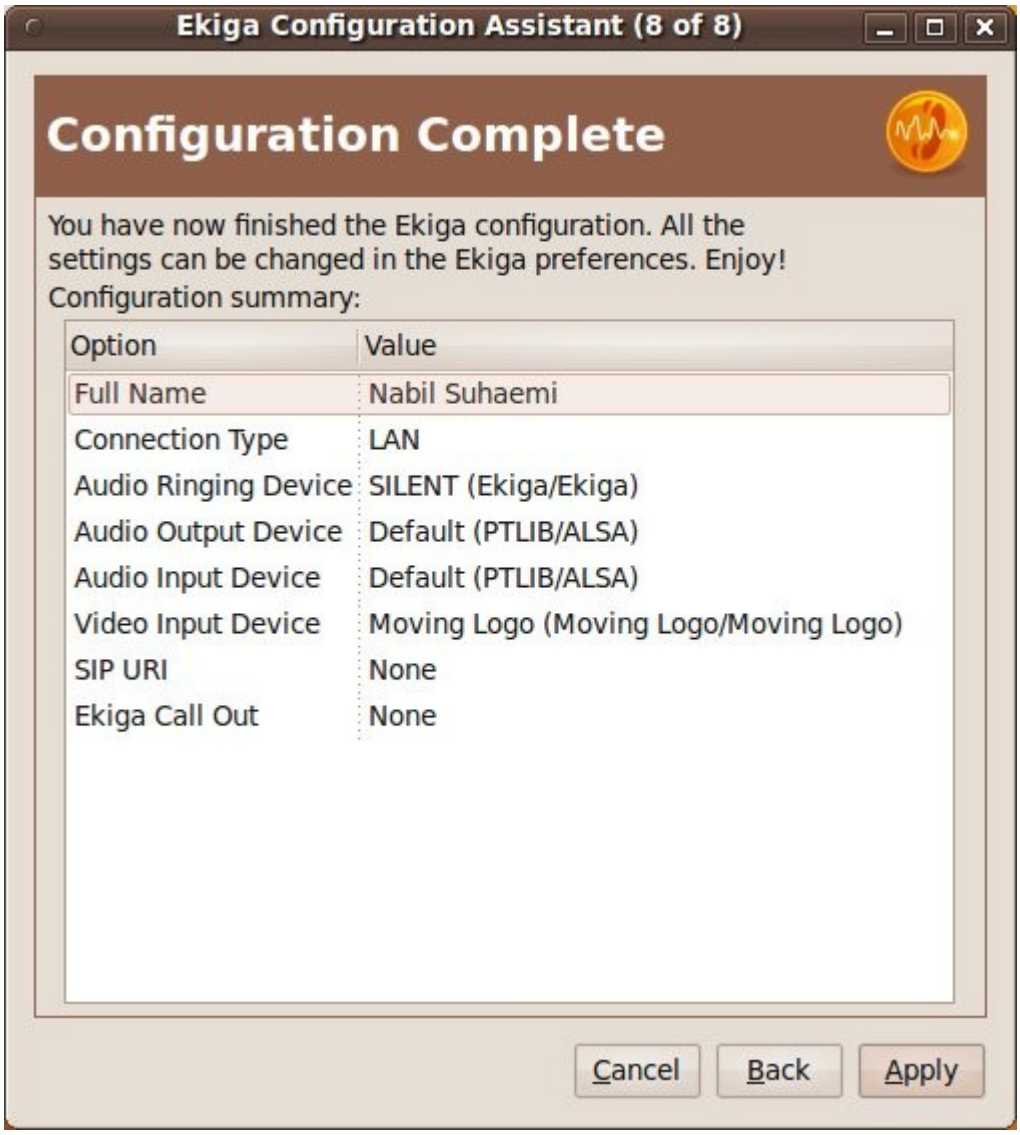


Figure 2.14 Finish.

Finally, the configuration process of Ekiga is completed. It will show the summary of the parameter in Ekiga. Press “Apply” to begin uses Ekiga.

Configuring Account in Ekiga

Configuring an Account in Ekiga may be done through menu

Ekiga -> Edit -> Accounts

or

Ekiga -> Ctrl-E

The detailed of VoIP Account configuration in Ekiga is as follows,

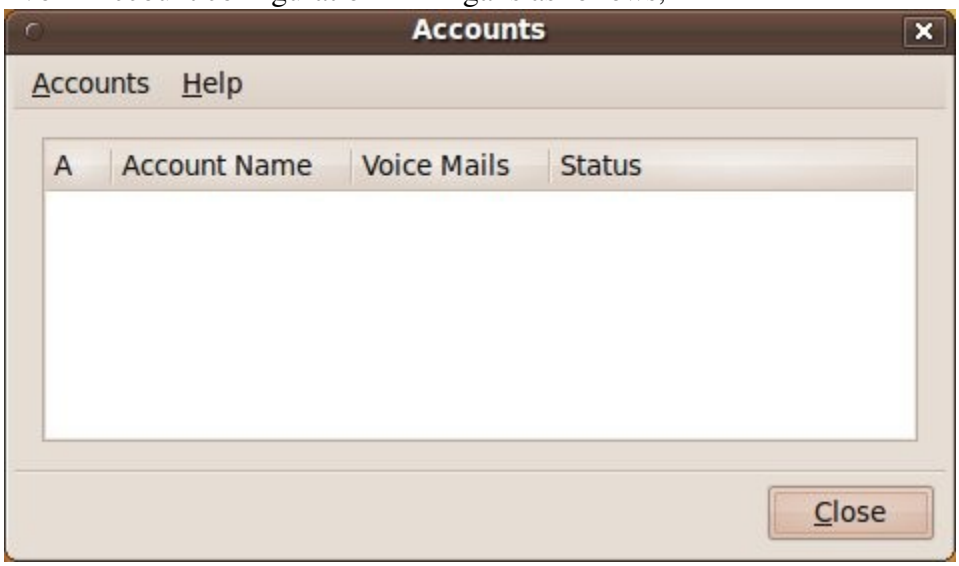


Figure 2.15 Start Account Configuration.

After the Account menu is activated, we will see the above figure.

Figure 2.16 Add SIP Account.

Click on Accounts -> Add a SIP Account

Figure 2.17 Add SIP Account information.

In the above Example, we enter the parameter to use SIP account in VoIPRakyat. Enter the data, namely,

Name	Nabil Suhaemi
Registrar	voiprakyat.or.id
User	123456
Authentication User	123456
Password	<your password in voiprakyat.or.od>

Figure 2.18 Add SIP Account information.

In the above figure, we set the parameter for local VoIP softswitch at IP address 192.168.0.3.

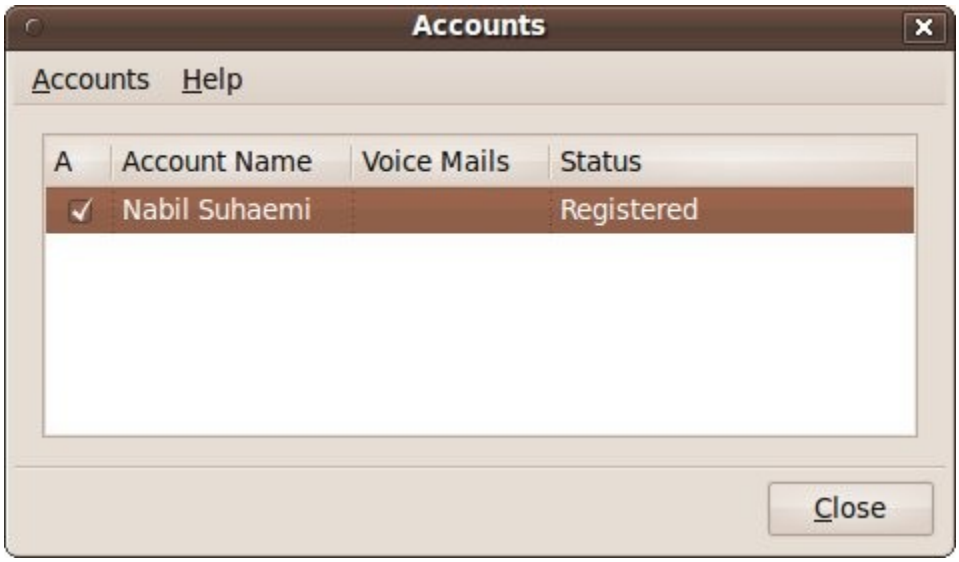


Figure 2.19 Enable SIP Account.

Make sure the account is activated by click-in on the A column. To use the account, we need to make sure the account is registered to the softswitch.

Figure 2.20 Ekiga Ready to use.

Shown in the above figure is Ekiga after it successfully registered to the softswitch. At the bottom of the softswitch we can really see that it "Registered sip: …...". At this point, we can make a call by putting the destination number in after the sip: field.

Test your SIP Softphone

Now that you have adjusted both softphones (or just one of them), the next important thing you have to do is to test whether it could run properly. Note that the quality of the voice produced by the softphone during the test may have been just fine, but when your softphone is connected to a VoIP provider, the voice quality could be poor, depending on many other things such as bandwidth availability and the type of codec run by the softphone. For this test purpose, VoIP providers usually provide the telephone number to which you can dial.

If your computer is connected to an internet behind a firewall, the firewall might block your connectivity. In order to make your VoIP connectivity working behind the firewall, you have to open Port 5060-6060 to enable Session Initiation Protocol (SIP) and Port 8000-20000 for voice data delivery using Real Time Protocol (RTP). But if you're not sure what to do, you can simply ask your network

administrator to do what is told here.

VoIP Rakyat - Dunia dengan telepon gratis, telepon murah, telepon voip - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://voiprakyat.or.id/services/

Most Visited Getting Started Latest Headlines

VoIP Rakyat - Dunia dengan tel...

Register (Free!)

Forgot Password

VoIP Rakyat Members

- Online Phones
- Phonebook
- Locations
- Call Detail Record
- Test Me Please

Words

This server and all our services entirely made up and managed by Open Source Software. They are: Linux, Asterisk, Apache, MySQL and those superb tools that comes with the distribution. So.. now you know that you too CAN build the same like us :) Happy learning.

onno w. purbo:
VoIP Rakyat cikal bakal Telkom

Service Number

Termination Point

Prefix	Layanan
01 002	Terminasi ke PSTN/GSM dan iMaxindo. Contoh: bila ingin menghubungi nomor telepon PSTN Jakarta (021) 8613027 dari jaringan VoIP Rakyat maka dial 0100262218613027 (62 adalah kode PSTN negara Indonesia, 21 adalah kode PSTN kota Jakarta). Halaman rate dan cara pembayaran masih dalam pembangunan, untuk sementara dapat ditanyakan di VoIP Forum.
01 014	Terminasi ke jaringan SIP Broker (http://www.sipbroker.com). Contoh: untuk memanggil layanan Automated Time pada nomor 612 di FWD melalui SIP Broker maka dial 01014393612 (01014 adalah prefix SIP Broker di VoIP Rakyat, 393 adalah kode FWD di SIP Broker tanpa tanda *, 612 adalah layanan Automated Time di FWD).

Layanan Umum

Nomor Telepon VoIP	Layanan
901	Menunjukkan waktu DKI Jakarta dan sekitarnya.
902	Noise test.
903	Echo test.

VoIP Rakyat Network Peering

Prefix	Network
62 848 1001 0848 1001	VoIP Rakyat Network

Online phones :281
* Monitored since Jan 2006

KITA BERHEMAT
Rp. 79650625,-
(79 juta rupiah)
* Asumsi Rp. 125,- per menit

Getting Started Guide

VoIP Rakyat easy to start with. Just follow these steps:

1. Register
2. Download softphone here
3. Configure softphone here
4. Start calling, try to call: 901

Mudah untuk memulai VoIP Rakyat. Ikuti langkah-langkah berikut:

1. Daftar
2. Download softphone disini
3. Konfigurasi softphone disini
4. Mulai bertelepon-ria, coba telepon: 901

Recommendations:

- Softphone: VRC v0.3.6.2
- IM: yes
- Conference room: yes
- OnNet IP-to-IP: yes

Done

Figure 2.21: Just like other VoIP Providers, VoIP Rakyat provides its users with some numbers with which the users can use for testing their VoIP quality

Go to VoIP Rakyat's Service Number page, http://voiprakyat.or.id/services/. This page provides you with some numbers that can be used to test your VoIP connection and their functions. Some of them are:

901 – which indicates the time Jakarta's time and nearby countries.
902 - noise
903 - echo test

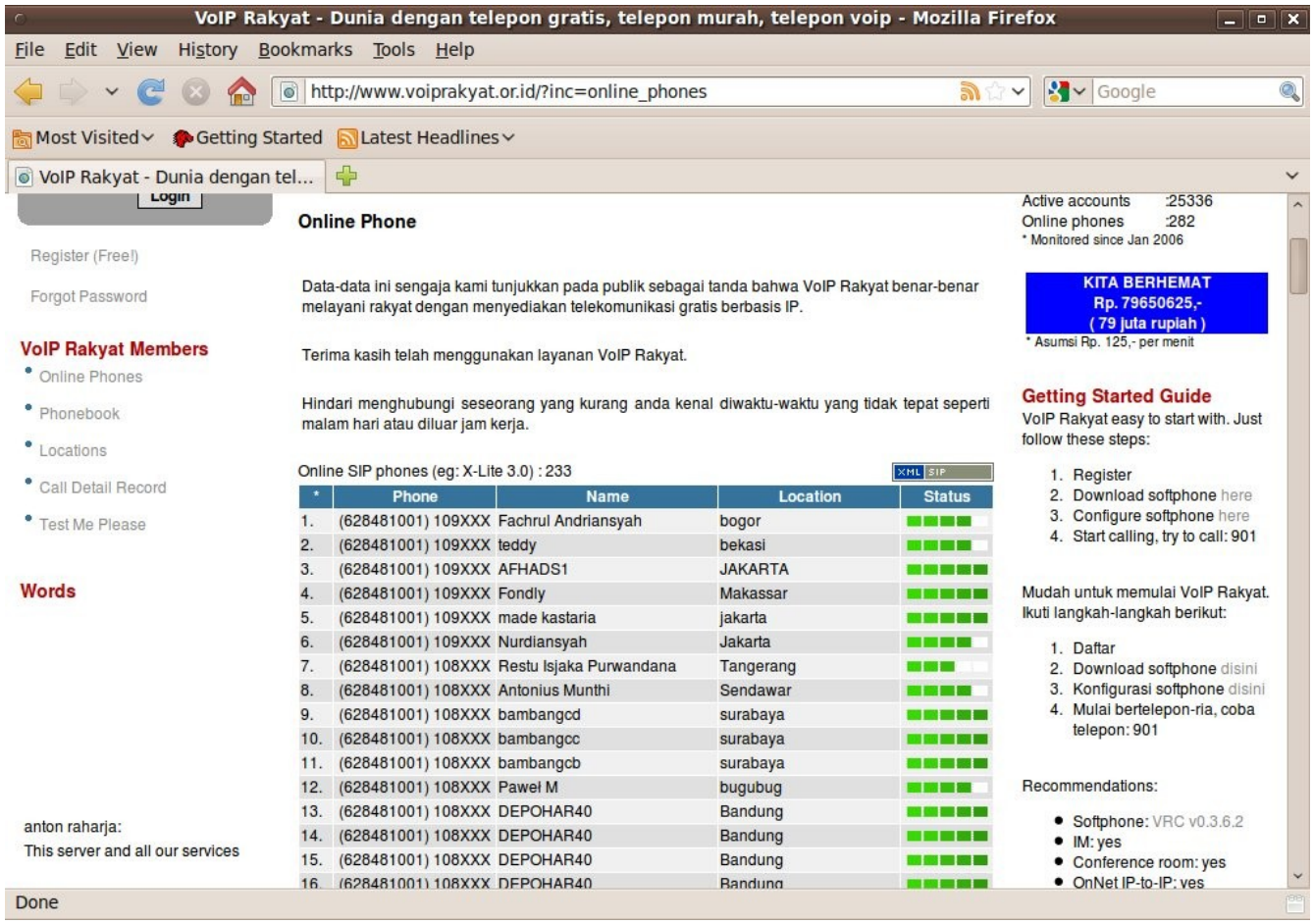


Figure 2.22: Through VoIP Rakyat's Phonebook, you can see who's online

In testing this connectivity, what users will often do is to call anyone found online in http://www.voiprakyat.or.id/?inc=online_phones. So don't be surprised if someone dials your number. Depending on where the users are, the call comes from a variety of countries, including the U.S.

There are of course other VoIP phone numbers which you can use to test your VoIP connection. These are provided in a long list available in http://www.voip-info.org/wiki/view/Phone+Numbers. If you want to call using SIP address format (sip@domain.com), the following is a table of some numbers you may use:

Function	SIP Provider	SIP	Enum
Autoattendant	BC Wireless (http://www.bcwireless.net/moin.cgi/NetworkServices/VoiceServices/PublicConferenceRoom).	1000@mutual.bcwireless.net	1 604 484 5289 x8600 through E164.org
	Enum2go (http://enum2go.com/)	878107472000010@sip2go.com	
Echo Test	N3 Network Lab. (http://www.n3network.ch/)	Echo test sip: echo@n3network.ch sip: 905100@n3network.ch (no G.729)	
	Mouselike.org (UK) (http://www.mouselike.org/)	904@mouselike.org	+441483604781
	VoipTalk UK (http://www.voiptalk.org/)	904@voiptalk.org	
Reread Called ID		95861111@mutual.bcwireless.net	
Welcome Line	FWD	55555@fwd.pulver.com	
	Ewing IT	611300766674@sip.like2fone.com	
	Xmission (http://xmission.com/transmission)	xmission@pbx.xmission.com (tidak ada G.729)	
	UCLA (http://internet2.edu/sip.edu)	13108254321@ucla.edu (tidak ada G.729)	
	TELL	18005558355@proxy01.sipphone.com	
	U. Philippines	0116329818500@proxy01.sipphone.com	
	Personal Telco (http://wiki.personaltelco.net/moin.cgi/	274185@fwd.pulver.com	

	SipPhoneDirectory)		
	Patton Electronics (http://www.patton.com/support)	support@patton.com (tidak ada G.729)	
	Party Line	17475552663@proxy01.sipphone.com (VoIP conference setiap sabtu jam 20:00 GMT)	
	Ingate (http://www.ingate.com/trysip.php)	music@trysip.ingate.com	
	MIT (http://sipphone.com/numbers)	16172531000@proxy01.sipphone.com	

CHAPTER 3: VoIP Hardware for experienced Users

Once you are experienced in using softphone, you may start wondering whether there is an easier way to communicate through VoIP, as using softphone via a computer is not practical—you need to turn on your computer each time you want to communicate through VoIP or keep your computer running for a long duration just to receive incoming call. This may not be prudent at all, since the purpose of using VoIP is to minimize your cost. Besides wasting electrical energy, the computer in which the softphone is running could crash.

So instead of using a computer to communicate through VoIP, you could use VoIP hardware, equipments that enable you to communicate through VoIP efficiently and as easy as you use your conventional phone.

Called Internet Telephone appliance, these hardware typically have the following characteristics:

- it is physically simple, with its dimension slightly bigger than the size of a cigarette box.
- There are ports for connecting to the network or computer, such as LAN/UTP, USB or wireless at 2.4 GHz frequency.
- There is one port or more for connecting to telephones with RJ-11 port.
- It can be configured through the web.

However, VoIP hardware is not free, as you still have to spend some money for buying the equipment. For about US$ 100, you can get a set of decent VoIP hardware produced from China or Taiwan. But despite this cost, VoIP hardware are highly recommended, as you may find the benefits the hardware bring outweigh the cost you have to cover, in terms of ease of use and energy efficiency.

This Chapter will explain several hardware available in the market and how to configure them: IP Phone, Internet Telephone Gateway or better known as Analog Telephone Adapter (ATA), and Wireless IP Phone. They way you configure VoIP hardware is not much different from what you do with softphone. Basically all you have to configure are the IP settings (IP address, subnet mask, and gateway) and registration to SIP server or proxy server (Username or telephone number, password and hostname server). Often, IP settings is configured automatically using DHCP server operating in a network, so you don't have to set the IP address, subnet mask and gateway.

Linksys PAP-2 Analog Telephone Adapter

Figure 3.1: With ATA, you can use your PSTN phone for VoIP communication

The simplest type of VoIP hardware is the Analog Telephone Adapter (ATA), which can easily be connected to a conventional telephone. The ATA used as an example in this book is the Linksys PAP2, which has two RJ-11 ports (FXS ports) that can be connected to two conventional phones. Each of these ports can be registered to a SIP Proxy server individually. As a result, we could have two SIP accounts, each connected to a conventional phone.

What we have to understand is that an ATA has two type of RJ11 connections, namely,

- FXO to be connected to PSTN / Telco line / PABX extension.
- FXS to be connected to Telephone line / FAX.

After all UTP, LAN, power and telephone cables are plugged in, you have to first of all find out the IP address of the Linksys PAP2 so we will be able to configure using the web, by carrying out the following steps:

- Press “*” repeatedly on the phone keypad until you hear someone talking through your phone.
- Press “110#” to listen to the IP address for the Linksys PAP2 configuration.

The next step is to configure your PC so that you can configure Linksys PAP2 through the web. All you have to do is match the family IP address to PAP2's, by doing the following: Go to Start, Open Control Panel, Network connections, local Area Connection, Internet Protocol (TCP/IP) and Properties. Then go to Web Linksys PAP2 from your PC through this address http://ip-address-pap2/.

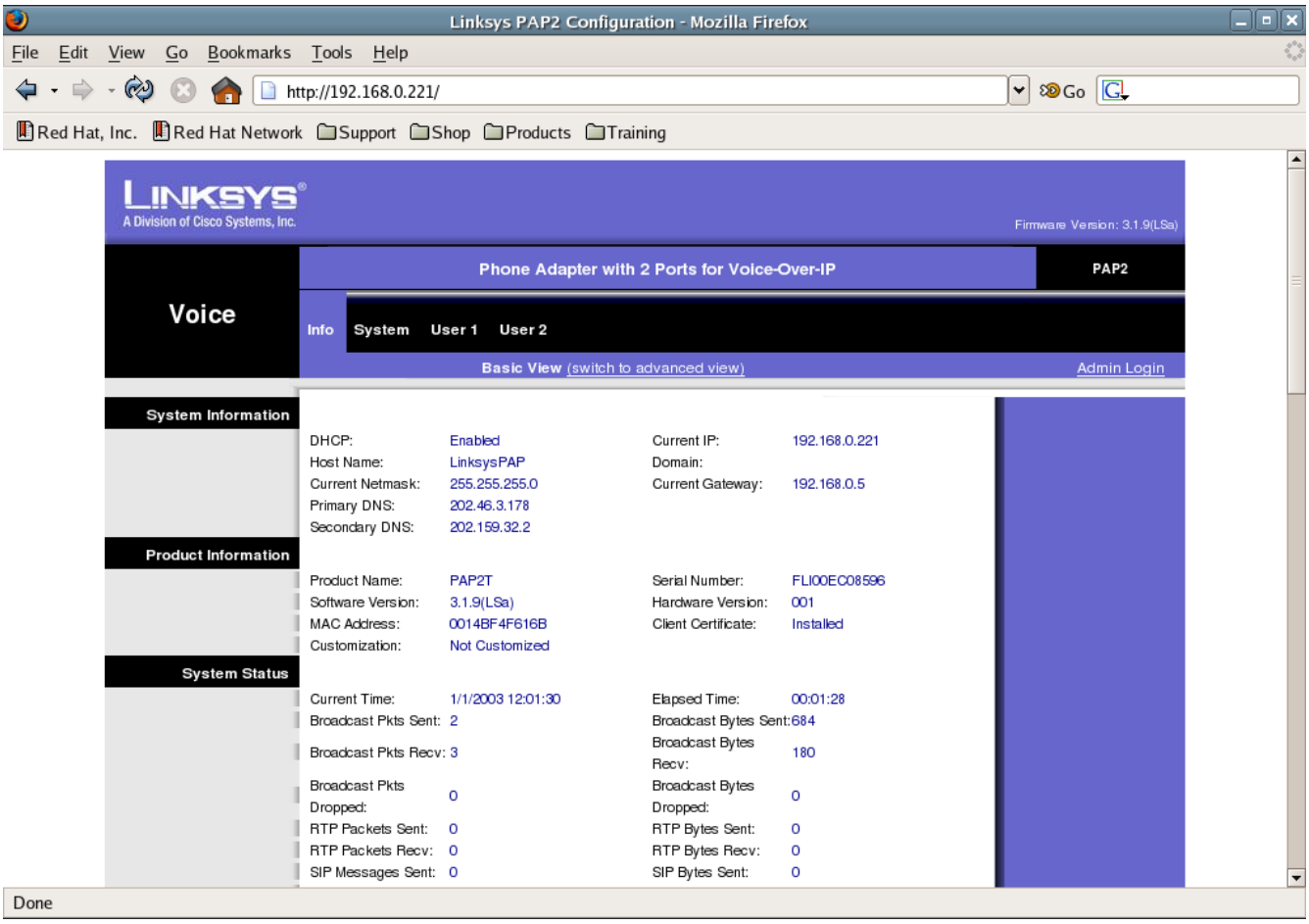


Figure 3.2: The initial menu that will appear is the status of Linksys PAP2

Click Admin Login, which is on the top right to begin the configuration as an administrator.

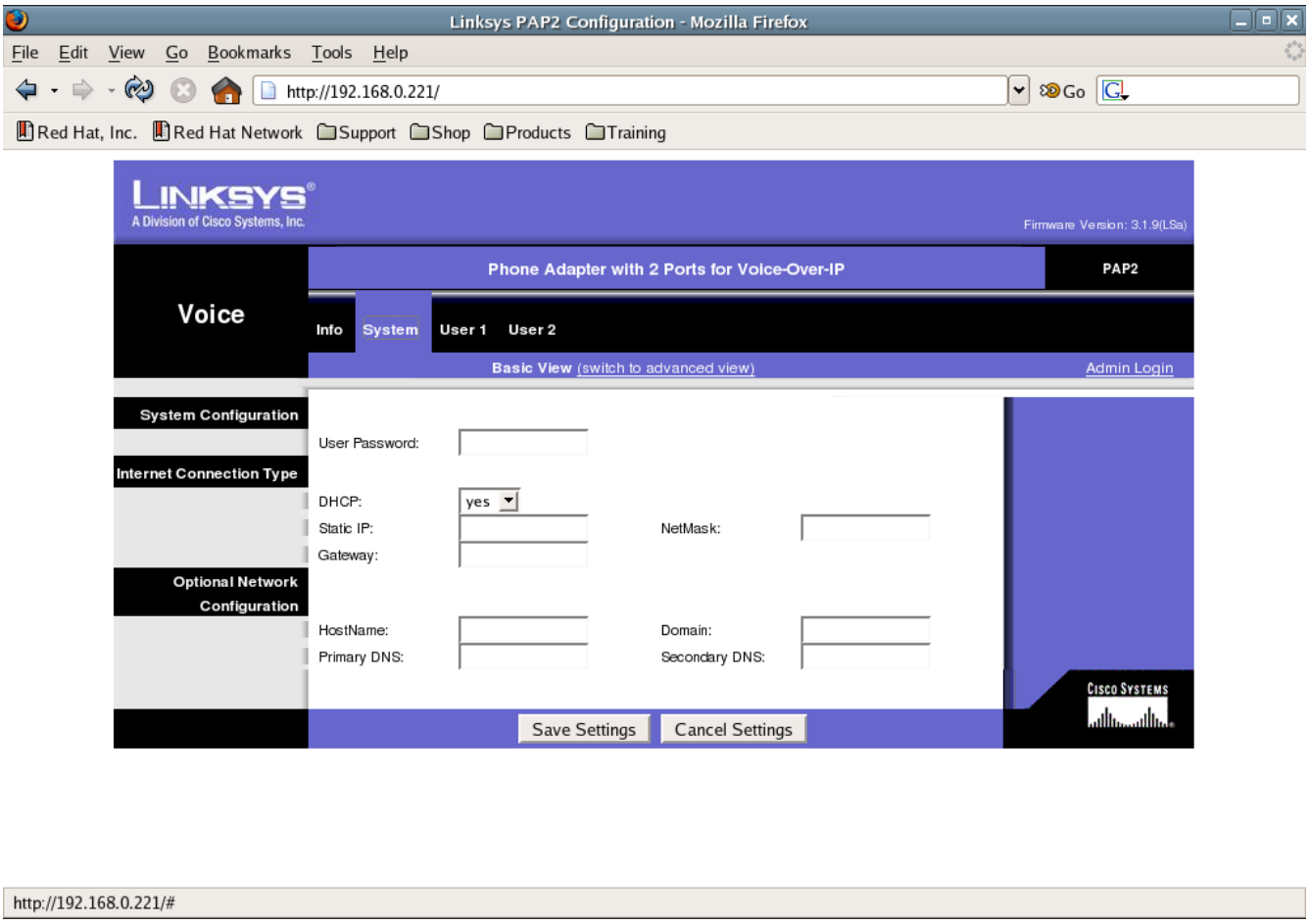


Figure 3.3: You can determine whether you want to use dynamic or static IP address

To view or change the IP address configuration, click System. Check whether the IP address, Gateway and DNS put in place are correct. Alternatively, set DHCP to “yes” so Linksys PAP2 will use the IP address that is obtained automatically.

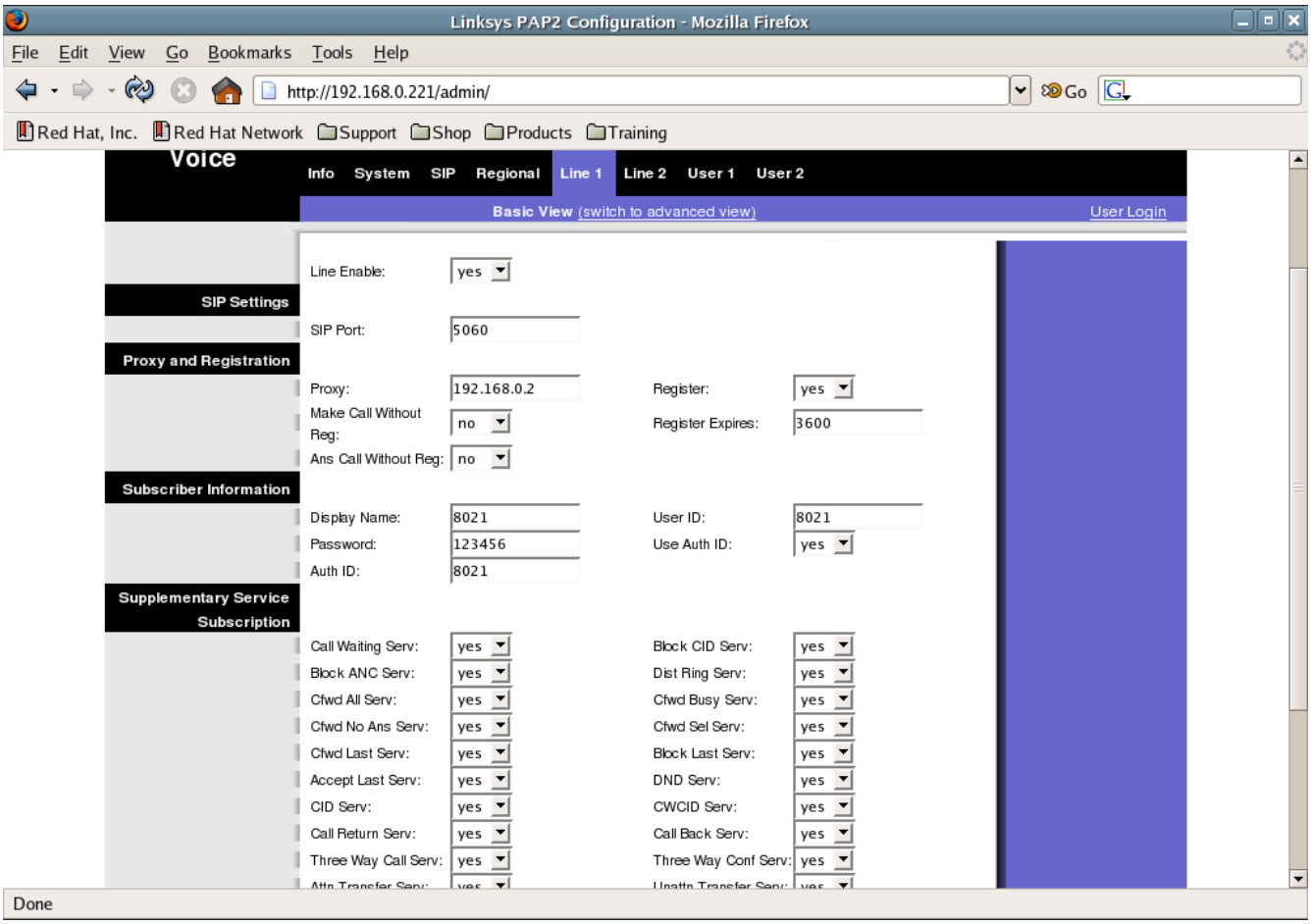


Figure 3.4: Each line (line 1 and 2) has its own settings in the administration panel

In Linksys PAP2 we can set up two SIP accounts registered with the SIP Proxy, with each account connected to a phone. The account settings can be done in menu "Line 1" and "Line 2".

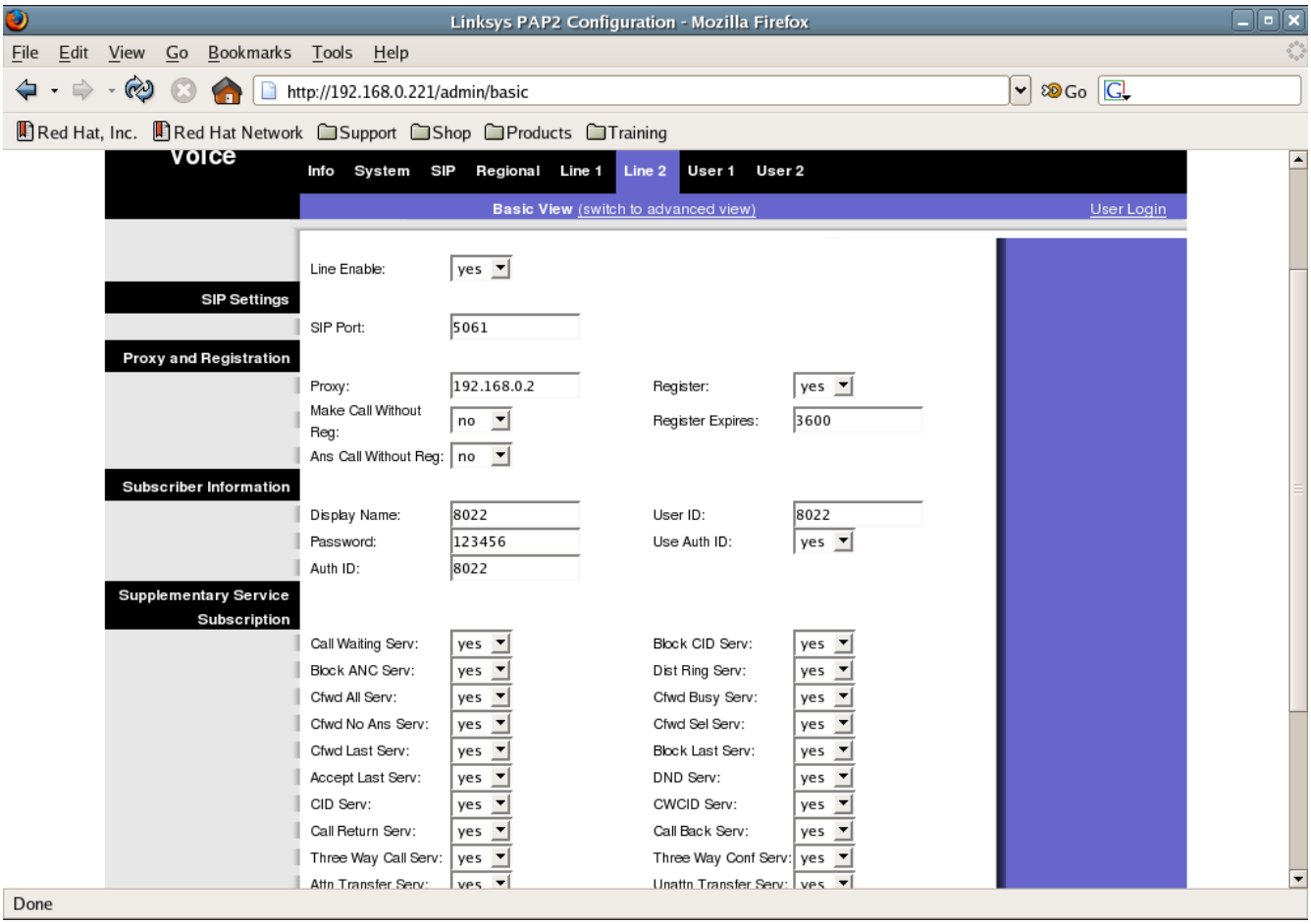


Figure 3.4: Line 2 tab of the administration panel

Few important steps to do in activating an account in both menus:

- Set Line Enable to yes.
- Fill in your account using the following parameters:

Proxy	voiprakyat.or.id
User ID	telephone number given by Voip Rakyat
Password	the password given by voiprakyat
Use Auth ID	no

If you set Auth ID to yes, then fill in Auth ID with the telephone number given by VoIP Rakyat. Do the same process for your other SIP account, the one registered with PAP2 Line 2. Actually there are many

other parameters that can be configured, but for a normal operation, it is not necessary to configure them. So it is sufficient for us to use the default configuration values.

Linksys IP Phone SPA 941

Using IP Phone might be more attractive to most us who want to use VoIP, as compared to using Softphone, IP Phone is much simpler to use. It is physically similar to a conventional phone, but unlike conventional phone, IP Phone is designed to use for VoIP, so there is no RJ-11 port like the one available in conventional phones.

Figure 3.5: An IP Phone

Figure 3.6: IP Phone typically has two RJ-45 ports

What IP Phone has instead is the RJ-45 port for its LAN connection (ethernet socket). As you can see at the back of IP Phone (shown in figure 3.6), both ports are of RJ-45, one to be connected to a LAN while another to the computer. This allows us to use the phone while using the computer for the internet. However keep in mind that your bandwidth may not be sufficient for both. So only use both at the same time when you think you have enough bandwidth to ensure the quality of your VoIP communication remains good. An IP Phone can usually be configured through the web.

There are abundant types of IP Phone in the market. You can find them at the following link: http://www.voip-info.org/wiki/view/VOIP+Phones.

The sort of IP Phone we use as an example is SPA 941. To obtain its IP address, we have to do the following:

- Click Menu (illustrated as paper icon below the mail button)
- Click the cursor so it will provide a drop-down menu
- Find network

There you will find the IP address of SPA 941.

Next you have to configure your PC so that you will be able to configure Linksys SPA941 through the web. Go to PC, match the IP address to that of SPA 941 by choosing Start, Control Panel, Network connections, Local Area connection, Internet Protocol (TCP/IP) and Properties.

Figure 3.7: The first appearance you will see is the status of Linksys SPA941.

Go to Linksys SPA941 web through http://ip-adress-spa941.

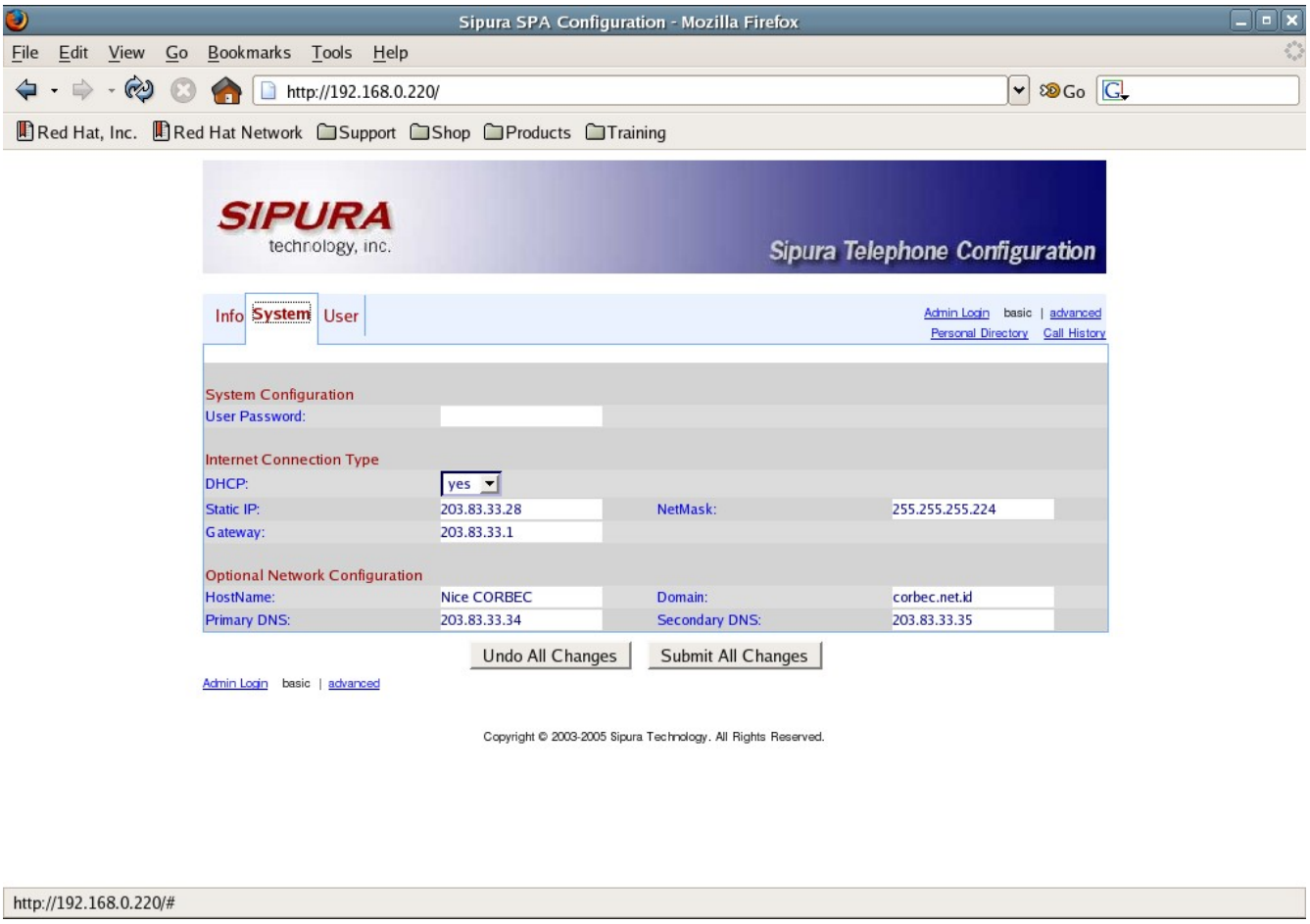


Figure 3.8: Choose which internet connection type you want to have

In the system menu, we can configure our IP address, netmask, gateway and DNS of SPA941. If you wish to have the IP Address be detected automatically using the information obtained from DHCP server, you can just set DHCP to “yes”.

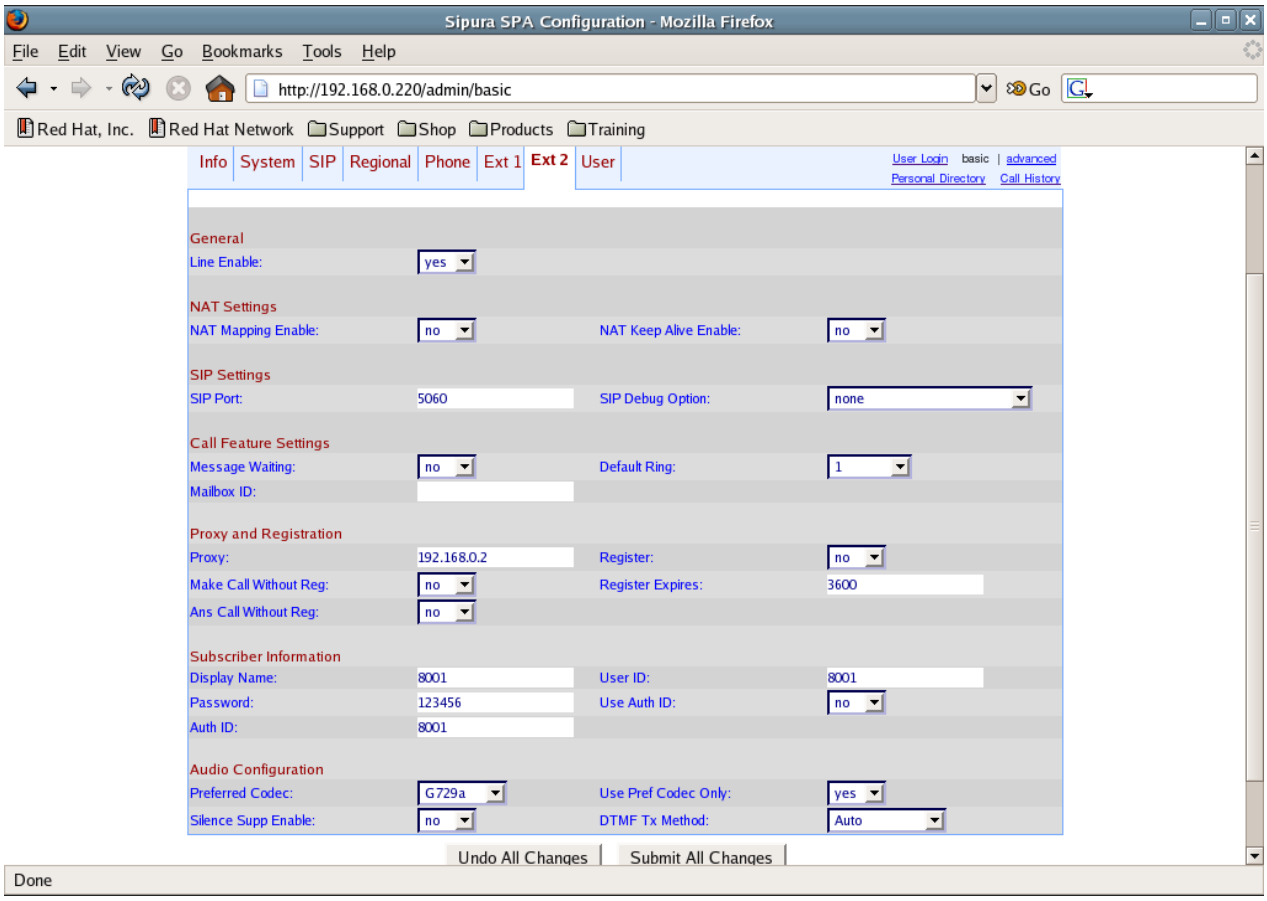


Figure 3.9: By clicking on Ext 2 tab, you can set some important parameters of Ext 2 line.

By logging in as Admin, we will see that SPA941 has two external lines: Ext 1 and Ext 2. Each of them can be configured so as to be registered to different SIP proxy.

Figure 3.10: By clicking on Ext 1 tab, you can set some important parameters of Ext 1 line

There are two steps needed to activate an account at menu Ext 1 or Ext 2:

- Set Line Enable to yes.
- Fill in the the following parameters with the information pertaining to your account:

Proxy	voiprakyat.or.id
User ID	the telephone number given by VoIP Rakyat
Password	the password given by VoIP Rakyat
Use Auth ID	no

If "Use Auth ID" is set to yes, then fill the in Auth ID with the telephone number given by VoIP Rakyat. Do the same for the other SIP account you want to register to Ext 2 of Linksys SPA941.

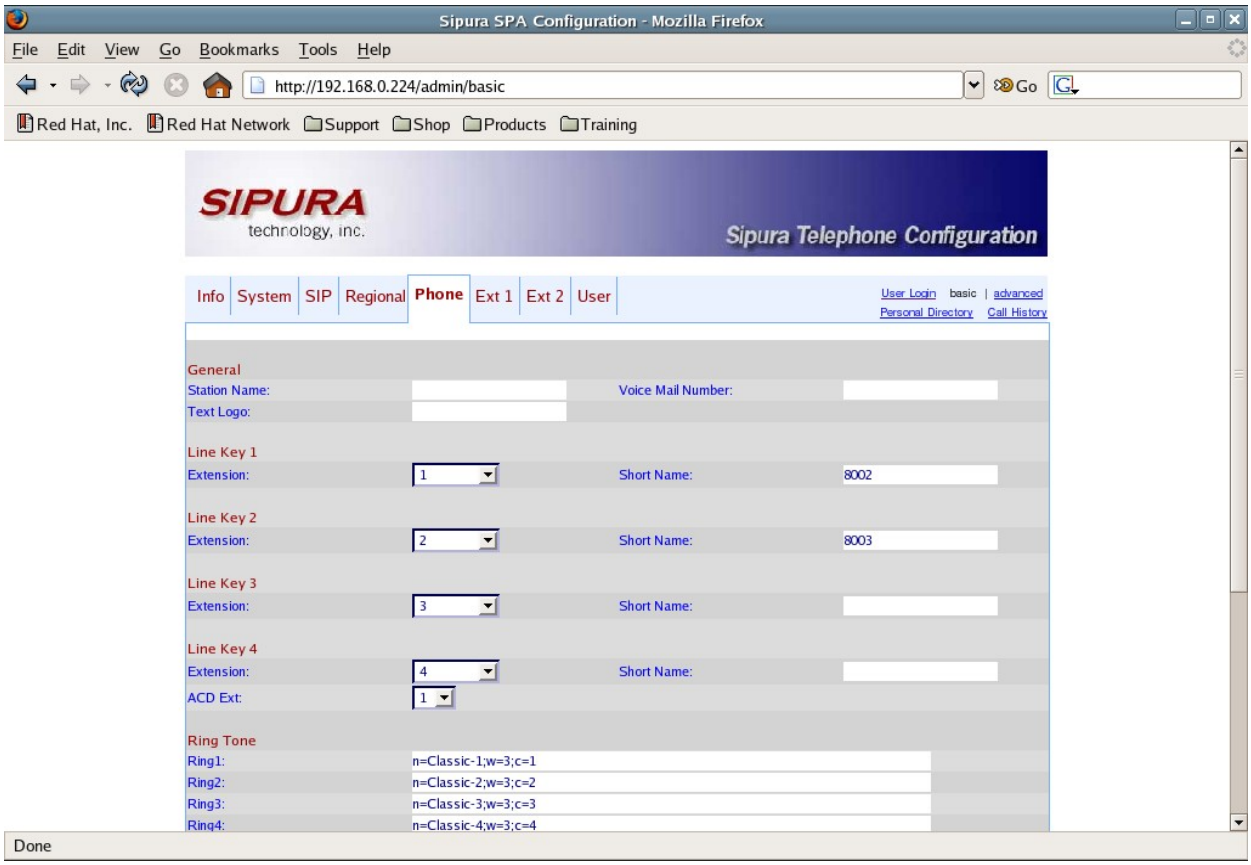


Figure 3.11: The phone tab of administration panel

In Linksys SPA941, we are given the facility to open a specific Ext by using the Line Key button on the right side. Four Line Key buttons are available. One may program these four buttons – two for each Ext Line. To do the programming, you have to be an admin, by carrying out the following steps: choose an extension (either 1 or 2) for each Line Key and show number and fill in it with the number or User ID given by the SIP Provider.

WiFi IPPhone

WiFi Phones can be used for internet telephony connected to IP PBX via WiFi or HotSpot. In other words, the phone can be used as an extension of a PABX or a phone which is connected to a hotspot. Some of these WiFi Phone may have dual functions—GSM mode and VoIP—it allows the possibility of receiving a GSM call or VoIP call through WiFi mode as an extension to an IP PBX.

Operating WiFi Phone is not difficult. All you have to do are configure your SIP account by entering the name of the server, telephone number and password; searching any available WiFi access point; and connect to a WiFi Access Point and get an IP address.

Now that you understand what WiFi Phone is and how to operate them, we will provide some example on how to configure and operate WiFi Phones.

Linksys Wireless-G IP Phone

Linksys launched a Wireless-G IP Phone a dedicated WiFi Phone. It is not a PDA nor ordinary cellphone (See figure 3.12). If you have the WiFi Phone properly configured, connected to the Wireless Access Point and registered to a VoIP Softswitch, then what should appear on the screen of the phone is the name of the access point and the telephone number of the phone. Under this circumstance, the WiFi Phone is ready to be used for calling.

Figure 3.12: Wireless-G IP Phone

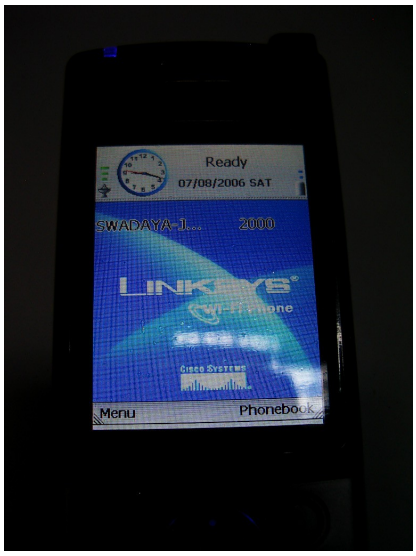


Figure 3.13: WiFi Phone can be used for VoIP call when it is properly configured

In Linksys WiFi Phone main menu, there are at least two (2) things you have to configure so that your phone will function well. Firstly, the wireless settings, by which we can scan any access point wireless frequency and connect our phone to the access point so we can be connected to the internet. Secondtly, the Phone Settings, allows us to configure the SIP server that we use to call. For the latter, you need to fill in the information pertaining to phone numbers, passwords and the servers used. Since we are using VoIP Rakyat as an example, the information should be those of VoIP Rakyat.

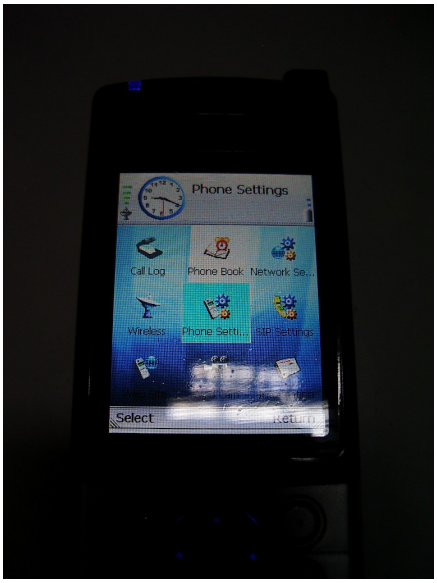
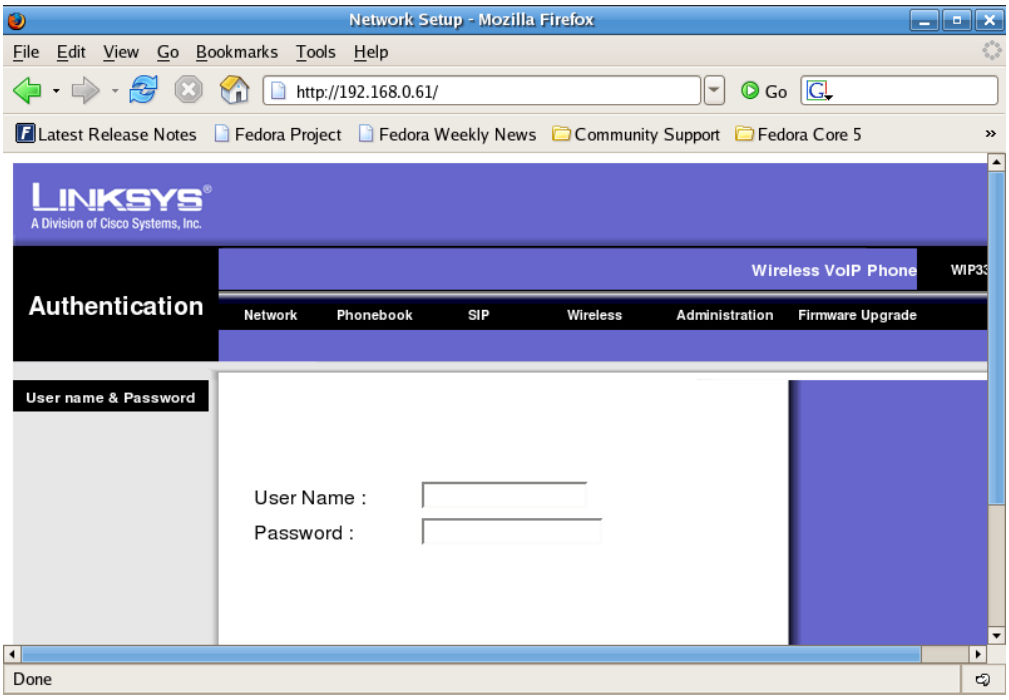


Figure 3.14: Through the phone menu, you can make direct configuration in order to make your phone VoIP enabled

Configuring the WiFi Phone using menu shown in the figure 3.14 is easy, but since there is no software that could help us capture the screens for configuring the phone, we use the web configuration instead for the purpose of helping you understand how to configure the WiFi Phone. The same result should otherwise be similar to that of direct phone configuration. In contrast to WiFi Phone that is combined with PDA or GSM, Linksys WiFi Phone can be configured using the web, in addition to feature allowing you to directly make configuration using the menu available in the phone screen.

Figure 3.15: Enter the user name and password to log in the administration panel so that you can configure the phone



The web will appear as what is shown as figure 3.15. It is the display prompting you to enter your user name and the password required to authenticate your account. The default for the username and password is admin and admin respectively.

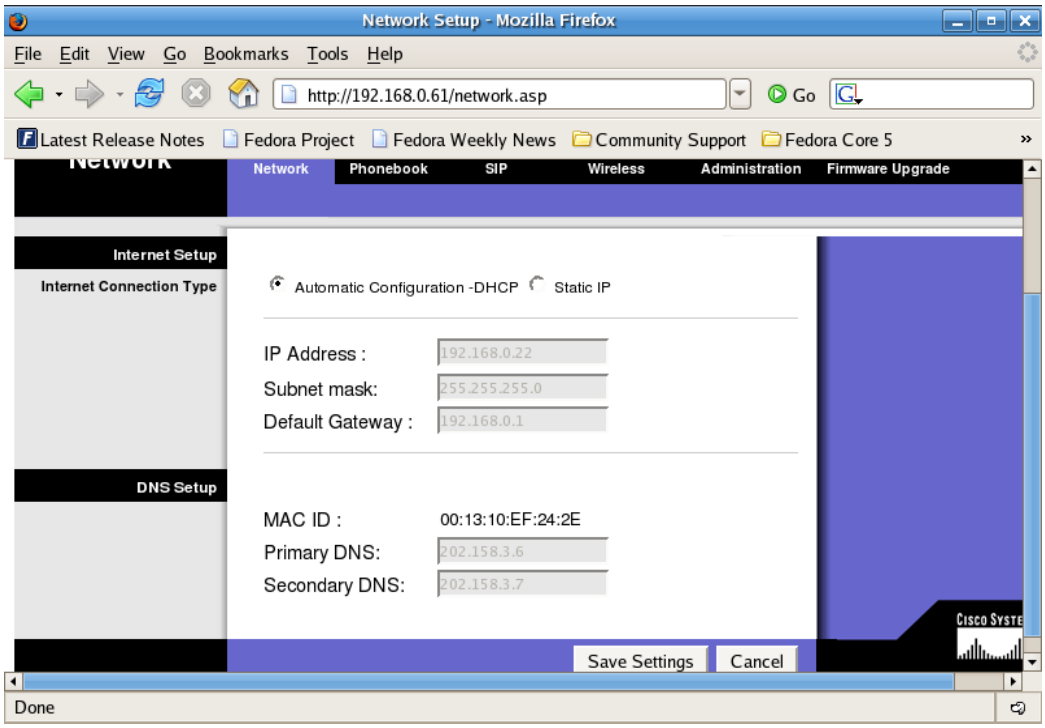


Figure 3.16: Through the Network Tab of the administration panel, you can set how you will obtain your IP address

Once you have entered the username and password, you will be brought to the administration panel whereby you can configure the IP address. Normally, in a hotspot that provides any user connecting to it with free IP address, we just need to set the configuration to "Automatic Configuration-DHCP". In the case where the IP address is not provided automatically by the hotspot, you have to manually enter the information pertaining to the IP address, subnet mask, gateway, primary and secondary DNS. The MAC address of the WiFi Phone appears by default. Once you have finished entering these information, click Save Settings to save them into the memory.

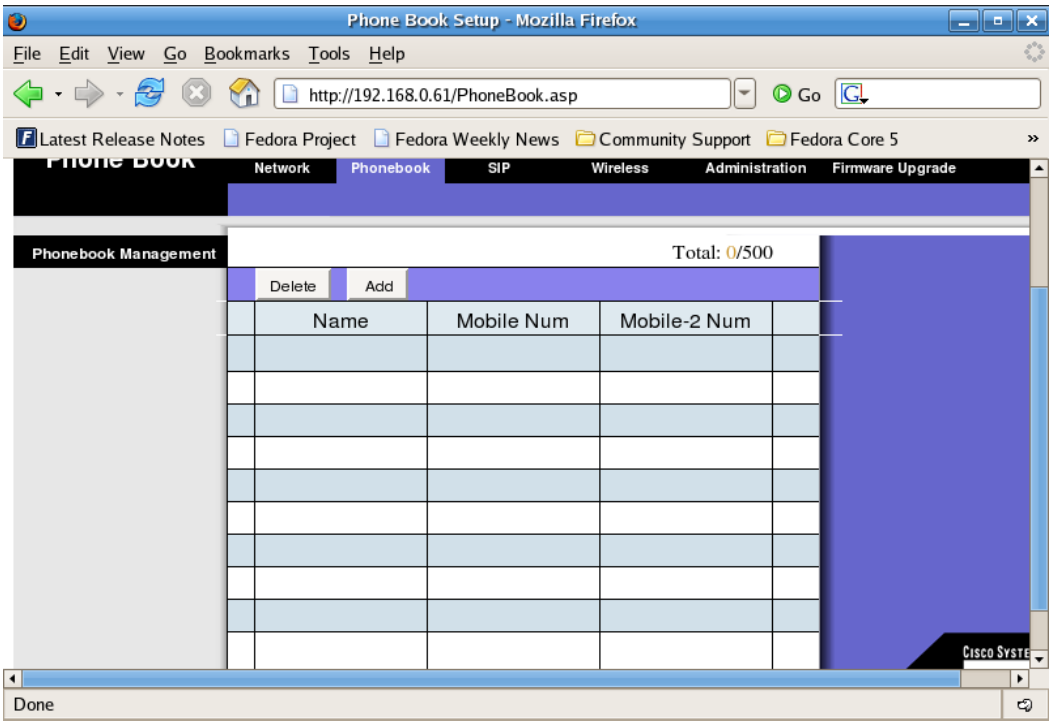


Figure 3.17: By clicking on the Phonebook menu, you can Add new phone numbers or delete existing ones

In the Phonebook menu, we can add new numbers or delete the ones already listed there. We can also include multiple numbers for each person.

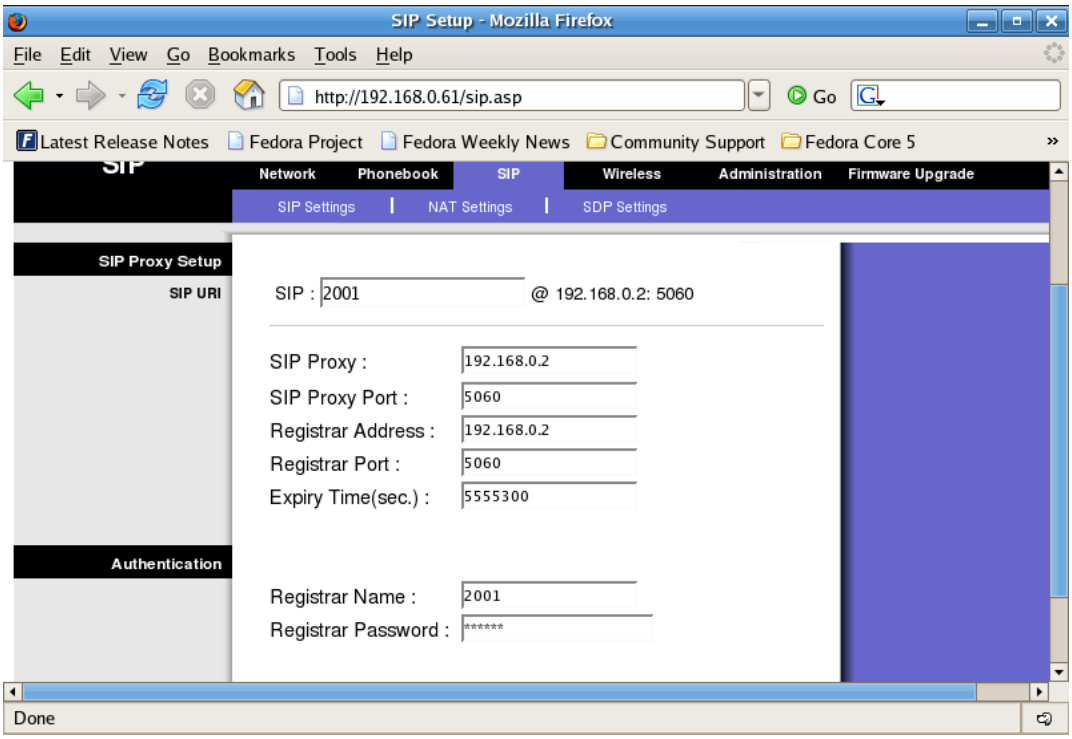


Figure 3.18: SIP Settings

In the SIP Settings tab, we can configure the IP address of the SIP Proxy, SIP Port (usually 5060), the IP address of the SIP Registrar (usually the same as that of SIP Proxy), Registrar Port (also usually 5060), and SIP account number consisting of telephone number and the password related. As for VoIP Rakyat SIP information, fill in the SIP Proxy and SIP Registrar with voiprakyat.or.id.

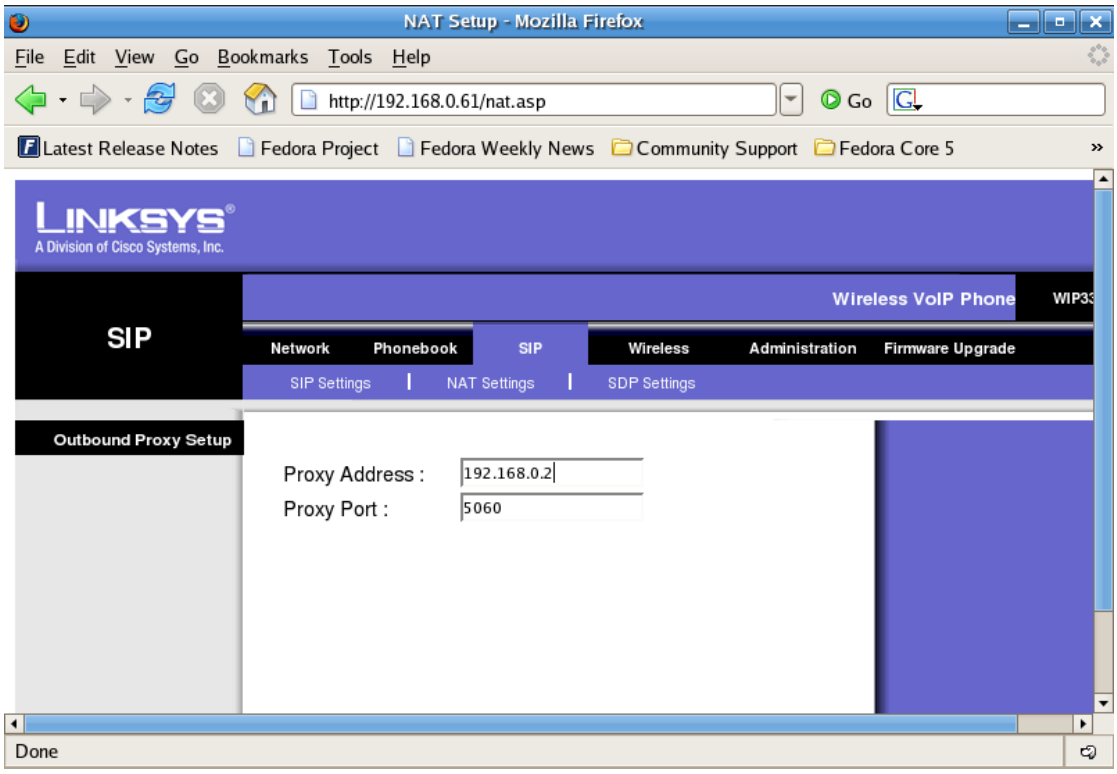


Figure 3.19: The NAT Settings

In the NAT Settings menu, you can configure the Proxy address and Proxy Port. The Proxy address for VoIP Rakyat is voiprakyat.or.id. And the Proxy Port normally used is 5060.

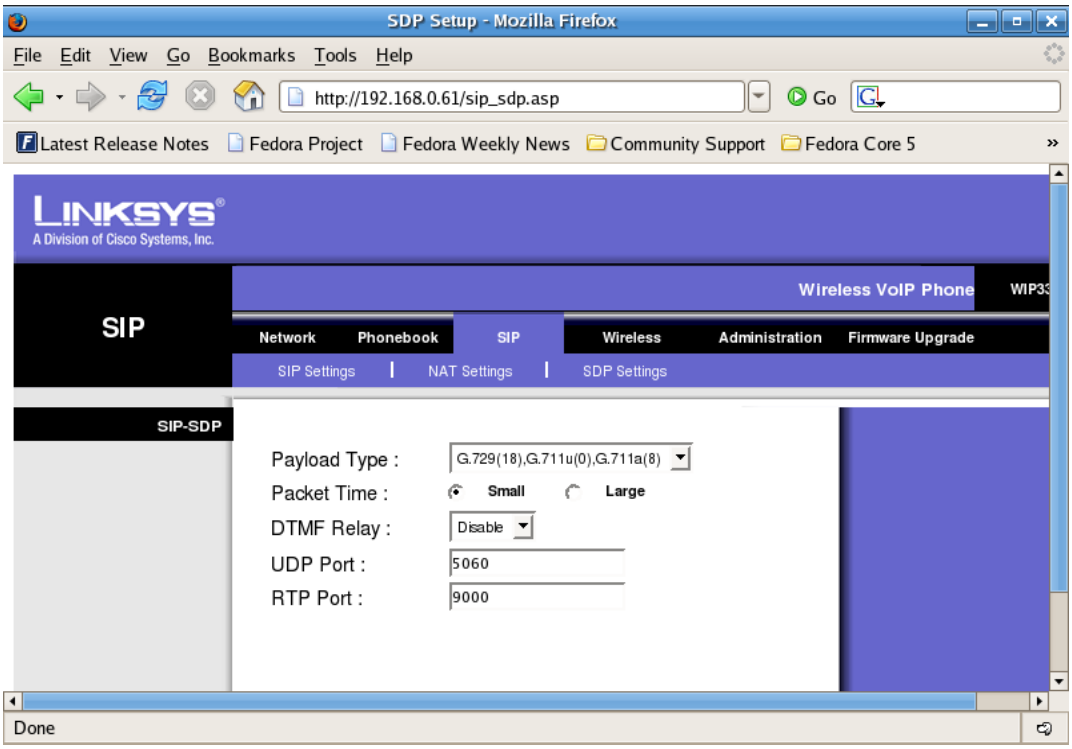


Figure 3.20: The SIP-SDP Settings tab

In SIP-SDP Settings tab, we can configure several things related to the type of Codec, packet time, DTMF Relay, UDP Port and RTP Port. These parameters are good by default, so just leave them as is.

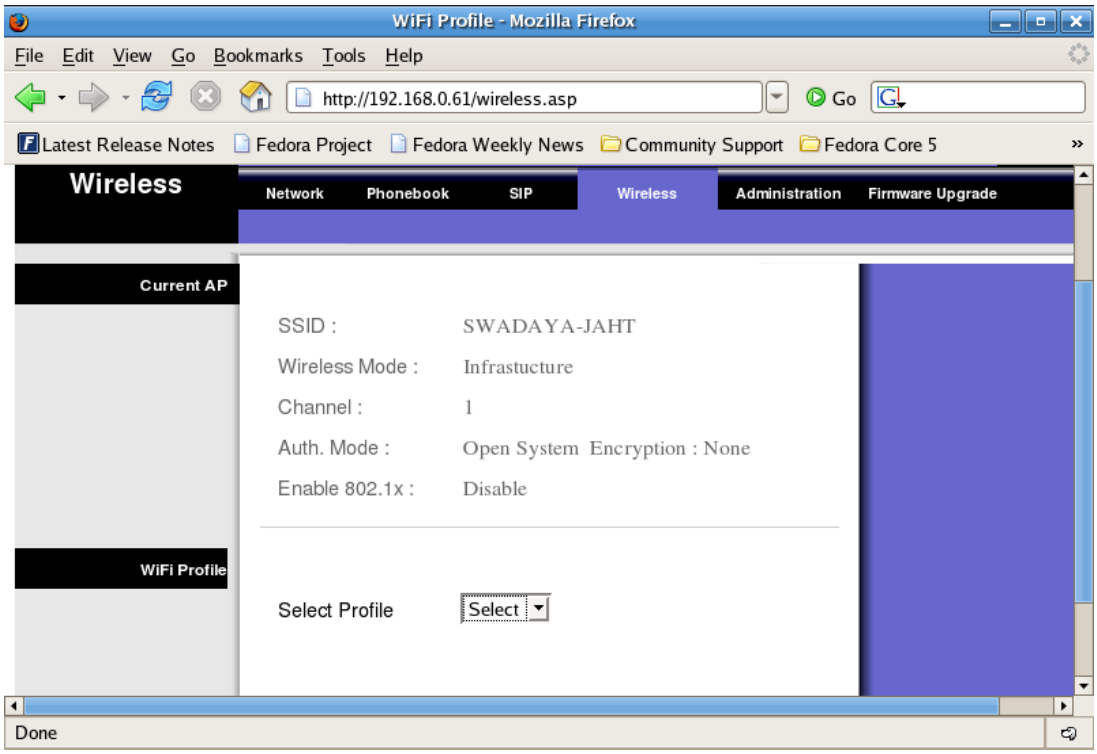


Figure 3.21: By clicking on the wireless tab of the administration panel, you can find out to which hotspot the phone is connected

In the Wireless section we can see to which Access Point the WiFi Phone is connected.

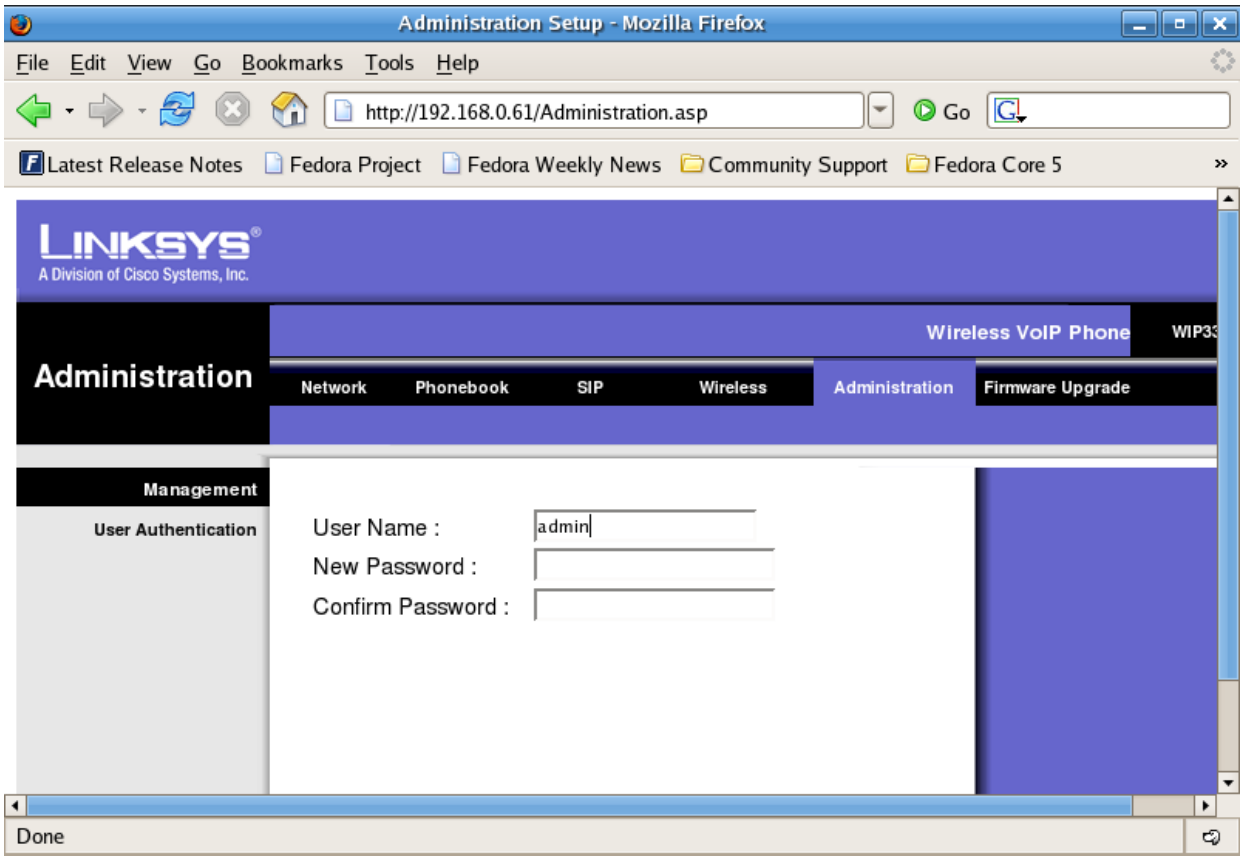


Figure 3.22: By clicking on the Administration tab of the administration panel, you can change your password

In administration section, we can set the administrator's username and related password for the WiFi Phone. The default configuration for username and password are both admin.

Hewlett Packard Ipaq 6395

Personal Digital Assistance (PDA) which uses Pocket PC (PPC) operating system, such as Ipaq 6395 or other kind of Ipaq having WiFi capability, can be used for VoIP communication. One of the software that can be used for this PDA is SJPhone PPC, which can be downloaded from http://www.sjlabs.com/sjp.html. Also available in this site are the manuals necessary for operating the softphone. SJPhone installation can be done in the following steps: connect Ipaq to PC through the provided USB cable and run the software on PC, and SJPhone PPC will be automatically installed in Ipaq.

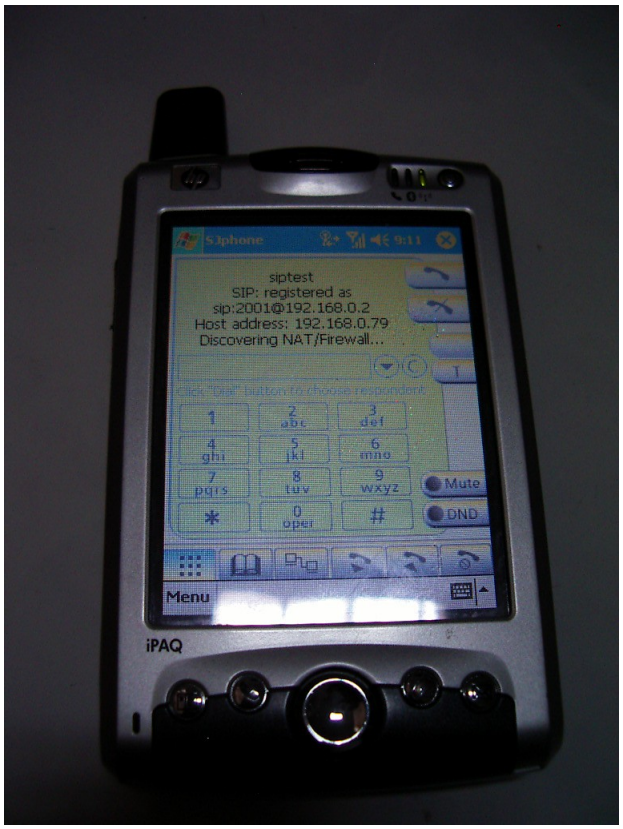


Figure 3.23: Hewlett Packard Ipaq 6395

Activating Ipaq 6395's Wireless Capability

In order to access internet telephony using PDA Ipaq, we need to activate the wireless connectivity feature available in Ipaq. Through Ipaq Wireless menu, press the WiFi button so the wireless connectivity becomes active.

Figure 3.24: Through iPAQ Wireless Settings, you can enable the phone's WiFi feature

If all goes well, the color of the WiFi button will turn green, a sign which indicates that the device is properly connected to the wireless network.

Figure 3.25: The green WiFi icon indicates that you're connected to a wireless network

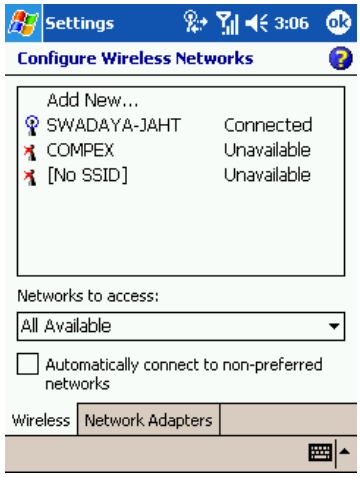


Figure 3.26: By clicking the Settings icon next to the WiFi icon, you can see to which network your phone is connected

If you want to make further configuration on how you use the WiFi access, click the Settings Icon, which will bring you to a menu showing various access points monitored by Ipaq 6395. Choose the access point to which you want to be connected.

Running SJPhone

Figure 3.27: In order to run SJPhone, tap the icon

SJPhone Software can be found as a program of Pocket PC. To run it, simply press the button. Note that the technique for operating SJ Phone through Pocket PC is not so different from that which runs in on PC.

Figure 3.28: Enter your account number and the required password in order to initialize the profile

Figure 3.29: The appearance of SJPhone dialing console

If the SIP account has been properly configured in SJ Phone, what will be asked first when you activate SJ Phone is the account number and password required to access such SIP account. SJ Phone will appear like what is shown in Figure 3.29, with its dialing keypad and all the buttons needed for dialing up and hanging up.

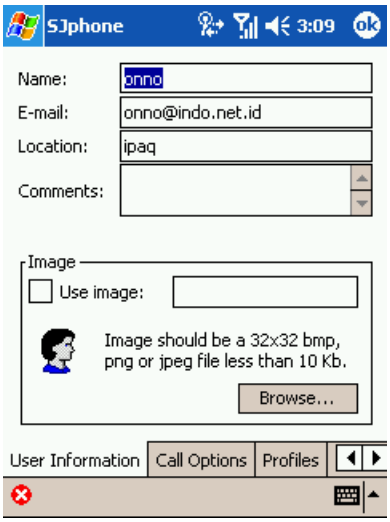


Figure 3.30: Through the user information tab of SJPhone menu, we can enter our name, email and location. We can even include comments and our image

Tap the menu button. In menu, we can enter the information pertaining to the user, which includes name, email address, location and even any picture we want to use as our image.

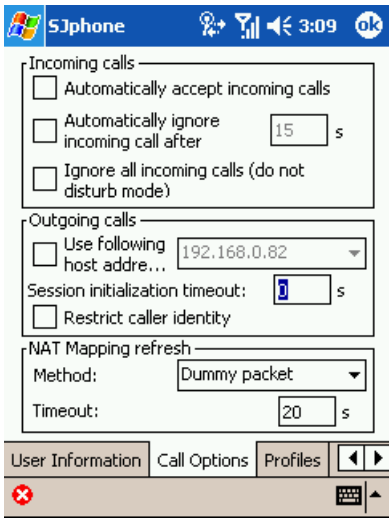


Figure 3.31: Settings of incoming and outgoing calls, and NAT Mapping refresh

Tap the call option tab. Through this tab, you can configure some things like:

- whether we want to automatically receive all incoming calls. This menu is in fact very useful for the sort of Ipaq with small screen that makes us difficult to receive VoIP calls manually.
- Whether we want to be left undisturbed, ignoring all incoming calls.
- The IP address used for outgoing calls.
- Limiting the Caller ID in use.

In general, these parameters do not need to be changed, possibly except for the "Automatically Accept Incoming Calls" to compensate for the small PDA screen.

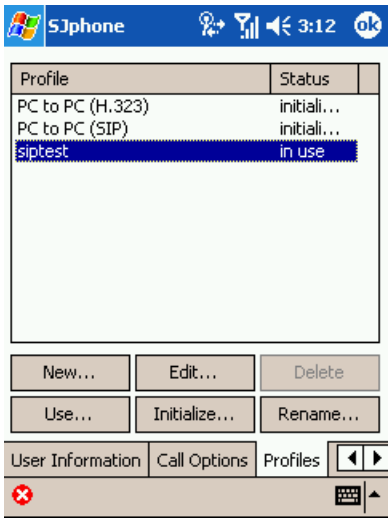


Figure 3.32: Under the profiles tab, you can either make new profile; edit, use, initialize, rename or delete existing profile

In the profile dialog, we can make detail configuration for each account. Basically, a profile defines an account., which can be either a SIP account or H.323 account. The latter is a technology once used by many VoIP providers. The former is a technology used in VoIP Rakyat. There are several options available in the profile menu:

- New to create new profile
- Edit to edit existing profile
- Delete to delete existing profile
- Use to use existing profile
- Initialize to initialize a profile
- Rename to change the name of existing profile

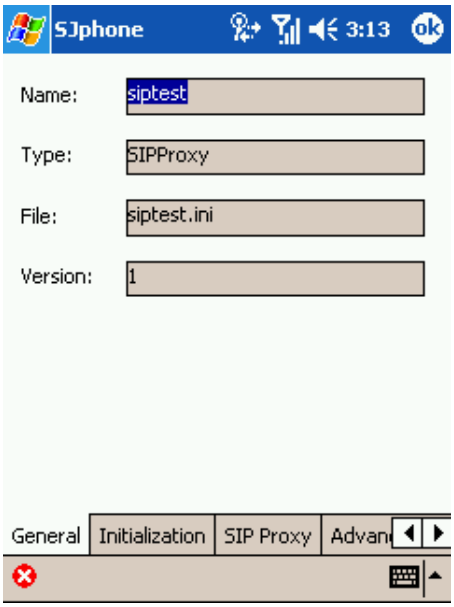


Figure 3.33: Enter the name of the profile, the type of interface it uses, and the name of the profile file

When editing a profile for the first time, we will be brought to the general tab of the profile. Here we can define the name of the profile, the type and name of the profile file. For VoIP Rakyat, the interface type we use is SIP Proxy.

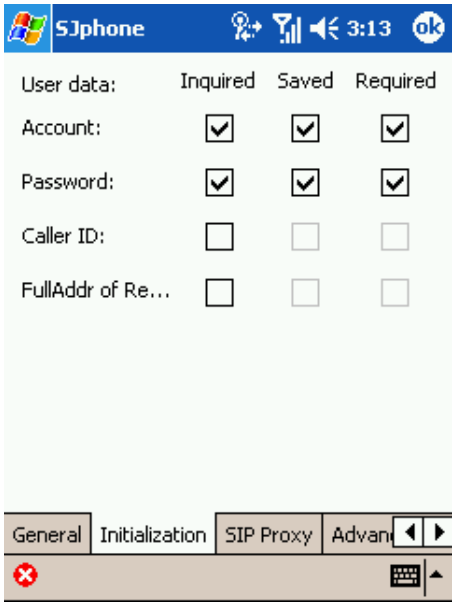


Figure 3.34 Through the initialization tab, configure what needs to be inquired, saved or required

In initialization tab, we can set the user data initialization process, including phone number/account, password and Caller ID, whether users will be inquired, the data pertaining to these parameters need to be saved or required. It is recommended that you use the default setting, leaving the the settings as is.

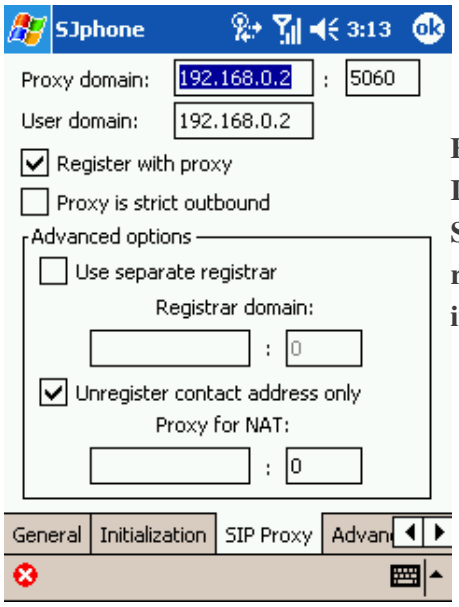


Figure 3.35: In order to enable SIP Proxy, enter the required information in the SIP Proxy tab

Of all menus required for configuring a profile, SIP Proxy is perhaps the most important. The information entered there will determine whether the SIP softphone can actually be used or not. The information you have to enter are as the following:

- Proxy Domain is your SIP Proxy server. For VoIP Rakyat, the proxy domain is voiprakyat.or.id.
- The Proxy Domain Port is usually 5060.
- User Domain for VoIP Rakyat is voiprakyat.or.id.
- Click Register with proxy

Figure 3.36: Additional settings available in the Advanced tab

In Advanced tab, we can configure more sophisticated features such as voice mail number, removing fancy characters from phone numbers, accept redirection replies etc. However, to operate SJPhone in a standard mode, we don't have to change these parameters.

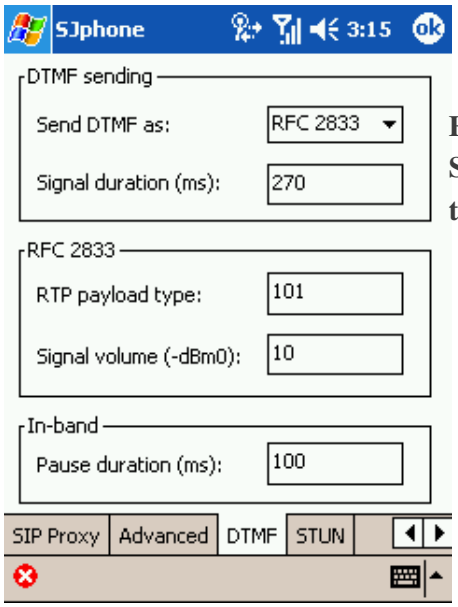


Figure 3.37: Settings of DTMF tab

In Dial Tone Multi Frequency (DTMF), we can choose several things related to DTMF:

- DTMF is sent as voice or text data using RFC2833.
- The duration of the tone. The default value used is 270 ms.
- Type of Real Time Protocol used in RFC 2833 is 101.
- The default DTMF signal volume is 10 -dBm0
- The pause duration during which the signal is sent in in-band mode. The default value is 100 ms.

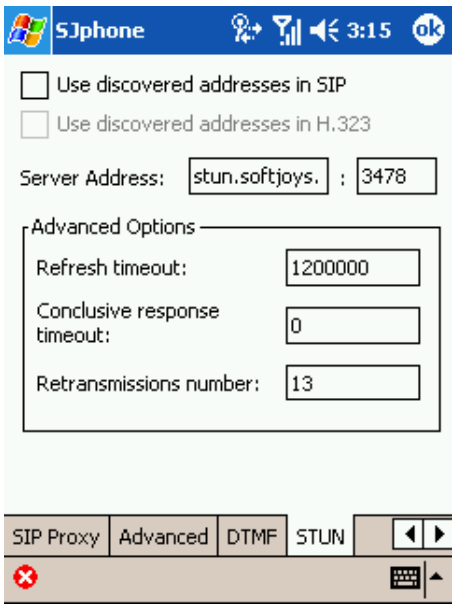


Figure 3.38: STUN Settings

The STUN tab allows us to determine which server will be used to help SIP find the IP address we use. The default STUN server used is stun.softjoys.com, with port 3478. So if you want to apply STUN to VoIP Rakyat, you can use UDP Port 3478 and 3479.

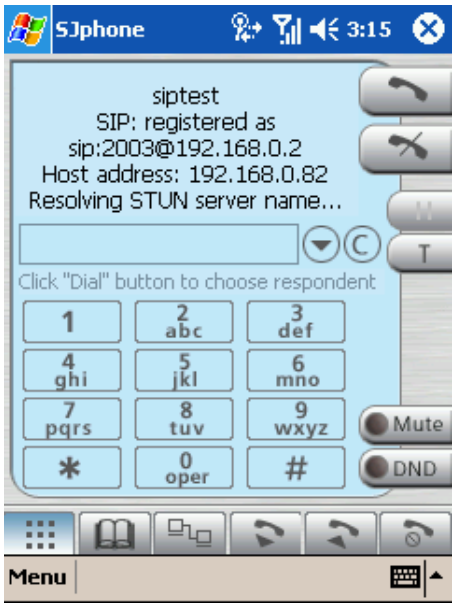


Figure 3.39: The appearance of the console showing successful SIP registration

If it is successfully registered, then the display of the screen will say “SIP: registered as number@server SIP”, with the host name also shown on the screen. Under this circumstance, SJPhone is ready to be used. We can place a call the way we use a regular cellphone with a PDA.

SJPhone Features

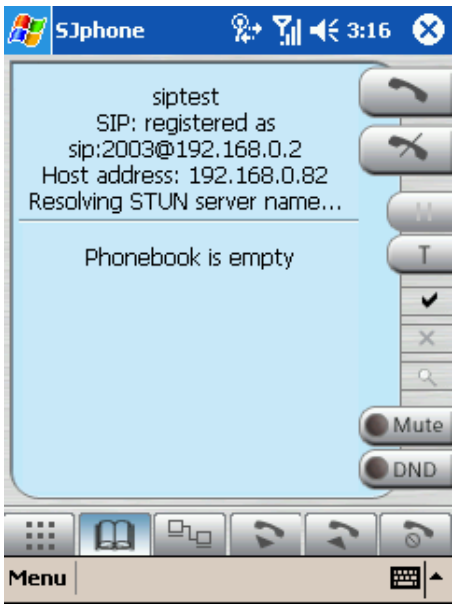


Figure 3.40: Tap the phonebook icon in order to save contact numbers and call them

There are several features provided by SJPhone to help users in using the phone, one of them is the phonebook icon (looklike an open book), which is located at close to the bottom of the screen. Through this option, we can enter the names and number of our friends.

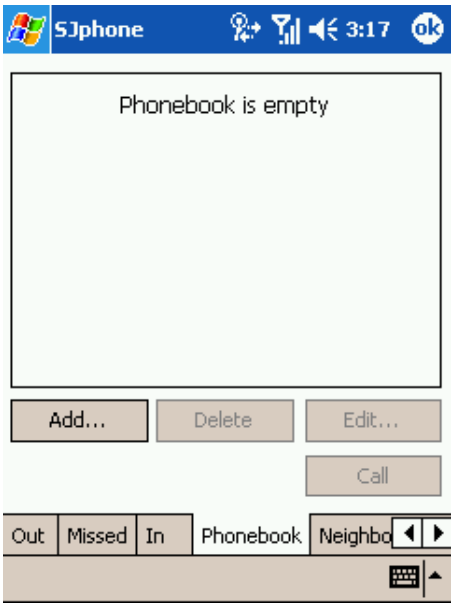


Figure 3.41: The phonebook is still empty, with the Add icon the only available option in the phonebook tab

To add a contact, simply tap Add, which is available in Phonebook tab.

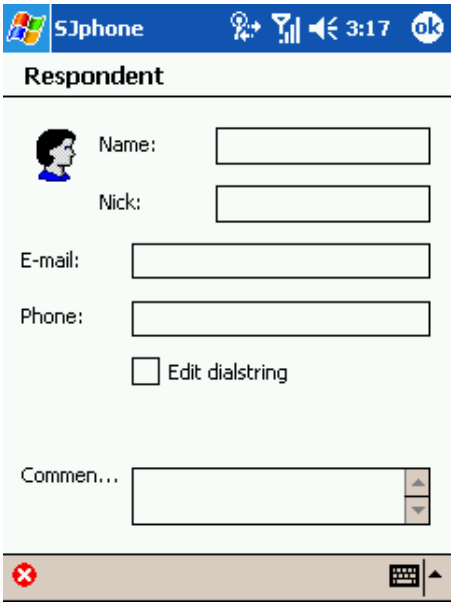


Figure 3.42: Enter the information pertaining to a contact

With the respondent dialog properties open, we need to enter the name, nickname (optional), email and phone number. You can also comment on the user, perhaps just in case you will forget who this person is.

Using SJPhone to place call through Ipaq 6395

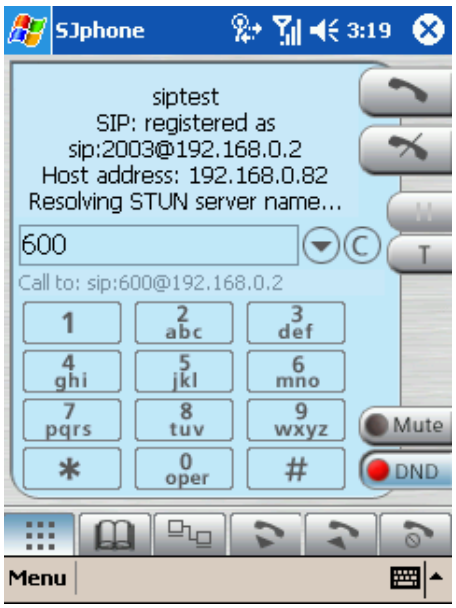


Figure 3.43: Dial some numbers in order to place a call

Figure 3.44: A call is successfully connected

To place a call using SJPhone in Ipaq 6395 is not difficult. All we have to do is to enter the destination phone number and press the dial key located on the top-right. If the call is connected, a message saying so, the duration time of the conversation and the codec in use will appear on the screen.

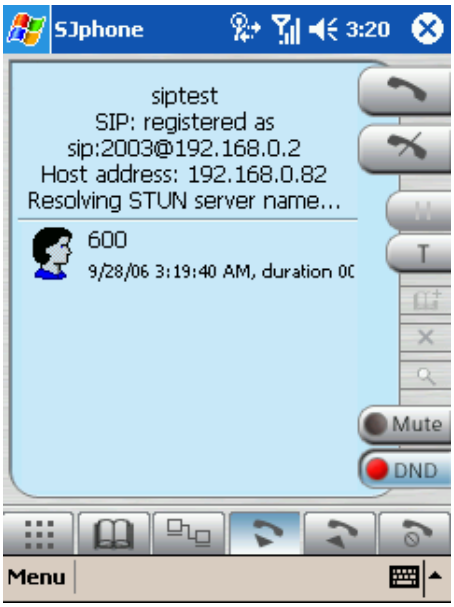


Figure 3.45: By tapping on the outgoing call icon, you can see the list of the numbers you have called and the duration of the conversation

Outgoing Call Statistic can be viewed by tapping on the phone icon with a triangular arrow pointing downward.

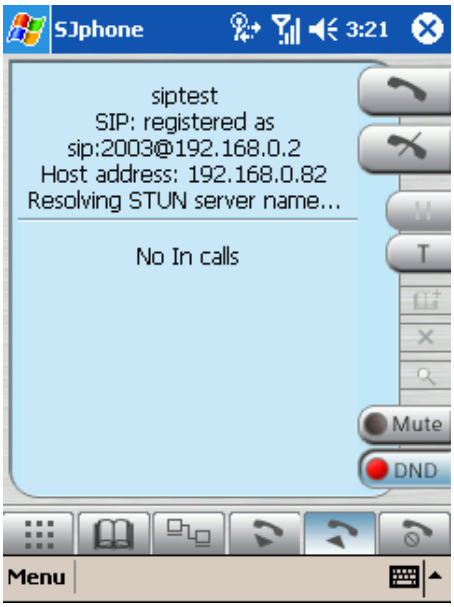


Figure 3.46: By tapping on the incoming call icon, you can see the list of the numbers dialing your number and the duration of the conversation

Incoming call statistics can be accessed on the tab available at the bottom of the screen, with the tab appearing as a phone with a triangular arrow pointing toward the phone.

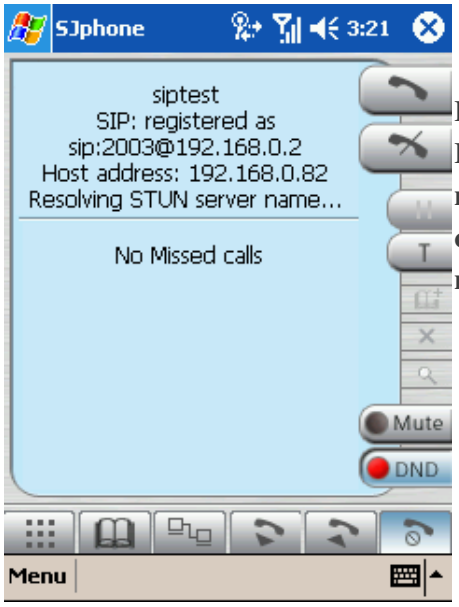


Figure 3.47: By tapping on the missed call icon, you can see the list of missed calls

Missed Call statistics can be viewed on the menu available at the bottom of the screen, with the icon appearing as a phone with a stop sign below it.

Nokia

As part of cellular major industry, Nokia seems to have recognized that internet telephony will be instrumental in the future. As such Nokia makes it possible for Symbian operating system to operate in Nokia handphone, providing customers with a cellular that can be readily used for internet telephony. In the example, we will use several Nokia handphone, such as, Nokia E61, Nokia E71 and Nokia N80. The former is more of PDA-type cellular phone while the latter is small in terms of dimension. Nokia E61, Nokia E71 and Nokia N80 are WiFi Phone.

The WiFi phone configuration for all Nokia is somewhat similar, with minor differences in terms of menu appearance. So generally, those who are used to Symbian should not encounter significant challenges in turning to these phones.

Figure 3.48: Nokia N80

Figure 3.49: Nokia E61

Nokia Wireless Configuration

Figure 3.50:

Nokia's console

Nokia's console display looks like what is shown in Figure 3.51. There are things to be configured so that Nokia can be connected to both WiFi and VoIP:

- Enable WiFi and create a profile of an access point that can be accessed.
- Create SIP account.
- Create a Profile from internet telephony facility.

Click the globe icon to open the menu.

Figure 3.51: By clicking the menu icon, we can select a variety of options.

With the menu open, select tools. Through this option, we can configure WiFi, SIP, internet phone and other settings.

Figure 3.52: There are many options available in Tools menu.

With the Tools icon selected, select Settings in order to access connection menu allowing us to configure WiFi, Internet Telephone and SIP settings.

Figure 3.53:
Under menu Settings, configure the Connection

In Settings, there are several options we can choose: Phone, Call, Connection, Date and Time and Security. To configure WiFi Access Point, SIP Settings, and internet telephony, we need to configure using the Connection submenu.

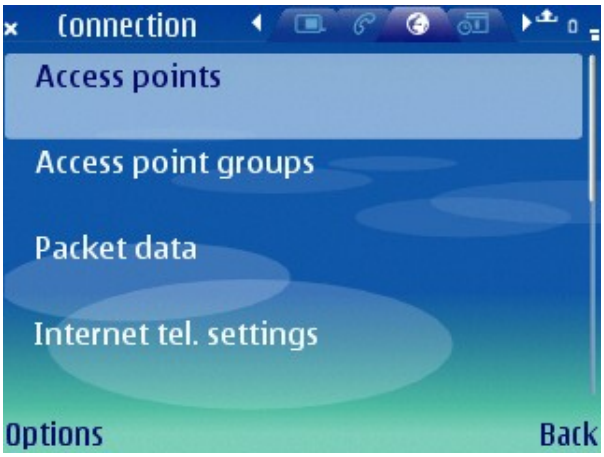
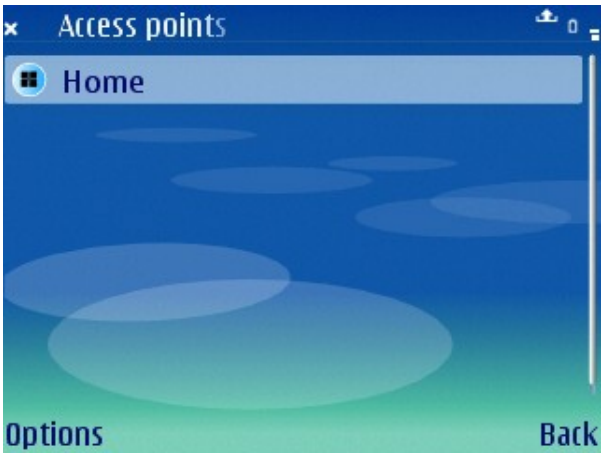


Figure 3.54:
Options available under Connection

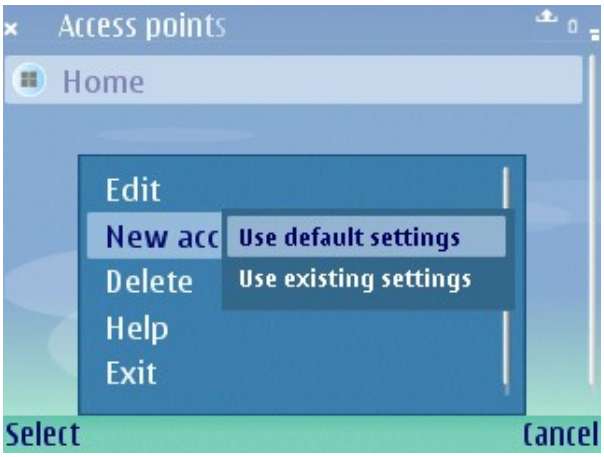
In the Connection menu there are a few more options. We need to configure only three of them: Access Points, SIP Settings and Internet Telephony Settings. Select Access points.

Figure 3.57:
There is no access point yet shown on the screen



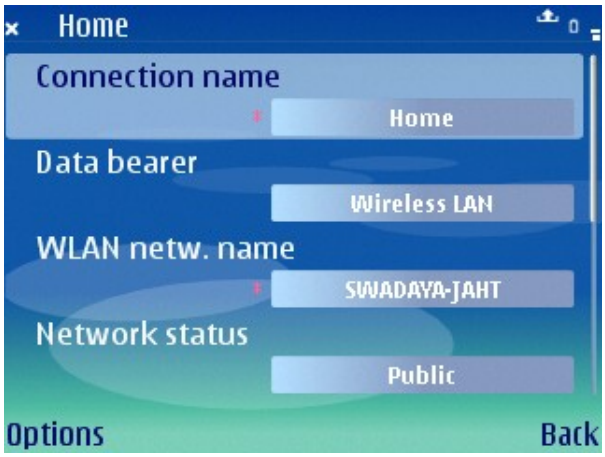
With the Access point menu open, we can add Access Point, by selecting the Options menu located at the bottom-left f the display.

Figure 3.58:
You can either make new access point or edit or delete existing access points.



There are several options available in the Access Points menu: Edit, New Access Point, Help, Delete and Exit. To add a new Access Point, select New Access Point, which will bring two more options: Use default settings and Use existing settings. Assuming that this is the first time you're using the phone, select Use default settings.

Figure 3.59:
Creating an access point profile



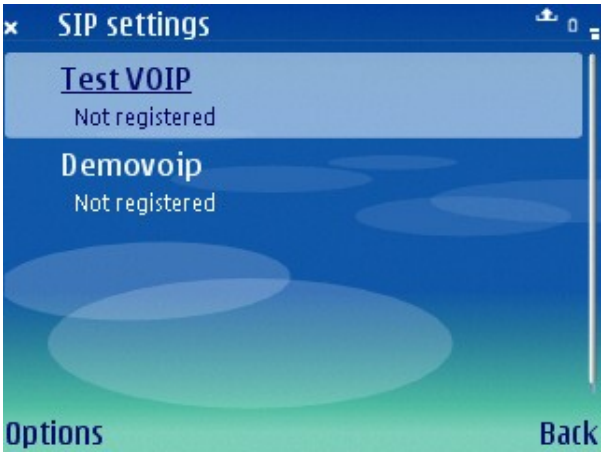
For creating an Access Point profile, we need to set the Connection name, type of connection (Data bearer), and the name of WLAN network. For data bearer, choose Wireless LAN.

Figure 3.60:
You can either enter the network information you already know or search for any networks reached by your cellular

If you know the name of the network, enter it manually, by selecting Enter manually. Otherwise, let the phone find any available network, by selecting Search for network.

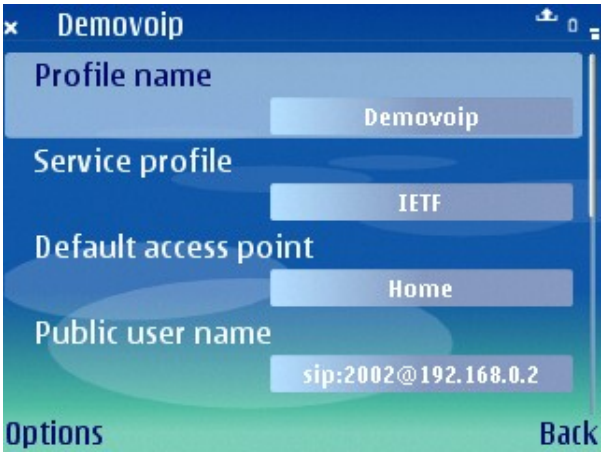
SIP Server and Account Configuration in Nokia E61

Figure 3.61:
You can do the configuration of SIP server and account by selecting Options



Through SIP Settings, we can configure SIP accounts that will be used for calling. The settings is done through Options menu in SIP Setting.

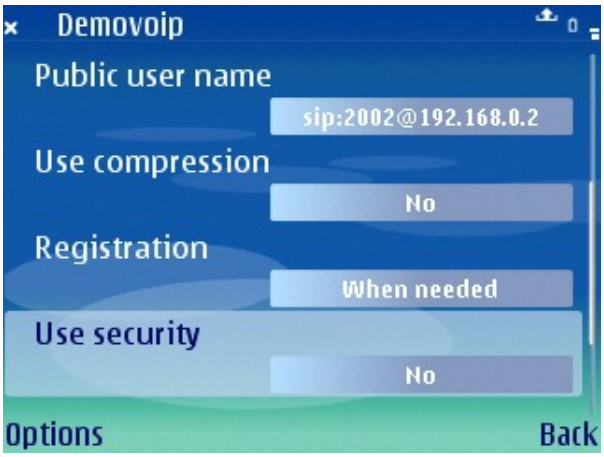
Figure 3.62: Settings Demovoip Profile.



There are some parameters of SIP Settings that need to be configured correctly:

- Create a name for Profile name.
- Choose IETF for Service Profile.
- Fill Default Access Point with information of Access Point profile we use to connect to the internet network through WiFi.
- Make sure that you fill Public user name parameter with the proper format of SIP number you use. For example, 23123@voiprakyat.or.id or 2002@192.168.0.2. The prefix "sip" will be added automatically in case that you forget to include it.

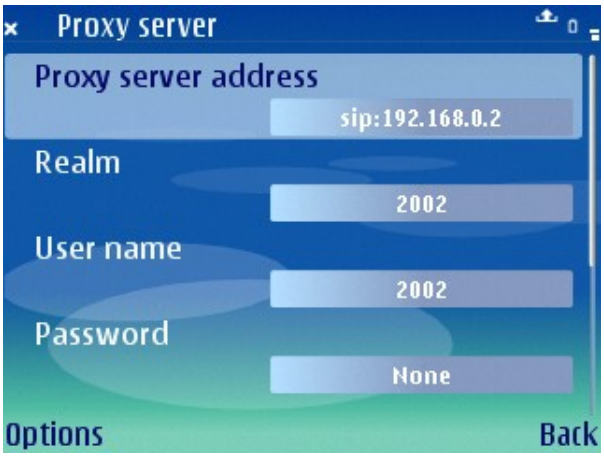
Figure 3.63: SIP Profile in SIP Settings in Nokia E61



Next we need to set the following parameters:

- Set Use compression parameter to No.
- Set Registration paremeter to When needed so that Nokia will prompt us whether we want to connect to a SIP softswitch each time we will use SIP Phone.
- Set Use Security parameter to No.

Figure 3.64:
Proxy server settings



Through the Proxy Server Address menu, we need to configure the following:

- Proxy Server Address.
- Realm – for some reason, it is best to fill this parameter with a telephone number similar to our username. In Asterisk IP PBX, the default realm is asterisk.
- Username – telephone number or SIP username.
- Password – leave this blank.
- Set allow loose routing to Yes.
- Fill in Transport Type with UDP.
- Fill in Port with 5060.

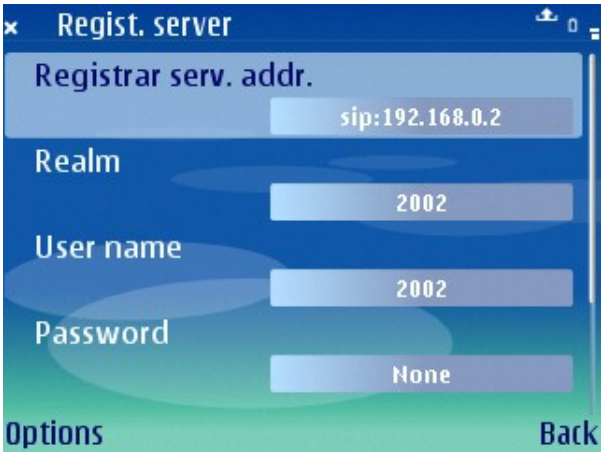


Figure 3.65:
Regist. server settings

In Registrar Server, we need to configure the following parameters:

- Fill in Registrar Server Address with hostname or IP address of our SIP server. For VoIP Rakyat, enter voiprakyat.or.id.
- Fill in Realm with the telephone number or username.
- Fill in Username with SIP telephone number.
- Leave password blank

Internet Telephone Configuration in Nokia

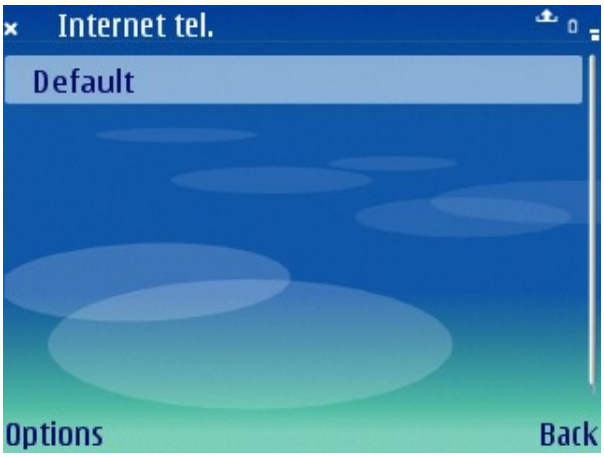


Figure 3.66: Internet Telephony settings

In Internet Telephony Settings, we can create a profile of Internet telephony facility that will be used using Nokia. To set the profile, select Options of the Internet Telephony Settings.

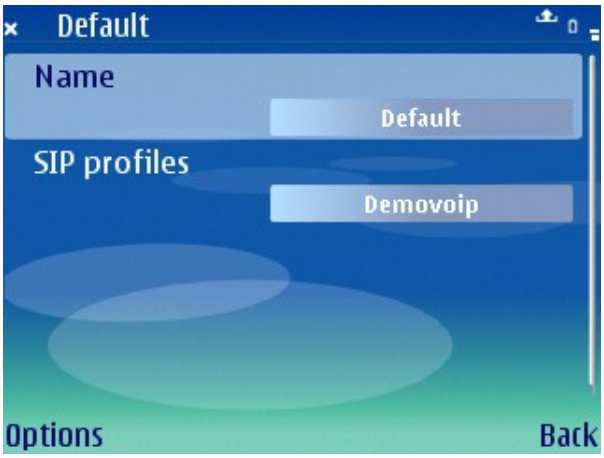
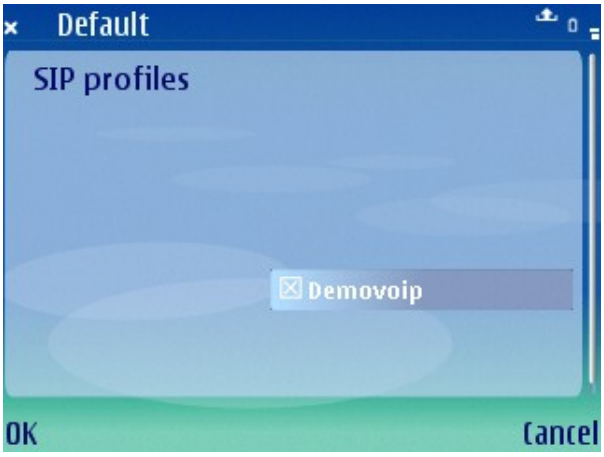


Figure 3.67: Make sure the profile chosen is to be used as a default profile

In Internet Telephony Profile Settings, we need to include only the profile name and SIP profile that will be used for Internet telephony. Be careful when you're doing so. Make sure that the profile selected is to be used as a default profile, otherwise our call will be rejected when we attempt to dial using our cellular to the VoIP number. All this ordeal is unnecessary if we have just one SIP account.

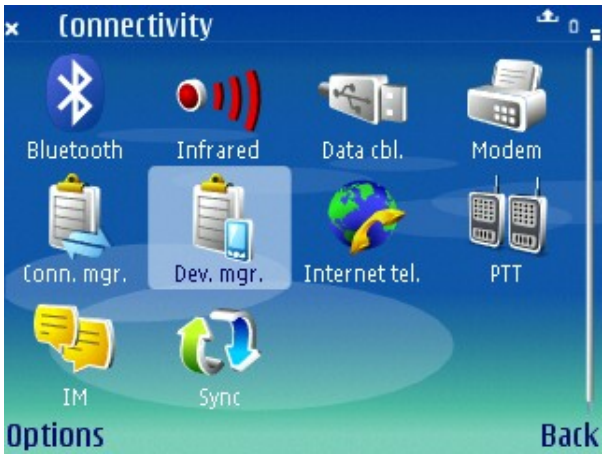
Figure 3.68:
Selecting a profile



The SIP Profile selection will be carried out manually by selecting a variety of SIP Profiles we have created through SIP Settings.

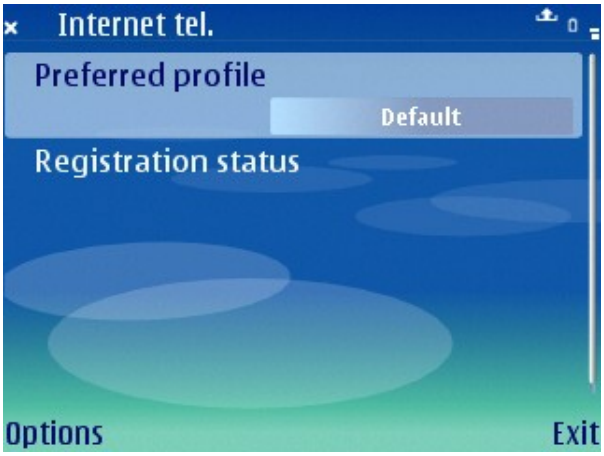
Registering to VoIP Softswitch

Figure 3.69:
Connectivity Settings



For establishing connection to VoIP, select Internet tel. (shown as a globe icon with yellow phone). This is assuming that you have properly configured Internet Telephony settings. Unlike Nokia E61, Nokia N80 connects to VoIP through option available in a folder labeled Internet. Go into the folder and choose Internet Telephone.

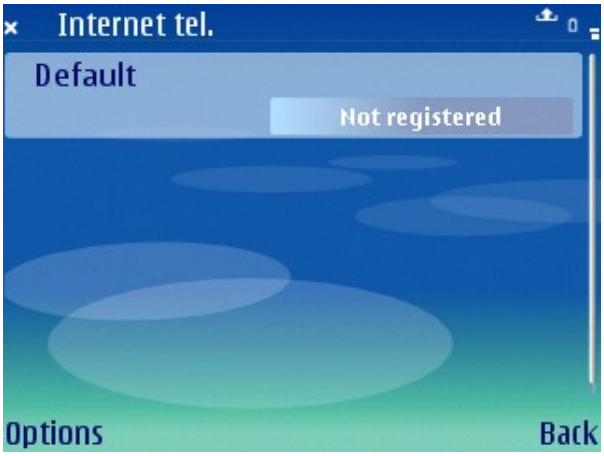
Figure 3.70: Internet telephony settings



In Internet telephony (shown as Internet tel.), we will be provided with two parameters:

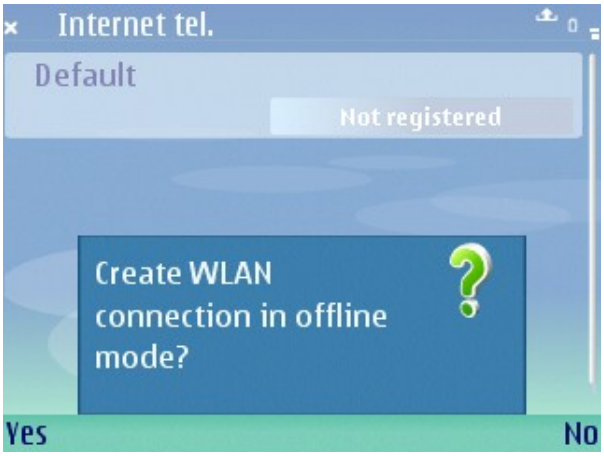
- Preferred Profile, the name of Internet Telephony Profile we use.
- Registration Status, the registration status of SIP account we set in SIP Settings.

Figure 3.71: Internet telephony settings



If we choose "When needed" in the Registration parameter in SIP Settings, the status of initial condition of internet telephony setting, when Internet telephony is active, is Not registered.

Figure 3.72: Enable WLAN connection in offline mode so Nokia E61 can be connected to a WiFi network



If we attempt to change the status from Not Registered to Registered, what Nokia will firstly try to establish connection to the Access Point which we have configured in SIP Settings. When Nokia asks whether you want to create WLAN connection in offline mode, select Yes. This selection will connect Nokia E61 to a WiFi network.

Offline mode can somewhat be problematic, because if we are in offline mode, it means that although we are registered with the SIP server, people are still unable to contact us. To make sure that we can be contacted via GSM, we need to activate Nokia so it becomes online mode. Online mode will be possible only if we are using SIM card in the phone and are connected to a cellular network. In online mode, other users will be able to contact us through both VoIP or GSM.

Figure 3.73:
A registration attempt in progress

Once connected to a WiFi network, we have to wait for a while to let Nokia register itself with the Softswitch.

Figure 3.74:
The registration is completed

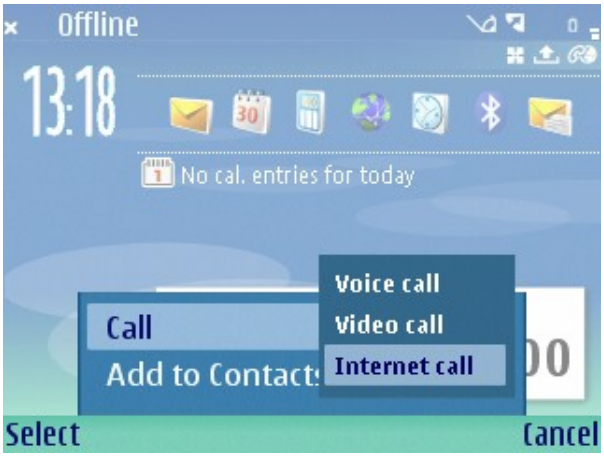
If registration with the internet telephone is completed, there should be a notification saying so, as shown in figure 3.73. Such notification indicates Nokia can now be used for internet telephony.

Calling using Internet Telephone in Nokia E61

Figure 3.75: Initial display of Nokia E61

Placing a call using internet telephone in Nokia is similar to how we call using other phone: We just need to type the phone number to which we want dial.

Figure 3.76: Once the number is dialed, we need to choose the type of call.



Then Nokia will ask whether the call is of Voice call (GSM), video call or internet call. Select Internet Call to place a call using internet telephony. If we choose Voice call, then the mode of communication used to connect our call is of GSM.

Figure 3.77:
The phone icon with a small globe next to it indicates that the call is established

When the call is established, we will get a notification on the screen that our telephone number is connected to the destination number.

Figure 3.78:
You can either mute the sound, activate handset, end active call, hold the call, make the call open active standby and place new call

To disconnect a call, simply select End active call.

VoIP in ADSL Modem

Even when VoIP is widely used nowadays, there are only few ADSL modems with built-in VoIP equipment. One of them is Linksys WAG54GP2, a small modem that has two VoIP equipments. Configuration can be done entirely using the web, making it very convenient for both users and system administrators.

Figure 3.79: an ADSL modem

ADSL Modem Configuration

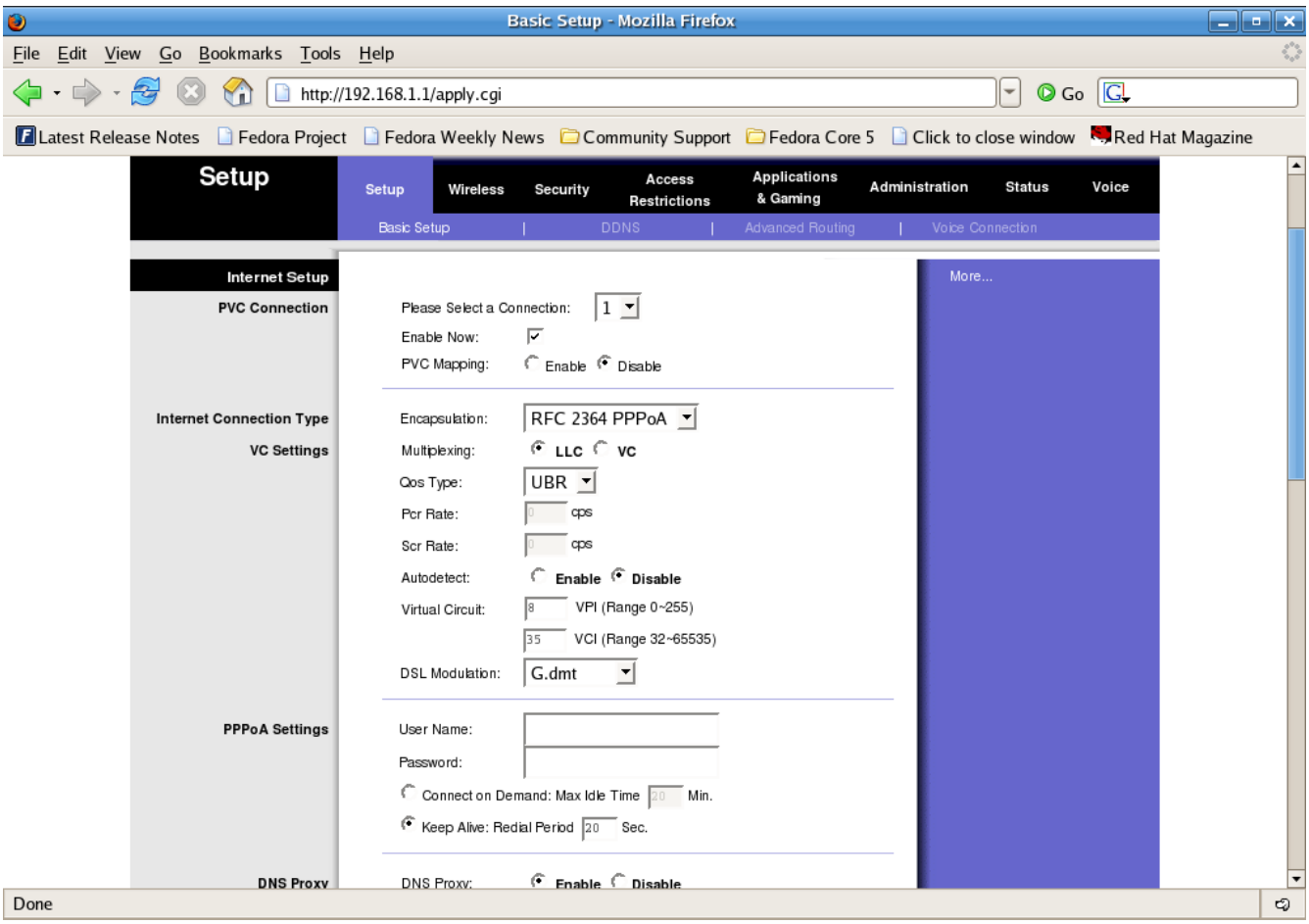


Figure 3.80: The Basic Setup sub-tab under the Setup Tab of the modem administration panel

After entering the administrator password and username (default is admin for both), we will be directed to the setup page of WAG54GP2 Linksys ADSL Modem. Through this page you can configure several things such as:

- Configuring the connection to the Internet, type of modulation used, encapsulation, multiplexing techniques used, VCI and VPI value of the ADSL connection.
- Configuring PPP, username and password
- DNS Proxy Server

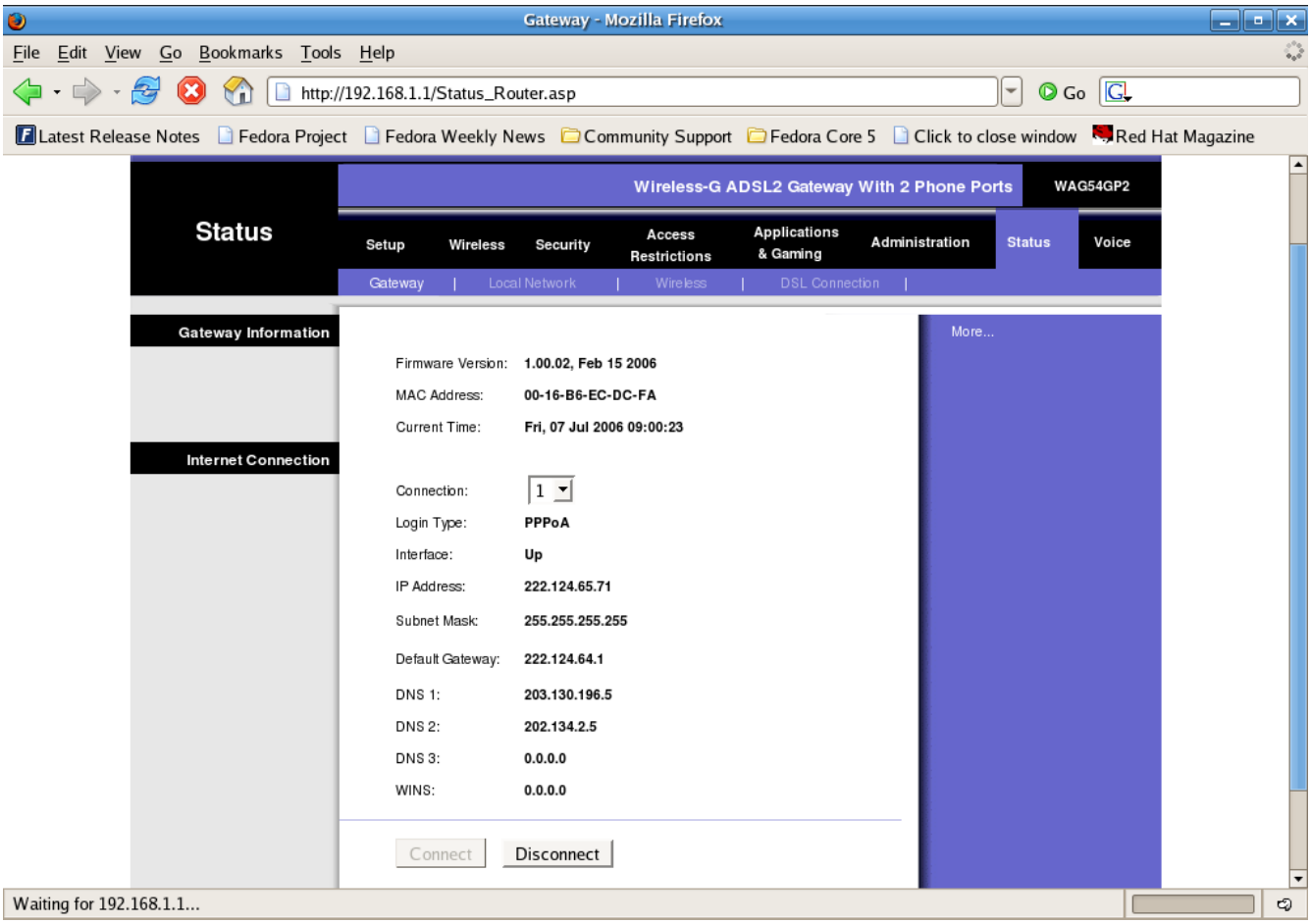


Figure 3.81: The status tab of the modem administration panel

The setup page also provides information on gateway, connectivity connectivity condition in PVC status, and internet connectivity condition including the IP address, Subnetmask, Default gateway, DNS and facility used to connect or disconnect a connectivity.

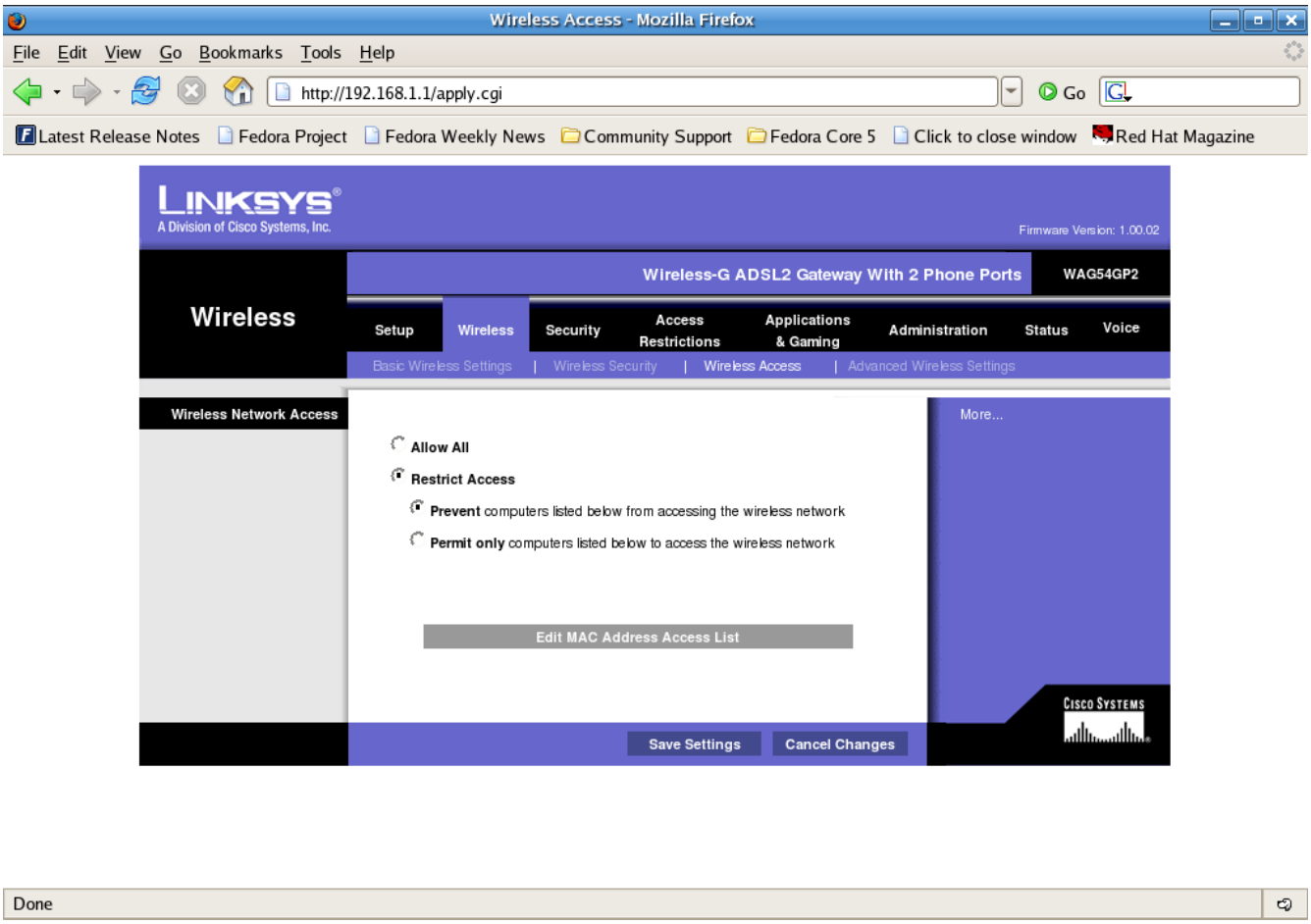


Figure 3.82: The Wireless tab of the administration panel

In addition we can also configure a variety of facilities available in Linksys WAG54GP2 ADSL Router through the web:

- Wireless
- Security
- Access Restrictions
- Application & Gaming
- Administrator

Each of these has submenu, which we will not explain any further, as we will focus more on the VoIP feature of the modem.

VoIP Configuration in Linksys WAG54GP2

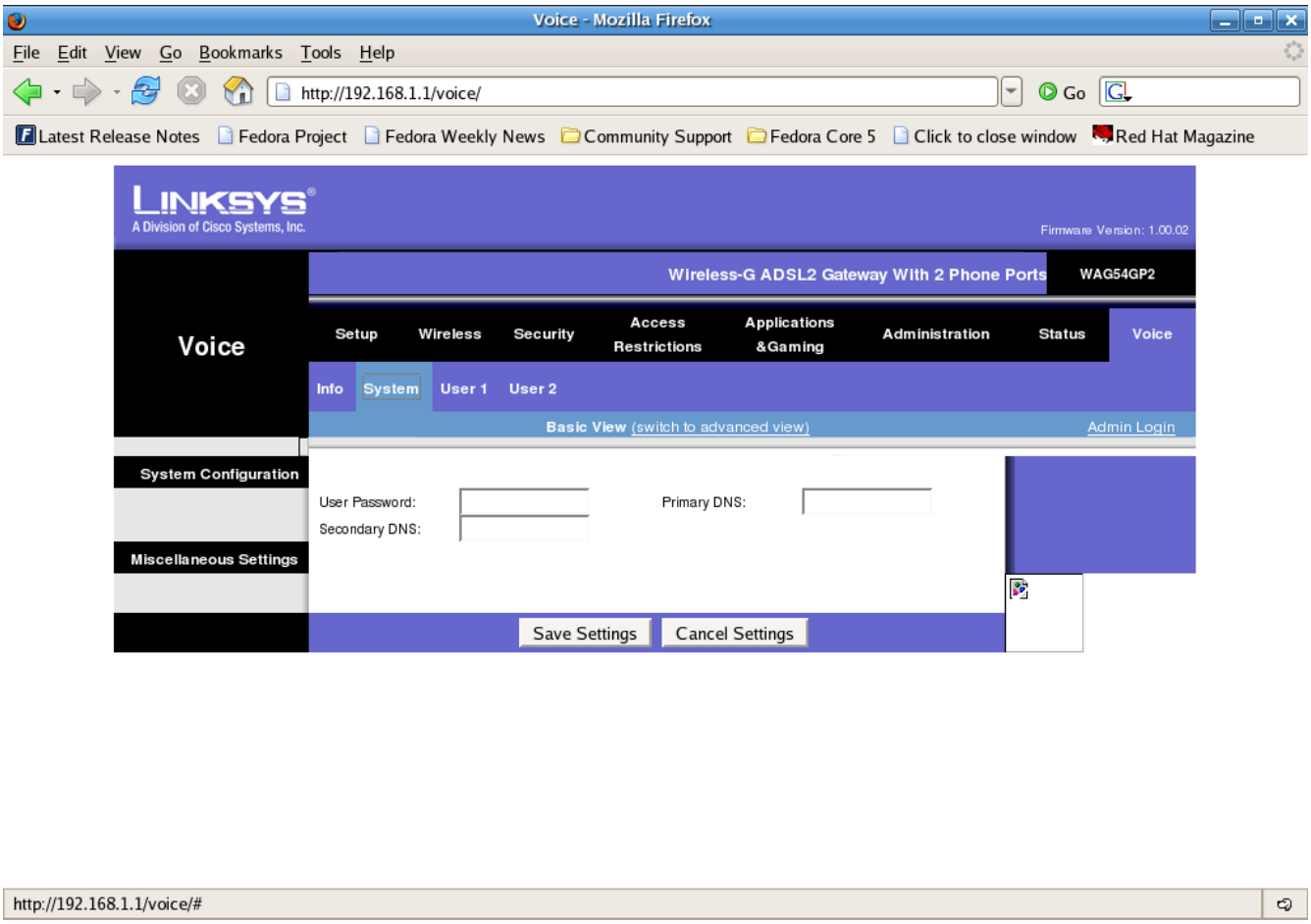


Figure 3.83: The System tab of the modem administration panel

The menu for configuring VoIP on WAG54GP2 can be found in its Voice menu. In general, how to configure the device is not different from the configuration other Linksys equipments, with the following steps:

- User mode is primarily used to view any existing configuration.
- Admin mode is mainly used to change the configuration.

In order to set the SIP account, we need to change the basic view to advanced view in the Admin mode.

The information required to set the SIP account are as follows:

- Username/ telephone number.
- Password.
- SIP Server address.

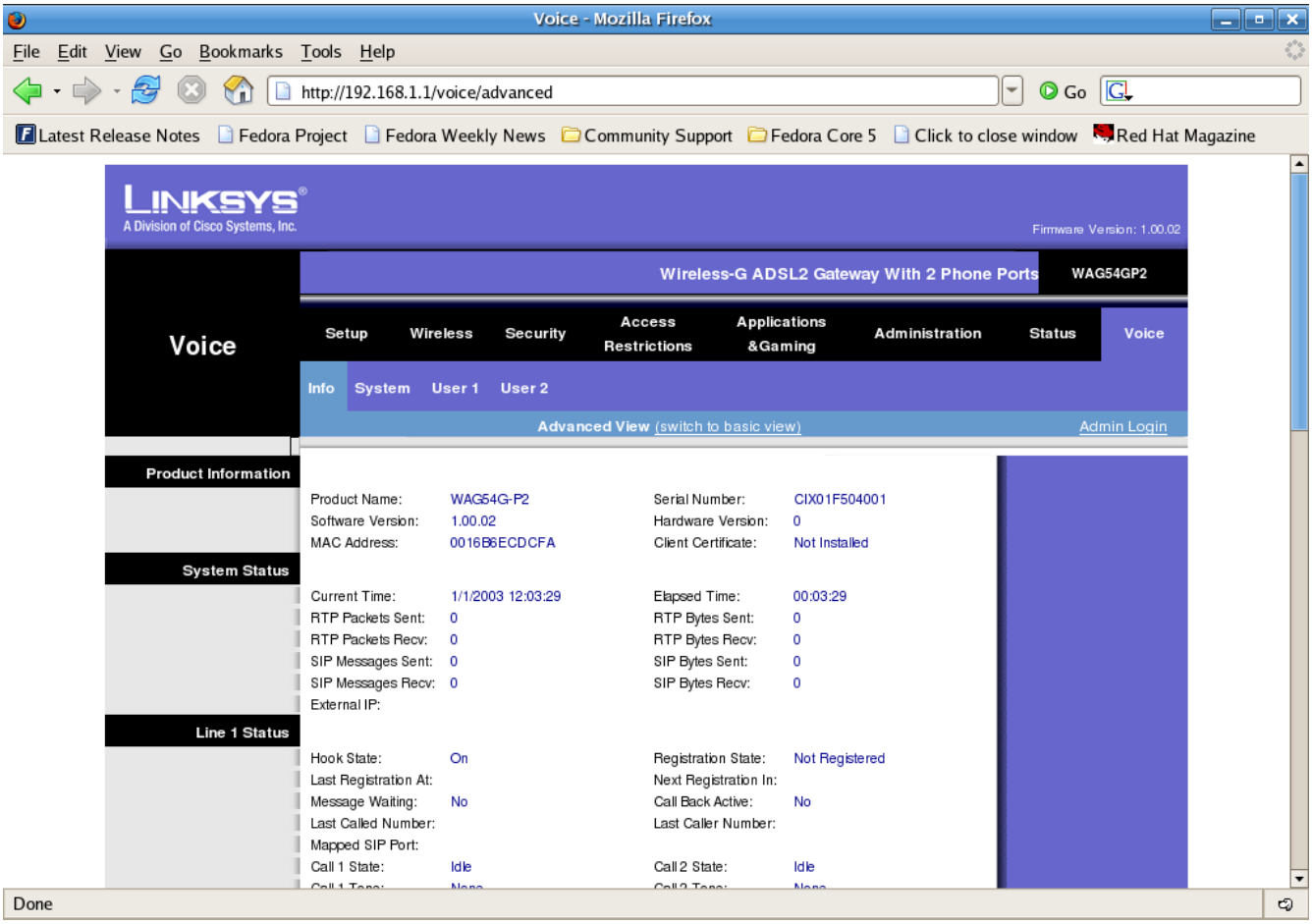


Figure 3.84: The Info tab of the modem administration panel

It is recommended that you look into the Info submenu available in Voice menu. What you have to look in particular is the Line status, specifically the registration state parameter. Once everything is properly configured, ensure that what is stated in the Registration State is Registered.

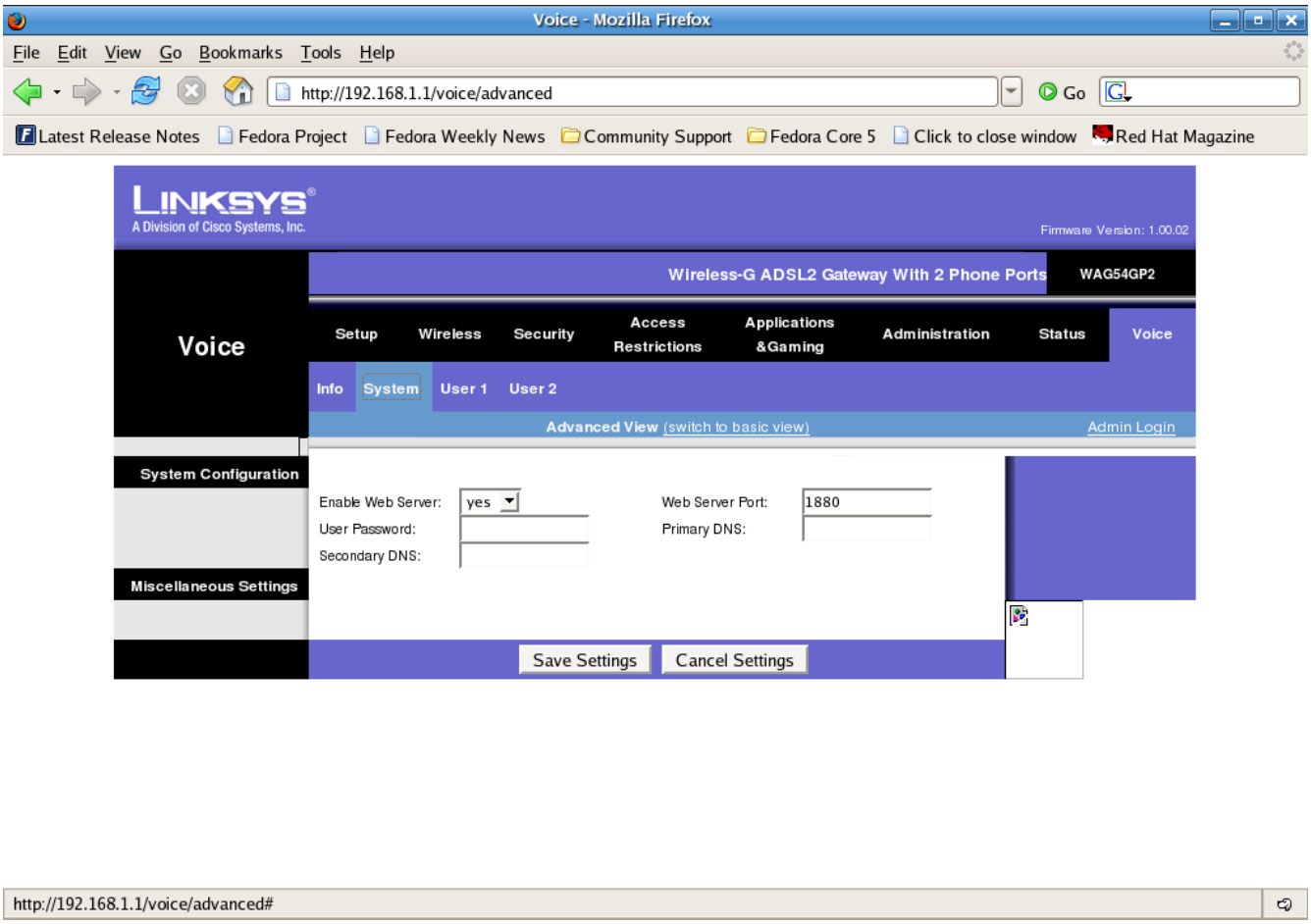


Figure 3.85: The System tab of the modem administration panel

System configuration uses web from VoIP Linksys WAG54GP2 through a specific port, with 1880 as its default value. This port can be enabled or disabled through system menu. Don't forget to click Save Settings to store the configuration settings.

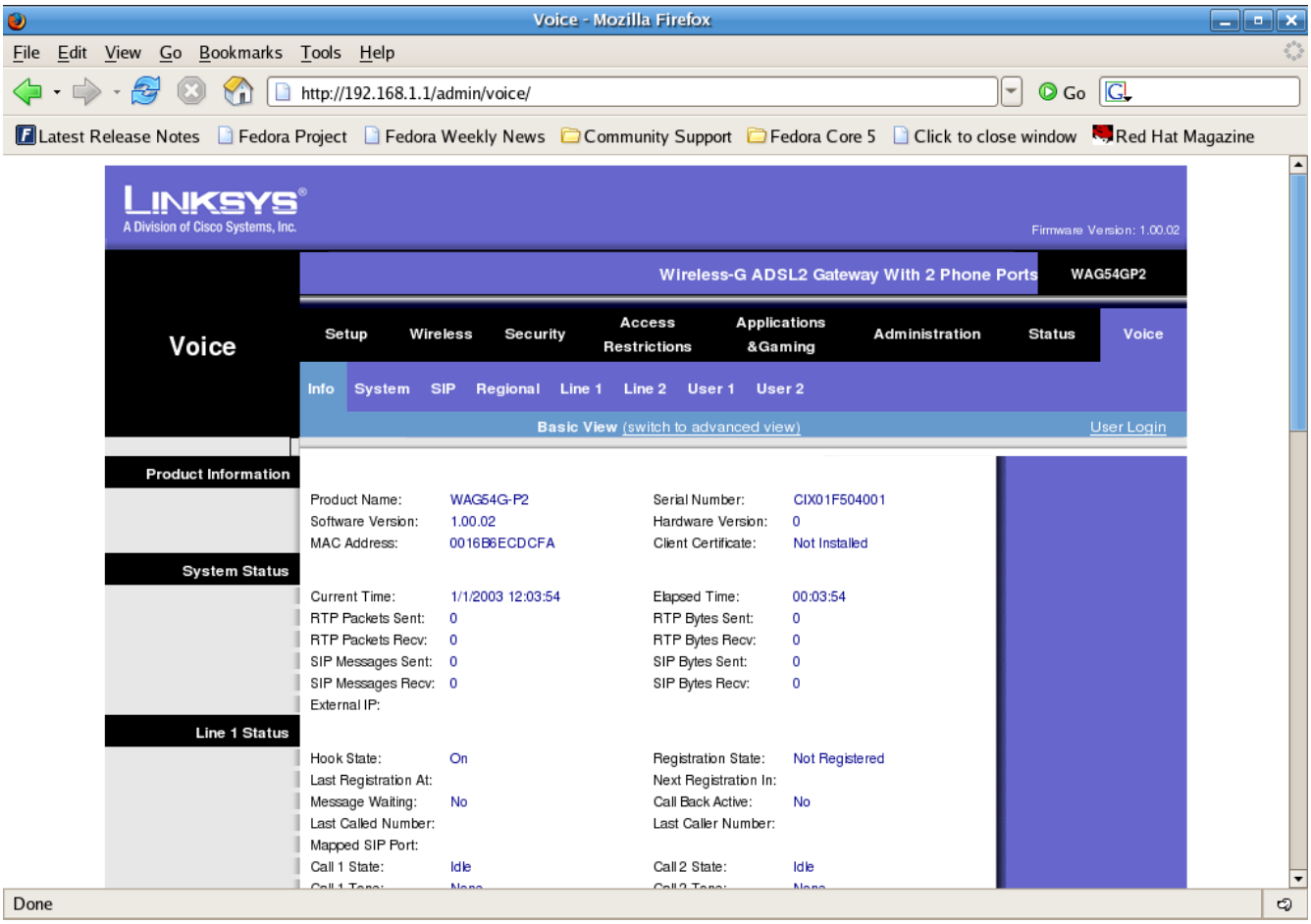


Figure 3.86: The Info sub-tab of Voice tab of the modem administration panel

In Advanced View, we will obtain more information. What we need to access is Line 1 and Line 2 menu so as to configure SIP account in SIP softswitch used. Other parameters in other menu beside Line 1 and Line 2 need not to be changed.

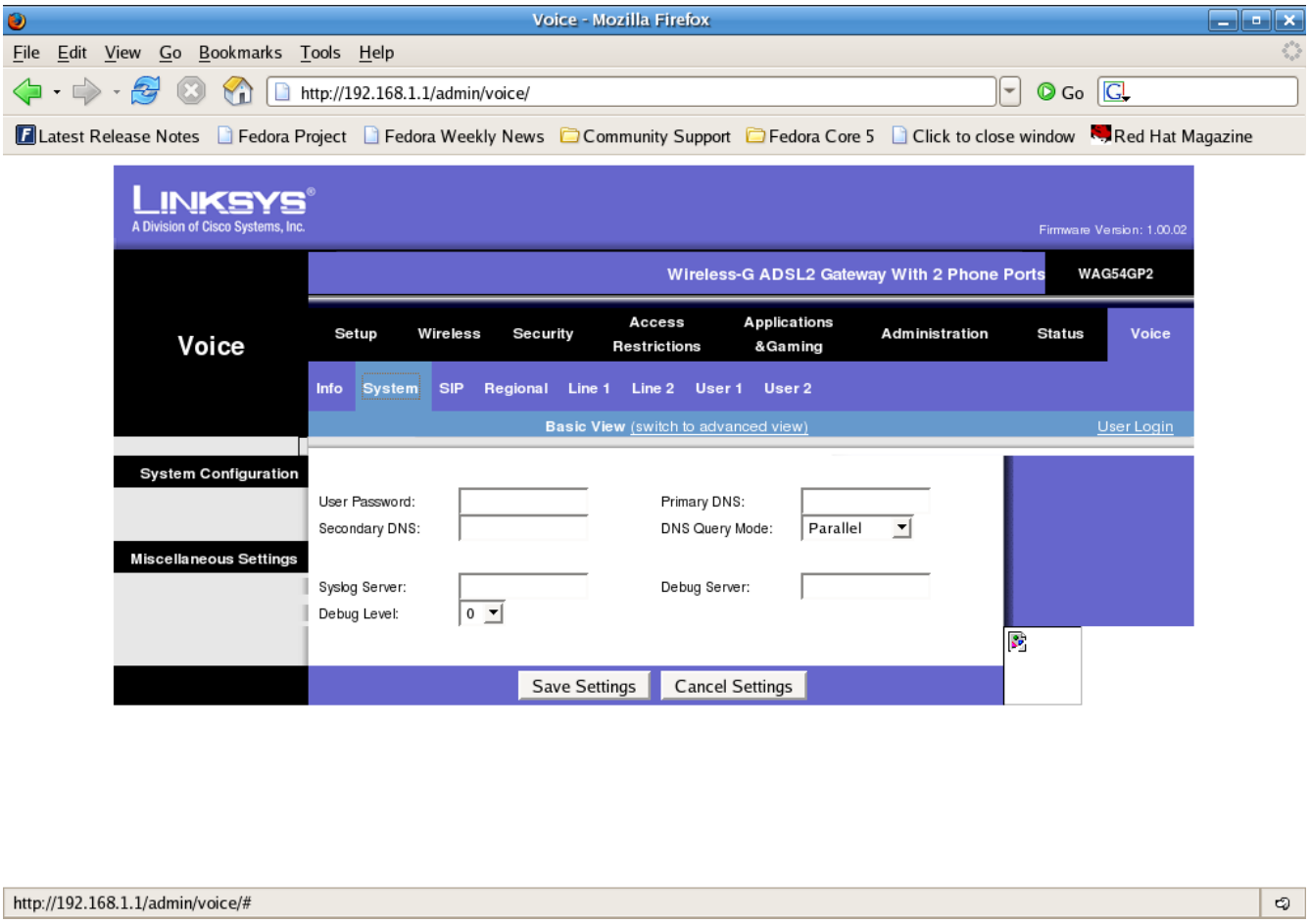


Figure 3.87: The System sub-tab of the Voice tab of the modem administration panel

In system menu, if necessary, we can include Primary and Secondary DNS parameters.

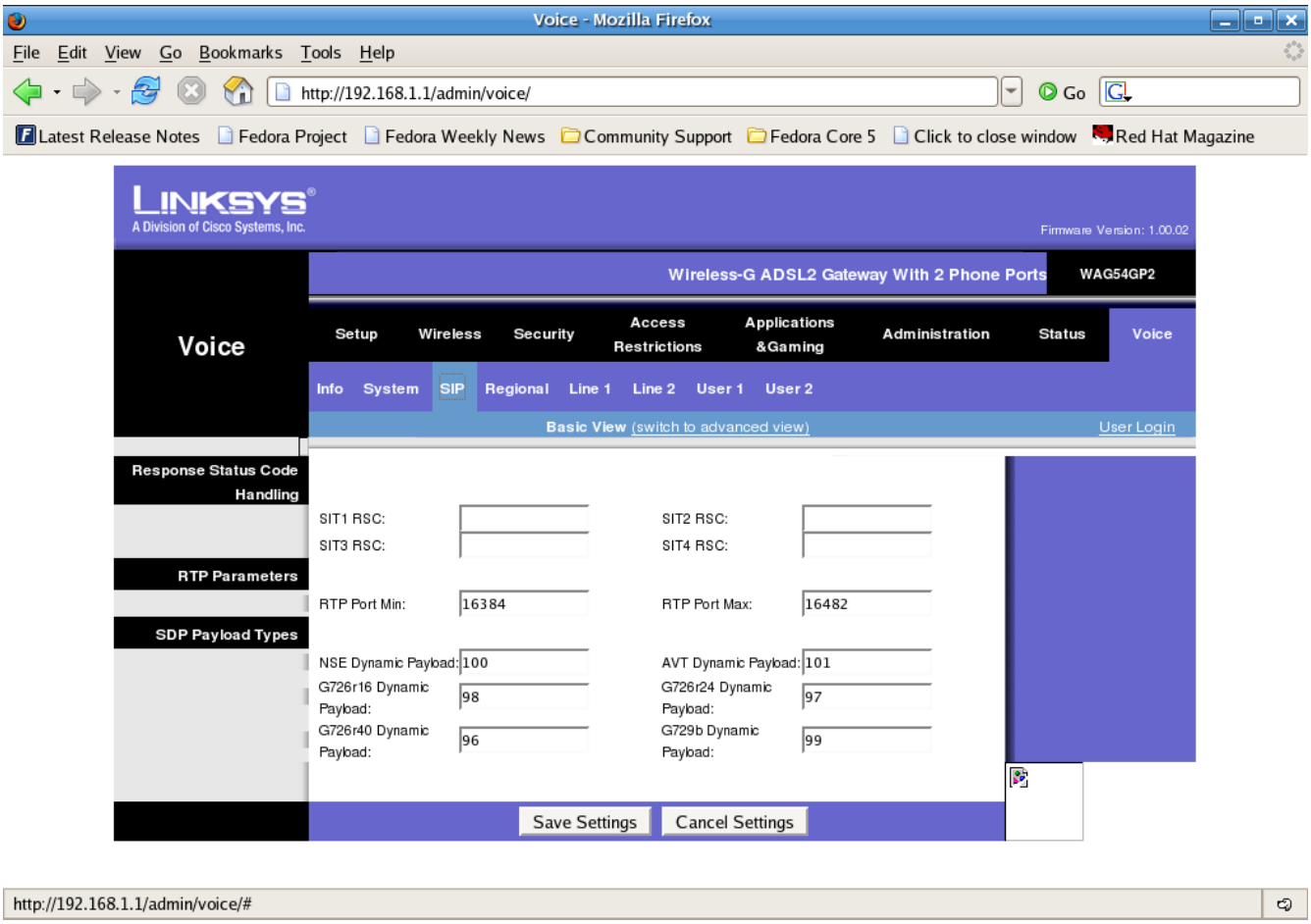


Figure 3.88: The SIP sub-tab of Voice tab of the modem administration panel

Through the SIP menu we can configure the ports, payload, CODEC, etc. Basically, these parameters need not to be changed. We can still use its standard parameters to achieve good results.

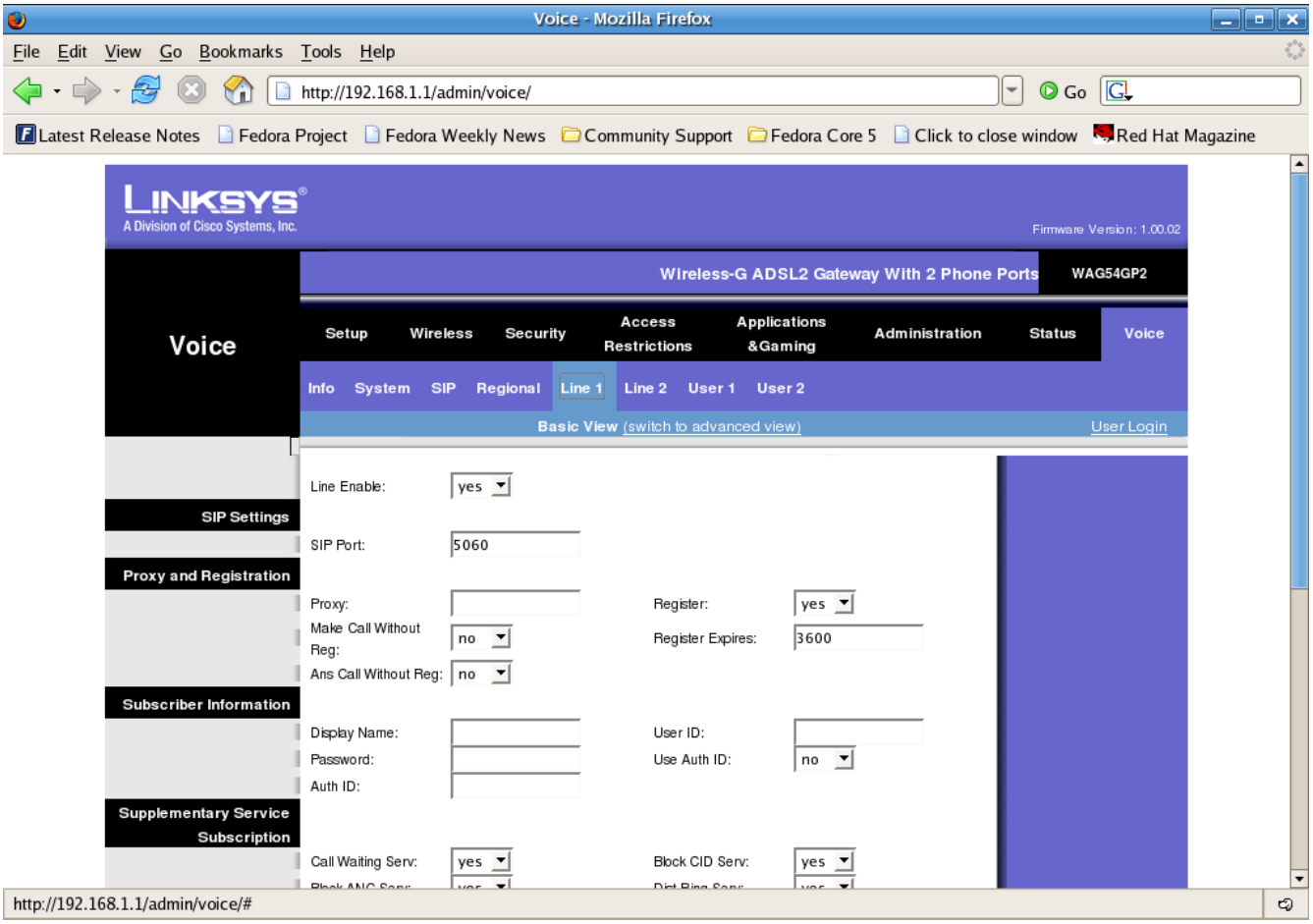


Figure 3.89: Line 1 sub-tab of the Voice tab of the modem administration panel

In Line menu parameter, we can set SIP account that is used to register with VoIP softswitch. The parameters we have to set are as follows:

- Line Enable – set to Yes so the line becomes active.
- Proxy – fill in with name/hostname/IP address of the softswitch to be used.
- Display Name – fill in with VoIP phone number.
- User ID – fill with VoIP phone number.
- Password – fill with VoIP password.

Use Auth parameter usually is set to No. If it is set to Yes, we need to fill in the Audth ID parameter with the VoIP telephone number. The same settings also applies to Line 2.

CHAPTER 4: Interconnectivity and Telephone Number Allocation

Questions mostly asked by VoIP users is whether VoIP can be used to dial—and receive calls from—a PSTN or cellular number, since one of our purposes in using internet telephony is that we want to have this two-way interconnectivity, particularly to be able to receive calls from PSTN or cellular. Unfortunately, this might not be as easy as we think, as calls originating from PSTN or cellular can reach often only telephone numbers legitimately recognized by the PSTN or cellular. These numbers are allocated by E.164, the official numbering system acknowledged by the International Telecommunication Union.

However, the good news is that there are some leeways making it possible for us to call PSTN or cellular numbers using VoIP. One of which is by subscribing to a VoIP provider that provides us with PSTN numbers. On the other hand, we can also register our PSTN number to international VoIP network so our VoIP account can be recognized as a legitimate PSTN number. This section will help you understand in detail the technique for obtaining the number. We could get a free phone number from Washington State, US, (http://www.ipkall.com) which enable cellular/PSTN users to call us on VoIP network.

Note that to receive a call from PSTN provider, it is sufficient to use a computer, softphone and internet connectivity, preferably kept online for 24 hours, as if we are using ordinary phone. Obviously, as we have explained earlier, it is preferable to use VoIP hardware such as IP Phone.

Getting Free Washington State Telephone Number

Figure 4.1: You can get a free phone number from IPKall

A website that provides Washington State telephone number for free is IPKall http://www.ipkall.com, with the number having +1 prefix, the conventional country code for United States of America. It is

interesting to note that this number, although available as a virtual number, can actually be called from other PSTN number in different countries, with each country's international rates applied to the call. To be able to enable the number, you need to have a SIP account from a SIP provider or use the one you have created in VoIP Rakyat.

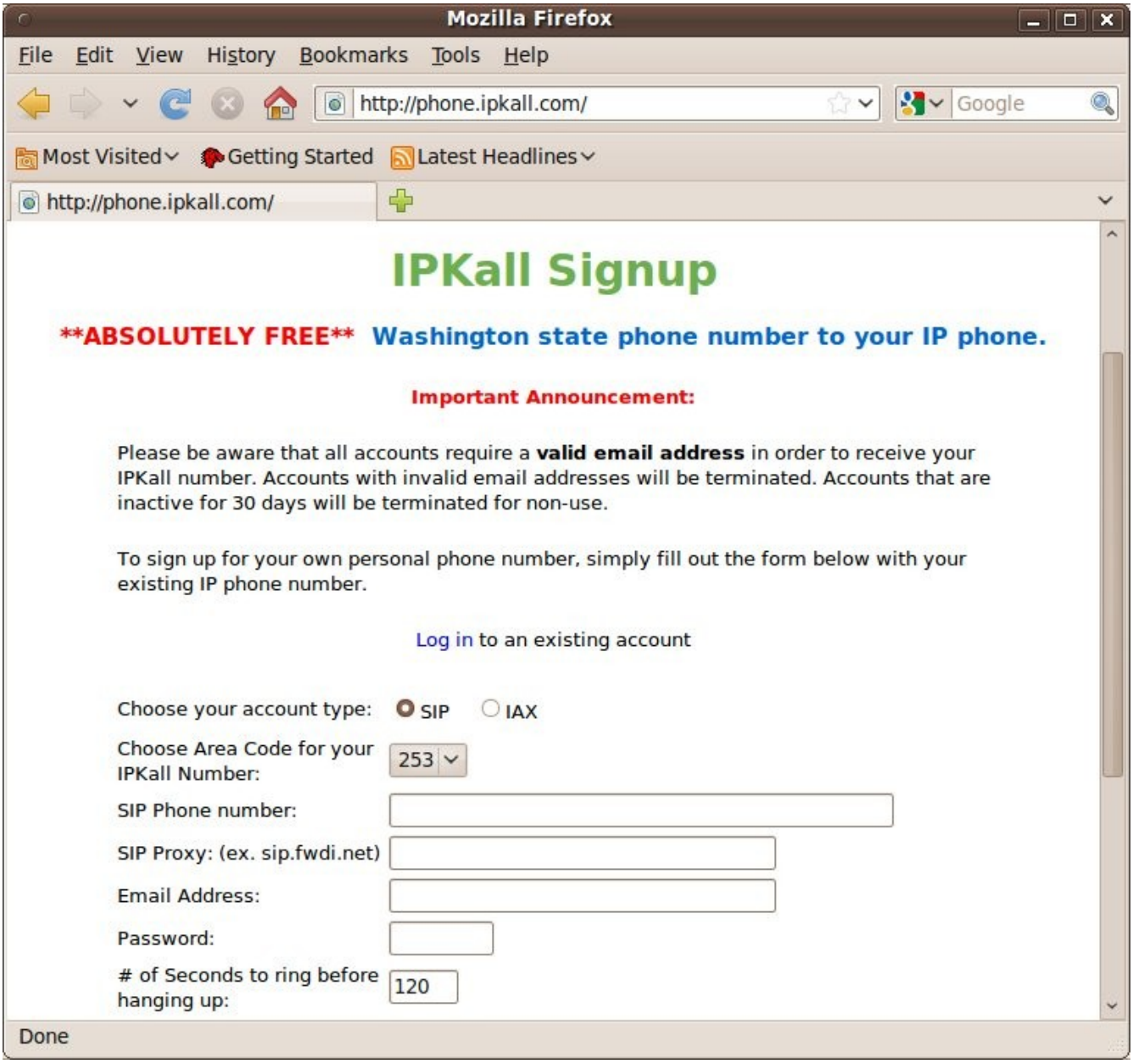


Figure 4.2: You can log on using an existing account, or create a new account on the spot

Once we have a SIP account, the next step we have to do is sign up to www.ipkall.com in order to get Washington State's telephone number. In the sign up pane, choose any of the following the area code: 206, 253, 360, and 425. Whichever number you choose, enter additional information on the SIP phone number given by a SIP Provider (in our case, it's the number given by VoIP Rakyat), SIP Proxy (voiprakyat.or.id), our email address for confirming the account we are creating, and the password for making changes in IPKall account. Type in the Captcha graphical words. After all parameters are filled correctly, click Submit to proceed.

Normally, we have to wait for about an hour to receive the confirmation sent through email. To activate your IPKall account, click the URL obtained from the email. With the account confirmed, you now have the State of Washington phone number with which you can receive calls from other PSTN across the world through your SIP account.

Free Internet Country: Country Code +882

One of the services important to VoIP is ENUM, which carries out mapping from IP address to telephone number using Domain Name System (DNS). The phone number system for telephone we are familiar with, the one with specific country code, is known as E.164 format. The mapping process is usually performed by a DNS machine in the Internet, through NAPTR special entry. There are two (2) main Top Level Domains used as reference in ENUM process: e164.arpa and e164.org. The former is the top level domain normally used by formal telecommunication institution controlled by the International Telecommunication Union. In your country, e164.arpa is possibly under the control of the country's ministry of telecommunication.

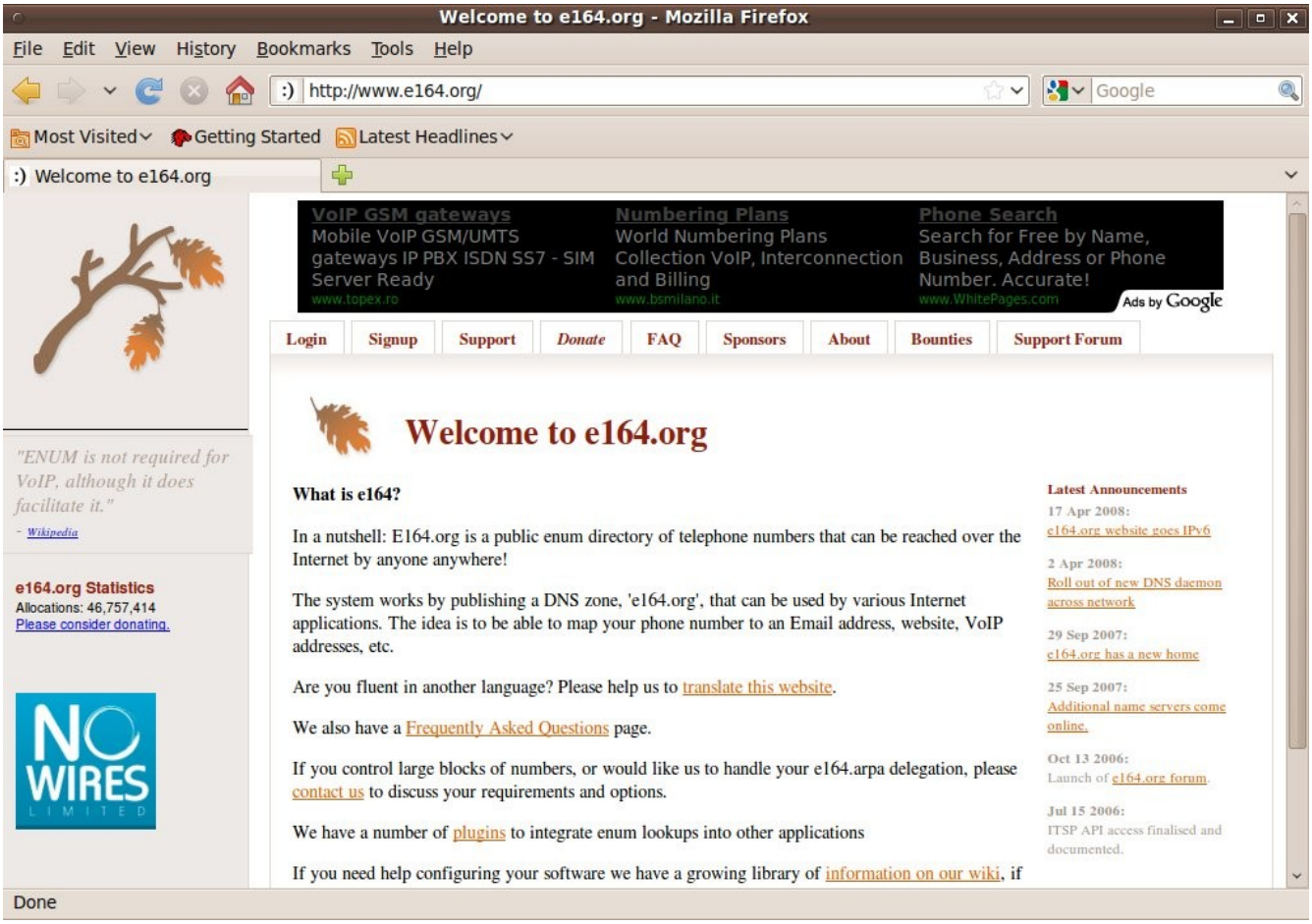


Figure 4.3: e164.org main window

The latter, e164.org, is the informal level domain provided by communities, the sort that are concerned with how people can minimize telecommunication cost. This is the domain we will use for our VoIP communication. We can register in http://www.e164.org to get an account that can be used to obtain a phone number and register the number.

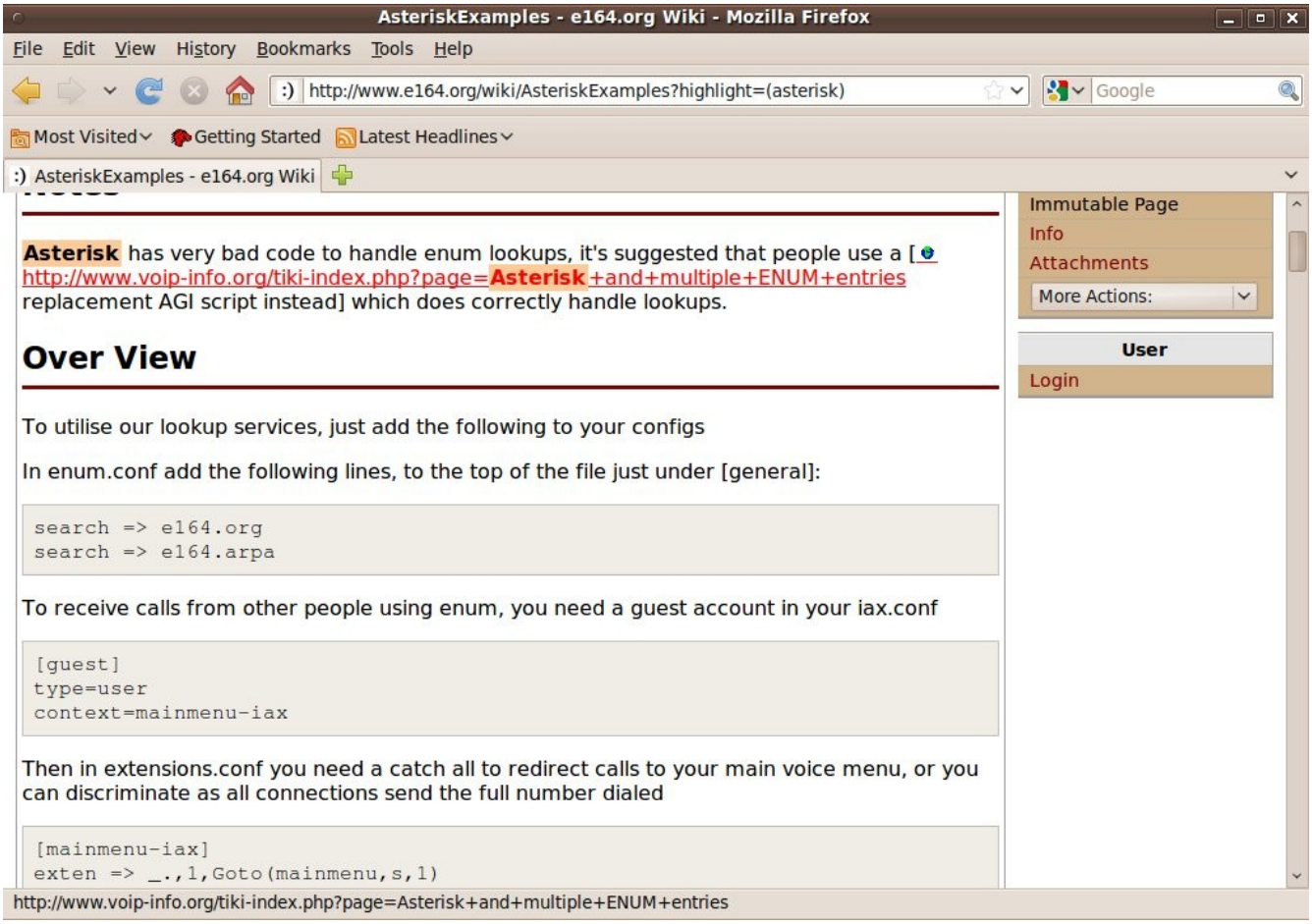


Figure 4.4: To use e164.org, simply follow the instructions shown in http://www.e164.org/wiki/AsteriskExamples

For smooth interconnection process between asterisk softswitch and e164.org, we need to configure /etc/asterisk/enum.conf so the Asterisk softswitch will be able to recognize the numbers listed in e164.org domain, by activating the following parameters:

search => enum.voiprakyat.or.id

search => e164.org
search => e164.arpa

Once these parameters are activated, the softswitch will automatically seek the PSTN numbers available in e164.org and e164.arpa. Since we are using VoIP Rakyat as an example in this book, we will refer you to enum.voiprakyat.or.id, an ENUM developed in Indonesia. You may later change the parameters to any ENUM provider that is suitable to your needs or even develop your own ENUM server, as running one requires only a DNS server.

Entries that needs to be incorporated into /etc/asterisk/enum.conf are:

search => enum.voiprakyat.or.id

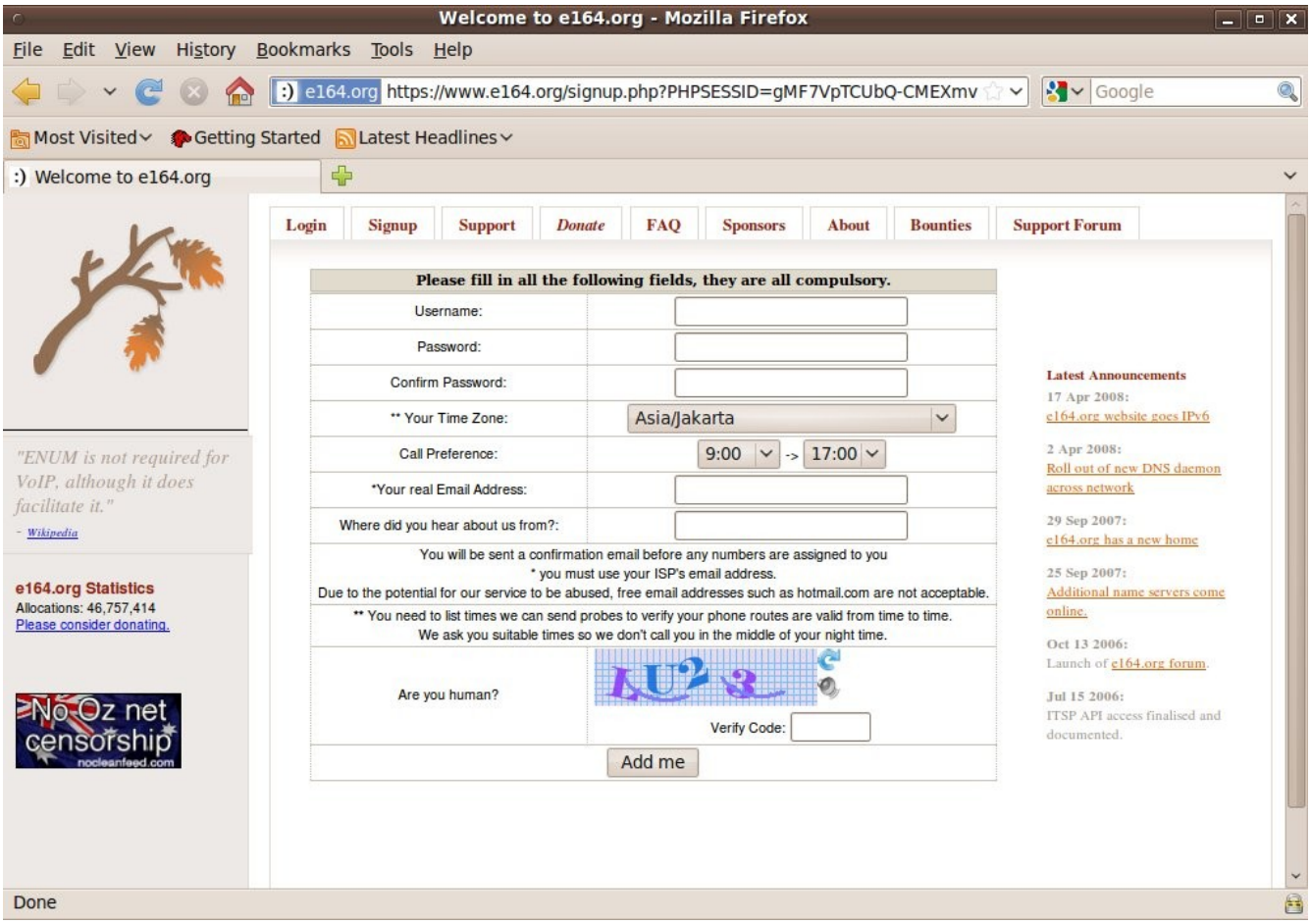


Figure 4.5: Before you can be connected to e164.org, you have to sign up first

Through the registration page of e164.org https://www.e164.org/signup.php, enter the required information in order to obtain a telephone number or register a telephone number. The information you have to enter are username, password, your email address, your time zone and Verifying code. Then click "Add me" to complete the registration. If the registration is successful, you will be able to use your newly-created account to get a telephone number assigned by e164.org or register yours.

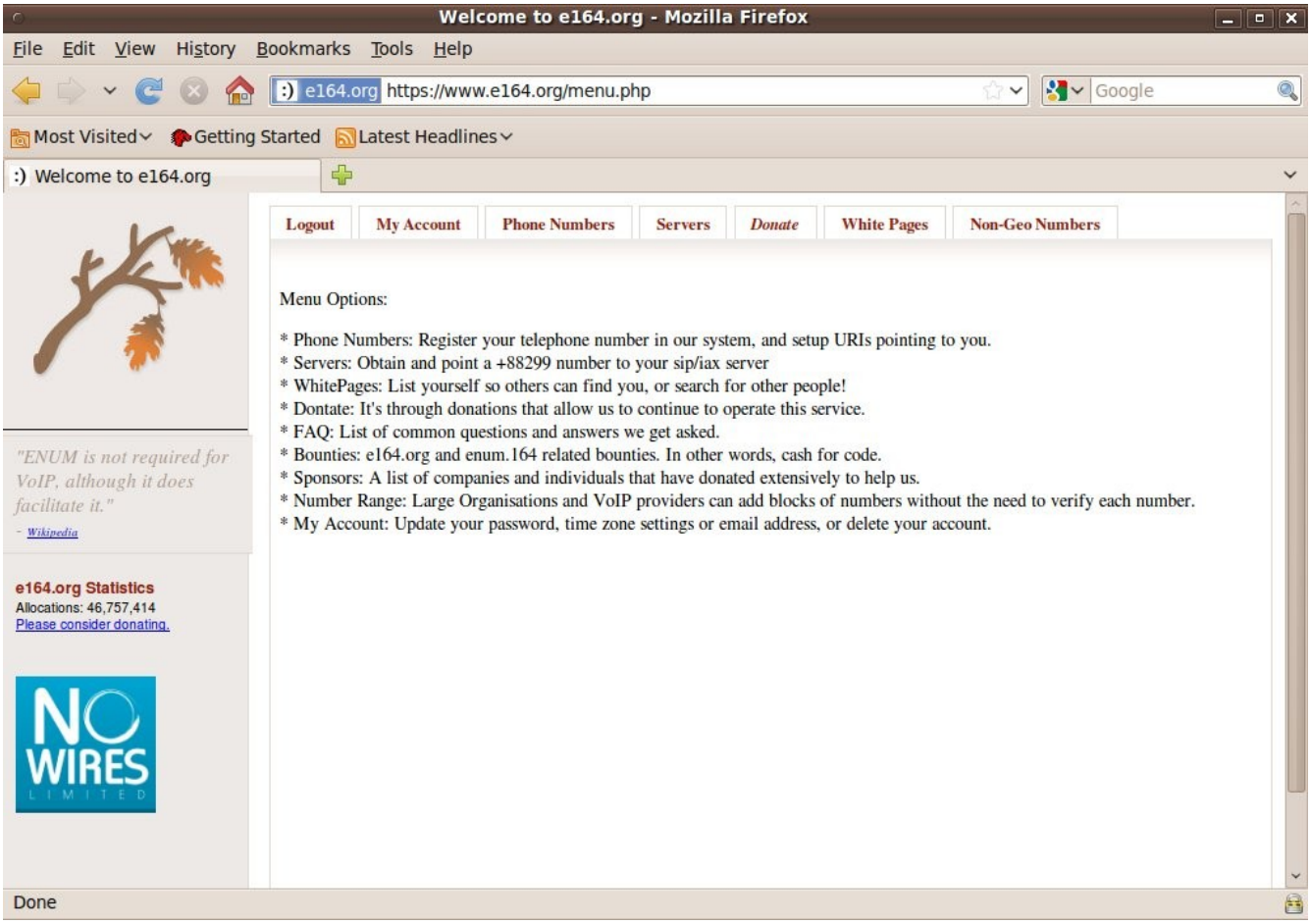


Figure 4.6: The web appearance after you are logged in

To obtain a telephone number or register your number, you need to log on to e164.org. Once you're logged in, there are some options you can choose from.

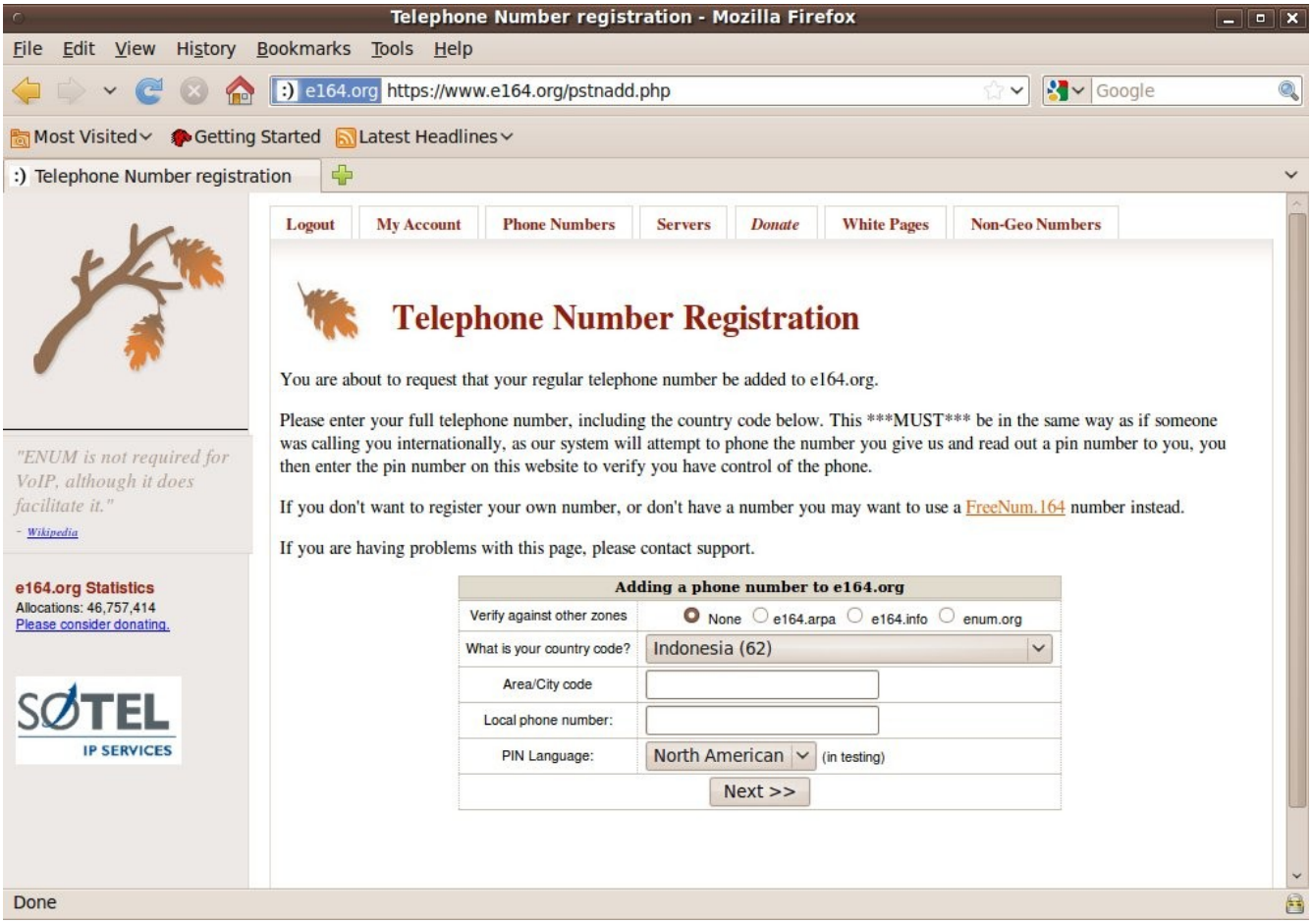


Figure 4.7: PSTN Phone Numbers can be added via https://www.e164.org/pstnadd.php

Access to https://www.e164.org/freenumadd.php will bring you to a default window whereby you can add a virtual phone number to e164.org. You will be assigned an internet telephone number with country code +822 from e164.org.

But if you're interested in adding a real PSTN number, access to https://www.e164.org/pstnadd.php will add PSTN number instead and register the number you use in your country. When registering, you need to have the PSTN number active as e164 will dial the number to authenticate that it is real. Once you received the activation code, go back to e164.org website to activate the number you have just

registered.

To register a PSTN number, you need to enter information such as country, area code, telephone number, and SIP account that will be called when someone places a call through VoIP network using the PSTN number. So the VoIP network will not reach your real PSTN number, but your SIP account using this PSTN number. Your SIP phone will ring, but not your PSTN phone.

Once all information are entered correctly, click Add me to register our PSTN number so it can be called through internet telephony network.

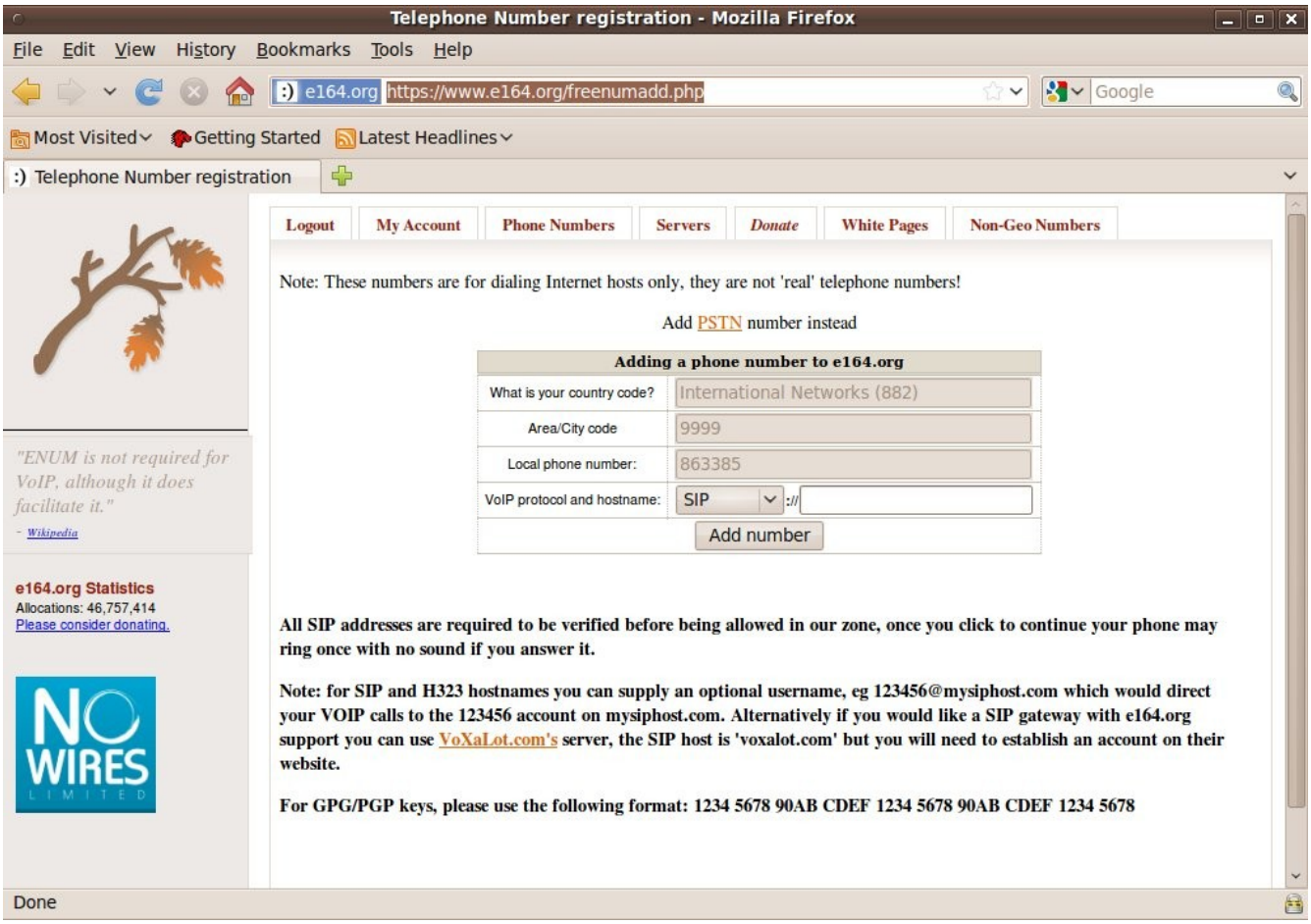


Figure 4.8: You can obtain +822 number assigned by e164.org via https://www.e164.org/freenumadd.php

The second option is much easier to do: simply request a VoIP number with country code +882 via https://www.e164.org/freenumadd.php. This number cannot be reached by PSTN numbers but will be

reached only through VoIP network. To obtain a +822 number, enter your SIP, IAX2 or H.323 number into the blanks. Since the account you created in VoIP Rakyat is of SIP, choose SIP in the drop-down menu. Once all information are entered properly, click Add number in order to obtain the country code +882.

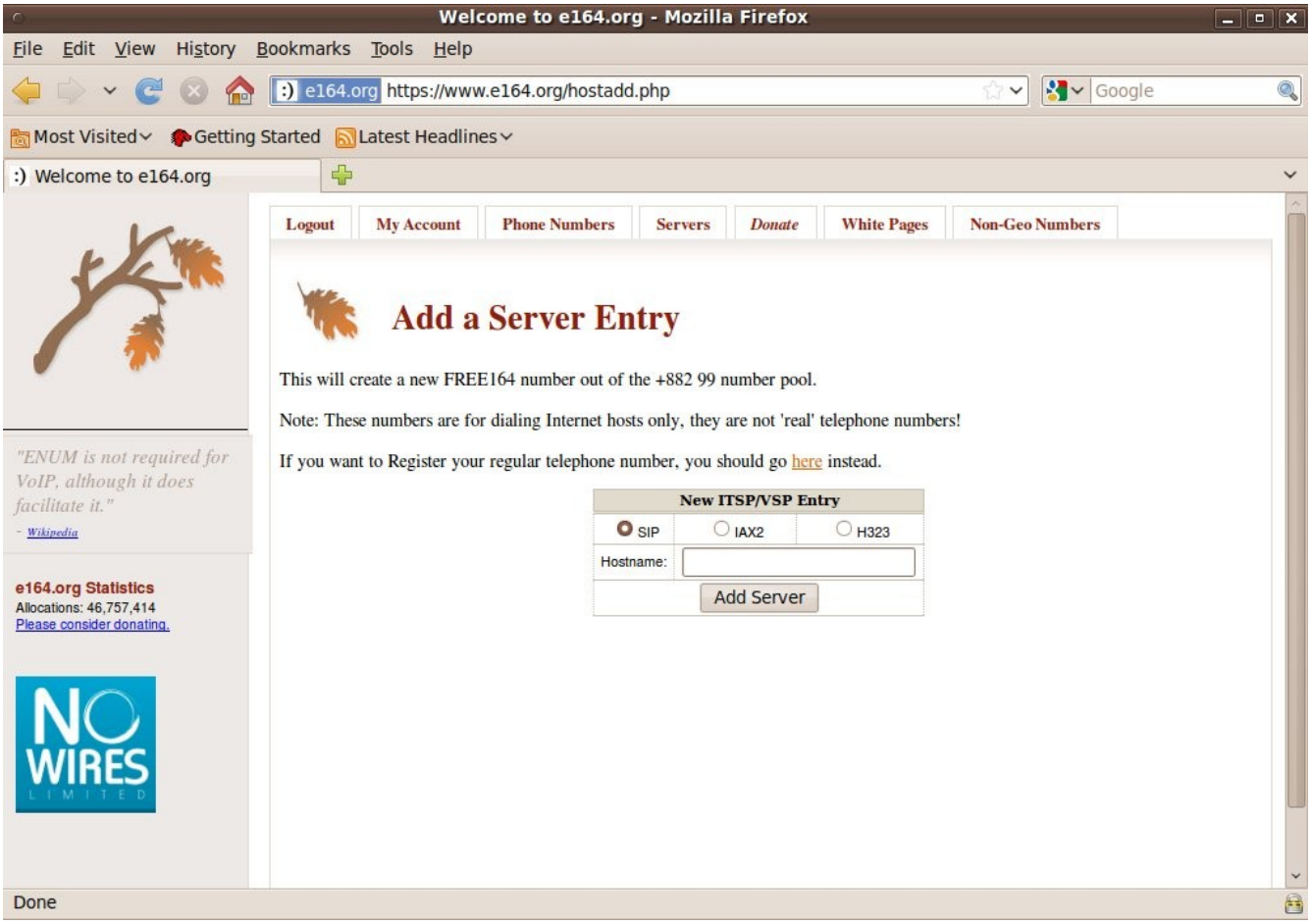


Figure 4.9: Request a block of number via https://www.e164.org/hostadd.php

The most interesting part of e164.org is its ability to obtain a block of numbers via https://www.e164.org/hostadd.php with area code +822 99, instead of having these numbers included one by one. To do this, click Server Add. Via add a Server Entry, choose the type of protocol used by the server and enter the name of the server. The server should have a Public IP address, not the one

used internally. Once all information are entered properly, click Add Server. This will make your SIP server be recognized by e164.org, with +882 being the country code assigned to the server. This also implies that you will have a bunch of numbers that you can further allocate to the users who are registered with your server.

Introducing your country code to International VoIP network

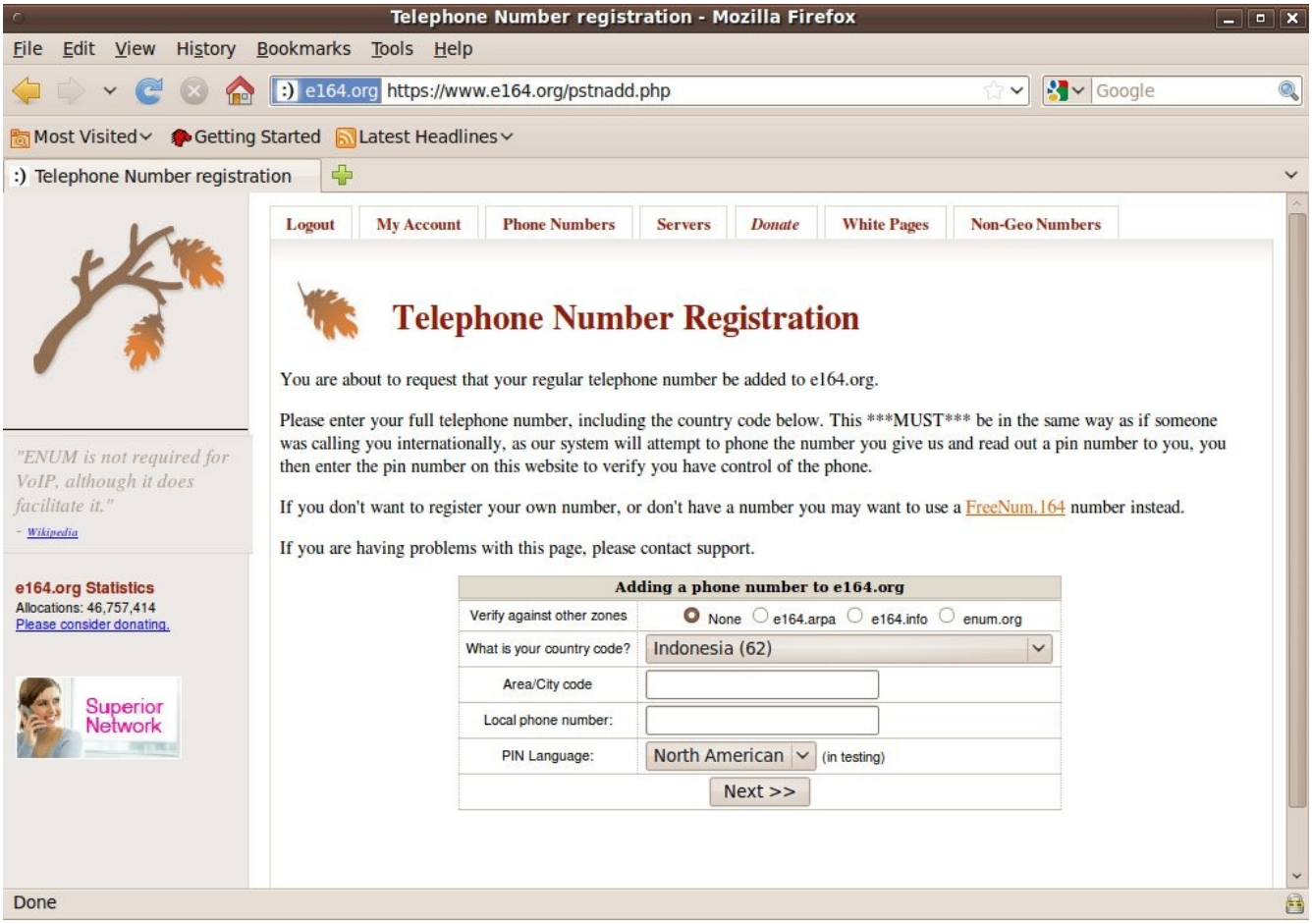


Figure 4.10: Add a telephone number via https://www.e164.org/pstnadd.php

If you want to introduce a PSTN or cellular number with a specific country code to this VoIP network, you can do so through menu available at http://www.e164.org/pstnadd.php. What you have to enter is the country of the PSTN or cellular number, area code, local telephone number, and the SIP account

registered with a SIP provider where the PSTN numbers are those of SIP.

When registering the phone number, you need to have the phone ready to receive calls, as within 15 minutes after you registered it, e164.org will dial your number to provide you with a Personal Identification Number (PIN) required to activate the account. Write them down somewhere so you don't have to memorize them. Go back to the Web and activate your account using the pin that has just been given to you. Once this is completed, your PSTN (or cellular) number can be recognized in the VoIP network, with all the numbers registered with the network capable of dialing your SIP account using your PSTN numbers.

VoIP Rakyat's ENUM

Besides e164.org, there is http://enum.voiprakyat.or.id, a mapping system developed by VoIP Rakyat.

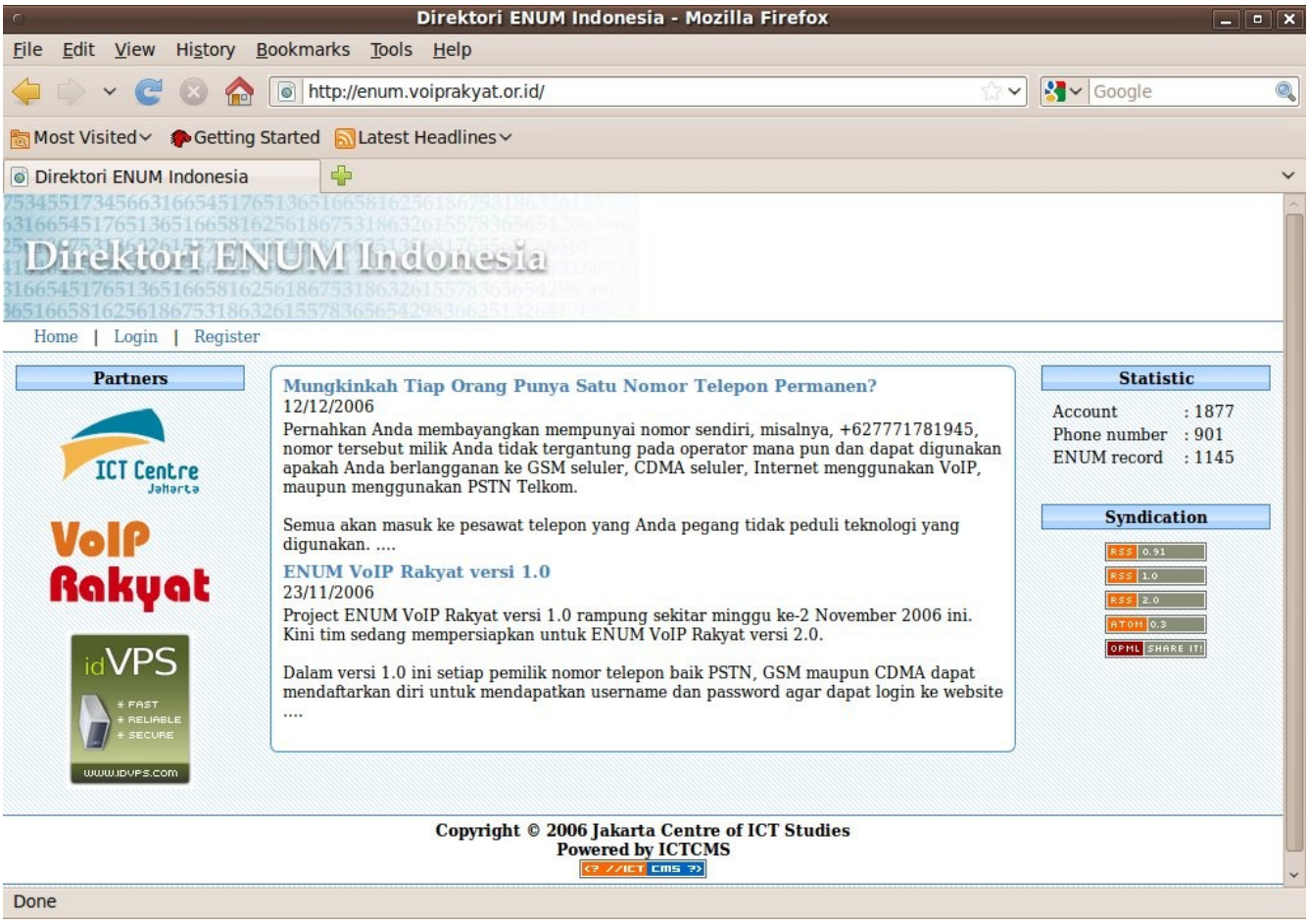


Figure 4.11: Indonesia's Enum directory developed by VoIP Rakyat

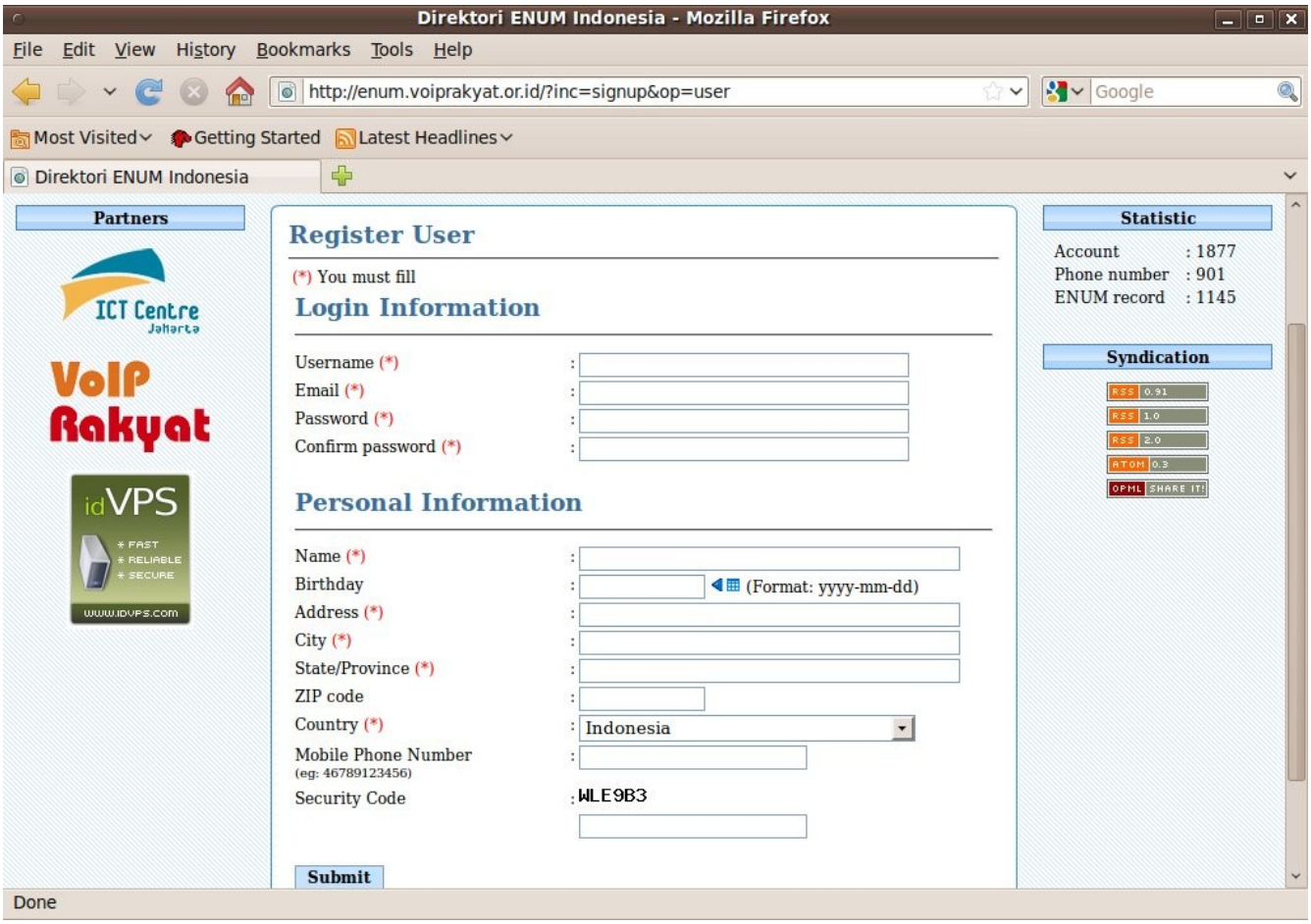


Figure 4.12: The sign-up page of VoIP Rakyat ENUM

Through VoIP Rakyat (VR) ENUM registration page, you can register yourself as a member. The information you need to fill in is username, email address, and password.

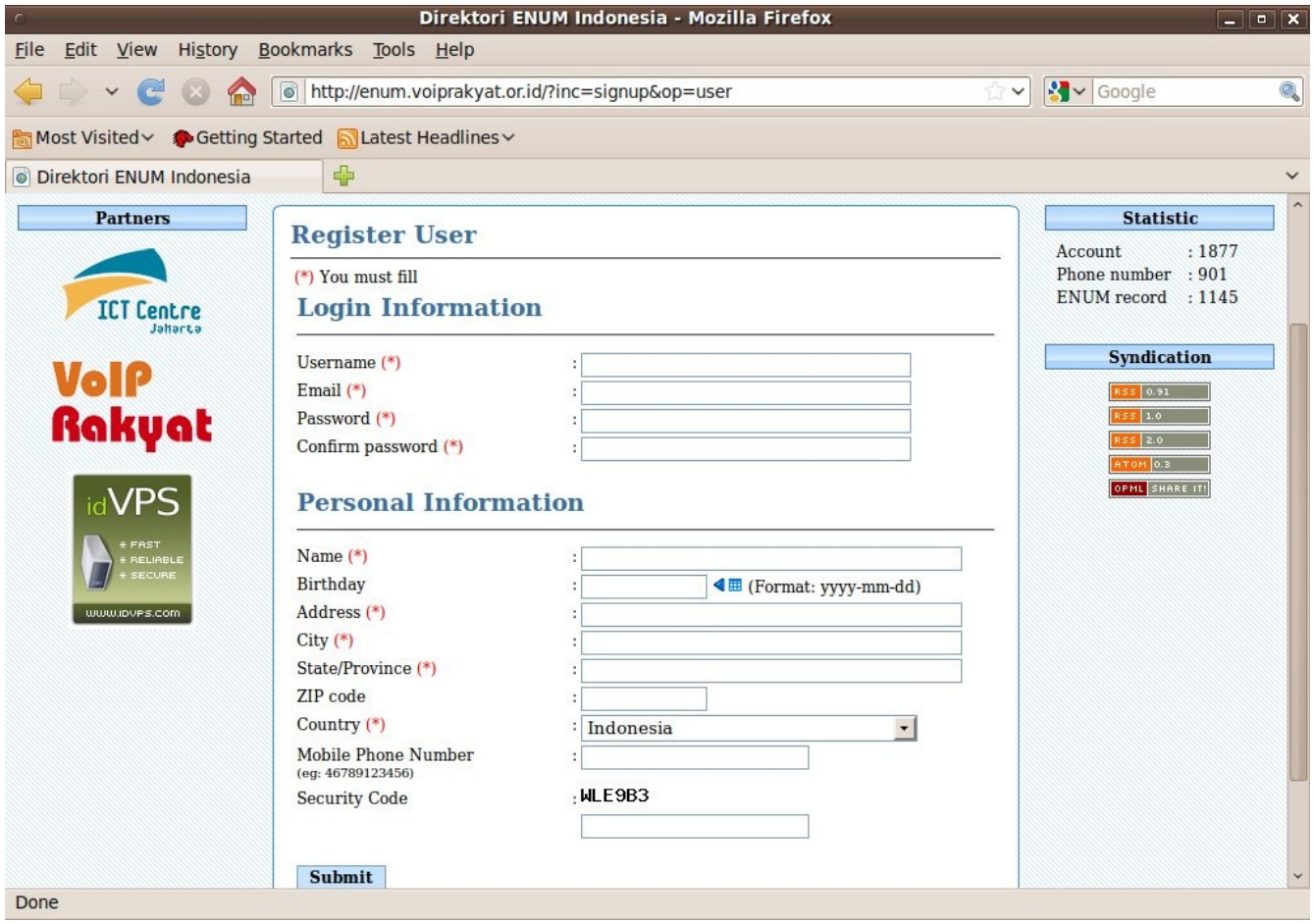


Figure 4.13: ENUM VoIP Rakyat sign-up page

Scroll the page down. Fill in all the information required: Name, Birthday, Address, City, State/Province, country and mobile phone number. For security reason, VR will verify that you are a real person, and not a spamming machine. Use the provided security code to fill in the blanks.

Once all information are entered correctly, click Submit to proceed.

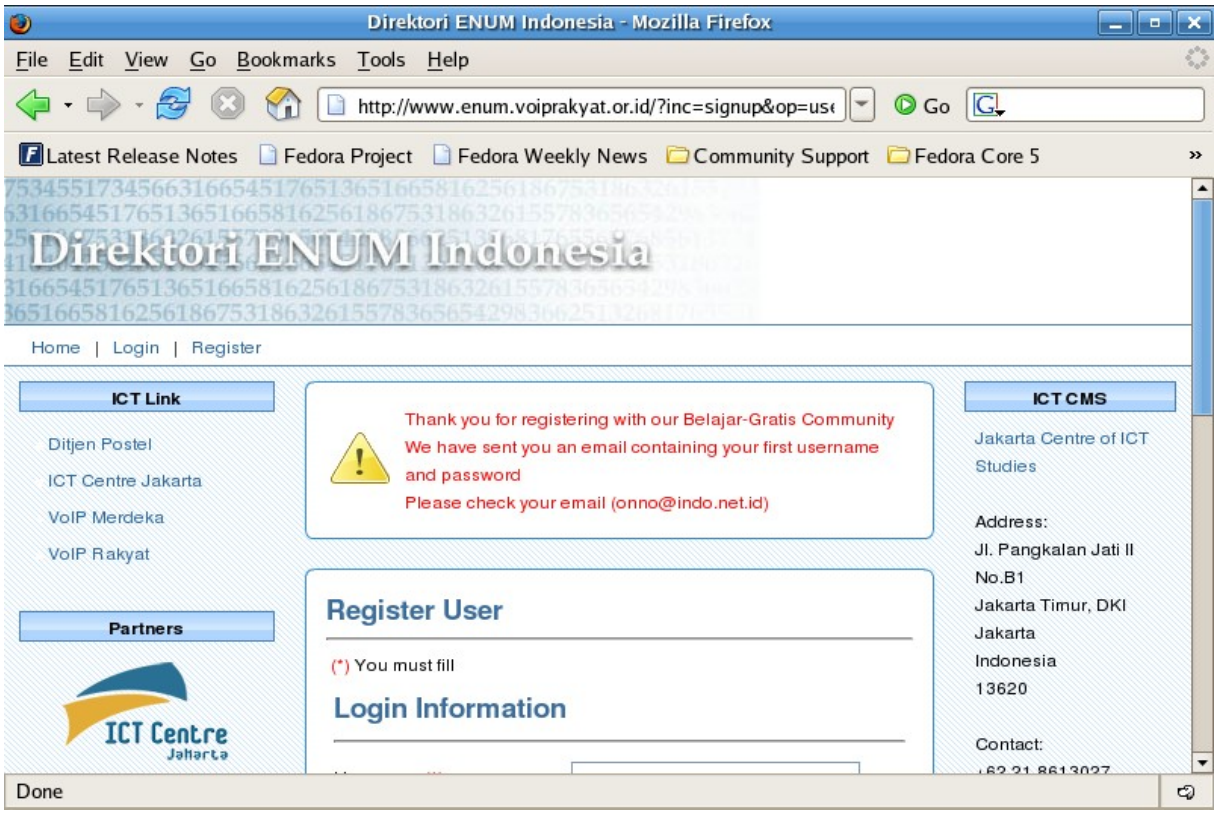


Figure 4.14: The notification informing that your registration is successful

Once the registration is completed, ENUM VoIP Rakyat will send us an email containing the username and password we set when registering to ENUM VoIP Rakyat.

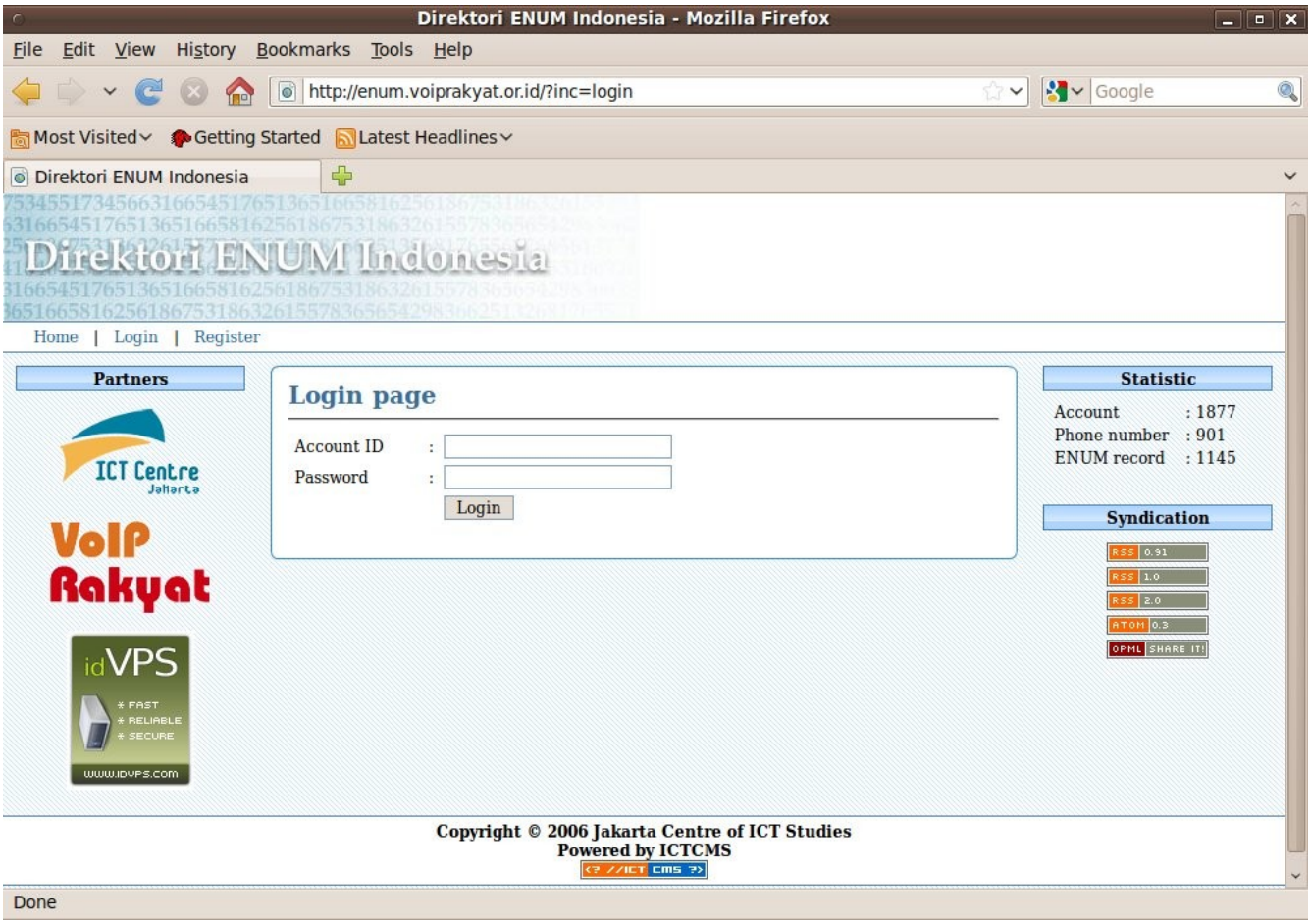


Figure 4.15: In order to access ENUM VoIP Rakyat, you need to enter your username and password

Now that your username has been registered, log on using it and the password provided. Click login to proceed.

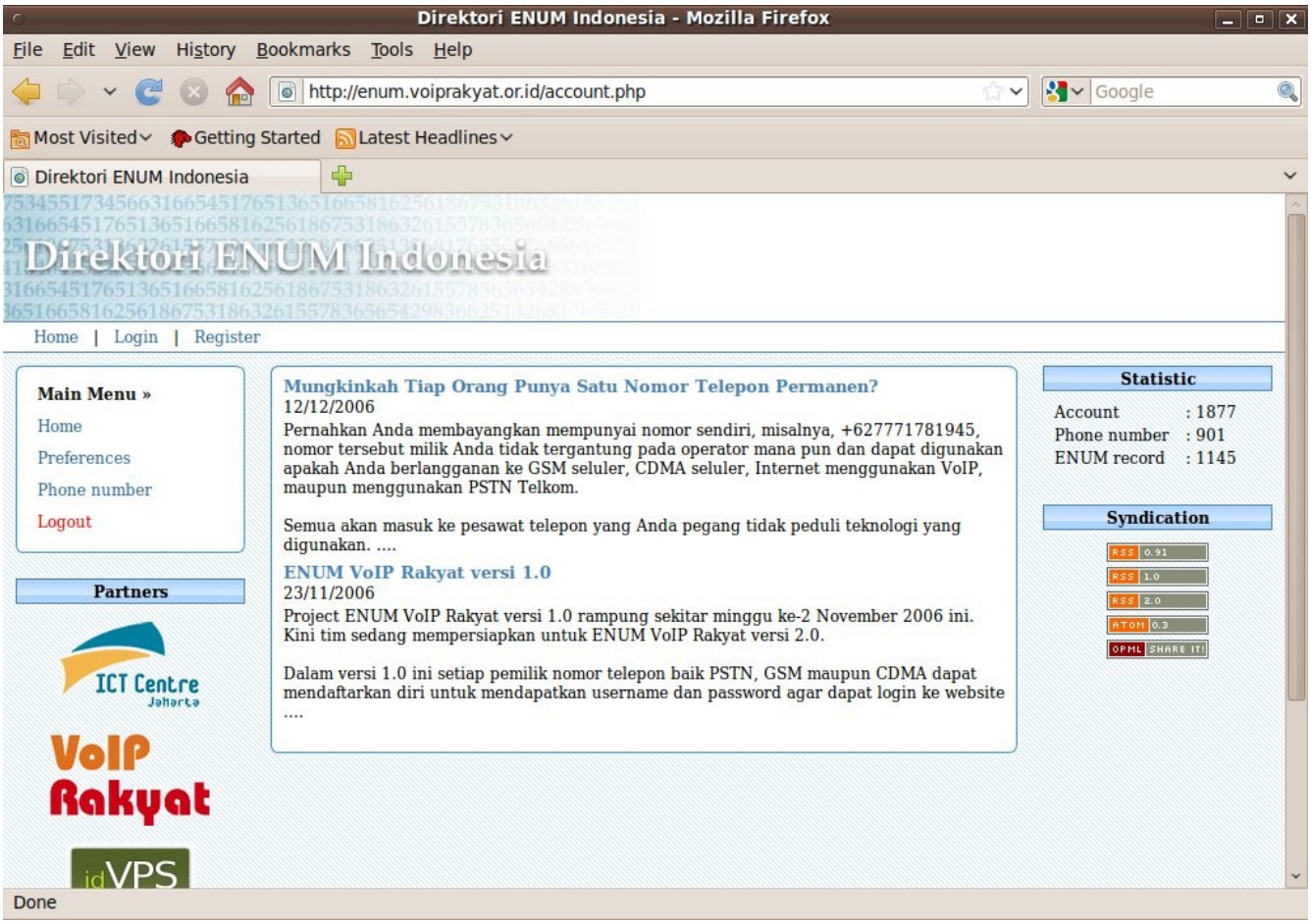


Figure 4.16: ENUM VoIP Rakyat main Window after you logged in

In ENUM VoIP Rakyat, on the left of the page, there are some useful options you can choose from: Preferences and Phone Number. First, click Preferences.

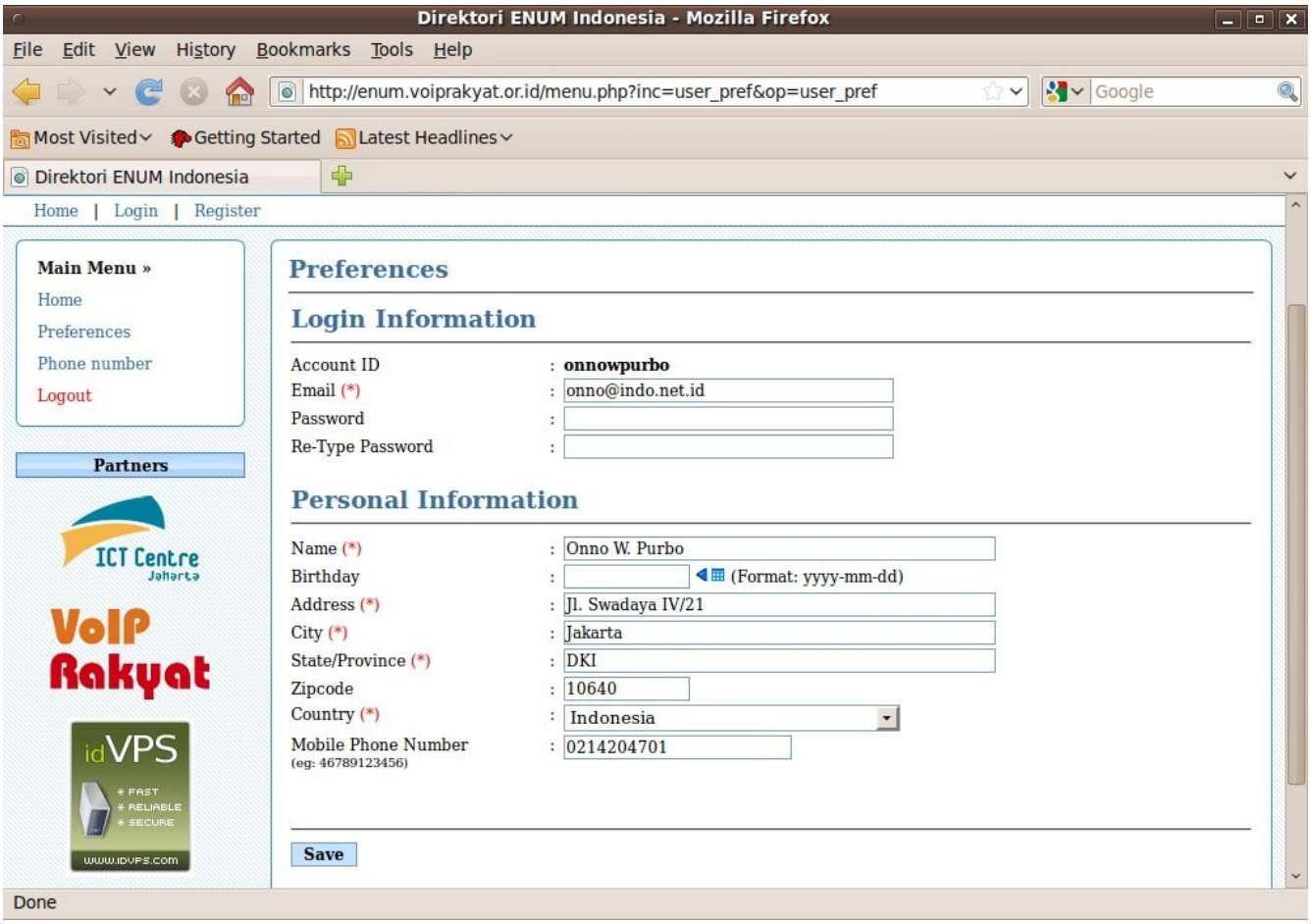


Figure 4.17: By clicking on Preferences, you can edit your login and personal information

With the Preferences option clicked, you can check the information you entered earlier when you did the registration, and make necessary changes.

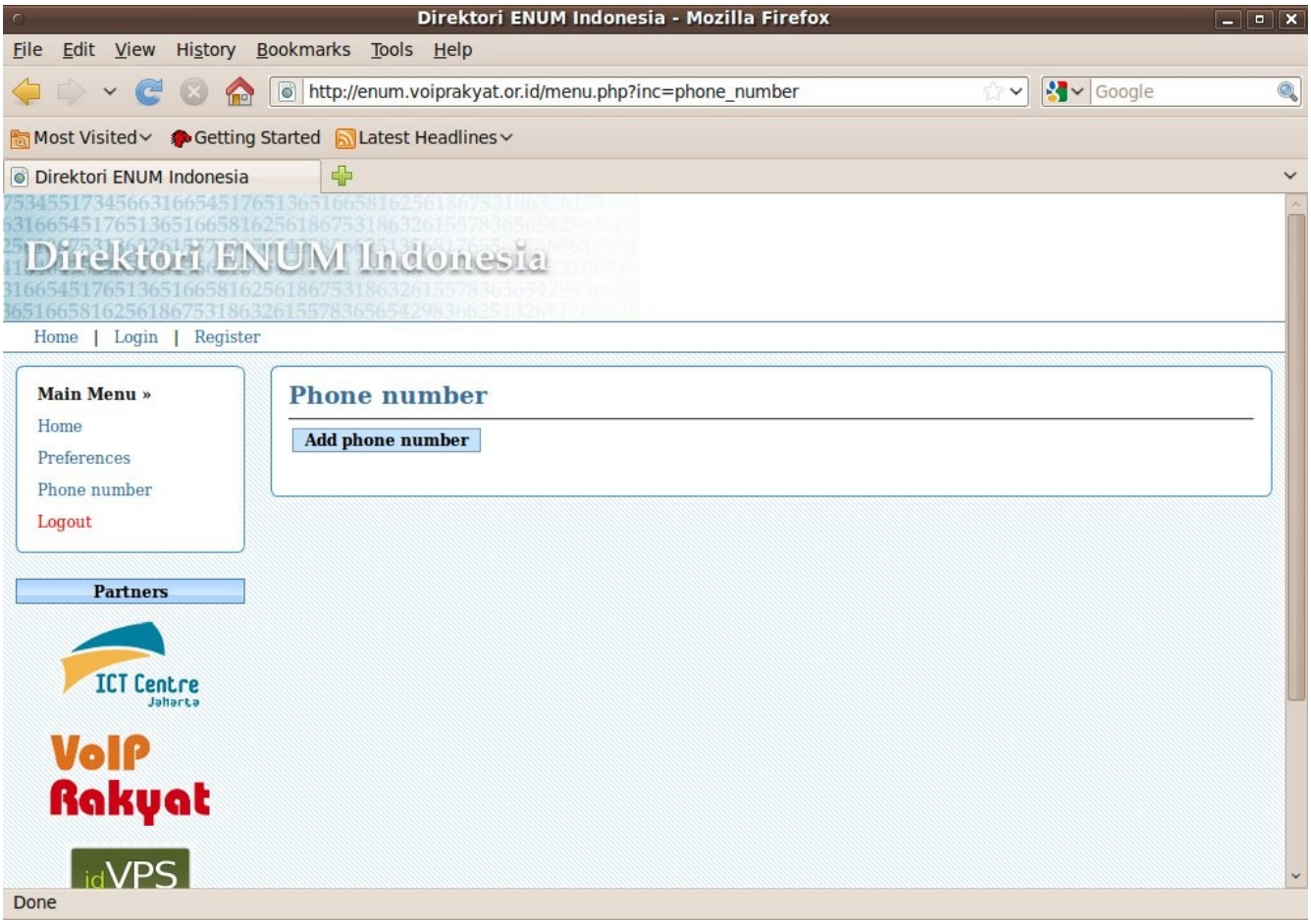


Figure 4.18: By clicking on Phone number, you can add your phone number

Click Phone number. Click Add phone number.

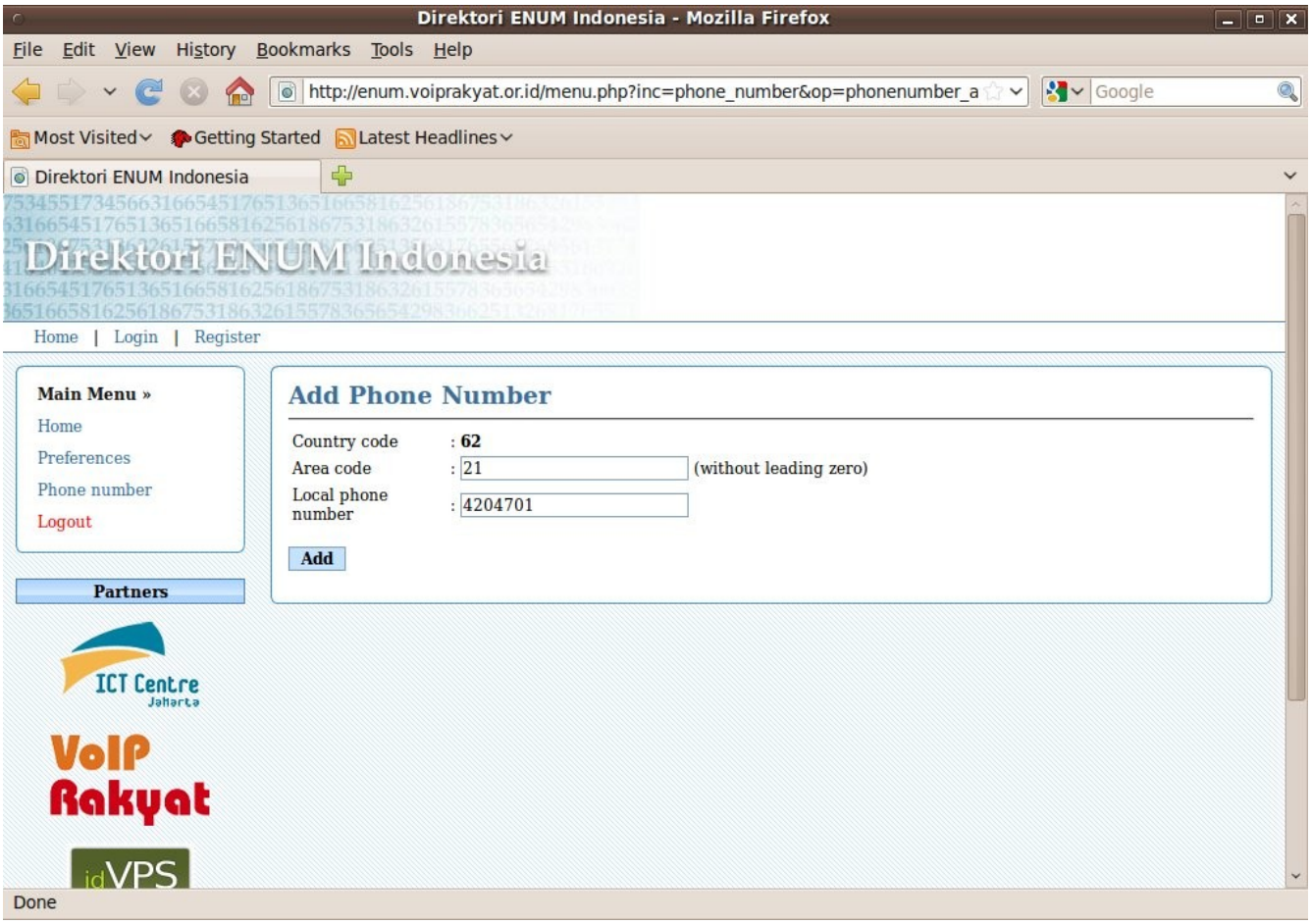


Figure 4.19: By clicking on Add Phone number, you will be able to register your phone

The information you need to enter is country code, area code and local number. Once these information are included, click Add so that the number will be added to VoIP Rakyat ENUM domain.

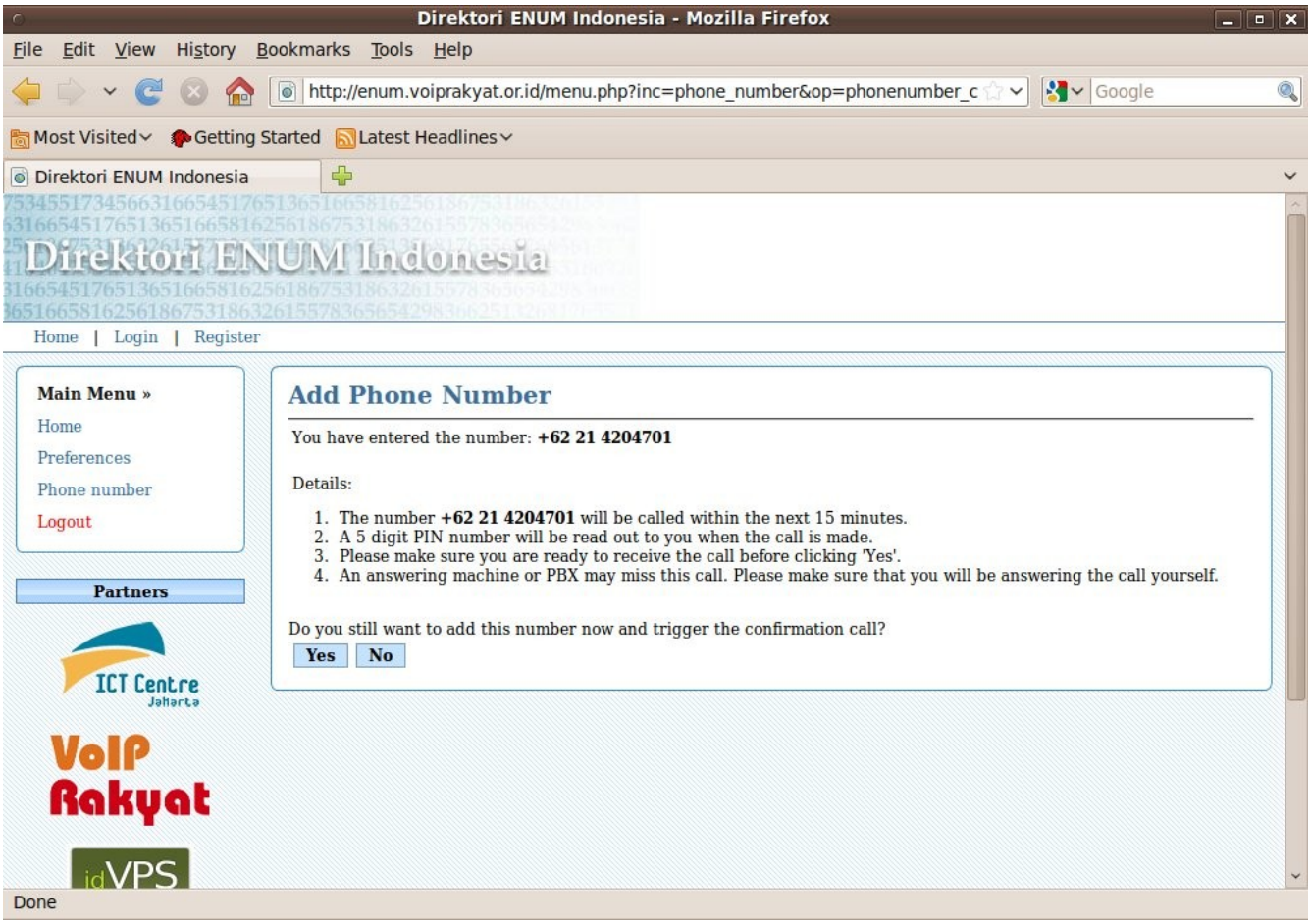


Figure 4.20: Before a number is added, ENUM VoIP Rakyat will confirm whether you really want to add the number

Just like e164.org, ENUM VoIP Rakyat is also designed to validate the number being registered. It will call your number and tell you the code required to authenticate the number. For this to happen, it is important that the number you provided earlier, when you did your registration, is neither of Fax machine nor of PABX. Otherwise you will not be able obtain the code given by VoIP Rakyat.

Connecting to PSTN and Cellular Using VoIP Discount

An alternative to registering your number to enum services such as e164.org or voiprakyat is to use a service called VoIP Discount (http://www.voipdiscount.com).

Figure 4.21: With VoIP Discount, you can make free or inexpensive calls over the Internet

By buying a certain amount of credit, we can obtain a telephone number that can be reached by PSTN telephone using the numbers of other countries such as Czech Republic, French, German, Holland, Swiss and England. With this credit, you will be able to make relatively inexpensive calls to PSTN or cellular. The rates vary, depending on where you are and the countries from which the number you're attempting to call originates. For the rates, go to http://www.voipdiscount.com/en/rates.html

To use VoIPDiscount, you need to:

- make sure that your computer meets the requirements for using them
- obtain VoIPDiscount software in http://www.voipdiscount.com/getfrommirror.php?file=voipdiscount&lang=en
- Install the software in your PC
- enter username and password if you use VoIPDiscount for the first time.

Once all these steps are completed, you will be able to dial any number the way you dial using your PSTN number, with country code, area code and telephone number.

If you use SIP IP Phone or ATA, you need to do the following configuration:

SIP port : 5060
Registrar : sip.voipdiscount.com
Proxy server : sip.voipdiscount.com
Outbound proxy server : leave empty
Account name : your VoipDiscount username
Password : your VoipDiscount password
Display name/number : your VoipDiscount username or voipnumber
Stunserver (option) : stun.voipdiscount.com

VoIP Cheap

Similar to VoIPDiscount, VoIP Cheap (http://www.voipcheap.com/en/index.html) also provides free or relatively inexpensive calls over the internet. The steps to use it is somewhat similar to VoIPDiscount, except that you need to download the software from http://www.voipcheap.com/getfrommirror.php?file=voipcheapCOM&lang=en. For VoIP Cheap calling rate, go to http://www.voipcheap.com/en/rates.html

Figure 4.22: With VoIP Cheap, you can make free or inexpensive calls over the Internet

If you use SIP IP Phone or ATA, you need to do the following configuration:

SIP port : 5060

Registrar : sip.VoipCheap.com
Proxy server : sip.VoipCheap.com
Outbound proxy server : leave empty
Account name : your VoipCheap username
Password : your VoipCheap password
Display name/number : your VoipCheap username or voipnumber
Stunserver (option) : stun.VoipCheap.com

In addition to providing free or inexpensive call service, VoIP Cheap, unlike VoIP Discount, also provides inexpensive SMS service, which is available at http://www.voipcheap.com/en/sms.html

CHAPTER 5: Asterisk Softswitch

One of the best IP PBX Open Source SIP Proxy software available in the internet is Asterisk, which has so many features that allows us to establish our own VoIP infrastructure. Some of these features are highly useful for telecommunication operators, making Asterisk suitable for many applications ranging from small-to-medium-scale IP PBX to IP PBX having hundreds of telephone extensions. However we will not list these features right now, as you will find what these features are as you read along the rest of the CHAPTER, in the syntax code we explain later.

However, for high performance softswitch, it seems OpenSIPS seems to be better in handling high traffic request.

Such Asterisk scalability is possible because of Asterisk function called Trunking, which integrates various VoIP equipments, protocols, cellular, PSTN and even SIP provider to a same network. The number of trunks, theoretically, depends on the amount of available bandwidth and the speed of the processing machine that runs Asterisk.

So what do we need so that we can use Asterisk? The answer to this question depends on how many telephone extensions you want to have in your system and, importantly, how many concurrent calls Asterisk can facilitate. Ideally, the Specifically, you have to understand the following parameters:

- The number of outbound connections and their type (Analog, ISDM, T1, VoIP).
- The number of internal and external concurrent calls (the ratio between calls).
- The type of phone that will be used (Analog, SIP, H.323, MGCP).
- The type of codec that will be used.
- Whether transcoding process will be necessary.
- How reliable the system is.
- How many Asterisk machine that will be placed.
- The condition of your computer network in terms of processing speed, Quality of Service (QoS), VLAN, and Power over Ethernet.

In general, a faster processor and the bigger the RAM, the more concurrent calls the server can facilitate. Since Asterisk seems to theoretically require around 30 MHz of CPU resources for every active channel, a 266-MHz CPU, for example, should ideally be able to facilitate about 8 concurrent calls, with the assumption that the Codec being used is G.711. Of course, in order to become an operator, you need to have a much more sophisticated server with faster CPU and higher RAM. But in order to understand what you really need, you can look into a variety of examples of hardware

configurations and their maximal capability, which are available at http://www.voip-info.org. Through this site, you will also find the scripts required to simulate a call and put some load on the system.

Based on these considerations, you will know how much money you really need to spend. Spend sometime browsing the internet to make some comparison on internet telephony equipments and how much they cost. However, manufacturers, normally, do not show the price of the items they sell in their site. These price tags are usually shown in sites selling internet telephony equipments, some of them are:

Digiumcards - http://www.digiumcards.com/
VoIP on solutions - http://www.voipon.co.uk/
The VoIP Connection - http://www.thevoipconnection.com/

The prices may vary, ranging from US$15 to US$50 per FXO or FXS. Meanwhile, IP Phone each cost between US$50 to US$150. You will of course get for less when you purchase them in large quantities. The cheapest you can get are the equipments produced in Taiwan or China. Some of them are LevelOne and Nexus.

Minimal Resource for Asterisk

Before you decide to invest in sophisticated equipments, it is beneficial to learn on how to install and use Asterisk, using a simple PC with Linux operating system as a server with Internet and LAN connection. This section will focus on minimal installation of Asterisk. Once installed, Asterisk will turn your PC into a simple softswitch.

This type of installation allows you to run Asterisk as simple as possible, at the expense of you being able only to run the following functions:

- User authentication with a phone number and password.
- Dial plan to manage what needs to be done for a call dialed to a specific number
- ENUM, so Asterisk will recognize numbers with specific country code. For example, in Indonesia, the country code in example would be +62XXX in the Asterisk configuration.

Asterisk Installation

Assuming, the Ubuntu repository at /etc/apt/sources.list has been correctly set. One can easily install Asterisk using command

```
# apt-get install asterisk
```

For a more complete command, you may use the following command.

```
# apt-get install asterisk asterisk-dev asterisk-config asterisk-sounds-main \
asterisk-sounds-extra dahdi gastman asterisk-mysql dahdi-firmware-nonfree \
asterisk-mp3
```

Ubuntu will start download and install asterisk as soon as the command invoked.

Compile Asterisk

For those who wish to compile asterisk softswitch from source code, we can do it through the followings,

Prepare the following applications

```
# apt-get install kernel-package libncurses5-dev fakeroot wget \
bzip2 g++ libssl-dev libxml2-dev doxygen
```

We can download most of the source code from Asterisk site, such as,

- http://www.asterisk.org
- http://downloads.asterisk.org/pub/telephony/dahdi-linux-complete/releases/dahdi-linux-complete-2.4.0+2.4.0.tar.gz
- http://downloads.asterisk.org/pub/telephony/libpri/releases/libpri-1.4.11.4.tar.gz
- http://downloads.asterisk.org/pub/telephony/libss7/releases/libss7-1.0.2.tar.gz
- http://downloads.digium.com/pub/asterisk/releases/

While mpg123 application can be downloaded from

- http://www.mpg123.de/download.shtml
- http://sourceforge.net/project/showfiles.php?group_id=135704
- http://sourceforge.net/projects/mpg123/files/

Copy all lastest source code to /usr/local/src/

```
cp asterisk-1.8.0.tar.gz  /usr/local/src/
cp libpri-1.4.11.4.tar.gz /usr/local/src
cp dahdi-linux-complete-2.4.0+2.4.0.tar.gz /usr/local/src/
cp libss7-1.0.2.tar.gz /usr/local/src/
```

```
cp mpg123-1.12.5.tar.bz2 /usr/local/src/
```

Open the source code

```
cd /usr/local/src
tar zxvf asterisk-1.8.0.tar.gz
tar zxvf libpri-1.4.11.4.tar.gz
tar zxvf asterisk-sounds-1.2.1.tar.gz
tar jxvf mpg123-1.12.5.tar.bz2
tar zxvf dahdi-linux-complete-2.4.0+2.4.0.tar.gz
tar zxvf libss7-1.0.2.tar.gz
```

Compile MPG123

```
cd /usr/local/src/mpg123-1.12.5/
./configure
make
make install
```

Compile Libpri

```
cd /usr/local/src/libpri-1.4.11.4/
make all
make install
```

Compile DAHDI. Make sure we have an Internet connection as we need to download the firmware during dahdi installation process.

```
cd /usr/local/src/dahdi-linux-complete-2.4.0+2.4.0/
make
make install
make config
```

Compile LibSS7. Do this after dahdi; before compiling asterisk.

```
cd /usr/local/src/libss7-1.0.2/
make
make install
```

Compile asterisk. Make sure we have Internet connection as we need to download the operation voice during asterisk installation process.

```
cd /usr/local/src/asterisk-1.8.0
./configure
make menuselect
make all
make
make install
make samples
```

Please note that “make menuselect” is optional, we can do the compalition process without “make menuselect”. If you like to install the documentation, please do

```
apt-get install doxygen
make progdocs
```

Configuring Asterisk

As the Asterisk installed, we need to configure it so Asterisk functions the way you want it to be. All files that you need to configure are stored in the folder:

```
/etc/asterisk
```

The minimal configuration files need to edited are:

```
sip.conf - for user authentication with a phone number and password.
extensions.conf - to set the dialplan.
enum.conf – for ENUM, for example, for country code +62.
```

Aside from these files, there are more configuration files for those who are seriously interested to study the asterisk. For now, it is sufficient for you to learn configuring those three files.

ENUM.CONF Configuration

There is not much to be changed in ENUM.CONF. However, you need to make sure that there are the following entries:

```
search => e164.arpa
search => e164.org
search => e164.id
search => enum.voiprakyat.or.id
```

This way, we can ensure that the information contained in ENUM e164.arpa, e164.org and e164.id will be recognized by Asterisk.

SIP.CONF Configuration

The user database is stored in /etc/asterisk/sip.conf. An example for an account with phone number 2099, password 123456, dynamic IP address using DHCP is as follows:

```
[2099]
context=default
type=friend
username=2099
secret=123456
host=dynamic
dtmfmode=rfc2833
mailbox=2099@default
```

To ensure that the dial tone is handled properly in Asterisk 1.6, we may add the following entry:

```
rfc2833compensate=yes
```

Enter the above entry for each user. At this point, each user may register his- or herself to the Asterisk. The registered users may call each other on the same Asterisk server.

To connect our Asterisk server to VoIP Rakyat or any other SIP proxy available in the internet, we need to register our Asterisk to the SIP proxy server. The commands used is:

```
register => 2345:password@sip_proxy/1234
```

which means user 1234 in our asterisk server that we operate is the user 2345 in sip_proxy logged in to the server using the password “password”. For example, user 2000 has an account 20345 in voiprakyat.or.id server with password “secret”, then the format used is:

```
register => 20345:secret@voiprakyat.or,id/2000
```

This way, calls made to VoIP Rakyat, specifically to account 20345, will be forwarded to number 2000 in our SIP server.

EXTENSIONS.CONF Configuration

The dial plan or routing table of a softswitch is normally stored in /etc/asterisk/extensions.conf. In extensions.conf we can configure what Asterisk needs to do as it receives a call on a certain extension. The simplest example of dial plan is:

```
exten => _20XX,1,Dial(SIP/${EXTEN},20,rt)
exten => _20XX,2,HangUp
```

which means that if there is someone who calls extension 20XX, then the first step carried out by the syntax is to have DIAL of the extension use SIP technology, wait for 20 seconds and if there is no response, carry out time-out (rt). The second step is to hang up. Of course you need to do a small configuration of the command so it will fit your circumstance in how you use your SIP server.

Some commands considered dangerous but often sought by user/admin are as follows:

```
exten => _0711.,1,Dial(SIP/${EXTEN:4}@2031,20.rt)
```

which means that there is someone who calls 0711. The dot "." implies that any number after 0711 is ignored. DIAL uses SIP technology to connect to 2031. Also note carefully the code {EXTEN:4} has to be read "omit the first 4 digits of the dialed number." For example: 07115551234 becomes 5551234.

If we use PABX between ATA and PSTN, the command used is as the following:

```
exten => _021X.,1,Dial(SIP/9${EXTEN:3}@2031,20.rt)
```

The syntax above implies that there is someone who calls 021X. Notice that the dot "." placed after X implies that any number placed after X is ignored. DIAL uses SIP technology to connect to 2031. Also note carefully the code 9{EXTEN:3} has to be read "omit the first 3 digits of the dialed number" and "add the prefix 9 in front of the number." For example: 0215551234 becomes 95551234

This means that if the number 2031 originates from an Analog Telephone Adapter (ATA) such as the SPA3000 located in the Jakarta and is connected to a PABX in Jakarta, anyone in such a VoIP network

will be able to call Jakarta without having to pay long distance or international call. What they need to pay is just the local rate for calling the intended number in Jakarta city.

The same way can be developed for calling mobile phone in Indonesia by connecting the ATA we use to PSTN or any Fixed Wireless Terminal (FWT) device. The command used is as follows

```
exten => _08X.,1,Dial(SIP/${EXTEN}@2031,20.rt)
```

Of course, an office that is connected to a public VoIP network will not open its access so that only certain users can call any mobile number or Telkom, and thus we usually do not use 021X. code, nor 08X. But we will enter each of the numbers allowed to be called through VoIP. For example:

```
exten => _0811567854,1,Dial(SIP/${EXTEN}@2031,20.rt)
exten => _0216575675,1,Dial(SIP/${EXTEN}@2031,20.rt)
exten => _0216755675,1,Dial(SIP/${EXTEN}@2031,20.rt)
```

This means that only number 0811567854, 0216575675 and 0216755675 can be contacted via VoIP numbers. Other than these numbers cannot be contacted.

To adopt the phone number format similar to Telco, e.g., +62 XXX or other numbers we may include ENUMLOOKUP command, for example,

```
exten => _00.,1,Set(enumresult=${ENUMLOOKUP(+${EXTEN:2},,,,e164.id)})
exten => _00.,n,Dial(SIP/${enumresult})

exten => _+.,1,Set(enumresult=${ENUMLOOKUP(${EXTEN},,,,e164.id)})
exten => _+.,n,Dial(SIP/${enumresult})
```

In an environment where there are many asterisk / SIP servers, sometimes we need to create an “area code” to be able to call to each other among these servers. For examples,

“Area Code”	SIP Server IP Address
“021”	203.159.31.99
“022”	203.159.31.123
“023”	203.159.31.48

The dialplan would be

```
exten => _021.,1,Dial(SIP/${EXTEN:3}@203.159.31.99,30,rt)
exten => _022.,1,Dial(SIP/${EXTEN:3}@203.159.31.123,30,rt)
exten => _023.,1,Dial(SIP/${EXTEN:3}@203.159.31.48,30,rt)
```

Notice ${EXTEN:3} that will remove the three (3) digits “Area Code” as we pass the extensions number to the destination SIP Server.

CHAPTER 6: Asterisk for Incoming and Outgoing calls

Defining SIP Channel in sip.conf

Every SIP client and server is identified with a text block such as the following:

```
[xxx]
type=yyy
parameter1=value
parameter2=value
```

Where “xxx” is the name associated with the SIP client, or any arbitrary name used by other configuration file to refer to a SIP device. Typically, a SIP phone with extension number 123 will have an entry that begins with [123]. Please note that you still have to activate extension 123 in extensions.conf file so that people can call to extension 123. The parameter Type must contain "user", "peer" or "friend".

Asterisk will match the incoming call with the name of the device and type=user of the SIP protocol conversation in column From: user name (without acknowledging the SIP domain). Another way also used is to match the SIP request and [xxx] in sip.conf file, through both the IP address of the requester and the peer information of part of [xxx] in sip.conf file. If host=dynamic, it's not possible to perform matching until the SIP client is registered.

Asterisk as SIP Client

Asterisk can register itself to another SIP server and becomes a client. For this, the command used in sip.conf under [general] for registration to the SIP server is:

```
register => user[:secret[:authuser]]@host[:port][/extension]
```

If you have problems with your computer network, such as an unstable connectivity, frequent connectivity breakdowns, and losing established registration to your SIP server, you can add parameter “registerattempts” and “registertimeout” before the generic definition of register. Setting registerattempts=0 will force Asterisk to keep registering until successful (default value is 10 attempts). The value of registertimeout determines the length of time in seconds between attempts for registering (the default value is 20 seconds).

Example:

register => 2345: password@mysipprovider.com/1234

The above command will register “2345” to mysipprovider.com and will be identified as extension 1234 in Asterisk which we operate. In the example above the parameters used are:

user – the user id for the SIP server (example: 2345)
authuser - user authorization (optional) to the SIP server
secret - the user password
host - server name (example: mysipprovider.com)
port – the SIP port in Server. The default is 5060.
extension - the local extension number in Asterisk (example: 1234).

The extension number is used to contact local extension of the Asterisk SIP server which we signed up. If there is no extension, Asterisk will automatically enter extension "s".

To see if Asterisk has successfully registered itself with the SIP Server, we can use Asterisk Interface Command Line, which can be accessed through the asterisk command “-r” in the shell.

asterisk -r

Registration status can be viewed through the command:

sip show registry

It seems that this command will be omitted in Asterisk version 1.4, and will be changed into

sip registry list

To see the phone/extension listed in Asterisk which we operate, we can use the following command

sip show peers

In Asterisk 1.6, the command seems to be replaced by

sip peers list

To make a call to a SIP server outside of Asterisk, we need to define sip.conf like the following example:

```
[mysipprovider-out]
type=peer
secret=password
username=2345
host=sipserver.mysipprovider.com
fromuser=2345
fromdomain=fwd.pulver.com
nat=yes
context=from-mysipprovider
; is further defined in extensions.conf
```

In extensions.conf, we need to add a command like:

```
exten => _9.,1,Dial(SIP/${EXTEN:1}@mysipprovider-out,30,r)
```

Please note that the variable ${EXTEN:1} here will take all the characters/ letters from the incoming extension except for the first character, which in this case, is the number 9.

Meanwhile, SIP extension configuration - extensions.conf – for receiving calls coming from the SIP server can also be developed using the following command:

```
[from-mysipprovider]
exten => 1234.1, Answer
; 1234 is the extension contact. The default extension contact is "s"
exten => 1234.2,Dial(SIP/111,25,Ttr)
; Incoming calls are redirected to a SIP telephone number 111
exten => 1234.3,Hangup
```

Generic SIP configuration

In [general] section in sip.conf, there are some variables that we can setup, some of which are

```
allow = <codec>
; a Codec that is allowed based on preferences. Prior to using this, use disallow=all.
```

disallow = all
; disallow all codecs to be used.

allowexternalinvites = yes|no
; Enable or Disable INVITE & REFER to non-local domain. The default is yes.

allowguest = yes|no
; Allows or rejects calls from guest (the default is yes).

allguest = yes|no
; Allows or denies the call from guests. The default is yes.

Autocreatepeer = yes|no
; If it is set to yes, everyone can easily log in as a peer without a password,
it is usually beneficial for operating with SER. The default is no.

autodomain = yes|no
; Enable/disable the ability of Asterisk to add local hostnames and
local IP address to domain list. The default is no.

bindaddr = IP_Address
; IP Address bound as a place for listening to connection. The default is 0.0.0.0 (any interface).

bindport = Number
; The UDP port in bind for listening to incoming connections. The default is 5060.

callerid = <string>
; Caller ID information that will be used if there is no other information. The default is asterisk.

canreinvite = update|yes|no
; The default is yes.

checkmwi = Number
; The interval in seconds to check the mailbox. The default is 10 seconds.

compactheaders = yes|no
; whether Asterisk will send a SIP header in compact or complete form. The default is no.

context = <contextname>

; This is the default context that will be used for telephones that do not have context.
The content of the context can be set in extensions.conf.

defaultexpirey = Number
; The default length of time (in seconds) of an incoming or outgoing registration.
The default 120 seconds.

dtmfmode = inband|info|rfc2833 (global setting)
; The default is rfc2833.

domain = domains
; list of domains separated by comma, a list for which Asterisk is responsible.

dumphistory = yes|no
; Enables support for dumping SIP transactions in LOG_DEBUG. The default is no.

externip = IP_Address or hostnames
; The address we will place in the SIP messages if we are behind NAT.
If the hostname is used, then the IP address associated with the hostname will be read once
at the time of reading sip.conf. If we want to use the hostname of the dynamic IP,
use externhost parameters.

externhost = hostname.tld

externrefresh = Number
; determines how often (in seconds) DNS checking is carried out for 'externhost'.
The default is 10 seconds.

ignoreregexpire = yes|no
; sets whether Contact information from a peer is still used even the information has expired.
The default is no.

language = <string>
; The default language used by Playback()/Background().

localnet = NetAddress/Netmask
; Local network and mask.

fromdomain = <domain>

; Set default From: domain in SIP message at the time it operates as a SIP ua (client)

insecure = very|yes|no|invite|port
; Set how to handle connections with peers. The default is no (authenticate all connections).

maxexpirey = Number
: Length of time (in seconds) of incoming registration. The default is 3600 seconds.

musicclass = one of classes that is used in musiconhold.conf

musdiconhold = similar to musicclass

nat=yes|no|never|route
; The default is no, which means that rfc3581 technique is used.

notifymimetype = mediatype/subtype
; Allows to override mime type in MWI NOTIFY used in voicemail online message.
The dafault is application/simple-message-summary.

notifyringing = yes|no
; Call notification is included in ringing stage. The default is yes.

outboundproxy = IP_address / DNS SRV name (excluding _sip._udp prefix)
; SRV name, hostname, or IP address of the outbound SIP Proxy.

outboundproxyport = Number
; UDP port number for Outbound SIP Proxy.

pedantic = yes|no
; enable a slow process to check Call-ID, SIP header with many lines,
and the URI-encoded headers. The default is no.

port = <portno>
; The default port for SIP peer. This port is not the port of Asterisk for listening to
incoming calls (see bindport).

progressinband = never|no|yes
; whether we should generate in-band ringing. The default is never.

promiscredir = yes|no
; Allows support for 302 Redirects; (Note: it will redirect all to local extension available in contact, not to extension on the final destination).
The default is no.

qualify = yes|no|milliseconds
; Check whether the client can be contacted. If set to yes, then the checking will be carried out every 2000 milliseconds (2 seconds).
The default is no.

realm = my realm
; Change authentication realm for the asterisk (default) to what we want.

recordhistory = yes|no.
; Enable logging of SIP transactions.
The default is no.

regcontext = context
; Default context used to respond to the SIP REGISTER of SIP Registrar.

register => <username>:<password>:[authid]@<sip client/peer id in sip.conf>/<contact>
; Register to SIP provider

registerattempts = Number
; the number of SIP REGISTER message sent to the SIP Registrar before giving up.
The default is 0 (no limit).

registertimeout = Number
; The number of seconds allocated to wait for responds from the SIP Registrar before the SIP REGISTER's time is up.
The default is 20 seconds.

relaxdtmf = yes|no
; The default is no.

rtautoclear = yes|no|number
; Auto-Expire friends made while operating. If it is set to yes, autoexpire will take place in 120 seconds.
The default is yes.

rtcachefriends = yes|no
; Cache realtime friends by adding them to the internal list like friends.
This is added to the config file.
Default is no.

rtpholdtimeout = Number
; Length of time in seconds during which there is no activity before disconnecting
a call on hold.
Default is 0 (no limit).

rtpkeepalive = Number
; Number of seconds of the interval for RTP keepalive packet if there is no passing traffic.
Default is 0 (no RTP keepalive).

rtptimeout = Number
; Number of seconds for waiting for RTP traffic before we hung up.
Default is 0 (no RTP timeout).

rtupdate = yes|no
; Send registry updates to the database when using Realtime support. The default is yes.

sendrpid = yes | no
; whether the SIP header Remote-Party-ID SIP should be sent.
The default is no.

sipdebug = yes|no.
The default setting that determines whether the SIP debug is enabled when loading sip.conf.
The default is no.

srvlookup = yes|no
; Enable DNS SRV checks when called upon. The default is no.

tos = <value>
; Set QoS of IP parameters for outgoing media streams
(numeric values are acceptable, such as tos = 184)

trustrpid = yes|no
; whether the SIP header Remote-Party-ID SIP can be trusted. The default is no.

useclientcode = yes|no:

usereqphone = yes|no
; Indicates whether we need to add ";user=phone" to URI. The default is no.

useragent = <string>
; Changes the SIP header "User-Agent". The default is asterisk.

videosupport = yes | no
; Enables support for SIP video. The default is no.

vmexten = <string>
; Dialplan extension to call mailbox. The default is asterisk. Configuring SIP - peer and client

The following variables can be used in every peer definition

accountcode = <string>
; the users who can be associated to accountcode. It is recommended
that you read the concept on Asterisk billing.

allow = <codec>
; the CODEC which is allowedbased on order preferences.
Use first disallow = ALL before allowing CODEC.

disallow = all
; Disallow all the CODECs to a given peer or user definition.

allowguest = yes|no
; Allow or reject calls from unknown person.
The default is yes. “OSP” can also be set if Asterisk is compiled to support OSP.

auth = <authname>
; The content of the Digest username= on a SIP header.

callerid = <string>
; The caller ID in use if no information is available. The default is asterisk.

call-limit = number

;The number of simultaneous telephone connections that can be made to a specific use/peer.

callgroup = num1, num2-num3
; Defines a call group that can call this tool.

callingpres = number| descriptive_text
; Set appearance of Caller-ID of a connection/call.
Descriptive text values that can be filled in are allowed_not_screened,
allowed_passed_screen, allowed_failed_screen, allowed, prohib_not_screened,
prohib_passed_screen, prohib_failed_screen, prohib, and unavailable.
The default is Allowed_not_screened.

canreinvite = update|yes|no
; whether the client is able to support SIP re-invites. The default is yes.

context = <context_name>
; If type=user, context is for the call going to the SIP user definition.
If type = peer, context in the dialplan is for outbound call of a SIP peer definition.
If type = friend, context is used for all inbound and outbound connections to
the SIP entity definition.

defaultip = ip.add.res.s
; The default IP address for the client host = if not specified as DYNAMIC.
This is used if the client had never been registered to use different IP address.
Only valid if the type=peer.

dtmfmode = inband|info|rfc2833
; How the client handles DTMF signal. Default is rfc2833.

fromuser = <from_ID>
; Determines the user tobe put in "from" other than the callerid (override callerid)
when conducting calls_to_peer (to another SIP proxy). Valid only for type=peer.

fromdomain = <domain>
; Set default From: domain in SIP message when conducting calls _to_ peer.
Valid only in the [general] or type = peer section.

fullcontact = <sip:uri_contact>
; SIP URI contact for realtime peer. Valid only for realtime peers.

host = dynamic|hostname|IPAddr
; Client - IP address or hostname. If you want the phone to register itself,
use dynamic keywords instead of host IP.

incominglimit and outgoinglimit = Number
; Limitation of the number of simultaneous active calls that can be performed by
a SIP client. Valid only for type = peer.

insecure = very|yes|no|invite|port
; Determines how to deal with peer connection.
The default is no (authentication for all connections).

ipaddr = ip.addr.from.peer
; Valid only for realtime peer.

language = language code as defined in indications.conf
; Defining a language for greetings

mailbox=mailbox
; Extension for Voicemail. Valid only for type = peer.

md5secret = MD5-Hash of "<user>: asterisk: <secret>"
; Can be used as a substitute to secret.

Musicclass = determines one of classes written in musiconhold.conf

name = <name>
; The name of the realtime peer. Valid only for realtime peer only.

nat = yes | no
; This variable determines the action pattern of Asterisk for clients behind the NAT.
But it still does not solve the problem if Asterisk is behind NAT.
The default is no, which means using the RFC3581 technique.

outboundproxy = IP_address or DNS SRV name
; SRV name, hostname, or IP address of the outbound SIP Proxy.
Valid only in the [general] and type = peer section.

progressinband = never|no|yes
; Do we generate ring in in-band. The default is never.

promiscredir=yes|no
; Allows support for 302 Redirects. The default is no.

qualify=yes|no|milliseconds
; Check whether the client can be reached.
If yes, a check will be done every 2000 milliseconds (2 seconds).
Valid only in the [general] and type=peer section.

regseconds = seconds
; Time in seconds between SIP REGISTERS. Valid only for realtime peer only.

rtpkeepalive=seconds
; The time, in seconds, of sending RTP keepalive packet if there is
no RTP traffic on the connection. Default 0 (no RTP keepalive).
Valid only for the [general] and type=peer section.

rtptimeout=seconds
; Disconnect a connection if within x seconds there is no RTP activity and
we are not in on hold position.
Valid only in the [general] and type=peer section.

rtpholdtimeout = seconds
; Disconnect a connection if within x seconds there is no RTP activity and
we are in on hold position.
Valid only for the section [general] and type=peer.

secret=password
; If Asterisk functions as a SIP Server, then SIP client must login using "password".
If Asterisk functions as a SIP client to a remote SIP server,
it requires SIP INVITE authentication, then the contents of secret is used
for SIP INVITE authentication that is sent by Asterisk to the remote server.

sendrpid=yes|no
; whether Remote-Party-ID SIP header should be sent. Default is no.

setvar=variable=value

; Variable channel which should be set for all connections to this peer / user.

subscribecontext = <context_name>
; Set a specific context for SIP SUBSCRIBE requests

trustrpid=yes|no
; whether Remote-Party-ID SIP header can be trusted. The default is no.

type = user|peer|friend
; connection to the client, outbound provider or a full client?

usereqphone=yes|no
; Showing whether to add "; user=phone" to the URI. Default no.
Valid only for the [general] and type=peer section.

username=<username[@realm]>
; If functioning as a SIP client to a remote SIP server that requires
SIP INVITE authentication, then this parameter is used for SIP INVITE authentication,
which is sent by Asterisk to a remote SIP server; for peers who will register to Asterisk,
the username is used in INVITE until they are registered.

vmexten = <string>
; Dialplan extension to reach mailbox. Default asterisk.
Only valid in the [general] or type=peer section.

DAHDI Usage For VoIP Cards

In Newer Asterisk software, DAHDI short for "Digium Asterisk Hardware Device Interface" is used to deal with VoIP hardwares. DAHDI is the new name for 'Zaptel' as of May 19th 2008. The post at http://blogs.digium.com/2008/05/19/zaptel-project-being-renamed-to-dahdi/ details the reason for the change. Asterisk 1.4 releases later than 1.4.21, and all releases of Asterisk 1.6, will automatically use DAHDI in preference to Zaptel, even if Zaptel is still installed on the system. More reference on this can be found at http://www.voip-info.org/wiki/view/DAHDI

Digium resources regarding zaptel to dahdi migration can be found at http://www.asterisk.org/node/48481. Basically /etc/zaptel.conf Becomes /etc/dahdi/system.conf and /etc/asterisk/zapata.conf Becomes /etc/asterisk/chan_dahdi.conf.

There are three (3) main configuration files, namely,

/etc/dahdi/system.conf
/etc/asterisk/chan_dahdi.conf
/etc/asterisk/dahdi-channels.conf

In /etc/dahdi/system.conf, unlike zaptel.conf, you have to explicitly set the echo canceller for each channel.

There are a number of other configuration files under /etc/dahdi

/etc/dahdi/init.conf
: Replaces /etc/default/zaptel (on Debians) and /etc/sysconfig/zaptel (on most other systems) - this is a shell script snippet that is sourced by the dahdi init.d script. All values there are optional (no need to explicitly define TELEPHONY=no). The variable MODULES, however, is no longer read from it. IT is read from:

/etc/dahdi/modules
: A list of modules to load. Replaces the variable MODULES from the above configuration file.

/etc/dahdi/genconf_parameters
: Fine--tuning parameters for dahdi_genconf (replaces zapconf and also deprecates genzaptelconf).

DAHDI Architecture

The package is composed of two sub-packages:

Kernel

Include kernel modules and minila helper files (firmwares)

Tools

The userspace tools to control DAHDI spans/channels:-

dahdi_cfg
: The DAHDI Configurator, which parses system.conf

dahdi_genconf

Generates /etc/dahdi/system.conf, so it's better that you don't hand edit system.conf. Uses /etc/dahdi/genconf_parameters to define it's actions.

dahdi_hardware
: Displays listing of DAHDI hardware detected

dahdi_monitor
: Monitors signal level on analog channel allows you to record audio from it
Usage: dahdi_monitor <channel num> -v -m -o -p -l limit -f FILE -s FILE -r FILE1 -t FILE2 -F FILE -S FILE -R FILE1 -T FILE2
example :- dahdi_monitor 1 -vv
note: extremly usefull, but otherwise not mentioned, that the raw format output is 8Khz 16bit signed. Use sox to convert to a wav. sox -r 8000 -s -w rx.raw rx.wav

dahdi_scan
: Generates a list of things DAHDI channels, with some details

dahdi_test
: Measures accuracy of the FXO/FXS board software digital signal processing

dahdi_tool
: A nice tool to see what your boards are doing.

DAHDI Sample installation

After compiling and installing of dahdi and asterisk, you have to perform some further steps to use your hardware. This example will show you a few steps how to get asterisk and two Digium cards enabled:

- Detect your hardware. This will generate /etc/dahdi/system.conf and /etc/asterisk/dahdi-channels.conf.

  # lspci -n

  You should see something like this for TDM410

  00:09.0 0200: d161:8005 (rev 11)

- Edit /etc/dahdi/system.conf and make sure there is

  loadzone = us
  defaultzone = us

- Check the channel type

  /etc/init.d/dahdi restart
  dahdi_scan

  We will see something like

  active=yes
  alarms=OK
  description=Wildcard TDM410P Board 1
  name=WCTDM/0
  manufacturer=Digium
  devicetype=Wildcard TDM410P
  location=PCI Bus 03 Slot 03
  basechan=1
  totchans=4
  irq=23
  type=analog
  port=1,FXO
  port=2,FXO
  port=3,FXS
  port=4,FXS

- Edit /etc/dahdi/system.conf to reflect the findings from dahdi_scan

  fxoks=3,4
  fxsks=1,2
  echocanceller=mg2,1-4

- Run modprobe as root and do dahdi_cfg

  # modprobe wctdm24xxp
  # dahdi_cfg -vv

- Check if it is correctly loaded

  # dmesg

  we will see something like

```
[  961.484269] wctdm24xxp 0000:00:09.0: PCI INT A -> GSI 17 (level, low) -> IRQ 17
[  961.940405] Port 1: Installed -- AUTO FXO (FCC mode)
[  962.576453] Port 2: Installed -- AUTO FXO (FCC mode)
[  964.209579] Port 3: Installed -- AUTO FXS/DPO
[  965.838703] Port 4: Installed -- AUTO FXS/DPO
[  965.842700] VPM100: Not Present
[  965.846981] Found a Wildcard TDM: Wildcard TDM410P (4 modules)
```

- This is not necessary, but if you like you can do generate etc/dahdi/system.conf and /etc/asterisk/dahdi-channels.conf.

  # dahdi_genconf

- Restart dahdi to unload and reload all modules and drivers

  # /etc/init.d/dahdi restart

- Point file /etc/asterisk/chan_dahdi.conf to /etc/asterisk/dahdi-channels.conf

  # open chan_dahdi.conf and include it under the section [channels]
  #
  # NOTE: You can edit and configure /etc/asterisk/dahdi-channels.conf at any time
  # to set up your specific options there.
  ...
  [channels]
  # include /etc/asterisk/dahdi-channels.conf
  ...

- In /etc/asterisk/dahdi-channels.conf we will see something like

  signalling=fxs_ks
  callerid=asreceived
  group=0
  context=from-pstn

channel => 1
callerid=
group=
context=default

signalling=fxo_ks
callerid="Channel 3" <4003>
mailbox=4003
group=5
context=from-internal
channel => 3
callerid=
mailbox=
group=
context=default

- Restart asterisk

  # /etc/init.d/asterisk restart

- Verify your current system status. You should get some output like this:

```
asterisk -r
asterisk*CLI> dahdi show status
Description                          Alarms  IRQ    bpviol CRC4   Fra Codi Options  LBO
DAHDI_DUMMY/1 (source: HRtimer) 1    UNCONFI 0      0      0      CAS Unk  YEL      0 db (CSU)/0-133 feet (DSX-1)
Wildcard TDM410P Board 1             OK      8      0      0      CAS Unk  YEL      0 db (CSU)/0-133 feet (DSX-1)
```

- Verify your configured channels

```
asterisk*CLI> dahdi show channels
   Chan Extension  Context         Language   MOH Interpret        Blocked    State
 pseudo            default                    default                          In Service
      1            from-pstn                  default                          In Service
      2            from-pstn                  default                          In Service
      3            from-internal              default                          In Service
      4            from-internal              default                          In Service
```

DAHDI extensions.conf

An example of DAHDI dial plan is as follows.

[from-internal]

```
exten => 1000,1,Dial(DAHDI/1,20,rt)
exten => 1000,2,Voicemail(1000,u)
exten => 1000,102,Voicemail(1000,b)

exten => 2000,1,Dial(DAHDI/2,20,rt)
exten => 2000,2,Voicemail(2000,u)
exten => 2000,102,Voicemail(2000,b)

exten => 8500,1,VoiceMailMain
exten => 8501,1,MusicOnHold

exten => 1001,1,Dial(DAHDI/3,20,rt)
exten => 1001,2,Voicemail(1000,u)
exten => 1001,102,Voicemail(1000,b)
exten => 1002,1,Dial(DAHDI/4,20,rt)
exten => 1002,2,Voicemail(2000,u)
exten => 1002,102,Voicemail(2000,b)

exten => _9.,1,Dial(DAHDI/g0/www${EXTEN:1})
exten => _9.,2,Congestion
exten => _91.,1,Dial(DAHDI/1/www${EXTEN:2})
exten => _91.,2,Congestion
exten => _92.,1,Dial(DAHDI/2/www${EXTEN:2})
exten => _92.,2,Congestion

[from-pstn]
exten => s,1,Answer
exten => s,2,Dial(DAHDI/g1,20,rt)
exten => s,3,Voicemail(1000,u)
exten => s,103,Voicemail(1000,b)
```

We have to make sure a couple of things:

- [from-internal] and [from-pstn] should be reflected in /etc/asterisk/dahdi-channels.conf
- [from-internal] and [from-pstn] must exists in /etc/asterisk/extensions.conf.
- If unsure, replace [from-internal] and [from-pstn] with default.
- Make sure DAHDI/1, DAHDI/2, DAHDI/3, DAHDI/4, DAHDI/g1 etc are correct as reflected in /etc/asterisk/dahdi-channes.conf

CHAPTER 7: Briker Softswitch

Briker is a softswitch built on a number of software, including Free PBX 2.4, Asterisk 1.4, Asterisk2Billing 1.3 and Webmin, all of which bundled into a linux software that runs on Ubuntu Platform. Briker may be freely downloaded from http://www.briker.org.

The first thing you need to do before installing the Briker is to set your BIOS configuration so you will boot your computer using the CDROM. Then insert the Briker IP PBX CD into the CDROM. Begin installation by typing “install” and press enter. Then the briker automatically erases the content of the hardisk and uses all the spaces available in the hardisk.

Briker's Installation Process

Figure 7.1: In the installer menu, there are many options: install, check, rescue, memtest, and hd

Once the installation process is completed, the system will create a default password for console login and web login, as well as configure the default IP address.

Default console login (SSH port 22):
Username : support
Password : Briker

Default web login (HTTP port 80):
Username : administrator
Password : Briker

default IP address :
IP address : 192.168.2.2
Subnet mask : 255.255.255.0

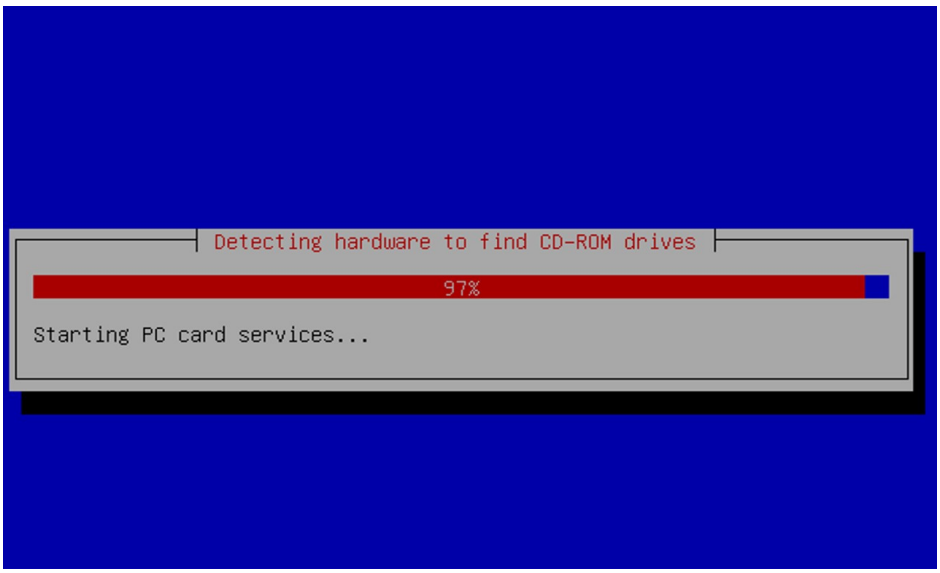


Figure 7.2: Briker checks whether there's a CD-ROM

Briker automatically checks the hardware components installed and finding the installer CD-ROM.

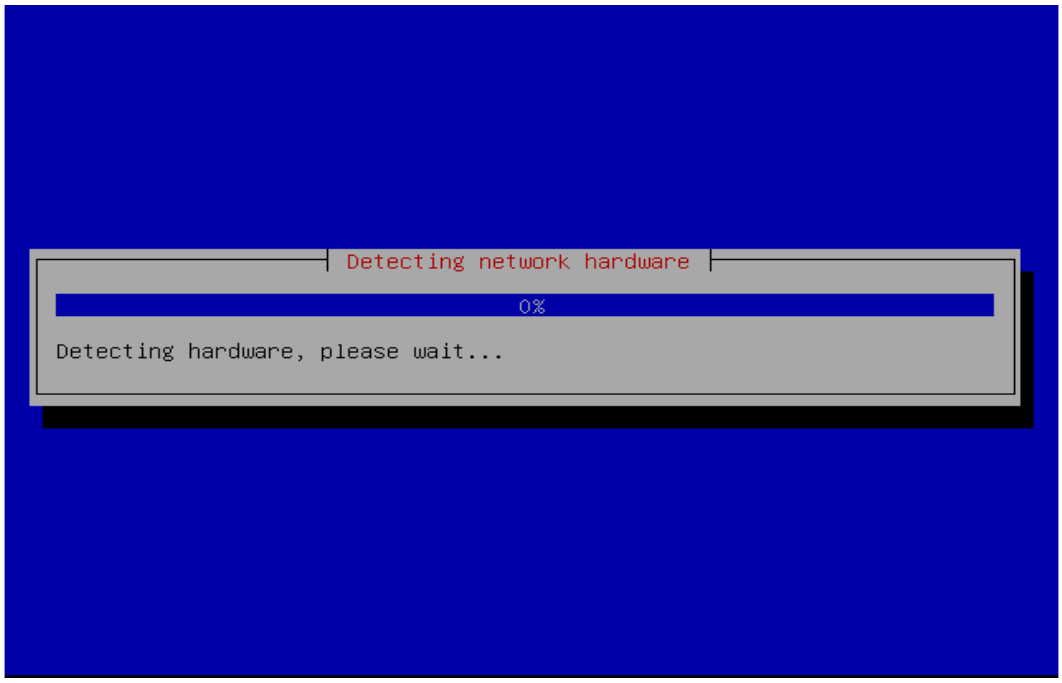


Figure 7.3: Briker also checks whether all the hardware required for networking are in place

Then the briker automatically checks the network hardware, and automatically configure the IP address.

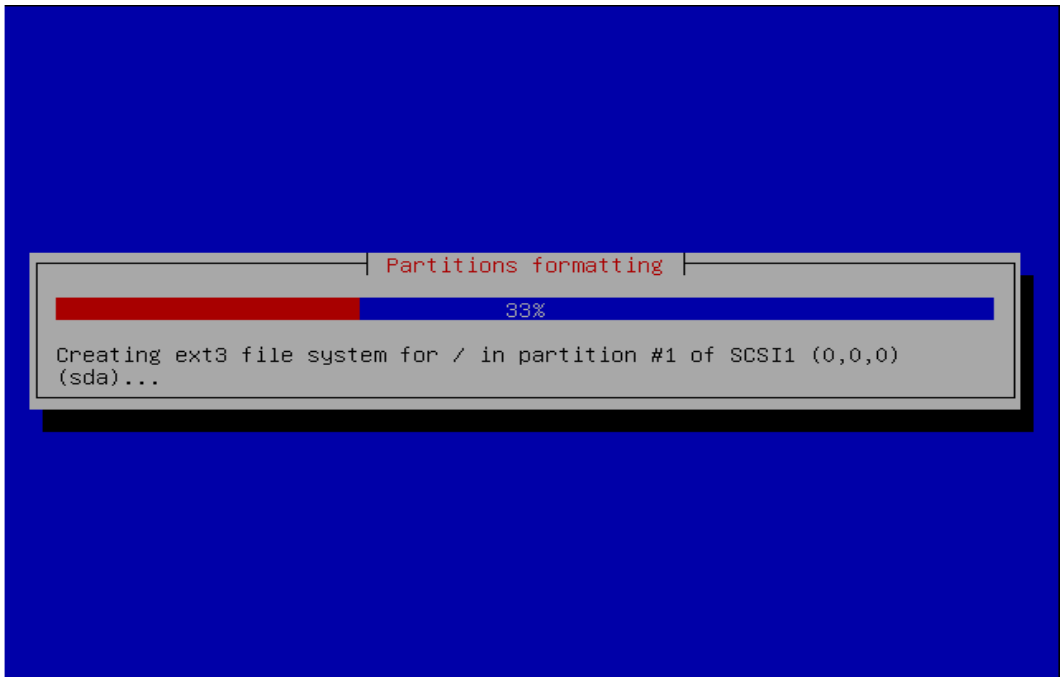


Figure 7.4: Formatting partitions

Then the Briker automatically erases the content of the hardisk and uses all the spaces available in the hardisk for it.

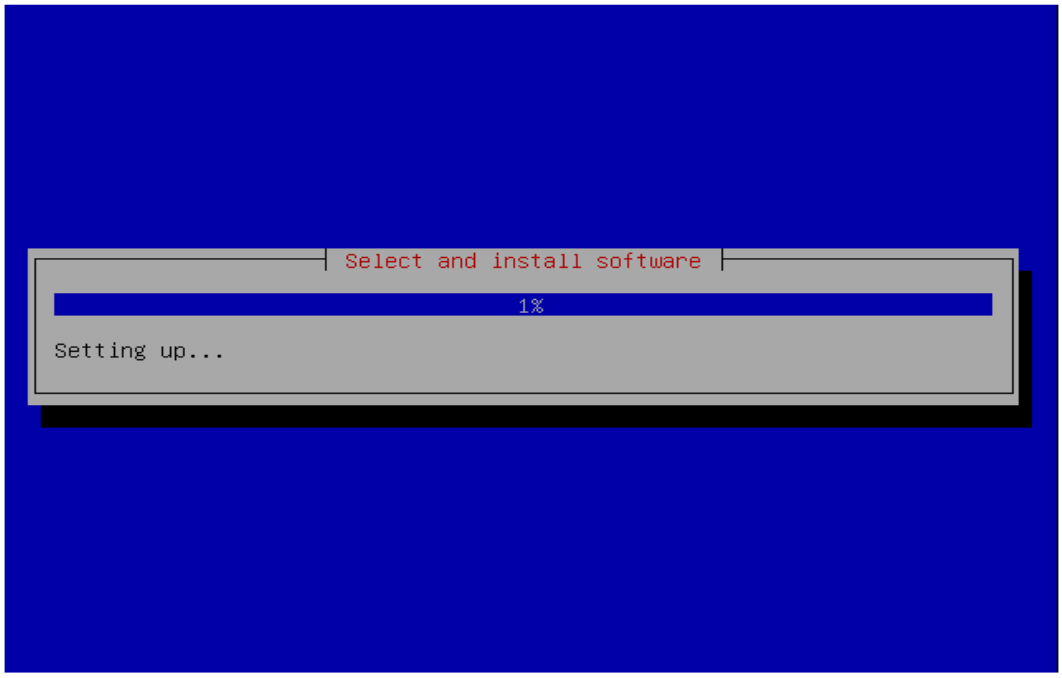


Figure 7.5: Installing required software

The Briker automatically installs the base system and other software required.

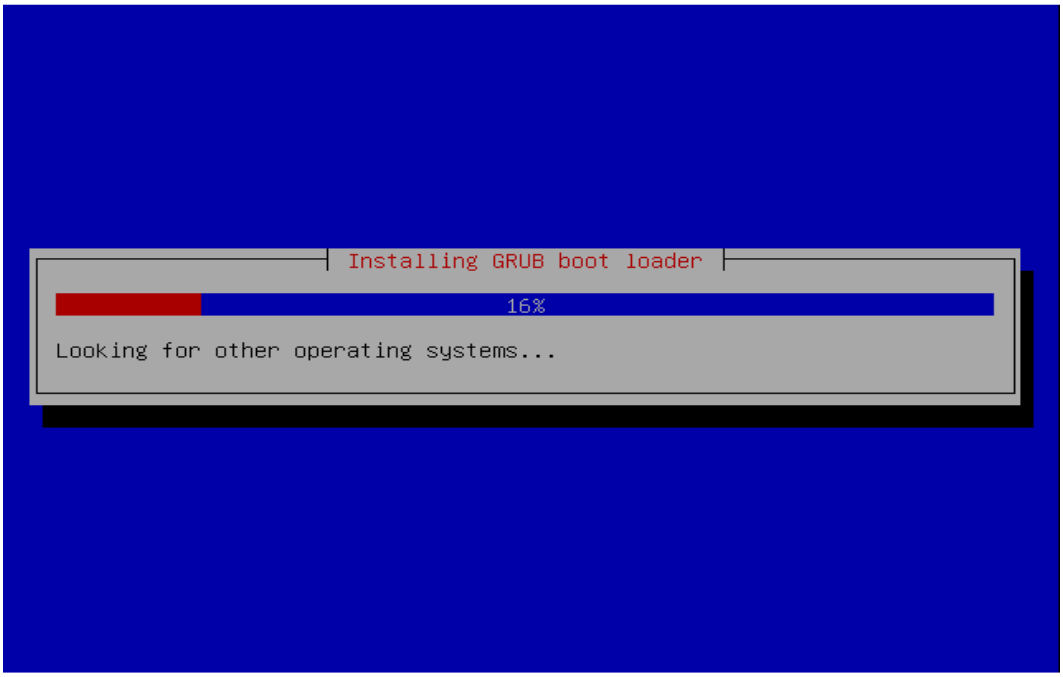


Figure 7.6: Installing GRUB boot loader

Finally, the briker will install GRUB boot loader. And once the whole installation process is completed, the CDROM will automatically eject the briker CD and the computer will restart.

Briker's Console

```
 wcte12xp.
 wct1xxp.
 wcte11xp.
 wctdm24xxp.
 wcfxo.
 ystdm8xx.
 ystdm16xx.
 wctdm.
 wcusb.
 xpp_usb.
No hardware timing source found in /proc/zaptel, loading ztdummy
Running ztcfg: done.
 * Starting periodic command scheduler crond                          [ OK ]
 * Starting web server apache2
apache2: Could not reliably determine the server's fully qualified domain name,
using 127.0.0.1 for ServerName
                                                                      [ OK ]
 * Running local boot scripts (/etc/rc.local)
nohup: appending output to `nohup.out'
                                                                      [ OK ]

Briker 1.0.2 "OWP" ippbx tty1

ippbx login: support
Password: _
```

Figure 7.7: With the installation completed, you will be able to begin the configuration process

After installing the software, we can begin configuring through the console, by changing the IP address, date etc. All commands for the login console can be carried out only after you authenticate yourself as a root user. The commands for configuration through the console will not work unless you enter the following entries:

$ sudo su -

The password you have to enter is the one similar to that of user support (default password). For security reason, you should change the default password by doing the following:

passwd

The default IP address of the Briker is 192.168.2.2. Change this address so that Briker will be able to adjust any network topology and obtain IP address allocation, by first of all editing file /etc/network/interfaces:

vi /etc/network/interfaces

```
/etc/netwo~interfaces   [----]  0 L:[  1+ 0   1/ 19] *(0   / 501b)= #  35 0x23
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
        address 192.168.2.2
        netmask 255.255.255.0
        network 192.168.2.0
        broadcast 192.168.2.255
        gateway 192.168.2.1
        # dns-* options are implemented by the resolvconf package, if installed
        dns-nameservers 192.168.2.1
        dns-search local

1Help  2Save  3Mark  4Replac 5Copy  6Move  7Search 8Delete 9PullDn 10Quit
```

Figure 7.8: The default IP address of Briker

The above figure shows that the IP address is 192.168.2.2. Make necessary changes and save the configuration by pressing F2 and exit the editing platform by pressing F10.

Then restart the networking services to activate the configuration, by executing the following syntax:

```
# /etc/init.d/networking restart
```

Next we have to make sure that the date and time of the Briker are set properly. Check them by typing the following syntax:

```
# date
```

If they are not set properly, then adjust them. For example, if we want to set the time to 08.00 and date to July 1, 2008, then the syntax would be:

```
# date -s "2008-07-01 08:00:00"
```

Setting the date and time properly is particularly important if you are using Briker for commercial use.

Briker's Web Configuration

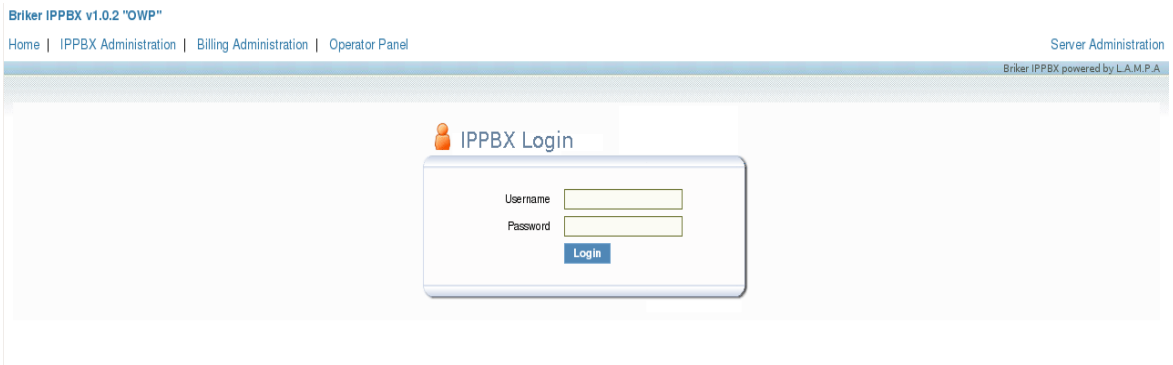


Figure 7.9: In order to configure Briker, you need to log on

With the console properly configured, you can now configure Briker administration. Browse to Briker's IP address through the web browser, a login window will appear as shown in Figure 5.9. Use the default username, “administrator” and default password “Briker”, then click on Login.

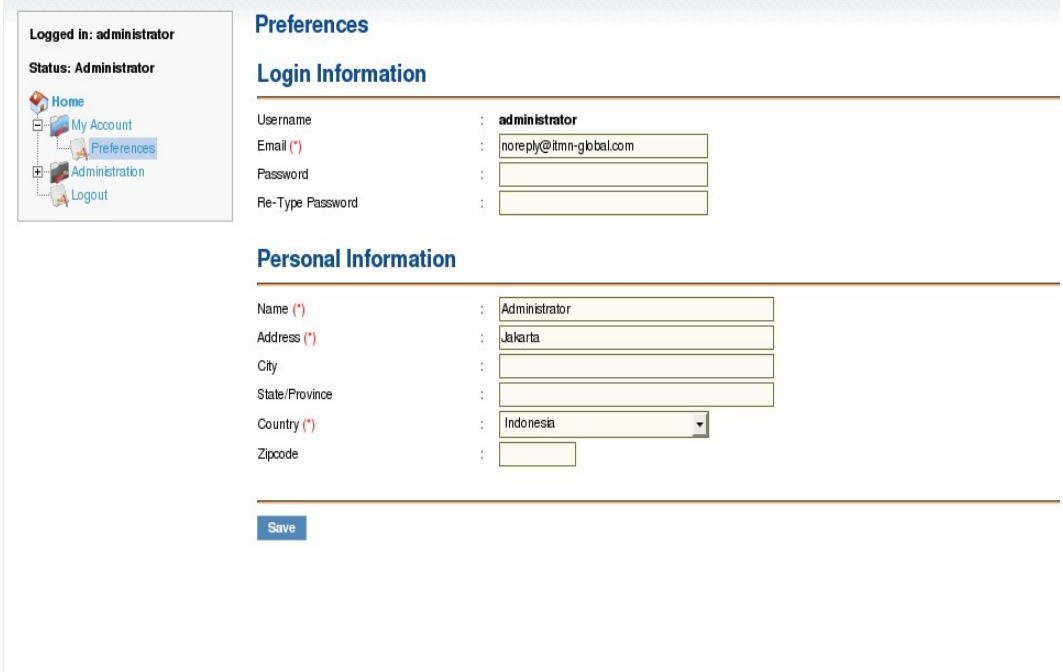


Figure 7.10: Preferences settings

To change the administrator password, click on My Account and choose Preferences. A menu as shown in Figure 5.10 will appear. Enter the new password in the password box and enter the same password in the Re-type Password box, then click on Save to activate the configuration.

Figure 7.11: Adding user

In Briker, we can have more than one administrator. Choose Administration and then Manage User. Then fill in Username, Email, Full Name, Password and User Level, and click Add.

To do IP PBX configuration, choose IP PBX Administration from the main menu, as shown in Figure 7.9.

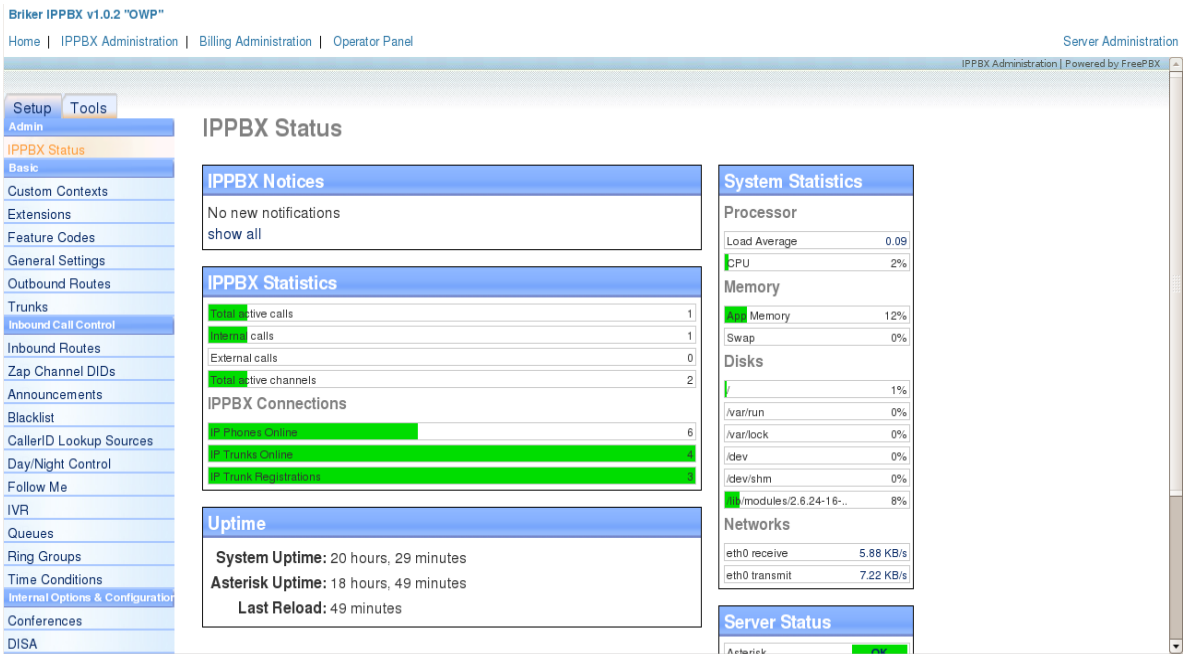


Figure 7.12: menu to configure IPPBX features is available in Briker, some of which are extensions, trunks and routes configuration.

IP PBX status indicates System Statistics showing the percentage of Load Average, CPU, Memory and Swap being used, the usage of harddisk space and the speed of Receive and Transmit Ethernet. Also availabe in this display is IPPBX Statistics showing Total Active Calls, Internal Calls, External Calls, Total Active Channels, and Uptime Briker. These data are realtime, updated periodically and automatically, a process that consumes a considerable amount of CPU resources. So it is recommended that you do not keep accessing this main page.

When you are familiar with the main display, it is time for you to add Extension, user who will use Briker services. Click Extension on IP PBX Administration menu. Through this option, you will be able to add new account, omit or replace any existing one.

Click Add Extensions. Then choose the sort of protocols used by the account: SIP, IAX2, ZAP, or Custom (protocol other than the first three). With any of these protocol selected, click submit (shown in Figure 7.13).

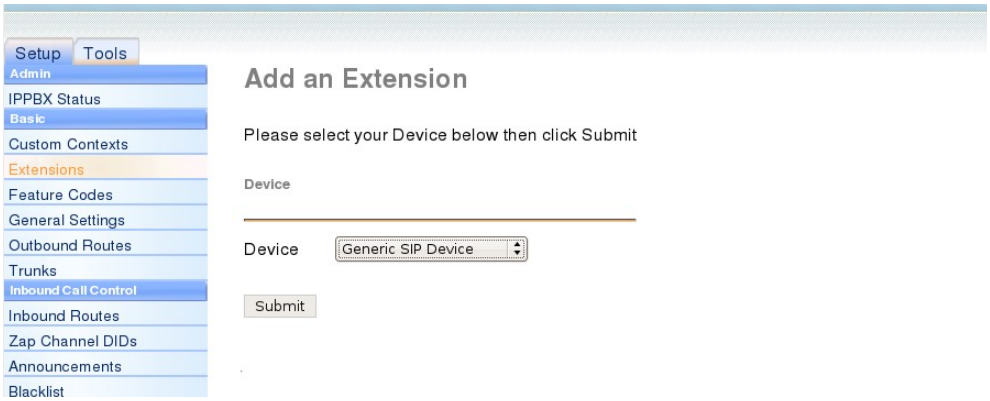


Figure 7.13: Extension differs by the type of device used by an account

Then Dialog properties as shown in Figure 7.14 will appear, prompting you to enter all the information required for adding extension.

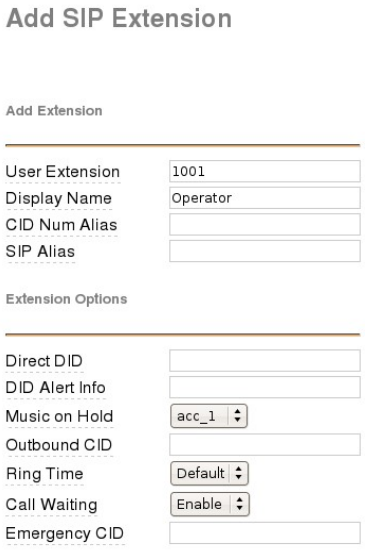


Figure 7.14: Dialog Properties of Adding SIP Extension

Device Options

This device uses sip technology.

secret	123456
dtmfmode	rfc2833

Fax Handling

Fax Extension	IPPBX default
Fax Email	
Fax Detection Type	None
Pause after answer	0

Privacy

Privacy Manager	No

Recording Options

Record Incoming	On Demand
Record Outgoing	On Demand

Voicemail & Directory

Status	Disabled	
Voicemail Password		
Email Address		
Pager Email Address		
Email Attachment	yes	no
Play CID	yes	no
Play Envelope	yes	no
Delete Vmail	yes	no
VM Options		
VM Context	default	
VmX Locater™	Disabled	

Submit

Fill in user extension with extension number, e.g. 1001. This is usually just numeric. Then fill in the display name, the name that will be used as Caller ID when dialing. Fill in secret with the password used by user for authentication process in registration extension at User Agent layer. Click Submit.

Zaptel Configuration

Zaptel is a collection of tools and drivers detecting hardware in the form of analog and digital telephony card installed on PCI or mini-PCI slot. The telephony card is used to connect the briker to Plain Old Telephony System (POTS) network or to analog telephone.

For example, connecting the briker to analog PBX requires analog telephony card. So does the briker when it is connected to Public Switch Telephone Network (PSTN), connected through a telephone cable provided by telecommunication operator. The analog or digital card to be used, however, depends on the type of technology being used by the operator.

To configure Zaptel, first of all, log in through the console. As this installation requires root privileges, log in as a root by executing the following commands:

```
$ sudo su
```

Then run genzaptelconf command

```
# genzaptelconf
```

To check whether zaptel has successfully detected what it is looking for, do checking by executing the following command:

```
# ztcfg -vvv
```

Then restart zaptel, by executing the following command:

```
# /etc/init.d/zaptel restart
```

SIP Trunk

Figure 7.15: Adding a Trunk

In *IPPBX Administration* menu, choose *Trunks menu,* choose *Add SIP Trunk.*

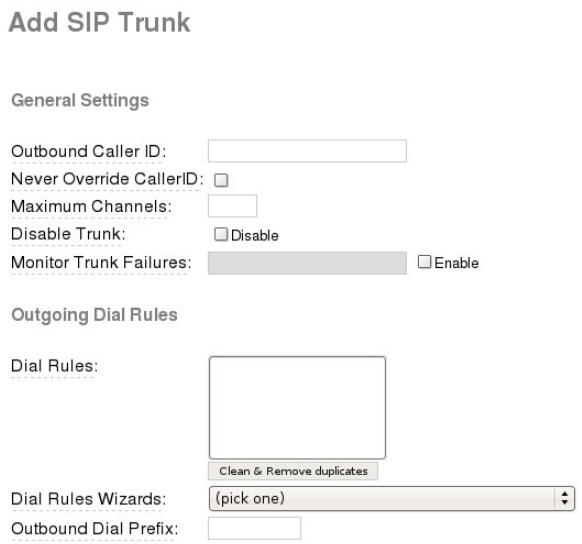


Figure 7.16: The general settings of Add SIP Trunk

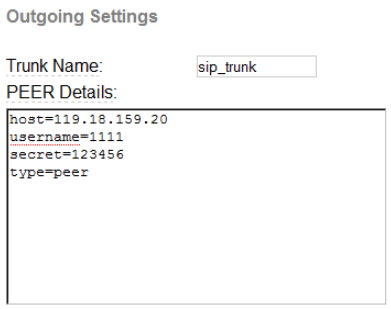


Figure 7.17: The general settings of Add SIP Trunk

Fill in the Outgoing Settings, as shown in Figure 7.17, by using data account from different server. Add particular options whenever necessary, such as fail to connect or unable to receive and make a call through trunk. Other particular options are:

context = from-trunk
qualify = yes

insecure = port,invite
authuser = <similar to user contact or meeting its trunk needs>
fromuser = <similar to user contact or meeting its trunk needs>
fromdomain = <similar to host or meeting its trunk needs>

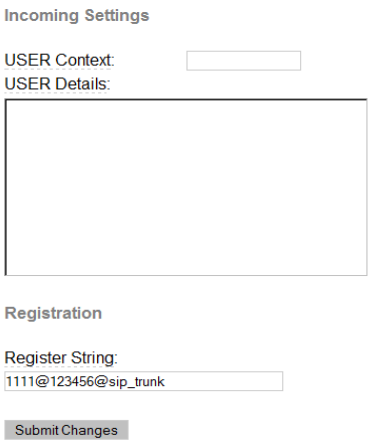


Figure 5.18: The general settings of Add SIP Trunk

For Register String, obtain the data from Outgoing Settings, with the format username:secret@<Trunk Name>. Save the configuration by clicking Submit Changes.

IAX2 Trunk

Go to Trunk menu, as if you like to configure the AIX2 Trunk. Choose Add IAX2.

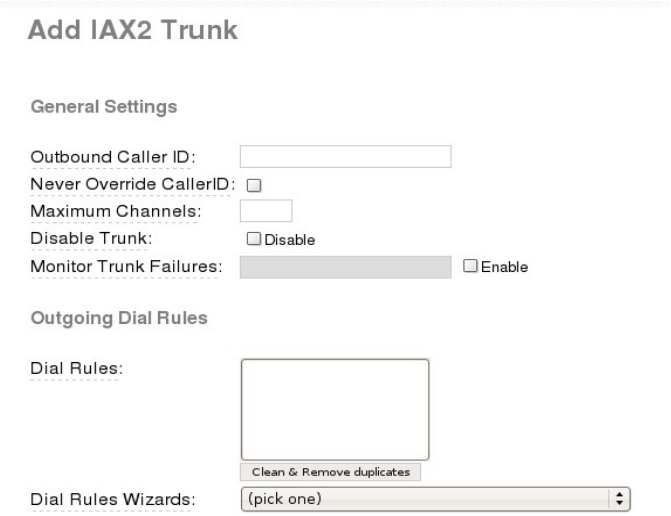


Figure 7.19: The general settings of Add IAX2 Trunk

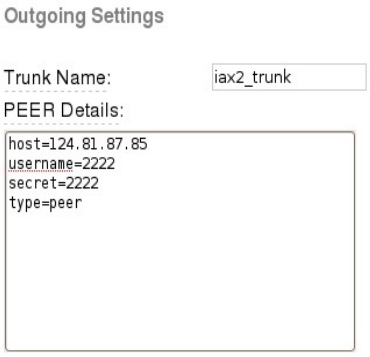


Figure 7.20: The general settings of Add IAX2 Trunk

Incoming Settings

USER Context:

USER Details:

```
secret=***password***
type=user
context=from-trunk
```

Registration

Register String:

2222:2222@iax2_trunk

Submit Changes

Figure 7.21: The general settings of Add IAX2 Trunk

For IAX2 Trunk, make the same configuration as shown in Figure 7.19, 7.20, and 7.21.

H323 Trunk

Go to the Trunks menu in IPPBX Administration menu.

Figure 7.22: There is no option specifically for H.323. So you have to choose Custom Trunk

Then choose Add Custom Trunk.

General Settings

Outbound Caller ID:
Never Override CallerID: ☐
Maximum Channels:
Disable Trunk: ☐ Disable
Monitor Trunk Failures: ☐ Enable

Outgoing Dial Rules

Dial Rules:
Clean & Remove duplicates
Dial Rules Wizards: (pick one)
Outbound Dial Prefix:

Outgoing Settings

Custom Dial String: H323/119.18.159.20/$OUTNUM$

Submit Changes

Figure 7.23: The general settings of Add Custom Trunk

For customized Trunk, fill in the Custom Dial String by using the format H323/<h323-gateway-address>/$OUTNUM$. As shown in Figure 5.23, the gateway address of H323 is 119.18.159.20. Then click Submit Changes.

Open a terminal console, then edit the /etc/asterisk/h323.conf file:

```
# mcedit /etc/asterisk/h323.conf
```

Edit the following options available in the /etc/asterisk/h323.conf file:

```
Port = 1720
bindaddr = <IP Briker address>
```

Then restart asterisk, by executing the following command:

```
# /etc/init.d/amportal restart
```

ZAP Trunk

This type of Trunk is connected to PSTN line, through analog card (TDM xxx) or digital card (TE xxx). After doing the zaptel configuration, do the configuration in IPPBX, by first of all logging in to IP PBX Administration.

Figure 7.24: Adding ZAP Trunk

Choose Trunks menu and choose Add Zap Trunk. A menu for trunk configuration should appear as shown in Figure 7.25.

Figure 7.25: General settings of Add Zap Trunk

Fill Zap Identifier (trunk name) with g0, which means group 0. The description of the group's name (for example, group 0) can be found at /etc/asterisk/zapata-channels.conf file.

Outbound Routes

Outbound routes are used to manage where the call should go to, the one going out through the trunk. It is these Outbound routes that define all the outgoing calls. For example, for connecting to PSTN, Briker uses prefix 9, which is followed by the destination number. The following is an example of its configuration.

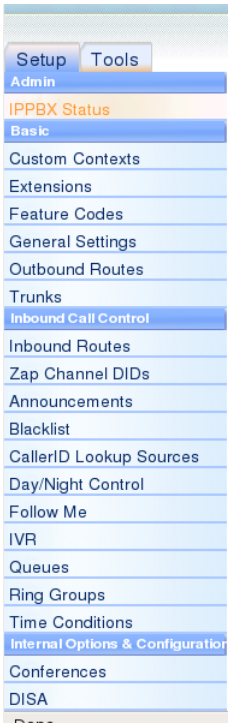
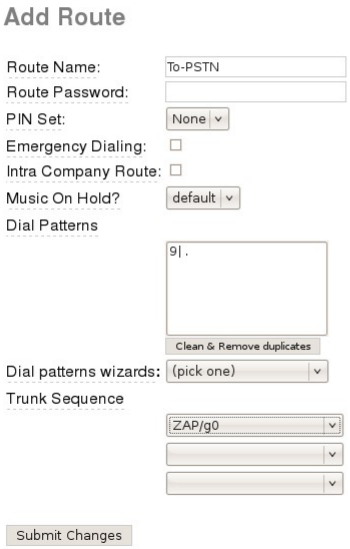


Figure 7.26: Setting Outbound Routes

In IPPBX Administration, choose Outbound Routes. Choose Add Route.

Figure 7.27:
Setting Outbound Routes



Fill in the configuration using the settings shown in Figure 7.27. Of the parameters in the settings, route name, dial pattern (the initial code to connect to other server), and Trunk Sequence (Trunk being used. Look at Trunks section) are most important. Once you have completed the settings, click Submit Changes.

Inbound Routes

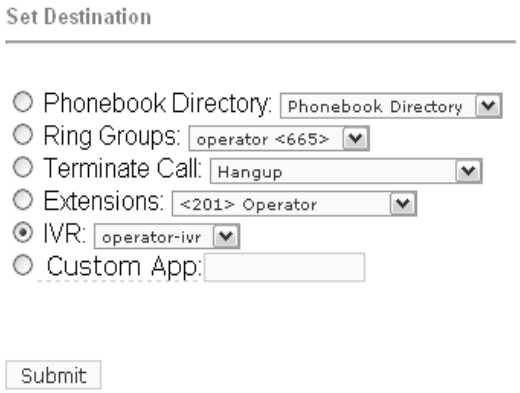
Inbound Routes functions to manage the destination of the call coming from the trunk. When a call comes from the trunk, the system will check whether the call is in compliant with the Inbound Routes configuration. If it is, then the call will be forwarded to its destination according to the configuration.

In IP PBX Administration menu, choose Inbound Routes. Then choose Add Incoming Route.

Figure 7.28: Setting Inbound Routes

For default configuration, you can leave Add Incoming Route blank. In Set Destination, you can direct any incoming calls to a certain destination. In the example shown in Figure 7.29, all incoming calls are directed to the IVR.

Figure 7.29:
In this example, all calls are directed to IVR



Then click Submit

Interactive Voice Response

Interactive Voice Response, commonly abbreviated as IVR, is a feature for managing automatic response whenever there is a call coming in. The following are the steps necessary to do IVT configuration in the briker.

Setup Recordings

Make a record for IVR that you will use (you can use the MS. Recorder application). For example, you can record “Welcome to PT Jelajah Media Information, press 1 for operator,” and set the encode to 16 bit, 8,000 Hz, and save it using the .wav extension (i.e. Welcome-jmi.wav). Upload the .wav file you have just created to the menu: IPPBX Administration > System Recordings, upload and name the file, for example, welcome-jmi, and save it.
IVR Setup

In IPPBX Administration menu, choose IVR. Then choose Add IVR.

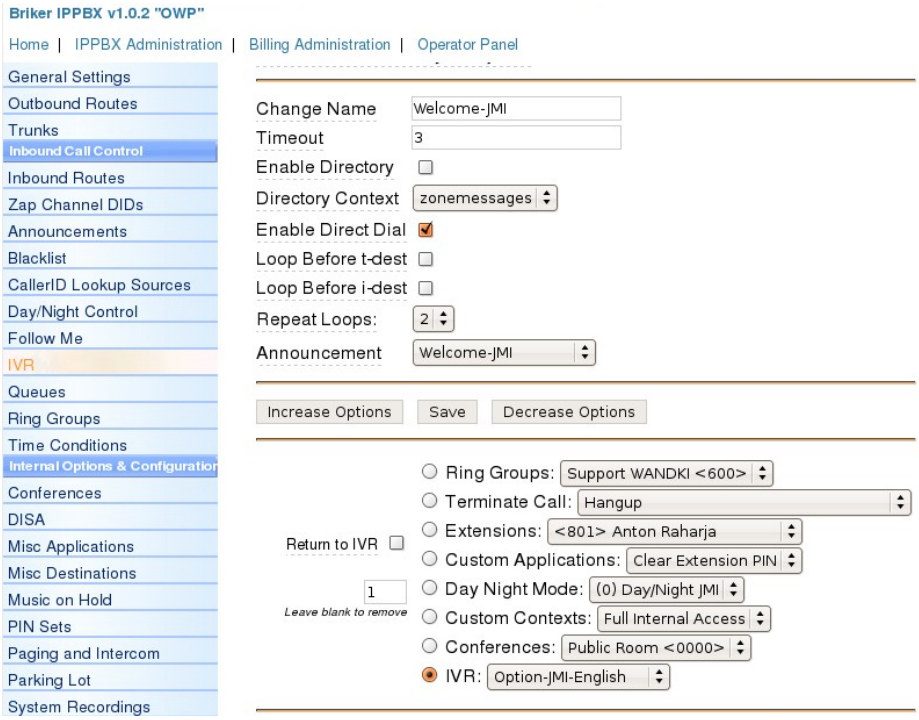


Figure 7.30: IVR Settings

Fill the parameters with the following data:

Change Name : Welcome-JMI
Timeout : 10
Enable Directory : no/unchecked
Directory Context: default/empty
Enable Direct Dial : yes/check
Announcement : Welcome-JMI (recording)

Options available in the Figure 7.30 imply that a user who call the IVR could press 1 and be forwarded to Option-JMI-English, provided that the IVR Option-JMI-English is activated. Once the data and options are configured, click Save and choose Apply configuration changes.

Ring Groups

Ring Group is one of many features used to manage group call. For example, in a company with 5 telephone operators/agents, the five operators can be included as a group, which is named, for example, 'operator help.' Whenever there is an incoming call, the call will be directed to the Ring Group 'operator help.' When the first operator is busy, the call will be forwarded to the second operator and so on. The following is the Ring Group configuration in the briker.

Choose Ring Groups in IPPBX Administration menu. Then choose Add Ring Groups

Briker IPPBX v1.0.2 "OWP"

Home | IPPBX Administration | Billing Administration | Operator Panel

Setup Tools

Admin
IPPBX Status
Basic
Custom Contexts
Extensions
Feature Codes
General Settings
Outbound Routes
Trunks
Inbound Call Control
Inbound Routes
Zap Channel DIDs
Announcements
Blacklist
CallerID Lookup Sources
Day/Night Control
Follow Me
IVR
Queues
Ring Groups
Time Conditions
Internal Options & Configuration
Conferences

Add Ring Group

Add Ring Group

Ring-Group Number:	600
Group Description::	
Ring Strategy:	ringall
Ring Time (max 60 sec)	20
Extension List:	1001 1002
Extension Quick Pick	(pick extension)
Announcement:	None
Play Music On Hold?	Ring
CID Name Prefix:	
Alert Info:	
Confirm Calls:	☐
Remote Announce:	Default
Too-Late Announce:	Default

Figure 7.31: Ring Groups settings

Use the following configuration

Add Ring Group

Add Ring Group

Ring-Group Number:	600
Group Description::	
Ring Strategy:	ringall
Ring Time (max 60 sec)	20
Extension List:	1001 1002
Extension Quick Pick	(pick extension)
Announcement:	None
Play Music On Hold?	Ring
CID Name Prefix:	
Alert Info:	
Confirm Calls:	☐
Remote Announce:	Default
Too-Late Announce:	Default

Destination if no answer:

- ○ Ring Groups: Support WANDKI <600>
- ○ Terminate Call: Hangup
- ○ Extensions: <801> Anton Raharja
- ○ Custom Applications: Clear Extension PIN
- ○ Day Night Mode: (0) Day/Night JMI
- ○ Custom Contexts: Full Internal Access
- ○ Conferences: Public Room <0000>
- ◉ IVR: Welcome-JMI

Submit Changes

Figure 7.32: In this example, the caller will be directed to IVR "Welcome-JMI" if the a group operator does not respond

Use the settings shown in Figure 7.31. The settings implies that if a group operator does not respond, then the caller will be directed to IVR 'Welcome-JMI.'

Pin Sets

Pin Sets functions as system authentication, a feature activated when a user does his or her call through the trunk and entered the password required.

Choose Pin Sets in IPPBX Administration menu, then choose Add Password Set.

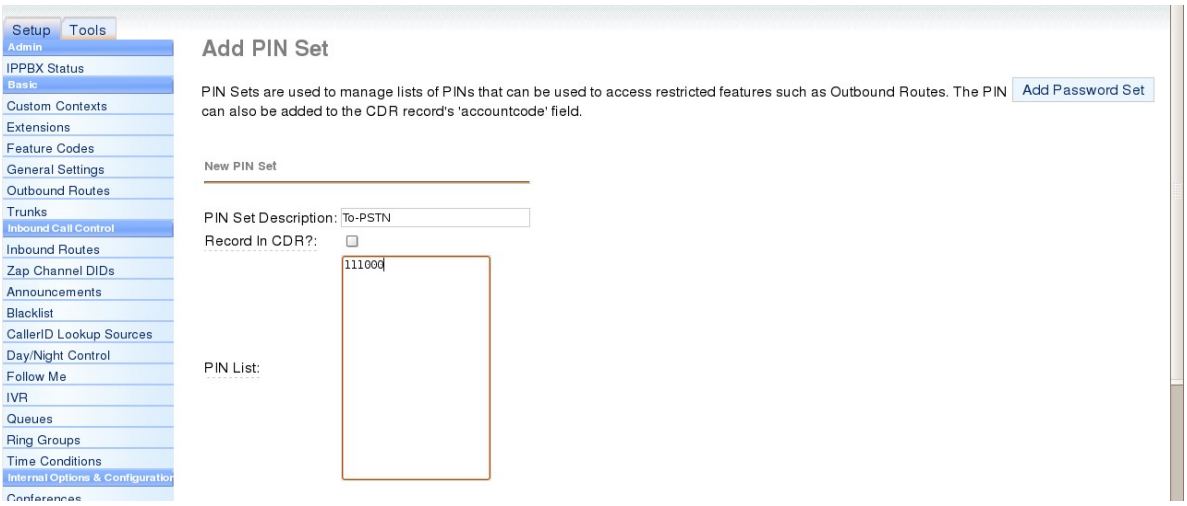


Figure 7.33: Setting Add PIN Set

The following menu are configuration menu for PIN Sets.

PIN Set Description : description of the name of PIN
Record In CDR : choose this if you want to have the PIN entered into Call Detail Record whenever the PIN is used
PIN List : password to be used

CHAPTER 8: OpenSIPS High Performance Softswitch

In this chapter, we will discuss on OpenSIPS. OpenSIPS (Open SIP Server) is a mature Open Source implementation of a SIP server. OpenSIPS is more than a SIP proxy/router as it includes application-level functionalities. OpenSIPS, as a SIP server, is the core component of any SIP-based VoIP solution. With a very flexible and customizable routing engine, OpenSIPS 'unifies voice, video, IM and presence services in a highly efficient way, thanks to its scalable (modular) design.

What OpenSIPS has to offer, comes in a reliable and high-performance flavour - OpenSIPS is one of the fastest SIP servers, with a throughput that confirms it as a solution up to enterprise or carrier-grade class. It performs much better than that of Asterisk. However, it lacks of feature that rich in Asterisk. Thus, in reality, it would be beneficial to use both Asterisk and OpenSIPS.

Compile OpenSIPS

Prepare the supporting software. In Ubuntu 9.10 and Ubuntu 10.04, it can be prepared by using the following command.

```
# apt-get install flex bison gcc make libperl5.10 libperl-dev libxmlrpc-c3 libxmlrpc-c3-dev \
unixodbc unixodbc-dev libradiusclient-ng2 libradiusclient-ng-dev libxml2 openssl libsctp1 \
libsctp-dev libexpat1 libexpat1-dev libldap-2.4-2 libldap2-dev libsnmp15 libsnmp-dev \
libconfuse0 libconfuse-dev libmysqlclient16 libmysqlclient-dev mysql-client-5.1 mysql-server \
zlib1g zlib1g-dev libmysql++3 libmysql++-dev libpcre3 libpcre3-dbg libpcre3-dev
```

In Ubuntu 10.10, it can be done as follows

```
# apt-get install flex bison gcc make libperl5.10 libperl-dev libxmlrpc-c3 libxmlrpc-c3-dev \
unixodbc unixodbc-dev libradiusclient-ng2 libradiusclient-ng-dev libxml2 openssl libsctp1 \
libsctp-dev libexpat1 libexpat1-dev libldap-2.4-2 libldap2-dev libsnmp15 libsnmp-dev \
libconfuse0 libconfuse-dev libmysqlclient-dev mysql-client-5.1 mysql-server zlib1g zlib1g-dev \
libmysql++3 libmysql++-dev libpcre3 libpcre3-dbg libpcre3-dev
```

Get source code of OpenSIPS, such as, opensips-XXX-tls_src.tar.gz ,from

http://opensips.org/pub/opensips/
http://www.opensips.org/index.php?n=Resources.Downloads#osippub
http://www.opensips.org/index.php?n=Resources.Downloads#osipsf

If we would like to use opensips with TLS, we need to do the followings.

```
$ sudo su -
# cp opensips-1.6.4-2-tls_src.tar.gz /usr/local/src/
# cd /usr/local/src/
# tar zxvf opensips-1.6.4-2-tls_src.tar.gz
# cd opensips-1.6.4-2-tls
```

Compile and install the following modules, i.e., "acc", "mysql", "textops", "sl", "db_mysql" and "enum" using the following command,

```
# cd opensips-1.6.4-2-tls
# make all && make include_modules="acc mysql textops sl enum db_mysql" modules
# make install
```

It seems, we need to copy some scripts to /usr/local/src/opensips/opensipsctl

```
# cp -Rf /usr/local/src/opensips-1.6.4-2-tls/scripts/* /usr/local/lib/opensips/opensipsctl
```

That's it. OpenSIPS is compiled and install and ready to use. OpenSIPS configuration file is located at

```
/usr/local/etc/opensips
```

Check for any problem in the configuration file can be done using the following command,

```
# opensips -c -f /usr/local/etc/opensips/opensips.cfg
```

To Run opensips, put in /etc/rc.local

```
opensips -f /usr/local/etc/opensips/opensips.cfg
```

Please note we remove -c switch

Prepare User Database Server

OpenSIPS uses database server, such as, MySQL for handling user registration. Install MySQL Server

and make sure it works using the following command,

```
# apt-get install mysql-server libmysqlclient-dev mysql-client-5.0
# /etc/init.d/mysql restart
```

To setup the database server, we need to edit /usr/local/etc/opensips/opensipsctlrc or /etc/opensips/opensipsctlrc, such as,

```
# vi /usr/local/etc/opensips/opensipsctlrc
```

or

```
 vi /etc/opensips/opensipsctlrc
```

Make sure,

```
DBENGINE=MYSQL
DBHOST=localhost
DBNAME=opensips
DBRWUSER=opensips
DBRWPW="opensipsrw"
DBROUSER=opensipsro
DBROPW=opensipsro
DBROOTUSER="root"
```

Copy scripts to /usr/local/lib/opensips/opensipsctl

```
# cp -Rf /usr/local/src/opensips-1.6.4-2-tls/scripts/* /usr/local/lib/opensips/opensipsctl/
```

Initialized the user database using opensipsdbctl command as follow,

```
# cd /usr/local/lib/opensips/opensipsctl
# opensipsdbctl create
```

Follow the following commad

```
MySQL password for root:  <enter MySQL root password>
INFO: test server charset
INFO: creating database opensips ...
INFO: Core OpenSIPS tables succesfully created.
```

```
Install presence related tables? (y/n): <y>

INFO: creating presence tables into opensips ...
INFO: Presence tables succesfully created.

Install tables for imc cpl siptrace domainpolicy carrierroute userblacklist? (y/n): <y>

INFO: creating extra tables into opensips ...
INFO: Extra tables succesfully created.
```

Use opensipsctl

Opensipsctl is a usefull tool provided by OpenSIPS, that can be used for,

- Adding users.
- Check who's online.
- Monitoring opensips activities.

To add user, for example, we can use

```
# opensipsctl add number@host password
# opensipsctl add 2000@192.168.0.3 123456
```

To see who's online, for example,

```
# opensipsctl ul show number@host
# opensipsctl ul show 2000@192.168.0.3
```

After opensips is running, to monitor OpenSIPS Softswitch activities

```
# opensipsctl monitor
```

Some Routing Technique in OpenSIPS

In the following sections, we will discuss how to route traffic to

- PSTN and Cellullar network
- “Area Code” for several interconnected SIP Servers
- ENUM network

How to route to PSTN and Cellular

Basically, we need an Analog Telephone Adapter (ATA) to interconnect a VoIP network to PSTN or Cellular network. In this example, we assume

- ATA is located at IP address 192.168.0.200
- ATA is using port 5061
- Area code for PSTN is 021
- Area code for Cellullar is 08

We need to add to the opensips configuration file

```
/usr/local/etc/opensips/opensips.cfg
```

For example, to be able to use the ATA (at 192.168.0.200:5061) to call PSTN from all host / domain

```
# attempt handoff to PSTN
if (uri=~"^sip:021[0-9]*@*") {
    rewritehostport( "192.168.0.200:5061");  ##  192.168.0.200:5061 is the ATA
    route(1);
   };
```

To restrict the call to PSTN only from mydomain.com

```
# attempt handoff to PSTN
if (uri=~"^sip:021[0-9]*@mydomain.com") {     ##  caller registered to mydomain.com
    rewritehostport( "192.168.0.200:5061");  ##  192.168.0.200:5061 is ATA
    route(1);
   };
```

To be able to use the ATA to call Cellullar from all host / domain

```
# attempt handoff to cellullar
if (uri=~"^sip:08[0-9]*@*") {
```

```
    rewritehostport( "192.168.0.200:5061");  ##  192.168.0.200:5061 is ATA
    route(1);
    };
```

To restrict the call to Cellular only from mydomain.com

```
# attempt handoff to cellullar
if (uri=~"^sip:08[0-9]*@mydomain.com") {      ##  caller registered to mydomain.com
    rewritehostport( "192.168.0.200:5061");  ##  192.168.0.200:5061 is ATA
    route(1);
    };
```

How to route using Area Code for interconnected SIP Servers

For example we have several SIP Servers in our network, such as,

“Area Code”	SIP Server IP Address
“021”	203.159.31.99
“022”	203.159.31.123
“023”	203.159.31.48

The dialplan for OpenSIPS would be something like,

```
if (uri=~"^sip:021[0-9]*@*") {
    strip(3);
    rewritehostport( "203.159.31.99:5060");
    route(1);
   };
if (uri=~"^sip:022[0-9]*@*") {
    strip(3);
    rewritehostport( "203.159.31.123:5060");
    route(1);
   };
if (uri=~"^sip:023[0-9]*@*") {
    strip(3);
```

```
rewritehostport( "203.159.31.48:5060");
route(1);
};
```

How to route ENUM Query in OpenSIPS

Steps to route ENUM query in OpenSIPS is as follows,

- Prepare ENUM modul in OpenSIPS configuration
- Create routing table for ENUM

ENUM query in OpenSIPS is basically transform the URI address from ENUM to URI SIP. Call process is normally done using the URI SIP.

To prepare the ENUM module in OpenSIPS configuration, we need to edit /usr/local/etc/opensips/opensips.cfg or /etc/opensips/opensips.cfg

```
# vi /usr/local/etc/opensips/opensips.cfg
```

Enter the following command

```
loadmodule "enum.so"
modparam("enum", "domain_suffix", "e164.arpa.")
modparam("enum", "i_enum_suffix", "e164.arpa.")
```

We can change e164.arpa to other ENUM top level domain, such as, e164.id or e164.th.

Test ENUM Query in OpenSIP

Assuming:

- An Asterisk Server Running at 192.168.0.2
- Echo test ready at number 600
- ENUM Server is ready to resolve ENUM Query for e164.id.
- Data in ENUM Server ready to map +62555666666600 to 600@192.168.0.2

Test test routing table would be

```
rewriteuri("sip:62555666666600@192.168.0.2");
prefix("+");
enum_query("e164.id.");
route(1);

route[1] {
    # send it out now; use stateful forwarding as it works reliably
    # even for UDP2TCP
    if (!t_relay()) {
            sl_reply_error();
     };
    exit;
}
```

ENUM Routing Table in OpenSIPS configuration

The short version

```
if (uri=~"^sip:00[1-9][0-9]*@*") {
   strip(2);
   prefix("+");
};

if (uri=~"sip:\+[0-9]+@*")
       enum_query("e164.id.");
```

The above example will allow all client from all server to access our ENUM query routing. A more complete version of ENUM query may be as follows,

```
# Somewhere in the route[x] section:

# if you want to make ENUM work with numbers starting with "00",
# use the following to convert "00" it into a "+"

if (uri=~"^sip:00[1-9][0-9]*@example\.net") {
```

```
    # strip leading "00"
    # (change example.net to your domainname or skip the stuff after the "@")
    strip(2);
    # (adjust, if your international prefix is something else than "00")
    prefix("+");
};

# check if request uri starts with an international phone
# number (+X.), if yes, try to ENUM resolve in e164.arpa.
# if no result, try in nrenum.net

if (uri=~"sip:\+[0-9]+@example\.net") {
    # (change example.net to your domainname or skip the stuff after the "@")
    if ( !enum_query("e164.arpa.") ) {
        enum_query("nrenum.net.");
    };
};
```

Another alternative that may be extended is as follows,

```
# is this an ENUM destination (leading +?)
if (method=="INVITE" && uri=~"sip:\+[0-9]+ at iptel\.org") {
        if (!enum_query("voice"))       # if parameter empty, it defaults to "e2u+sip"
          enum_query("");             # E2U+sip
    }
```

Yet another alternative that can be tried / expanded is as follows,

```
if (is_from_user_enum()) {
   enum_query("");
    }
```

CHAPTER 9: ENUM

ENUM is basically a mapping mechanism to map Telco number, such as, +628113334567 or +62555334567, to a number recognize in VoIP network such as, 20333@voiprakyat.or.id or 5007987@fwd.pulver.com. Thus, in principle, ENUM is merely a table.

ENUM is not limited to mapping only. ENUM recognize prioritizing. For example, a phone number +6255534567 may have several client with priority, such as,

+6255534567	priority 1	245678@voiprakyat.or.id
+6255534567	priority 2	6543686@fwd.pulver.com
+6255534567	priority 3	+62215678976 (nomor kantor)
+6255534567	priority 4	+62856789654 (nomor handphone)
+6255534567	priority 5	mail:oknum@salemba.co.id

The actual writing of ENUM on the Internet, for example using the top level domain e164.id, is as follows,

+6255512345678	8.7.6.5.4.3.2.1.5.5.5.2.6.e164.id
+6281812345678	8.7.6.5.4.3.2.1.8.1.8.2.6.e164.id

Please note that the ENUM number is reversed as oppose to the known normal phone number.

Example of ENUM Service

One of the best example of ENUM Service is the e164.org, we can register and has a good authentication mechanism by calling our phone number before it maps to their database.

e164.org not the only ENUM Server. Friends at VoIP Rakyat in Indonesia creates their own ENUM server located at http://www.enum.voiprakyat.or.id.

Delegation Concept in ENUM

We hope as we are ready each country will have their very own ENUM Server and receive a delegation from e164.arpa. For Indonesia, it would be 2.6.e164.arpa for handling country code (+62).

To understand how ENUM works, one needs to understand how a Domain Name System (DNS) works as ENUM uses DNS Server. Thus, ENUM works fairly similar to DNS but to map and to delegate a phone number. Please note that ENUM is different from a SIP Server.

Imagine at national level there is an allocation of area code for SIP network on +62555. It can be mapped to ENUM under the domain, for example, 5.5.5.2.6.e164.id. It may have several ENUM Name Server (NS) such as,

```
ENUM Server Domain 5.5.5.2.6.e164.id
+62555      ENUM NS    202.123.123.124
+62555      ENUM NS    235.123.123.234
```

Please note that at national level, the ENUM Server may not have a complete information on the subscribers.

For example, a community or a corporate or a telecomunication operator, assigned 4444 area code for its network, such that, it may use

+6255544440000 - +6255544449999

basically, it may allocate phone number for 10.000 subscribers. Thus, the community may run their own ENUM server under the sub-domain 4.4.4.4.5.5.5.2.6.e164.id, for example

```
ENUM Server Domain 4.4.4.4.5.5.5.2.6.e164.id
+62555444   ENUMNS     212.234.234.234
+62555444   ENUMNS     212.234.234.235
```

In the delegation process, the NS information of ENUM 4.4.4.4.5.5.5.2.6.e164.id must be written in ENUM 5.5.5.2.6.e164.id that tells

```
4.4.4.4.5.5.5.2.6.e164.id     IN NS  212.234.234.234
4.4.4.4.5.5.5.2.6.e164.id     IN NS  212.234.234.235
```

ENUM concept is not confined to operator, any corporate or community with smaller number of extensions, e.g., 100 extentions may use, for example, ENUM allocation for,

+6255566666600 - +6255566666699

Thus, this particular corporate must has its own ENUM server or collocate to other ENUM server to

handle 6.6.6.6.6.6.5.5.5.2.6.e164.id, such as

```
NUM Server Domain 6.6.6.6.6.6.5.5.5.2.6.e164.id
62555666666  ENUMNS      212.234.234.4
62555666666  ENUMNS      212.234.234.5
```

Delegation process for NS of 6.6.6.6.6.6.5.5.5.2.6.e164.id must be entered into the main ENUM Server for 5.5.5.2.6.e164.id tto tell

```
6.6.6.6.6.6.5.5.5.2.6.e164.id   IN NS  212.234.234.4
6.6.6.6.6.6.5.5.5.2.6.e164.id   IN NS  212.234.234.5
```

The ENUM delegation concept is clearly shown that is not limited to operator. Any entities may have their very own phone number. Thus, a more comprehensive authentication process may be needed to make sure the phone number is properly delegated.

ENUM Implementation

ENUM Server is principally a DNS Server. Thus, if one has a DNS Server, one may readily run an ENUM Server. To Install an ENUM Sever, one need to,

- Install a DNS Server. In Linux, we normally use BIND for DNS server.
- Add our ENUM allocation in /etc/named.conf.local.
- Include our ENUM data into the database file mentioned in /etc/named.conf.local.

BIND Installation

Install BIND as follows,

```
apt-get install dnsutils bind9
```

Setup BIND for ENUM Server

For example, we are assigned for +625XXXX. We need to edit,

/etc/bind/named.conf.local

Entry for domain 5.2.6.e164.id

```
zone "5.2.6.e164.id" IN {
        type master;
        file "/etc/bind/5.2.6.e164.id.db";
};
```

All subscriber numbers must be listed in /etc/bind/5.2.6.e164.id.db. An example of the DNS file of /etc/bind/5.2.6.e164.id.db is as follows,

```
$TTL    86400
@       IN SOA  ns.warnet.co.id admin.warnet.co.id. (
                        42              ; serial (d. adams)
                        3H              ; refresh
                        15M             ; retry
                        1W              ; expiry
                        1D )            ; minimum

        IN NS  ns.warnet.co.id.

0.0.0.2 NAPTR 10 100 "u" "E2U+sip" "!^.*$!sip:2000@192.168.0.3!" .
1.0.0.2 NAPTR 10 100 "u" "E2U+sip" "!^.*$!sip:2001@192.168.0.3!" .
2.0.0.2 NAPTR 10 100 "u" "E2U+sip" "!^.*$!sip:2002@192.168.0.3!" .
3.0.0.2 NAPTR 10 100 "u" "E2U+sip" "!^.*$!sip:2003@192.168.0.3!" .
4.0.0.2 NAPTR 10 100 "u" "E2U+sip" "!^.*$!sip:2004@192.168.0.3!" .
5.0.0.2 NAPTR 10 100 "u" "E2U+sip" "!^.*$!sip:2005@192.168.0.3!" .
0.2.0.2 NAPTR 10 100 "u" "E2U+sip" "!^.*$!sip:2020@192.168.0.3!" .
1.2.0.2 NAPTR 10 100 "u" "E2U+sip" "!^.*$!sip:2021@192.168.0.3!" .
2.2.0.2 NAPTR 10 100 "u" "E2U+sip" "!^.*$!sip:2022@192.168.0.3!" .
0.3.0.2 NAPTR 10 100 "u" "E2U+sip" "!^.*$!sip:2030@192.168.0.3!" .
1.3.0.2 NAPTR 10 100 "u" "E2U+sip" "!^.*$!sip:2031@192.168.0.3!" .
2.3.0.2 NAPTR 10 100 "u" "E2U+sip" "!^.*$!sip:2032@192.168.0.3!" .
3.3.0.2 NAPTR 10 100 "u" "E2U+sip" "!^.*$!sip:2033@192.168.0.3!" .
0.5.0.2 NAPTR 10 100 "u" "E2U+sip" "!^.*$!sip:2050@192.168.0.3!" .
1.5.0.2 NAPTR 10 100 "u" "E2U+sip" "!^.*$!sip:2051@192.168.0.3!" .
```

For example, it means the mapping numbers is as follows,

```
+6252000      0.0.0.2.5.2.6.e164.id   2000@192.168.0.3
+6252001      1.0.0.2.5.2.6.e164.id   2001@192.168.0.3
+6252002      2.0.0.2.5.2.6.e164.id   2002@192.168.0.3
```

After the editing process, please restar the DNS Server using the command

```
# /etc/init.d/bind9 restart
```

Test DNS for ENUM Query

We can use the dig command on the localhost of the DNS server to query the ENUM entries, for example,

```
$ dig NAPTR 0.0.0.2.5.2.6.e164.id @127.0.0.1
```

The output would be approximately

```
; <<>> DiG 9.6.1-P1 <<>> NAPTR 0.0.0.2.5.2.6.e164.id @127.0.0.1
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10744
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;0.0.0.2.5.2.6.e164.id.          IN      NAPTR

;; ANSWER SECTION:
0.0.0.2.5.2.6.e164.id.  86400  IN      NAPTR         10 100 "u" "E2U+sip" "!^.*$!
sip:2000@192.168.0.3!" .

;; AUTHORITY SECTION:
5.2.6.e164.id.          86400  IN      NS      ns.warnet.co.id.

;; ADDITIONAL SECTION:
ns.warnet.co.id.        86335  IN      A       76.163.126.2

;; Query time: 0 msec
```

```
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue Nov 24 08:12:11 2009
;; MSG SIZE  rcvd: 137
```

ENUM Delegation in BIND

In a country, for example, Bhutan (+975) may have ENUM top level domain

2.5.7.9.e164.bt

for example, running at a machine 203.159.31.41.

For example, we want to allocate phone numbers different organization, such as,

Organization	Number	ENUM	Server IP
Ministry of Eduction	+975 2 123 XXX	3.2.1.2.5.7.9.e164.bt	203.159.31.100
University A	+975 3 544 XXX	4.4.5.3.5.7.9.e164.bt	202.154.1.10
High School B	+975 3 342 XXX	2.4.3.3.5.7.9.e164.bt	222.119.6.45
Village C	+975 5 768 XXX	8.6.7.5.5.7.9.e164.bt	231.167.31.20
Dzong D	+975 7 243 XXX	3.4.2.7.5.7.9.e164.bt	204.19.1.5

At the ENUM Country Level DNS, in this example 203.159.31.41, the delegation process is done by configuring file /etc/bind/named.conf.local such that

```
zone "3.2.1.2.5.7.9.e164.bt" {
        type slave;
        masters {
                203.159.31.100;
                };
        file "/var/lib/bind/3.2.1.2.5.7.9.e164.bt.hosts";
        };
```

```
zone "4.4.5.3.5.7.9.e164.bt" {
        type slave;
        masters {
                202.154.1.10;
                };
        file "/var/lib/bind/4.4.5.3.5.7.9.e164.bt.hosts";
        };

zone "2.4.3.3.5.7.9.e164.bt" {
        type slave;
        masters {
                222.119.6.45;
                };
        file "/var/lib/bind/2.4.3.3.5.7.9.e164.bt.hosts";
        };

zone "8.6.7.5.5.7.9.e164.bt" {
        type slave;
        masters {
                231.167.31.20;
                };
        file "/var/lib/bind/8.6.7.5.5.7.9.e164.bt.hosts";
        };

zone "3.4.2.7.5.7.9.e164.bt" {
        type slave;
        masters {
                204.19.1.5;
                };
        file "/var/lib/bind/3.4.2.7.5.7.9.e164.bt.hosts";
        };
```

Where on 203.159.31.100, 202.154.1.10, 222.119.6.45, 231.167.31.20 and 204.19.1.5, we must run master DNS server for each respective ENUM.

CHAPTER 10: Conference Server on Asterisk

Establishing conference calls in PABX is complicated, as we have to configure them properly. Unlike Asterisk Conference feature, commercial conferencing facilities are easy to use but they can be exorbitant. So it is fortunate for us to have Asterisk as something to help us establish not only inexpensive ways to conferencing but also easier. Basically, there are two steps necessary for configuring conferencing in Asterisk:

- Make conference "room"
- Add "room" to dialplan.

Configuring Conference Room MeetMe

Asterisk's conference room is set up in /etc/asterisk/meetme.conf. All conference rooms should be listed under header [rooms]. The syntax used to configure conference room is:

```
conf => conference_number[,pin][,adminpin]
```

The following is an example of a conference room setup with the number 2500 and PIN 1234:

```
[rooms]
conf => 2500,1234
```

If we set up more than one room, it is recommended that you label them so as to remember which number for which conference.

```
[rooms]
; marketing team
conf => 2500,1234

; publisher team
conf => 2501,5678
```

Since being an administrator does not provide us with authorization that could use the features otherwise available when we use Asterisk for something much more complicated, it is not necessary to set the PIN.

There are several options that can be used, such as:

m - caller can listen but not speak.
t - caller can speak but cannot hear.
p - caller can get out of the Conference by pressing the # key.

There are two additional options that have not been implemented:

s - Asterisk provides menu to the user if * is pressed.
a – give the user administrator privileges on a conference.

Configuring Dialplan for Conference

To configure this dialplan, we have to edit file/etc/asterisk/extensions.conf and include the conference room we wish to have in it. This can be done by using a different context for each room, for example:

```
[marketing_team_conference_room]
exten => 300,1,MeetMe,2500(1234)

[publisher_team_conference_room]
exten => 301,1,MeetMe,2501(5678)
```

So the callers need only to call the phone number 300 or 301, enter the PIN when asked, and they will go into conference. MeetMe will ring a bell to inform those already in conference that there is someone entering into the conference.

We can certainly gather all the "room" conferences within a context. Or add a "room" to an existing context, with the following command.

```
[local-users]
exten => 250,1,Dial(SIP/alrac,10,r)
exten => 250,2,VoiceMail(u250@local-vm-users)
exten => 250,dial+101,VoiceMail(b250@local-vm-users)

exten => 300,1,MeetMe,2500(1234)
```

Or add a comprehensive context

```
include => marketing_team_conference_room
```

If the established conference gives the callers the opportunity to listen to speeches from the Boss without interrupting the speech, then we have to do the following:

```
[marketing_team_conference_room]
exten => 300,1,MeetMe,2500|m|1234
```

New callers who have joined the conference can find out how many people in the conference use MeetMeCount applications, by executing the following command:

```
[marketing_team_conference_room]
exten => 300,1,Playback(there_are)
exten => 300,2,MeetMeCount,2500
exten => 300,3,Playback(callers)
exten => 300,4,MeetMe,2500
```

Of course you need to save two sound files that somewhat reads "There are" and "Callers present in the conference". After editing extensions.conf, do not forget to reload the new configuration. In order to prevent anomalies encountered during operation, we can run asterisk console and execute the following command:

```
# asterisk -r
asterisk1*CLI> extensions reload
```

Activating Conference while Operating

One of needs that may arise is the setting of a conference room at anytime during operating VoIP. For this purpose, it is not necessary for us to change the content of extensions.conf file. However we need to set up a conference that its general context in the extensions.conf file and meetme.conf:

```
;meetme.conf
[rooms]
;general-purpose conference room
conf => 3500,1234

;extensions.conf
;generic conference room
```

```
[gen_conference]
```

If we need establish a new conference, we can immediately make it through CLI, with the following command:

```
localhost*CLI> add extension 400,1,Dial,3500 into gen_conference
Extension '400,1,Dial,3500' added into 'gen_conference' context
```

Here extension 400 will be added with priority 1 to gen_conference. Of course, this extension will disappear if we restart the asterisk, or we can delete it through the following command:

```
localhost*CLI> remove extension 400@gen_conference
Whole extension 400@gen_conference removed
```

In multi-line extensions, we can omit a single line or command by giving a priority, for example

```
localhost*CLI>remove extension 400@gen_conference 2
Extension 400@gen_conference with priority 2 removed
```

CHAPTER 11: Trunk Peering in Asterisk

One of the main reasons why we use VoIP is to have free long distance or international calls. Imagine that if you're having a branch office or working partners who often communicate with you. You have to establish a private network between office branches or those working partners so you can bypass the PSTN. There are a number ways you can do this through Asterisk.

- DUNDi, Distributed Universal Number Discovery protocol.
- Centralized directory, such as VoIP Rakyat

On this occasion, you will be shown a trunk peering process using VoIP Rakyat. The same mechanism can be applied to other SIP proxy across the world.

In addition, we will also discuss the real problems we face in configuring network involving NAT/Proxy Server, as most networks are protected by firewall that blocks VoIP signal.

We presume that we already have an account in VoIP Rakyat. In this sense, the given number and password are:

number 2012345 password abcdef
number 2055555 password 123456

Next we will do a comprehensive configuration of file sip.conf and extensions.conf, including providing the facilities required for testing.

In general, there are several important things in configuring trunk in Asterisk

- Registration to SIP account in voiprakyat (sip.conf)
- Creating username & password for various extensions (sip.conf)
- Configuring Dialout for a variety of configurations (extensions.conf)
- Configuration for inbound call (extensions.conf [inbound-sip])

With this configuration, we can now place outgoing calls using various available lines. In addition, we can also receive calls dialed from voiprakyat and the internet through inbound-sip module. The detail of each of a variety of configurations is available in the enclosed configuration.

CHAPTER 12: NAT and Firewall

Network Address Translation (NAT) or better known as proxy server is often troublesome for SIP configuration. The resulting problem usually found is either poor VoIP connectivity or no connectivity at all as the connectivity is blocked by a Firewall/ NAT. When something like this occurs, you have to do the following:

1. ensure that an antivirus software or firewall active in your PC is not blocking VoIP
2. Contact your network administrator and tell him or her to enable NAT for you or give you a Public IP, and open the following ports:

For SIP:

UDP port 3478 and 3479 (STUN, NAT type discovery)
UDP port 5060 and 5066 (SIP, signaling)
UDP port 8000 to 20000 (RTP, data, voice + video)

For IAX2:

UDP port 4569 (IAX2, signaling and data)

Asterisk may act as a SIP client or SIP server. To make it a client, do the following command:

register =>

in part of[general] in sip.conf.

We may have Asterisk function as a SIP server, by configuring the username, password, and other details of the SIP client that will be registered to Asterisk in sip.conf.

There are some scenarios of SIP NAT network channels:

1. Asterisk SIP client behind the NAT, with the client connected to a SIP proxy outside the NAT
2. Asterisk SIP client behind the NAT, with the client connected to a SIP proxy within the NAT
3. Asterisk SIP server behind the NAT, with the client outside the NAT connected to Asterisk
4. Asterisk SIP server behind the NAT, with the client within the NAT connected to Asterisk
5. Asterisk SIP client outside the NAT, with the client connected to SIP proxy outside the NAT
6. Asterisk SIP client outside the NAT, with the client connected to SIP proxy within the NAT
7. Asterisk SIP server outside the NAT, with the client outside the NAT connected to Asterisk
8. Asterisk SIP server outside the NAT, with the NAT client connected to Asterisk

In general, the setup can work with the existing configuration, of course, depending on the configuration of the client, NAT, server and many other factors, especially the firewall configuration. Of those setups, number 3 and 6 are difficult to do because SIP is a peer-to-peer protocol and most NATs allow only clients inside their network to connect to a server located outside but not vice versa.

1. Running with a proxy server that supports NAT
2. Running with no NAT in between
3. Running by doing port forwarding in the NAT/ proxy server
4. Running with no NAT in between
5. Running with no NAT in between
6. Running by doing port forwarding on the NAT/ Proxy server
7. Running with no NAT in between
8. Running with configuration nat=yes and qualify=xxx in sip.conf. Some clients using X-Like use STUN and send UDP keep-alive packets. Qualify will send a keep-alive packets from Asterisk to any client in the NAT

However, the worst case occurs when Asterisk is within NAT and the client is within different NAT. For this, we need to have a mediator that could see both ways simultaneously. To channel voice data or streaming, we need a media server. Asterisk which is placed outside the NAT would be able to function as a media server, and we can also add the feature to do Codec conversion.

CHAPTER 13: Voicemail in Asterisk

Voicemail is used in Asterisk to leave a message, when there is no one to receive the incoming call. Voicemail configuration for Asterisk is available in voicemail.conf file available in /etc/asterisk/. We can create a mailbox within the default mailbox context, either use the one already available or create another in different context. It is important to understand that the context in voicemail mailbox is not related to the context in extensions.conf

The command to create a voicemail mailbox is approximately as follows

```
mailbox_number => password, name, e-mail
```

mailbox_number is the number used in extension.conf for VoiceMail() command and for registering a user to sip.conf or iax.conf. Password is the password used to register a user to sip.conf or iax.conf. Name is the name associated with a mailbox, and email is the email used to inform if there is an incoming voicemail.

Sample of voicemail.conf content is as the following:

```
[mb_tutorial]
777 => 1212, ivan, ivan@voiprakyat.or.id
```

Here the mailbox context is created using the name mb_tutorial context, mailbox number 777 with password 1212, which is owned by ivan whose e-mail is ivan@voiprakyat.or.id. For a user who calls Ivan at extension 1234 to be able to leave a message in Ivan's mailbox, the following command can be used:

```
Voicemail (mailbox_number@context)
```

In extensions.conf file, the content is as the following:

```
exten => 1234.1, Dial (SIP/ivan, 30)
exten => 1234.2, VoiceMail (777@mb_tutorial)
exten => 1234.3, PlayBack(vm-goodbye)
exten => 1234.4, HangUp ()
```

In the example above, Asterisk will attempt to call SIP user ivan for extension 1234 and wait for 30 seconds. If nobody answers the phone, Asterisk will carry out the next priority, that is, Asterisk will

open mailbox 777 in mb_tutorial context. Once the caller has left a message, Asterisk will carry out playback (rewind the message) and hang up the call. Playback (vm-goodbye) will execute vm-goodbye file that should be available in /var/lib/asterisk/sounds/.

The Voicemail message is recorded in

/var/spool/asterisk/voicemail/<context>/<mailbox>/INBOX/

Therefore the full path to Ivan is

/var/spool/asterisk/voicemail/mb_tutorial/777/INBOX/.

To listen to the message stored in the mailbox, we can place a call by using VoiceMailMain command in Asterisk. The command is as follows:

VoiceMailMain(mailbox@context)

In the default configuration of Asterisk, if the sample configuration remains as what is, VoiceMailMain can be contacted using the number 8500.

The configuration sample of extensions.conf for accessing VoiceMailMain is:

exten => 9999.1, VoiceMailMain (777@mb_tutorial)

By dialing 9999, we will be able to go into mailbox 777, of course after we entered the correct password for this mailbox, which is 1212.

Various options are available when accessing mailboxes using VoiceMailMain:

0 Mailbox options
- 1 Record unavailable message
- 2 Record busy message
- 3 Record our name
- 4 Change our password
- * Back to main menu

1 Listen to old messages
2 Change folders
3 Advanced options
- 1 Send reply

2 Call back
3 Envelope
4 Outgoing call
5 Leave message
* Back to main menu

4 Play previous message
5 Repeat message
6 Play next message
7 Delete this message
8 Forward message to another mailbox
9 Save message in a folder
* Help; during message playback: Rewind
# Exit; during message playback: FastForward

When we listen to a voicemail message recording, we can use the following buttons to navigate, ie,

* to rewind (going back)
# to FastForward (forward)

Note: the '#' and '*' buttons work only when the message is in the process of playback.

CHAPTER 14: More on Asterisk's Dialplan

One of the most difficult parts in configuring a telephone exchange is the configuration of dialplan. Asterisk dialplan configuration is in the extensions.conf file, which is usually located in /etc/asterisk/extensions.conf. Dial controls what needs to be carried out when there is incoming or outgoing call. In other words, dialplan controls the pattern of calls in our softswitch.

Extensions.conf is configured based on a number of modules containing definitions or static parameter settings. These modules are also known as context, defined by the system administrator. In context, there two parts, general and global. The former, available in the the top part of extensions.conf, allows us to set a number of main configurations for extensions in Asterisk. The latter allows us to define a number of variables of global constants and initialize the values. Once these parts of context are set, the rest of content of extensions.conf file is taken by the Dialplan. So Dialplan consists of contexts, with every context consisting of extensions.

In addition, there is also Macro, a special type of context, labeled by the name defined by the user who typically uses macro-prefix. Macro can be executed repeatedly, behaving similar to the subroutine in programming language.

Each section in extensions.conf begins with a name written in square brackets, so as to make extensions.conf's structure similar to .ini file in Windows.

Pattern Extension

When we define extensions in a context, not only can we use ordinary numbers, names or letters, but we can also define the extensions that match a set of numbers dialed using extension pattern.

Attaching context

A context containing extensions can be incorporated into or associated with others. For example, consider the following context:

Context "default":

Extension	Note
101	Mark Spencer
102	Will Meadows

```
        0                       Operator

    Context "local":
        Extension               Note
        _9NXXXXXX               Local calls
        include => "default"

    Context "longdistance":
        Extension               Note
        _91NXXNXXXXXX Long distance calls
        include => "local"
```

We have defined three extensions:

- Operator. Default context allows us to dial 3 telephone extensions: Mark, Will and the operator
- Local context has only one extension that allows us to dial 7-digit number. In addition, if we incorporate the default context into the local one, we can also dial Mark, Will and the operator.
- The long distance context has an extension pattern allowing us to place a long distance call. This context also includes localcontext, and thus also allows us to call a local number or even the extension of Mark, Wil and the operator.

Using context in the extension, we can carefully regulate who can have access to a larger network. Be careful. If there is more than one pattern that match the dialed number, Asterisk may not use the numbers we want.

The Extension Pattern

When Asterisk accepts incoming connection through a channel, Asterisk will see the context defined for such a channel to see what commands that need to be carried out by Asterisk. Context will define a set of commands depending on the extension called by the user. For example, a given context may give a set of commands if a user calls the number "123", and a set of other commands if the user dials "9". We can also create another set of commands if the user calls number beginning with "555”.

If there are incoming connections - such as that coming from phone line outside – it implies that the user has not dialed an extension. In this case, Asterisk will act as a user dialing a particular extension called "s" (originating from the word “Start”). Asterisk will look for the extension "number" s in the definition of context for the channel and look for instructions that needs to be carried out for the "s" extension.

For example, we have a channel "Zap/1", which is connected to a telephone in an office. For example, in the Zap channel configuration (zapata.conf) we have defined context=john for Zap channel 1. Therefore, if we use a handset to dial a number, Asterisk will look for context with the name "john" in extensions.conf to see what has to be done. We can start a context by writing the name in square brackets

```
[john]
```

For each context, we can define one or more extensions that can be used by Asterisk to compare the numbers to be dialed. For each extension, we can tell Asterisk what needs to be done though a set of commands.

Extension

An extension can be a series of numbers or a pattern. Extension can be a series of number, like 123, and may also contain some standard symbols * and #, which are available on the phone keypad. So 34#76 is a valid extension number. Some keypads are labeled A, B, C, and D. Because of this, extension can also be defined based on letters. So basically an extension can be defined using both letters and numbers. Keep in note that there are many VoIP phones that can call extension numbers consisting of text Sembang, like "Office". Therefore it is not a problem to define such an extension name in Asterisk.

Are extension names case sensitive? Yes and no. Extension case are sensitive because when Asterisk attempts to match the extension dialed by a user to extension that is defined in context, the extension name should be precisely matched, including uppercase letters and small. Therefore, if a user calls extension "OFFICE" through their VoIP phone, Asterisk will not immediately run the commands we define for extension "Office". But in reality, extension names are not case sensitive in the sense that we cannot define different extensions based only on uppercase/ lowercase letters. It means we do not define the command for extension "Office" and "OFFICE" in a context.

Predefined Extension Names

Asterisk defines a number of extension names for specific needs. These extensions are:

```
i       : Invalid
s       : Start
```

h : Hangup
t : Timeout
T : AbsoluteTimeout
o : Operator

and many more.

Defining Extension

Unlike the extensions in traditional PABX, where the extension is usually associated with a phone, interface or menu in, the extension in Asterisk is defined as a set of commands to run. These commands are usually executed according to their level of priority. Some commands, such as Dial or GotoIf, have the ability to follow other commands depending on a certain circumstance.

At the time when the extension is dialed, the command marked as 1 will be executed, followed by command number 2 and so on, until the phone is hung up.

In the syntax used in extensions.conf file, a step in a given extension is written using the following format:

exten = extension,priority,Command(parameter)

The sign “equal to” = can also be written using “=>”, just like the form often used in many examples.

In conclusion, a "context" has a name, such as "john". In every of them, we can define one or more "extension". In an extension, we can define a set of commands. How do we define these extensions and the commands required to handle them? To define both, we need to edit extensions.conf file using a text editor. There are several tools that allow us to edit them using graphic/ web.

The components that build the stages of extension command or the command line are as follows:

- Extension is the label of an extension, which can be a string (containing allowed numbers, letters and symbols). Extension is a pattern that must be evaluated dynamically ir order to match many possible phone numbers. Every command line that becomes part of a particular extension should have the same label.

- Priority usually is of integer number. It is the sequence of a command that must be run within a given extension. The first command that will be run must begin with priority 1. If there is no

such thing as priority 1, then Asterisk will not execute the extension command. After running priority 1, Asterisk will then add another priority to the priority 2 and so on, of course, provided that there is no command that determines which subsequent priority which must be run. If the next command turns out to be undefined, Asterisk will stop the process running the command, notwithstanding there are still commands with higher priority.

- Command is the "application" to be run by Asterisk.

- Parameters are the parameters that must given to a command. Not all command requires parameter, as some of them can be executed without parameters.

For example:

```
exten => 123,1,Answer
exten => 123,2,Playback(tt-weasels)
exten => 123,3,Voicemail(44)
exten => 123,4,Hangup
```

With these definitions, an extension is numbered "123". When a call is dialed to this extension, Asterisk will respond to the call, executing a sound file with the name “tt-weasels” and give the caller the chance to enter voicemail into mailbox 33, and will be ended up with a hangup.

Asterisk itself does not really care about the order of line placement in extensions.conf. So with random placement of lines, the command we want to execute will still be carried out according to the order we want.

```
exten => 123,4,Hangup
exten => 123,1,Answer
exten => 123,3,Voicemail(44)
exten => 123,2,Playback(tt-weasels)
```

Another way in defining the command is to use Caller ID to match the caller.

```
exten => 123/100,1,Answer()
exten => 123/100,2,Playback(tt-weasels)
exten => 123/100,3,Voicemail(123)
exten => 123/100,4,Hangup()
```

With such command, compability with extension 123 will be possible only when the Caller ID of the

caller is 100. This can also be done through pattern matching process, such as the following:

```
exten => 1234/_256NXXXXXX,1,Answer()
and so on
```

This way, the compability with extension 1234 will only possible if the Caller ID begins with just the code area number 256.

We can even do the following:

```
exten => s,1,Answer
exten => s/9184238080,2,Set(CALLERID(name)=EVIL BASTARD)
exten => s,2,Set(CALLERID(name)=Good Person)
exten => s,3,Dial(SIP/goodperson)
```

In the second priority, it is shown that we can mark any person we dislike, while any person other than the one we dislike, after third priority, will return to the path specified.

An interesting Extension Examples

Asterisk is able to transfer calls. This can be done by adding the parameter “t” (lowercaps) to the user context, such as in the following syntax:

```
exten => 250,1,Dial(SIP/alrac,10,rt)
```

This way, the call transfer can be done by pressing "#", followed by the extension number. Asterisk will say "transfer" when you press "#" and sounds a dial tone until we enter the extension number to which we wish to call.

Asterisk has twenty parking spaces, number 701-720. Transfer the call that you want to park at extension # 700 and Asterisk will automatically park it at any empty lot and provide you with extension of where it is parked. To retrieve the call, you only have to dial the extension number. tempar extension mendial enough parking.

The steps necessary for parking calls are as the following:

- Add include => parkedcalls to the default context, or the one that you wish to have park call facility.

- You should have the file /etc/asterisk/features.conf which was created during installation. Make sure that you have the following syntax:

```
[general]
parkext => 700
parkpos => 701-720
context => parkedcalls
parkingtime => 180
```

You need to restart your Asterisk server through the console, as reloading is not sufficient. You can attempt it in the internal extension. So if there is an incoming call, the call can be parked by pressing #700, and Asterisk will say the extension number of where the call is parked. The caller will hear a beautiful music played through Music On Hold. When the parking time is up, then our extension number first dialed will be dialed again and we have the option whether to receive the call or not to receive and forward the call to voicemail.

The parameter "t" (lowercase) means that only the recipient of the call can transfer calls. This means we can only park a call just once. But if we add the parameter "T" (capitalized), such as:

```
exten => 250,1,Dial(SIP/alrac,10,rT)
```

then we can transfer the calls, whether as someone who receive the calls or as the caller. All this also means that we can unpark a call, park the call and transfer the call.

Asterisk can be configured for hunting telephone numbers. A hunt group is a list of phone numbers which will be rang consecutively until we pick up the phone. The example shows two phone extensions and a mobile phone number. The caller simply call extension 100 and Asterisk will do the rest of the tasks. Each phone will ring for 20 seconds, and when nobody pick it up, Asterisk will dial the next phone.

```
[alrac-followme]
exten => 100,1,Dial(SIP/350,20,r)
exten => 100,2,Dial(SIP/351,20,r)
exten => 100,3,Dial(Zap/1/1231234567,20,r)
exten => 100,4,VoiceMail(u350)
exten => 100,dial+101,VoiceMail(b350)
```

Other variation of the hunting technique above is that all numbers could ring at the same time. This is known as group ring. You can ring all the phones in a department if you wish them to do so. The

sample configuration is:

```
[customer service]
exten => 666.1,Dial(SIP/605&SIP/604&SIP/606,40,tr)
exten => 666.2,Voicemail(s699)
```

In the example, extension 604, 605, and 606 will be rang simultaneously when someone place a call to extension 666 from the Customer Service Department. If there is no one to answer the call within 40 seconds, the call will be forwarded to Voicemail.

Variable and Equation

In Asterisk there is support available for using variable with the name ${VARIABLENAME}. We can also write an equation using the construction ${EXPRESSION}, where the expression (equation) can be a regular expression, comparison, addition, subtraction and many more.

Reloading

After we made some changes to the dialplan and other things, we have to apply these changes we applied to asterisk by doing the following CLI asterisk command:

```
CLI> reload
```

A large configuration file size or many smaller file size?

Through the command #include <filename> in extensions.conf, other files can be included. This way, we can configure extensions.conf to be the main file, users.conf that contains local user, services.conf that contains various services like conferencing. By doing so, it is easier to maintain the dialplan we create.

Forwarding to another Asterisk

To forward calls to other Asterisk server, we can use the following syntax:

```
[iaxprovider]
switch => IAX2/user:[key]@server/context
```

The above command will carry out forwarding to other server. However, User and key have to be defined in iax.conf file of the server to which the calls will be forwarded. The context for this server is the same as that of extensions.conf of the server that does the forwarding.

CHAPTER 15: VoIP IP PBX Hardware

If you require a phone solution for a small office with 6-16 extension lines, you would normally use a PBX (Private Branch Exchange) machine to perform switching between these extension lines and to manage outbound connectivity. But as there are greater needs to extend the number of extensions (say hundreds), you will find switching using this conventional PBX rather complicated, with cables snaking around your premise. In VoIP, this is no longer a problem, as VoIP PBX is based on Internet Protocol, a from factor making IP PBX a much more compact solution with fewer port cables to connect.

Linksys SPA9000

Figure 15.1: IP PBX Hardware Physical Dimension is small

An example of IP PBX being used as an example is Linksys SPA9000, a device consisting only two LAN (RJ-45) ports and two telephone (RJ-11) ports. SPA9000 is actually a router or proxy server, with port connectivity to WAN and LAN. The RJ-45 enabling connectivity to WAN is labeled “Internet”, while the RJ-45 connection to LAN is labeled “Ethernet”. The default IP address for the LAN

connectivity is 192.168.1.1. Meanwhile, the two RJ-11 telephone ports are part of PBX port that can be connected to two conventional phones, including fax machine. Despite this simplicity, SPA9000 is a PBX that is capable of being connected to up to four PSTN lines or to a large VoIP infrastructure. Internally, SPA 9000 could accommodate 16 extension lines. When compared to a small conventional PBX, SPA9000 has only four ports compared to a small PBX that uses 20 telephone calbles. So all connections will be established through the internet infrastructure.

Linksys SPA9000 Configuration

The way Linksys SPA9000 operates is somewhat similar to other Linksys VoIP equipments. To configure Linksys SPA9000, we need to have:

1. information on IP address for both WAN and LAN ports.
2. SIP account of a provider in the internet to allow SPA9000 to register itself to four different SIP accounts.
3. Number allocation for extension lines of the PBX to SPA9000 to provide address up to 16 telephone numbers automatically.

Numbers allocated for each extension are specific, distinguishing an IP PBX from other VoIP appliances. Normally, a typical VoIP equipment does not provide telephone number allocation. So if you want to connect your conventional phone to Linksys SPA9000 through the internet or WAN port, the first thing you have to do is find the internet/WAN IP address of Linksys SPA 9000 so later you will be able to configure using the web, by doing the following steps:

- Press “*” on the conventional telephone keypad repeatedly until you hear a man talking through your telephone.
- Press “110#” and listen carefully to the SPA9000's IP address given by the man. Write it down so you dont have to memorize it.

Another easier way to obtain the IP address is to go into the Ethernet/LAN port. The IP address of SPA9000 ethernet LAN should be 192.168.1.1 by default, that is, provided you have not changed the settings.

The next step is to configure your PC IP address so it will match that of Linksys SPA9000 so you will be able to do the configuration through the web. Go to PC and match the PC's family address to that of SPA9000. Go to Start menu, control panel, network connections, local area connection, internet protocol (TCP/IP), and properties.

Now you will be able to log in to Linksys SPA9000's web interface from your PC via http://ip-address-spa9000/.

The first menu you will find is the status of the Linksys SPA9000.

Figure 15.2: Linksys SPA9000 Administration Panel

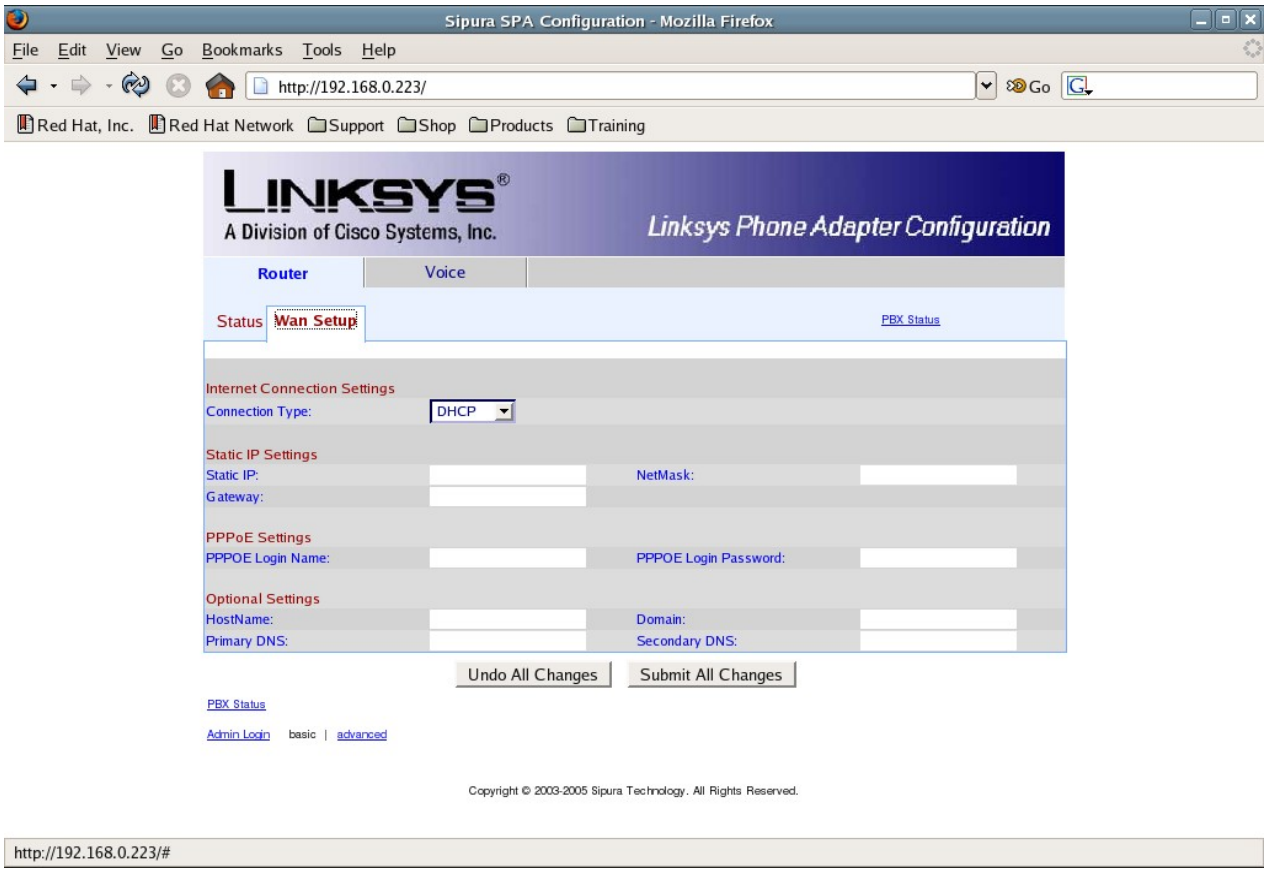


Figure 15.3: The WAN Setup Tab of Linksys SPA9000

In the WAN setup, we get two options:

- Using a static IP (requiring IP address, Netmask, gateway etc.)
- Using automatic IP (connection type should be set to DHCP).

In general, the IP address allocation method used in a WAN normally is dynamic. However, for a softswitch, it is recommended that you set the IP address allocation to static in order to make it easier for non-Linksys SIP client to register itself to the softswitch.

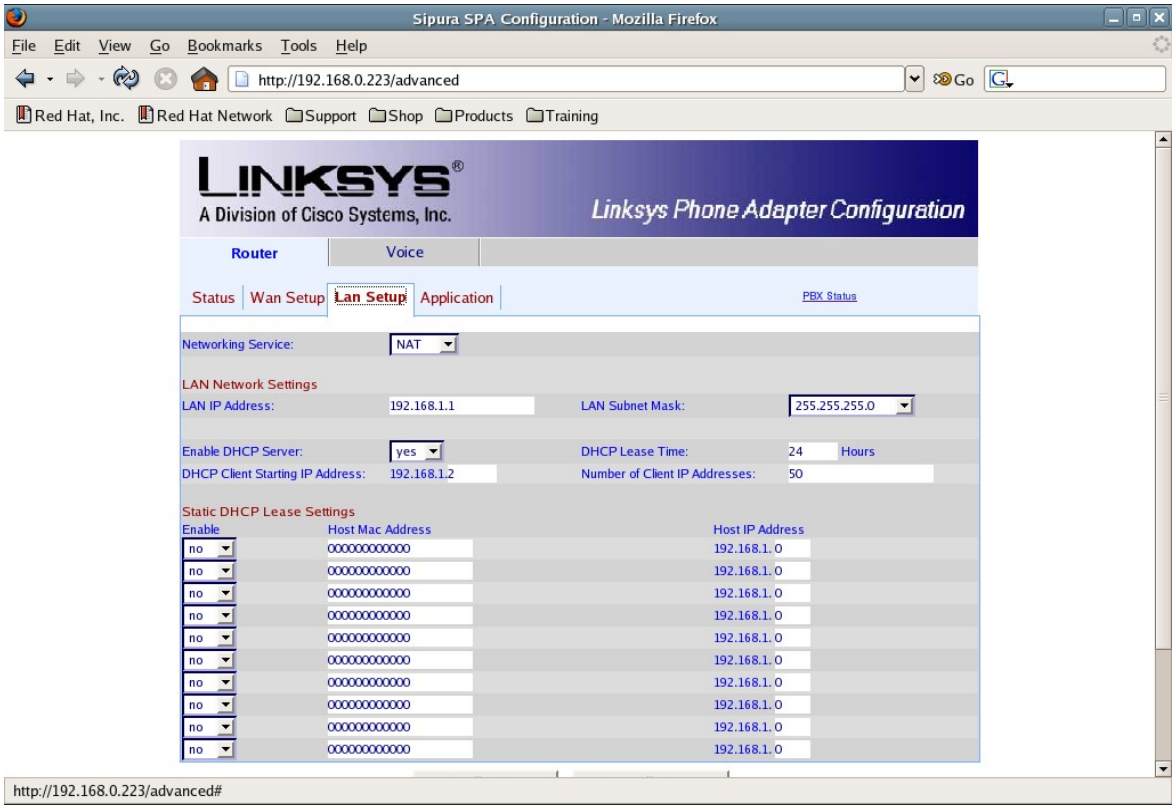


Figure 15.4: The LAN Setup Tab of Linksys SPA9000 Administration Panel

In the LAN Setup tab, we can configure the following:

- whether NAT/Router in Linksys SPA9000 should be activated or not.
- IP address. Put 102.168.1.1
- Netmask. Put 255.255.255.0
- DHCP Server for client in LAN

We can also configure the IP address to be allocated to a specific MAC address.

Configuring VoIP on Linksys SPA9000

Basically, there are several types of telephone connection in available in Linksys SPA9000:

- Two FXS or connections to the telephone. This connection basically needs not to be configured and by default, its number is 100 and 101.
- Four SIP connections to higher central level, to any SIP server
- There are 14 non-FXS extensions in form allocation for IP Phone. The number allocated ranges from 102 to 116. Still, configuration is not needed and, any IP Phone attempting to connect to Linksys SPA9000 can do so with a blank password.

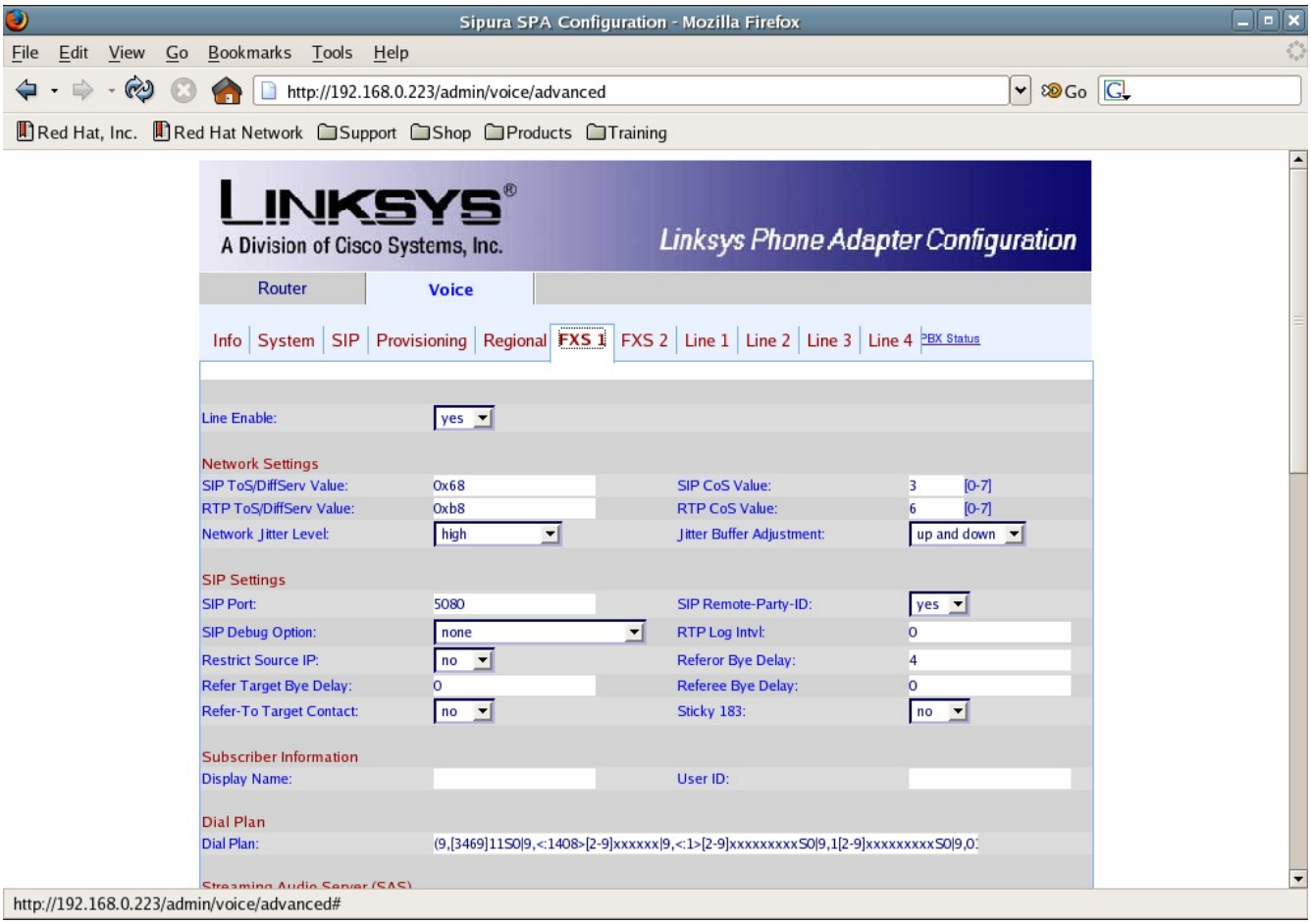


Figure 15.5: FXS 1 Tab under Voice Tab of Linksys SPA9000 Administration Panel

On the voice menu and admin menu, we can see the configuration for FX1 and FX2, or Line 1-4. On the FXS1 menu, we need to set Line Enable to Yes, so the FX1 will be able to receive calls dialed to it. Make sure that there is a phone line or fax machine connected to FXS1 or otherwise any incoming calls will not be received. Apply the same configuration to FXS2. Overall, the configuration allows us to connect two conventional phones, including fax machine, to Linksys SPA9000.

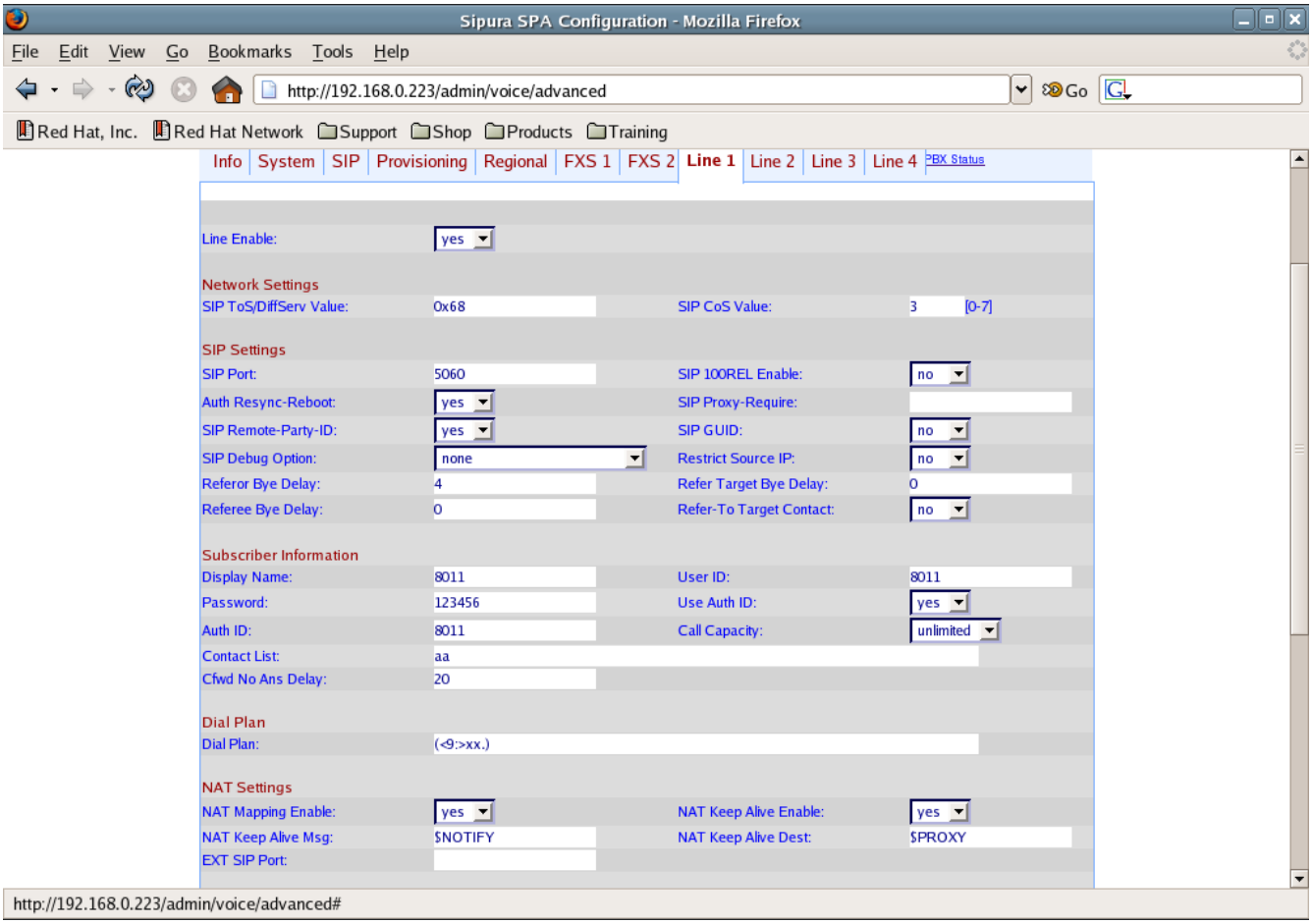


Figure 15.6: Line 1 Tab of Administration Panel

On the menu of Line 1 to Line 4, we can configure to which SIP server each of these lines will be registered to. Make sure Line Enable is set to Yes.

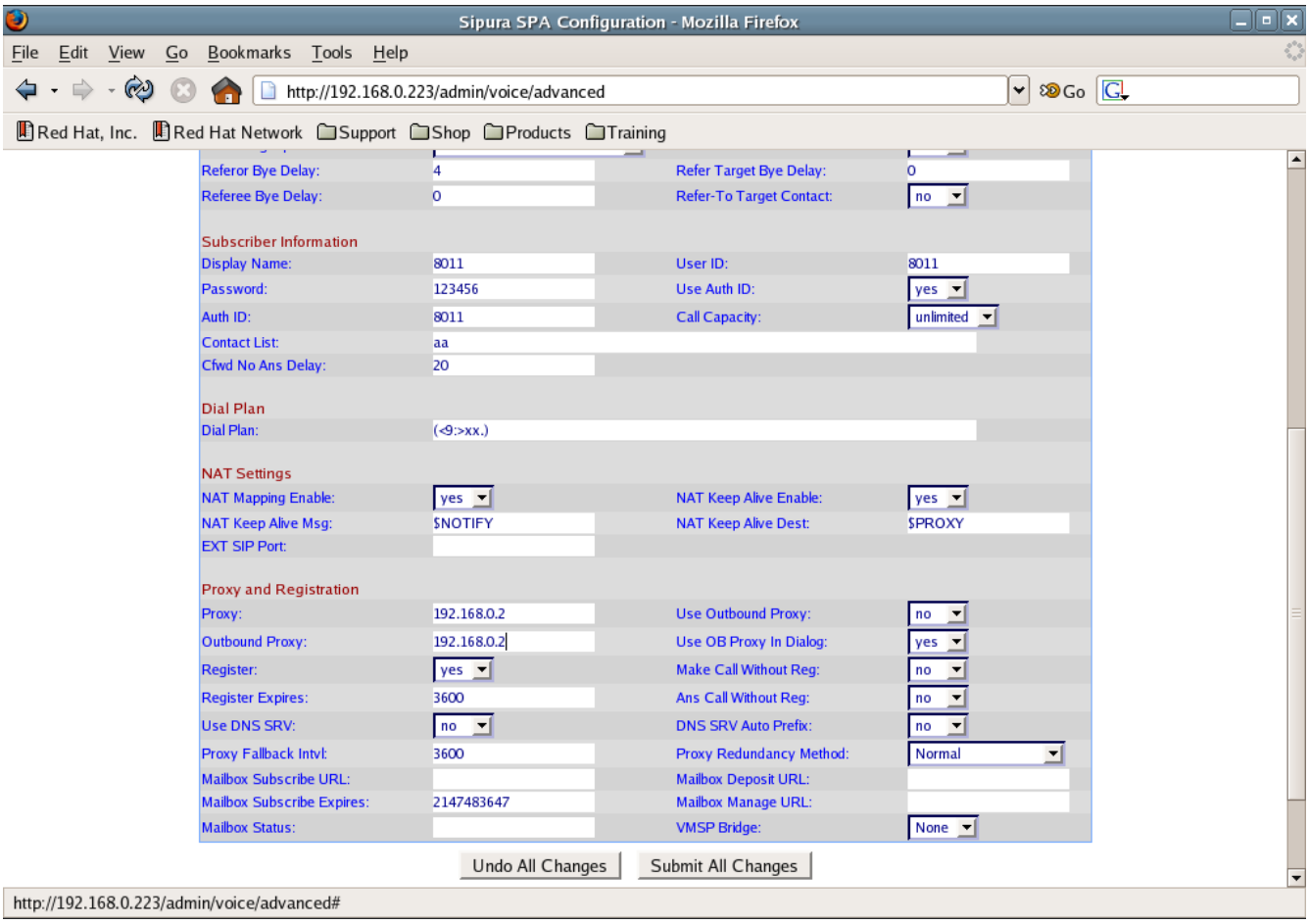


Figure 15.7: Line 1 Tab of Administration Panel

On Linksys SPA 9000 we could set four SIP accounts to be registered to any SIP Proxy, each account connected only to a line. Some important things to do this are as the following:

- set Line Enable to yes
- Fill the information pertaining to your account in the following parameters:

Proxy	voiprakyat.or.id
User ID	the number given by voiprakyat
Password	password of voiprakyat account
Use Auth ID	no

If you set “Use Auth ID” to ye, then fill that parameter with the number of your VoIP Rakyat account.

Do the same to your other SIP account(s) for the rest of the lines (Line2, Line 3 and Line 4).

CHAPTER 16: Analog Telephone Adapter for connection to PSTN

It is obvious that making both VoIP and PSTN to coexist is difficult, particularly when you are seeking to minimize your telecommunication spending by choosing either one of them. So this section will help you understand how to keep your PSTN line by using a trunking equipment to be connected to PSTN. Generally, the equipments available in the market have one FXO or two. Coincidentally, the Linksys SPA400, the equipment we use as an example, has four FXOs and is relatively less expensive. The interest here is that you want to have VoIP be connected to your conventional phone, either to a PABX or directly to your PSTN network. This usually does not require you to obtain a license, unless you intend to become a provider with commercial interest.

So for your purpose, you need to have an Analog Telephone Adapter (ATA), which can be connected to the cable of your PSTN cable. In VoIP, this is often called FXO (usually labeled as Line). The physical condition of this interface is similar to that of RJ-11 (conventional telephone jack). The difference is that in in RJ-11-FXO line, there is no voltage, something that is there when you have a RJ-11 Phone. With FXO, you can your VoIP be connected to PSTN and PABX extensions. And ATA phone can be connected to both a regular phone and PABX CO.

In short, there are two type of RJ11 connections in ATA, namely,

- FXO to be connected to PSTN / Telco line / PABX extension.
- FXS to be connected to Telephone line / FAX.

When connecting ATA to PSTN line, make sure that you do not connect it to the wrong plug. If you did, the PSTN's voltage, which usually is around 48 V, would collide with that of the ATA phone. This will damage your ATA equipment. So prior to connecting them, you have to set your ATA equipment so it recognizes whether the voltage in place is 48 V or 24 V.

Linksys SPA3000 Analog Telephone Adapter

Figure 16.1: ATA Linksys SPA3000 has two RJ-15 sockets on one of its sides

One of the smallest ATA we the author have ever seen is Linksys SPA3000. From the picture above, you can see two telephone jacks, each labeled Phone and Line. Connect the phone socket to your conventional phone while the Line socket to the PSTN cable.

Figure 16.2: ATA Linksys SPA3000 has a RJ-45 socket, power socket and LED indicator on the other side

On the back of Linksys SPA9000, there is RJ-45 plug that can be connected to LAN cable for computer and the Internet.

Configure Linksys SPA3000

Figure 16.3: Linksys SPA3000 Administration Panel

SPA3000 logical configuration is not much different from other VoIP equipment. In general, we need to configure: IP address, Netmask, Gateway, DNS, the telephone number, password and SIP server. The initial appearance of SPA3000 is somewhat similar to that of other Linksys products. So this is a plus for those who are already familiar with Linksys product.

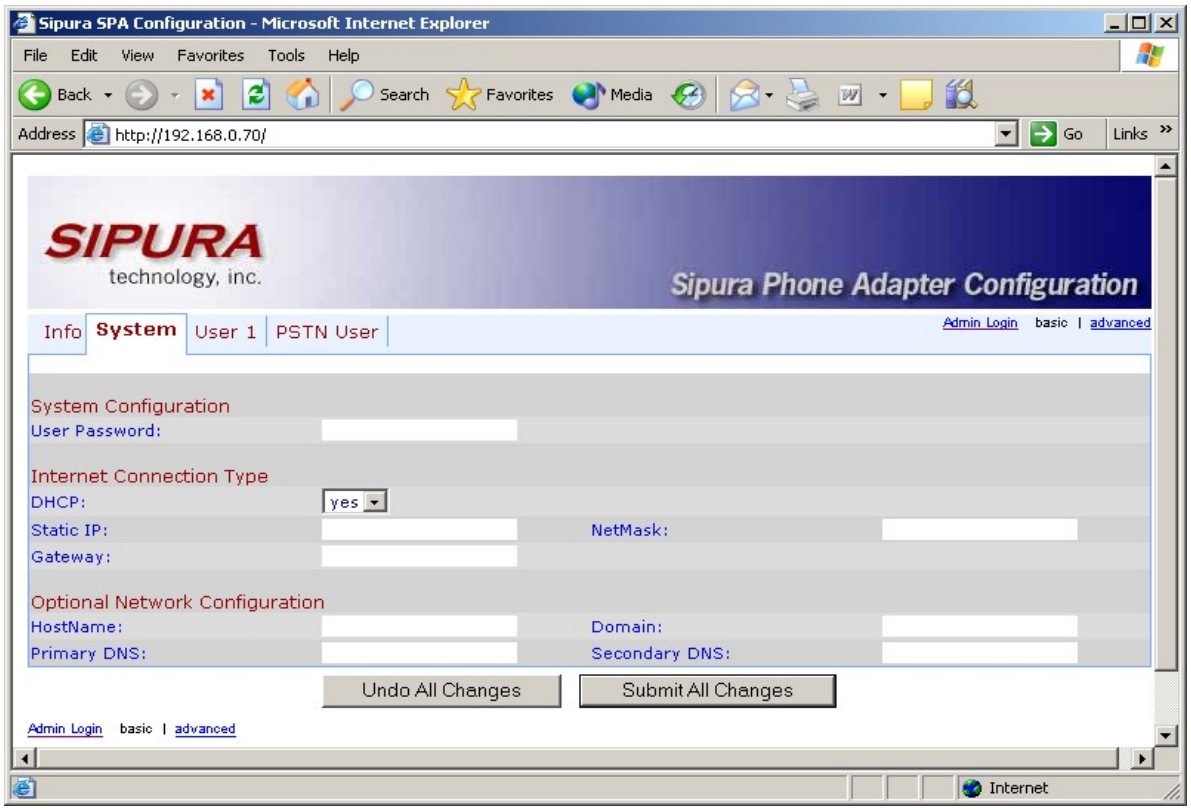


Figure 16.4: The System tab of Linksys SPA3000 Administration Panel

On the System menu you can set the IP address, Netmask, gateway, and DNS of the Linksys SPA3000. If you have a DHCP server, you can enable DHCP so ATA will get its IP address automatically from the server.

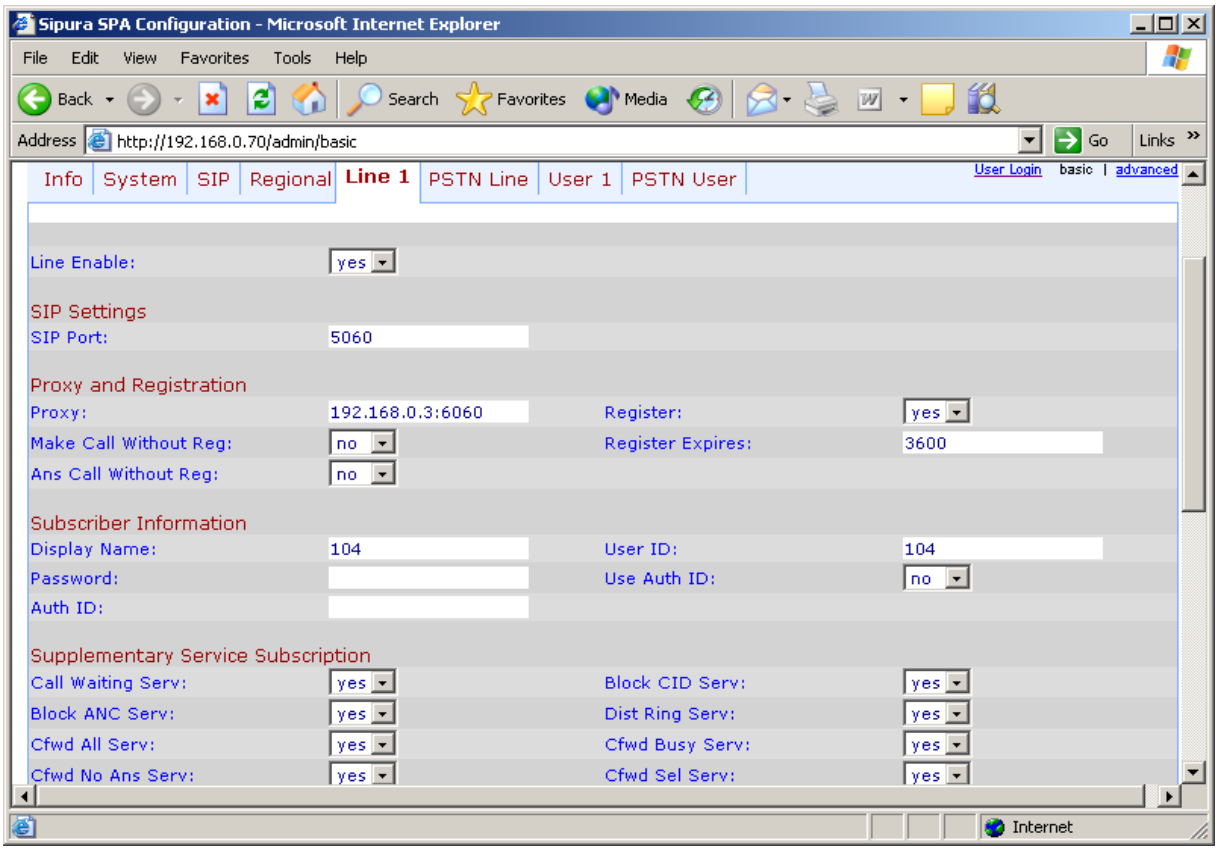


Figure 16.5: Line 1 tab of Linksys SPA3000 Administration Panel

Registration to the SIP server for the telephone is carried out through Line 1 menu. We need to enter some information:

Line Enable - yes
Proxy – the SIP Server.
Display Name - the phone number in the SIP server.
User ID – the phone number in the SIP server.
Auth ID – the phone number in the SIP server.
Password – the password to register to the SIP server.

Once you completed all these, the configuration for registration to SIP server for telephone connected to phone/FXS interface is completed.

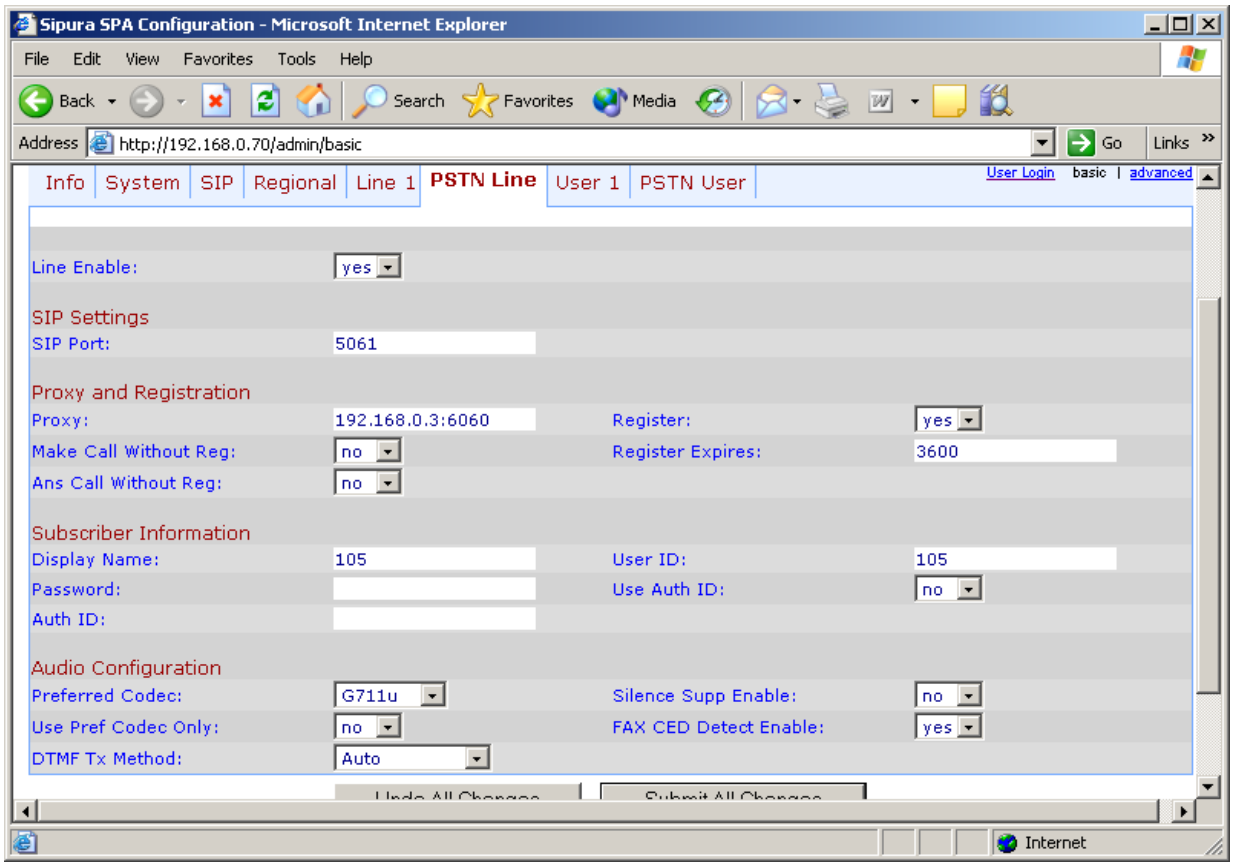


Figure 16.6: PSTN Line tab of Linksys SPA3000 Administration Panel

For connectivity to PSTN, the configuration for PSTN Line registration is similar to Line configuration, using the following configuration:

Line Enable - yes
Proxy - IP address / hostname of SIP Server.
Display Name - the phone number in the SIP server.
User ID - a phone number in the SIP server.
Auth ID - a phone number in the SIP server.
Password - password to register to the SIP server.

Once these are completed, so is the configuration for registering the PSTN Line to SIP server for telephone cable connected to FXO interface.

Linksys SPA3000 ATA Status

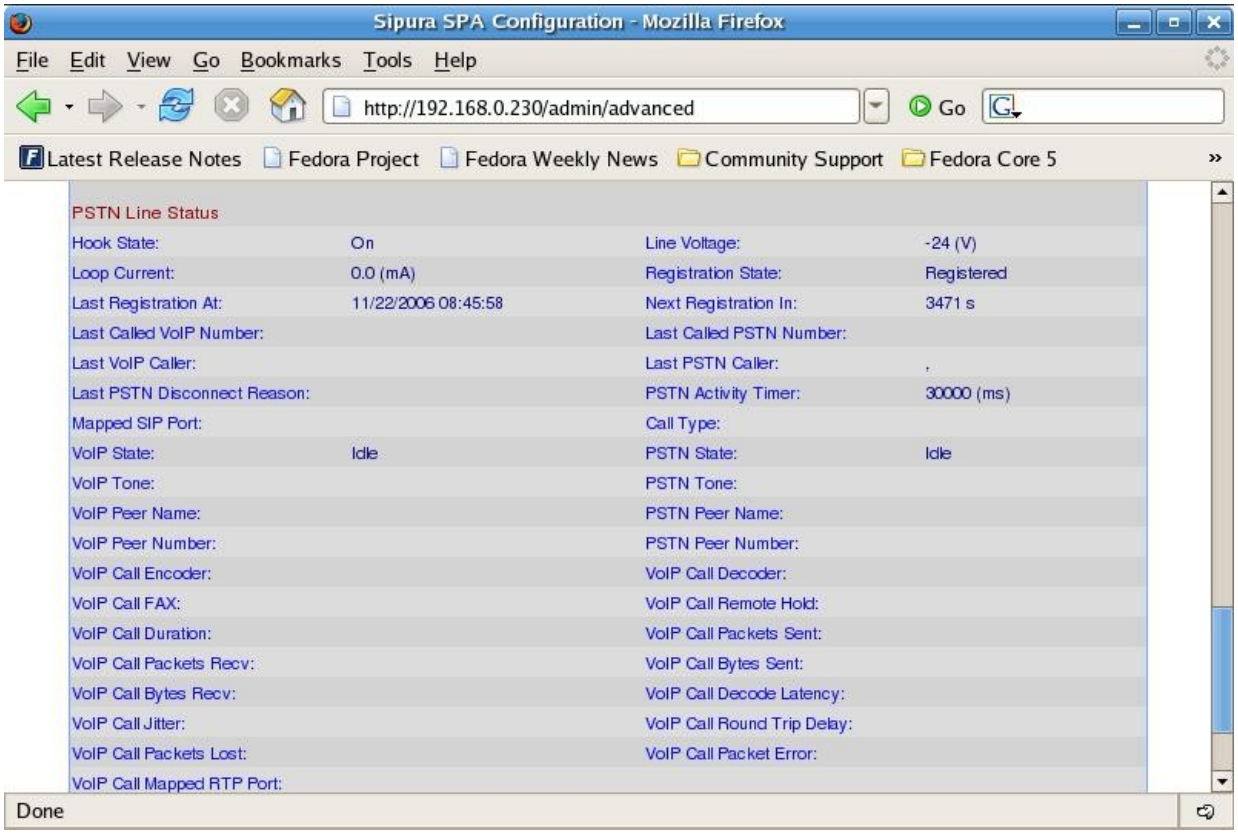


Figure 16.7: PSTN Line tab of Linksys SPA3000 Administration Panel

At the beginning of the SPA3000 configuration menu is the status and information menu. Slightly below, there is a status of SPA3000 PSTN line. Despite the many parameters available in this status, you have to be concerned with just two of them: registration state and Line Voltage. For the former, make sure that the parameter " registration state" says registered. This implies that SPA3000 is properly registered to a SIP proxy. For the latter, check the voltage level at the connection to PSTN/PABX. PSTN and a number of PABX usually have their voltage level at -48 V and -24 V respectively. While the voltage level of the PSTN is fine, PABX's voltage level will be problematic for SPA3000, as its default voltage threshold is configured only to have SPA3000 connected to PSTN or PABX when their voltage level is above -30V. When you do make a call using the line connected to the unrecognized PABX (or PSTN), SPA3000 will give a busy tone. To have SPA3000 recognize a PABX whose voltage level is below -30V, we have to change the parameter available in the PSTN Line menu, which you could access when you're logged in as admin.

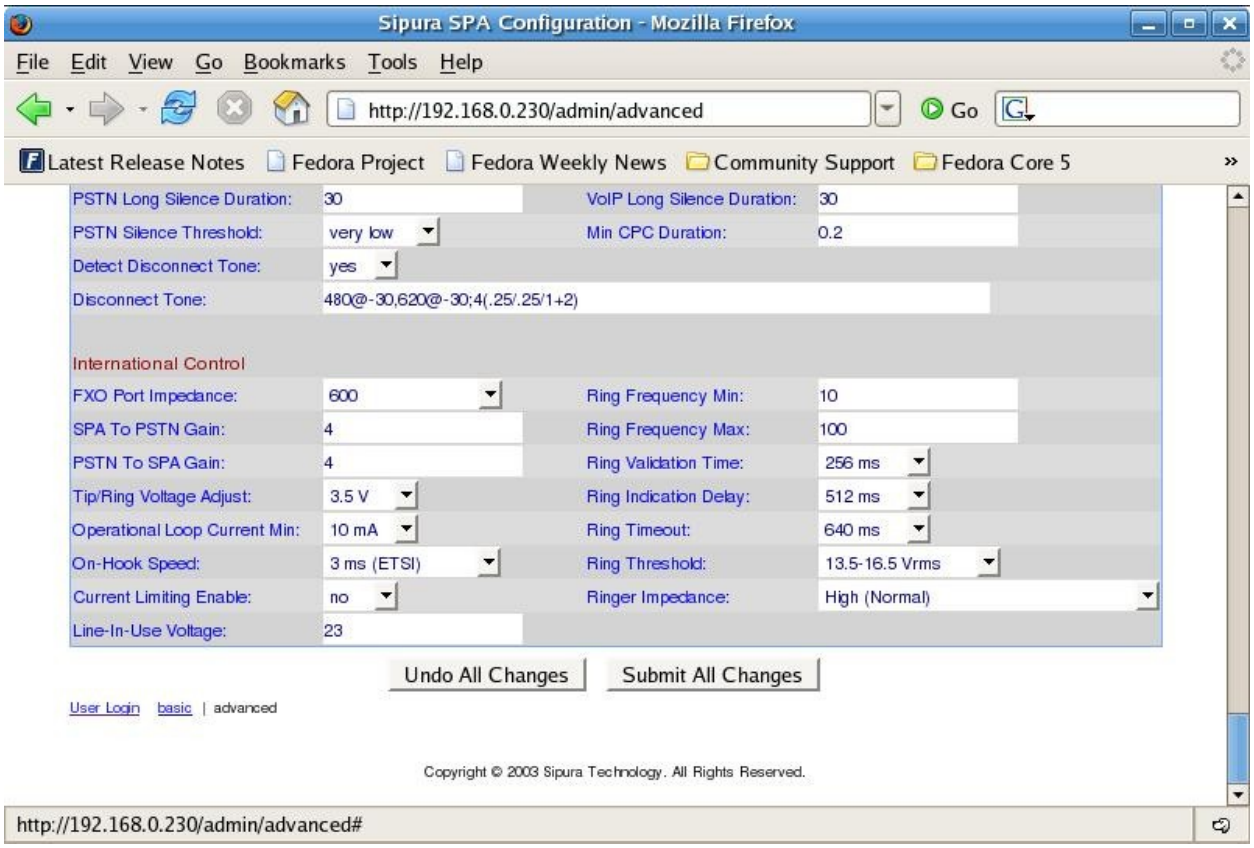


Figure 16.8: PSTN Line tab of Linksys SPA3000 Administration Panel

In the International control under the bottom, there is parameter "Line-in-Use" Voltage is its default value is 30. If the PSTN Line voltage of 24V the PABX only set parameter "Line-in-Use" Voltage of 30V will cause the SPA3000 think that the SPA3000 is not connected to the PSTN / PABX. Thus we need to the change the value to be smaller than 24V, such as 23 or 20 V. This way, SPA3000 will recognize that it is connected to a PSTN/PABX network even though the voltage line is only -24V.

LevelOne VOI-2100 Analog Telephone Adapter

Figure 16.9: LevelOne VOI-2100

LevelOne VOI-2100 is another type of ATA which can be used in SIP-based VoIP network. Similar to SPA3000, VOI-2100 has two RJ-11s, one for the connection to the telephone, while another to connect to PSTN or PABX cable. In contrast to SPA3000, VOI-2100 has an embedded router, NAT and DHCP server inside it. There are two UTP RJ-45 plugs, one can be connected to WAN while another to LAN.

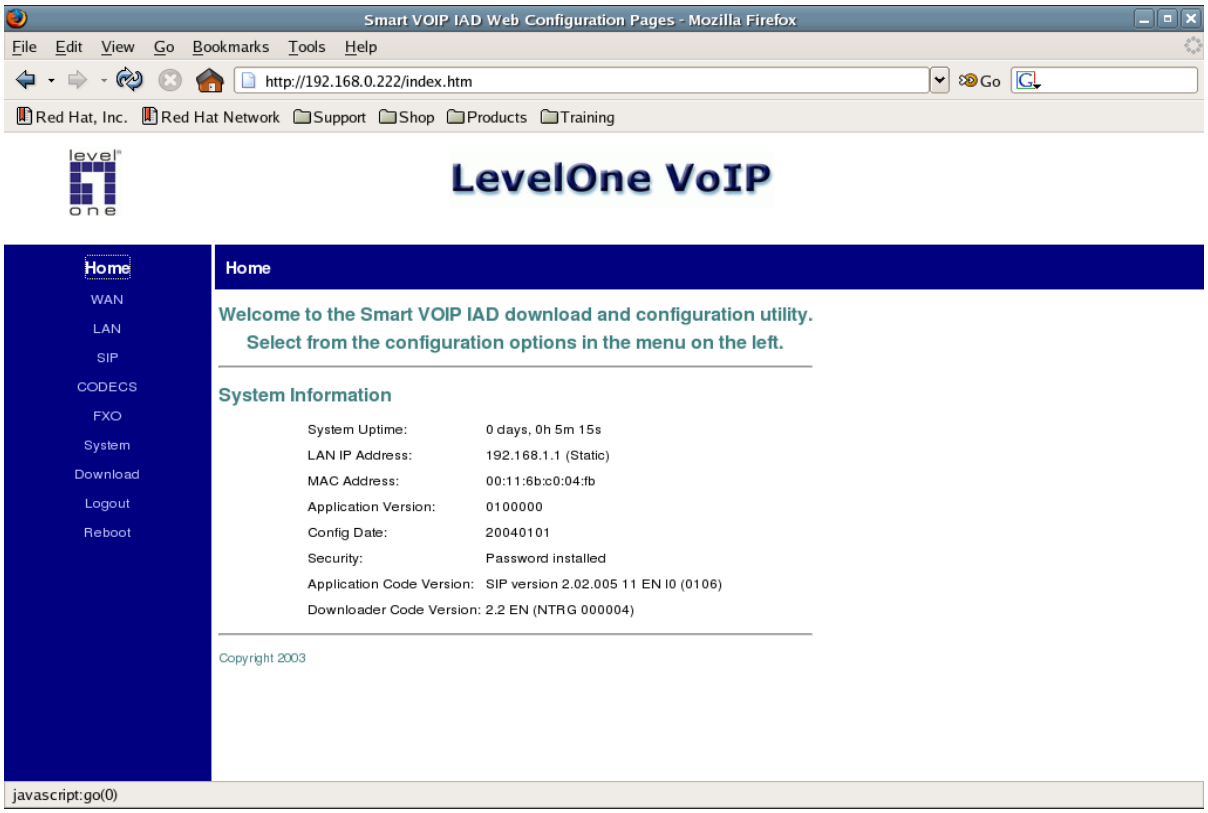


Figure 16.10: LevelOne VoIP Administration Panel

At the beginning of the LevelOne VOI-2100 menu is the status of the VOI-2100, such as MAC Address, System Uptime, etc.. Various configurations of VOI-2100 is available on the left.

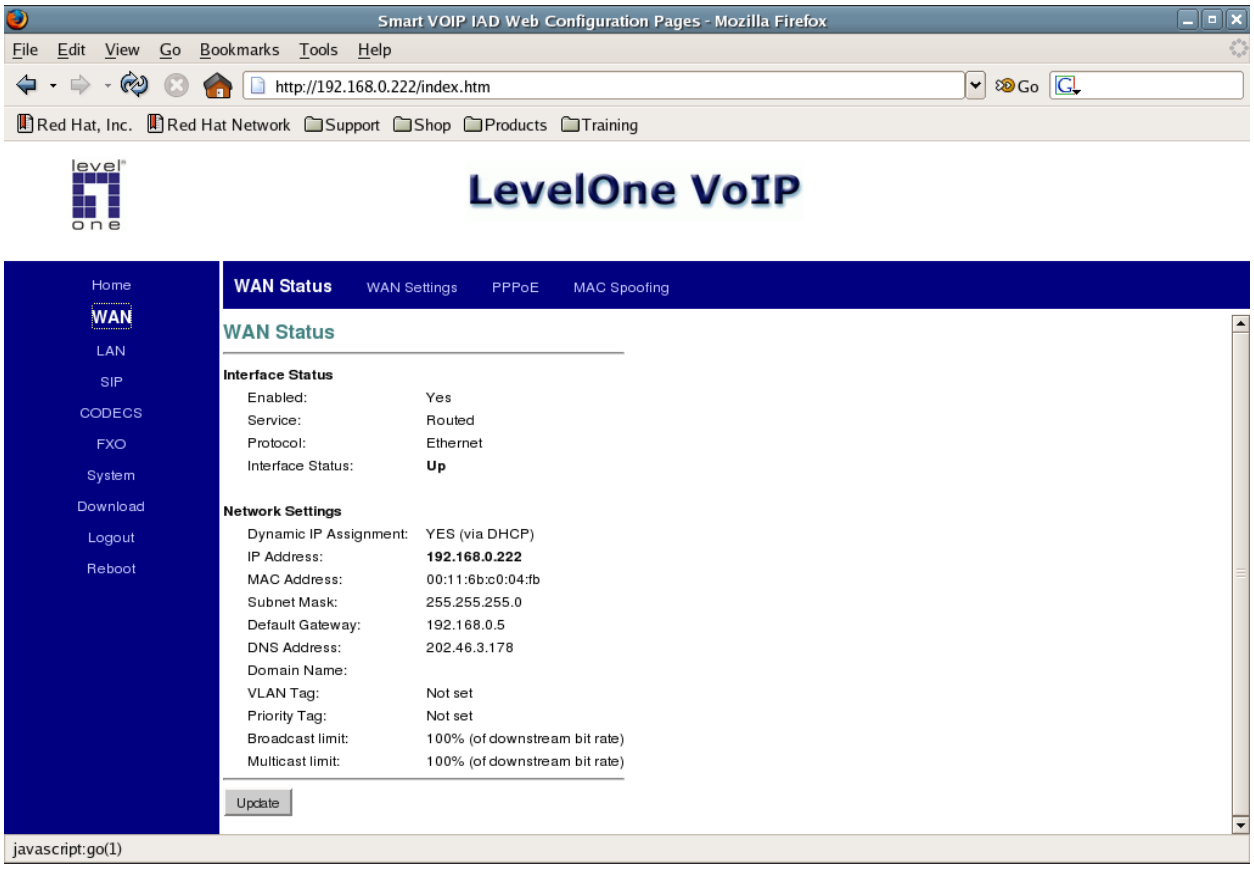


Figure 16.11: The WAN Status tab of LevelOne VoIP Administration Panel

On the WAN menu, click the WAN status. We can see the condition of WAN LevelOne VOI-2100, some standard information from the WAN connection, such as IP address, Subnet Mask, gateway, and DNS server. A number of tags that is possibly to be configured to improve VoIP performance are VLAN Tag and Priority Tag, both of which can be found also in WAN status.

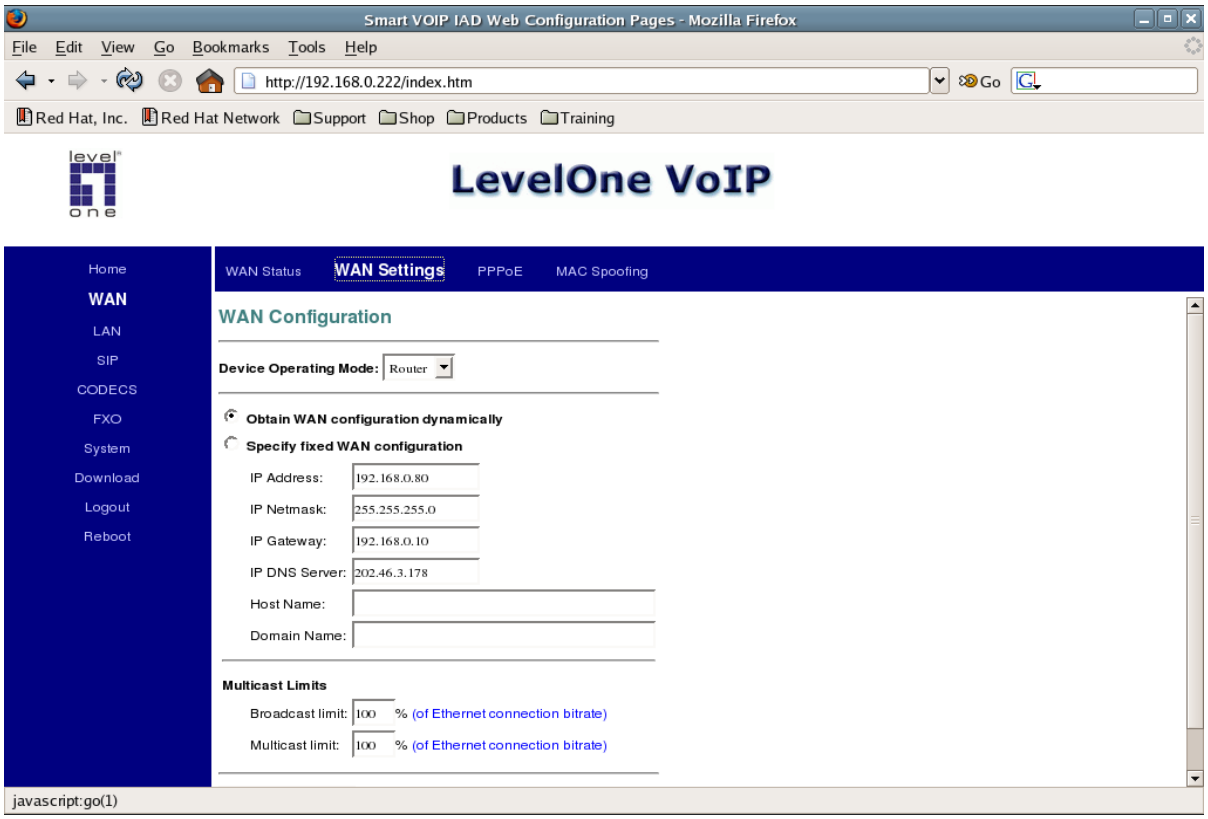


Figure 16.12: The WAN Settings tab of LevelOne VoIP Administration Panel

In WAN Settings. We can configure several parameters, such as,

IP address of the WAN Connection as static or dynamic.
Traffic limitation.

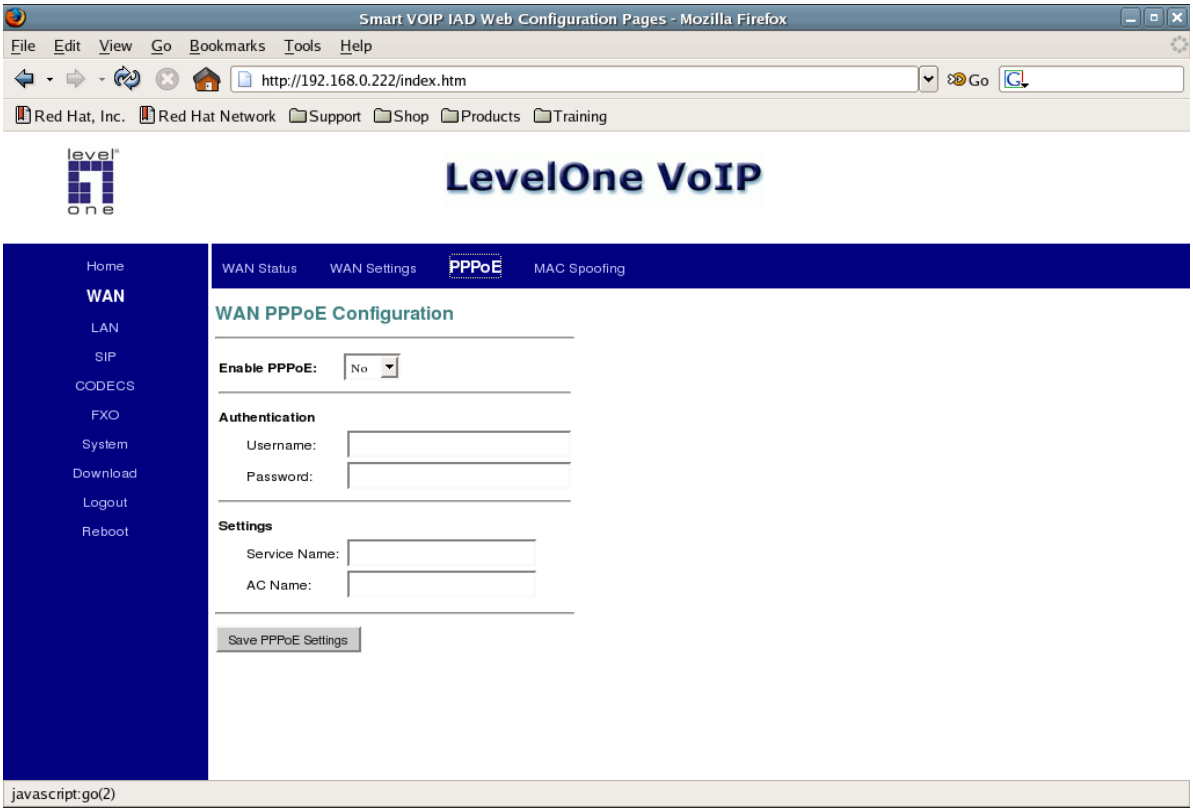


Figure 16.13: The PPPoE tab of LevelOne VoIP Administration Panel

In WAN menu, click PPPoE. Coincidentally, there is a feature to authenticate ADSL that uses PPPoE. Thus, if you like please feel free to enter the username and password of PPPoE.

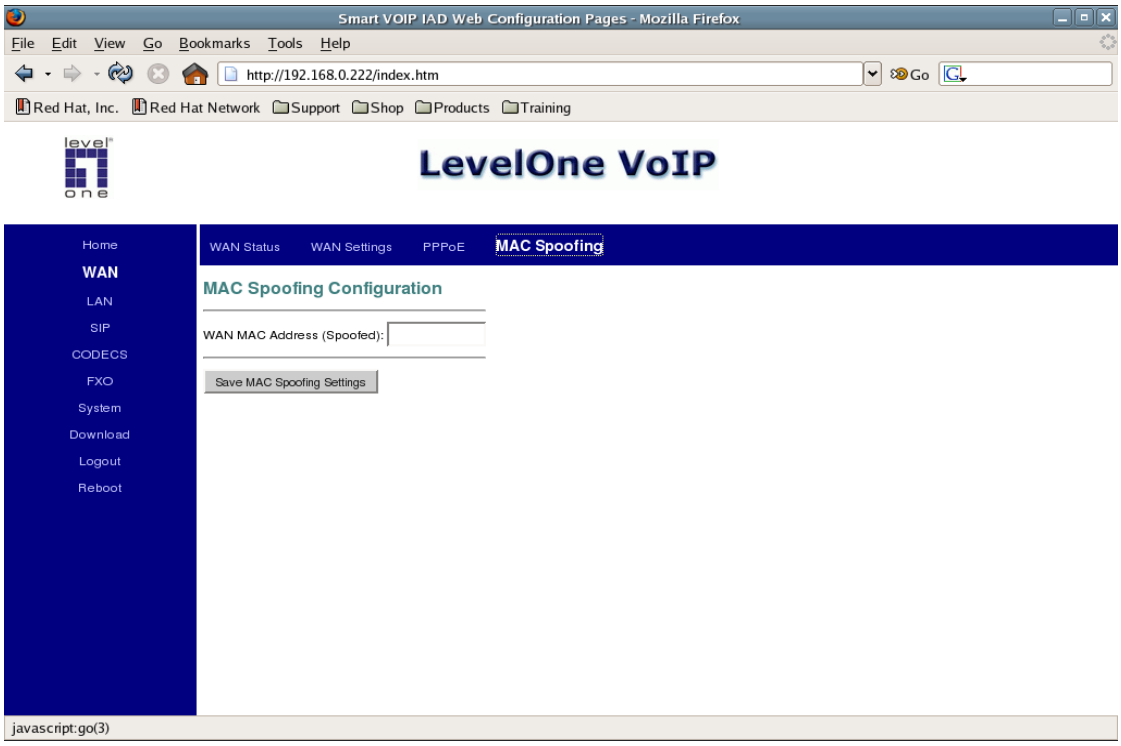


Figure 16.14: MAC Spooling tab of LevelOne VoIP Administration Panel

In the WAN menu, click MAC spoofing. This allows us to change the MAC address of the Ethernet WAN we want to use. This is often necessary to do when the ADSL provider to whom we subscribe our service sets only a certain MAC address capable of connecting to the provider. Through this MAC Spoofing menu, we can change the MAC address of the Ethernet WAN in order to use the MAC address approved by the provider.

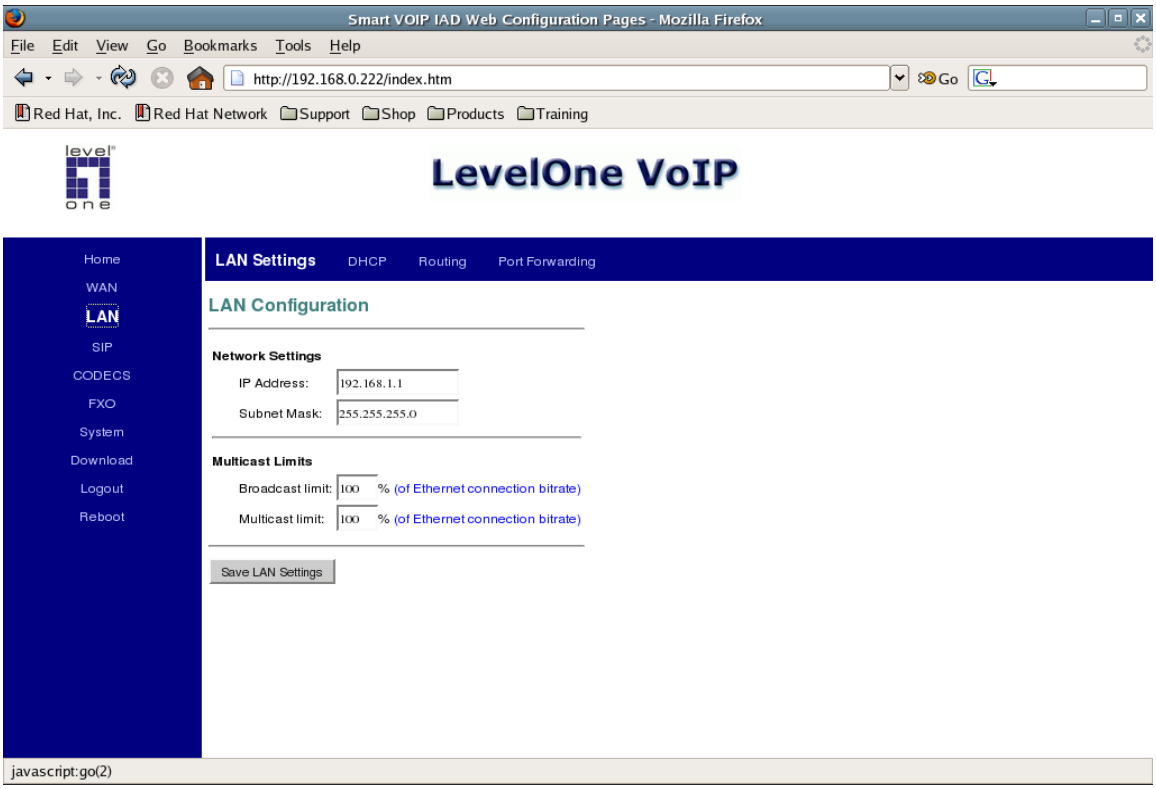


Figure 16.15: LAN Settings tab of LevelOne VoIP Administration Panel

On the menu LAN, click LAN Settings. Here we can set the IP address and Subnet Mask of the Ethernet LAN that we use.

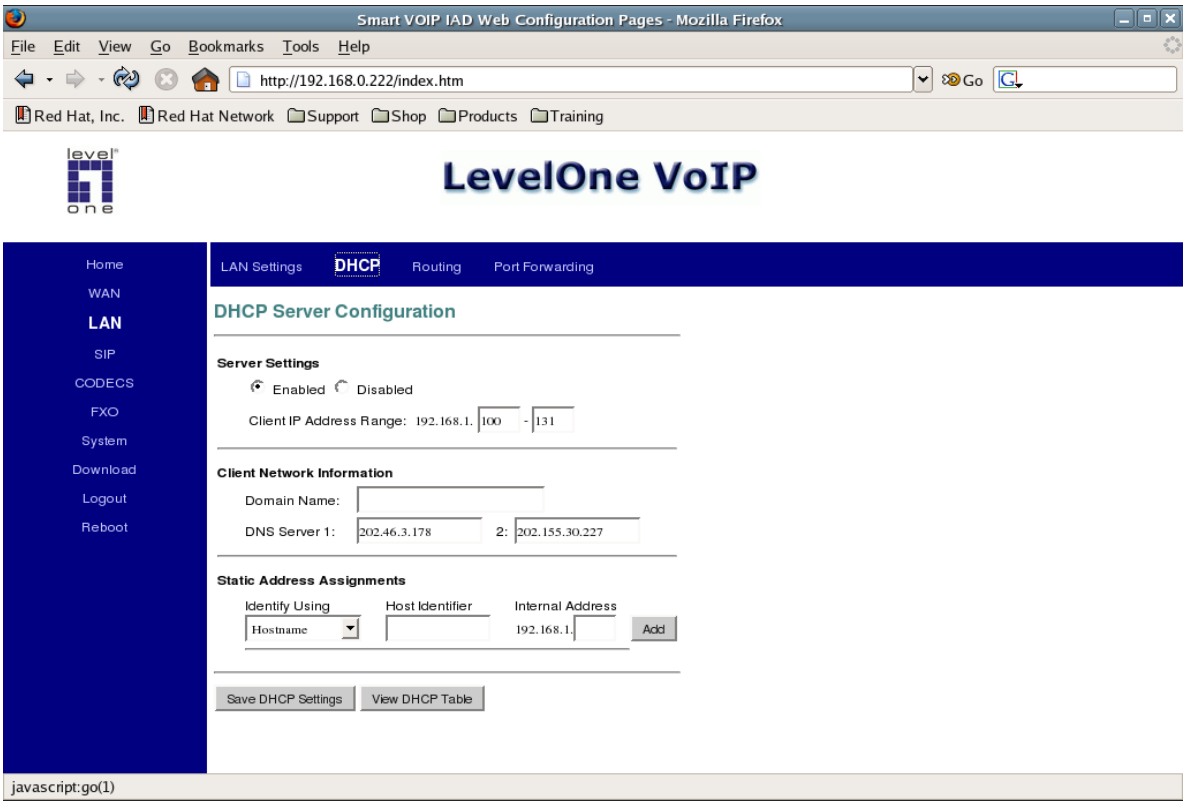


Figure 16.16: DHCP tab of LevelOne VoIP Administration Panel

On the LAN menu, click on DHCP. We can activate or deactivate DHCP server. We can also configure the range of client IP addresses that can be allocated to the network. Note that in a given network it is possible to have a number of DHCP servers. It is important to ensure a DHCP server's IP addresses allocated are not contradictory to those of different DHCP servers. Other information such as Domain and DNS Server can also be configured under DHCP tab.

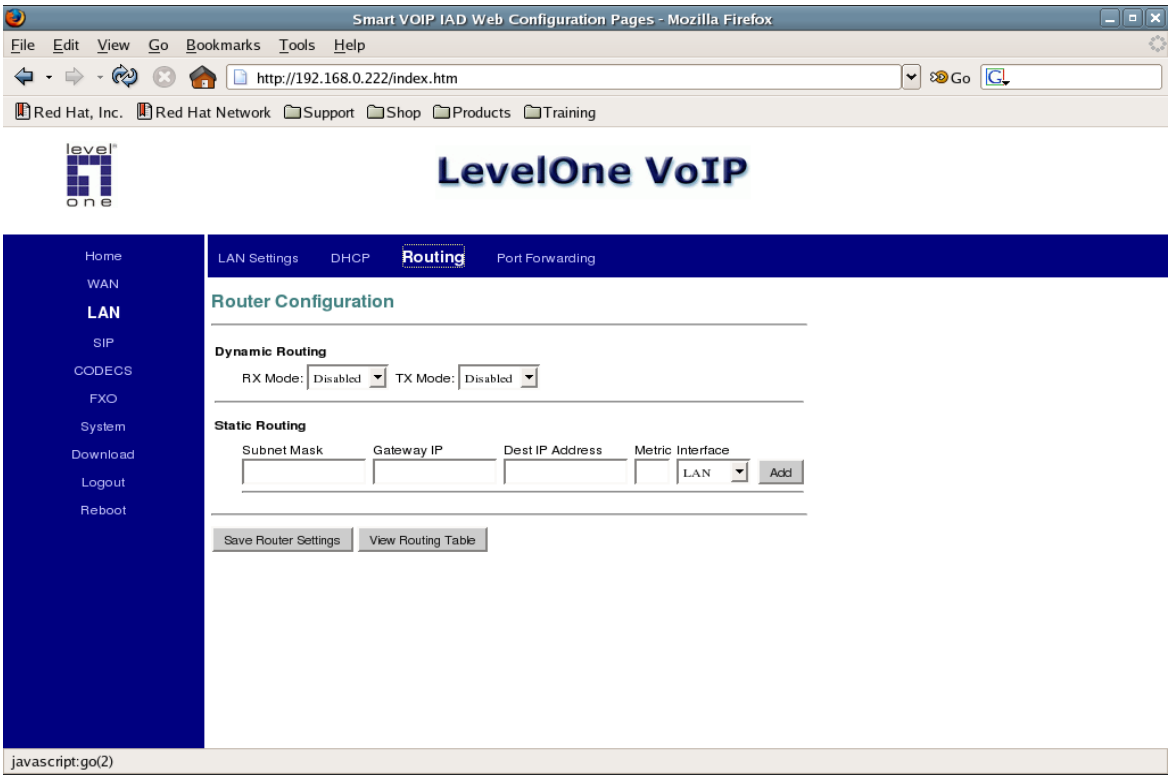


Figure 16.17: Routing tab of LevelOne VoIP Administration Panel

On the LAN menu, click Routing. We can add static routing to other networks if necessary. The information needed for this is just IP address destination, Subnet Mask, and Gateway router that connects to the network.

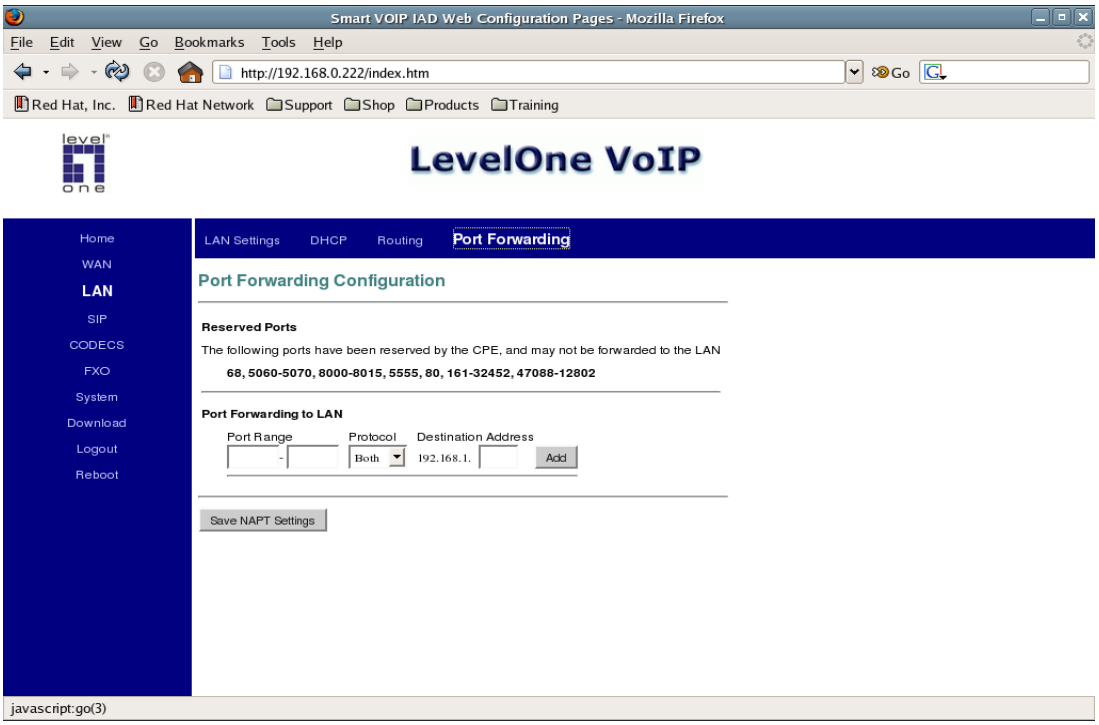


Figure 16.18: Port Forwarding tab of LevelOne VoIP Administration Panel

On LAN menu, click Port Forwarding. This feature allow us to do a forwarding from a port. For example, if we have Mail/SMTP Server behind NAT, then through this port forwarding, all traffic heading to port 25 (SMTP server) from outside/WAN can be forwarded by NAT to server behind NAT. Information you need to enter is port range and the server's IP address behind the proxy. For example, if we want to include just port 25, the port range should be just 25 to 25.

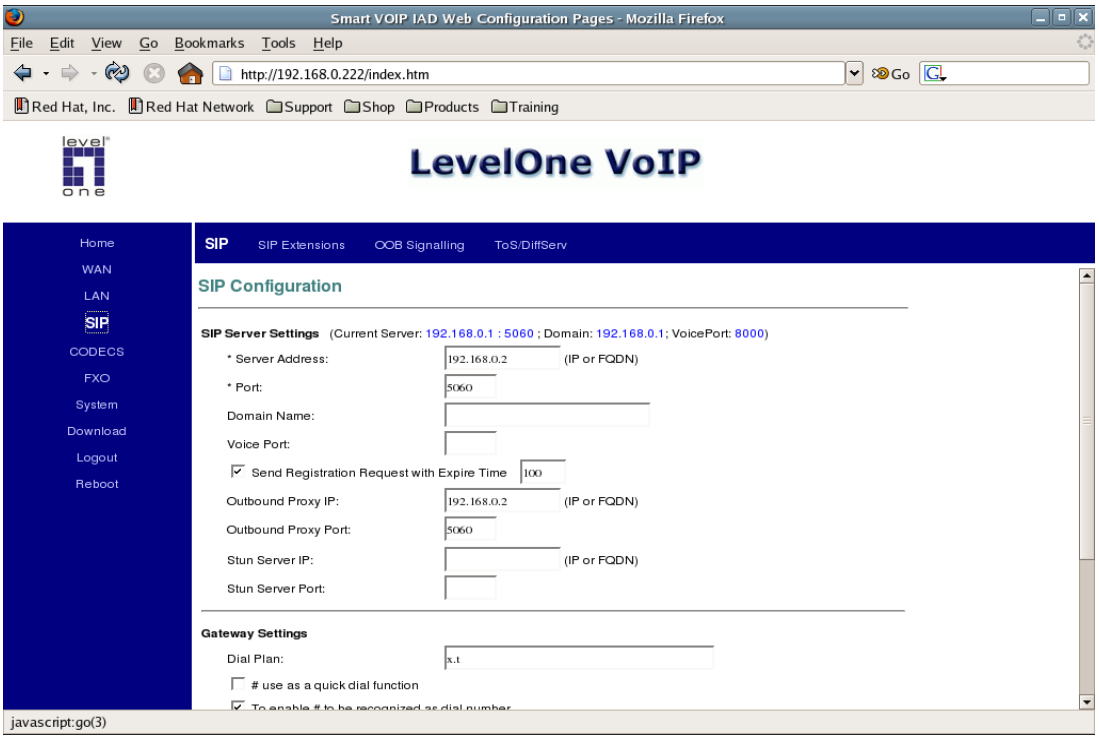


Figure 16.19: The SIP tab of LevelOne VoIP Administration Panel

The most important part of VOI-2100 is the SIP configuration. On the SIP menu, click SIP tab. This tabs allows you to change key parameters enabling VOI-2100 to enter SIP network. Some of these are:

- Server address – IP address/ hostname of the SIP proxy server
- Port – the port number. The value often used is 5060.
- Outbound Proxy IP – IP address/ hostname of outbound proxy is usually similar to that of SIP Proxy server.
- Outbound Proxy Port – which is usually similar to SIP Port, that is, 5060.

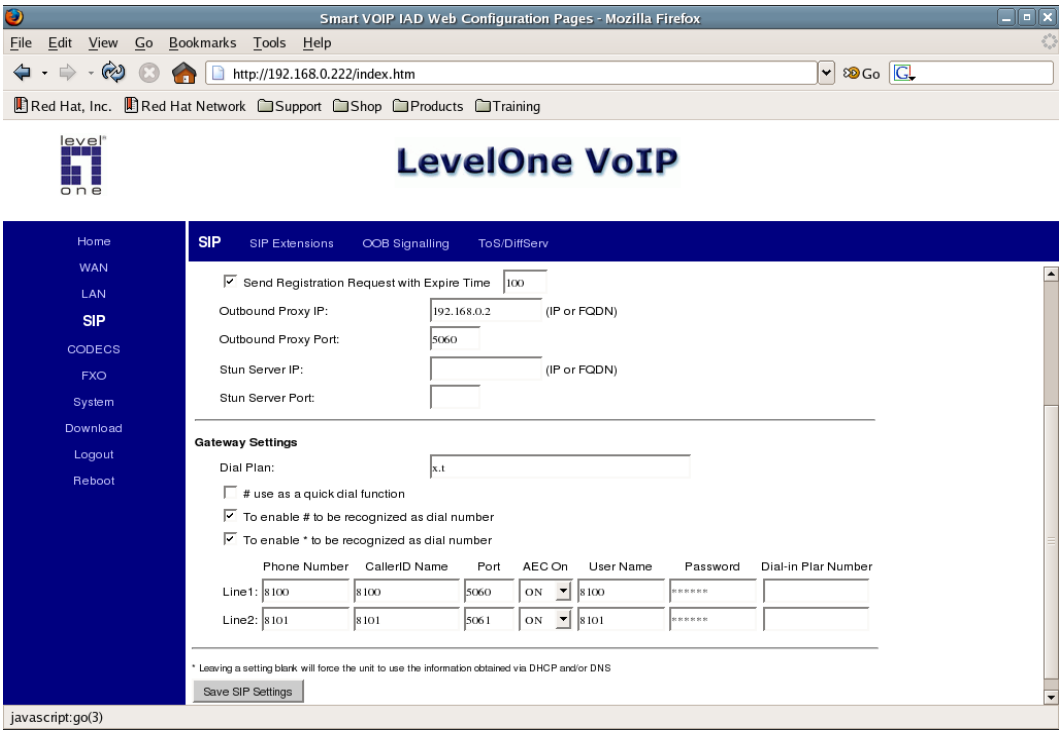


Figure 16.20: The SIP tab of LevelOne VoIP Administration Panel

Additional information pertaining to SIP account in a SIP Proxy server needs to be included also on SIP menu, a the very bottom of the menu. These information include:

Phone number - username in the SIP Proxy Server

- Phone number, which is the username of a SIP Proxy Server
- Caller ID, the caller ID we want to use
- Password, the one to be used to register to SIP Proxy server

There are two SIP accounts that can be registered with SIP Proxy Server: Line 1 can be connected to the telephone line while line 2 to PSTN line which plug is available in LevelOne VOI-2100. Under SIP menu, there are other sub-menus such as SIP Extension, Out of Band (OOB) Signaling, ToS etc. However, you don't have to change these parameters, as VOI-2100 can still operate without the need to change these parameters.

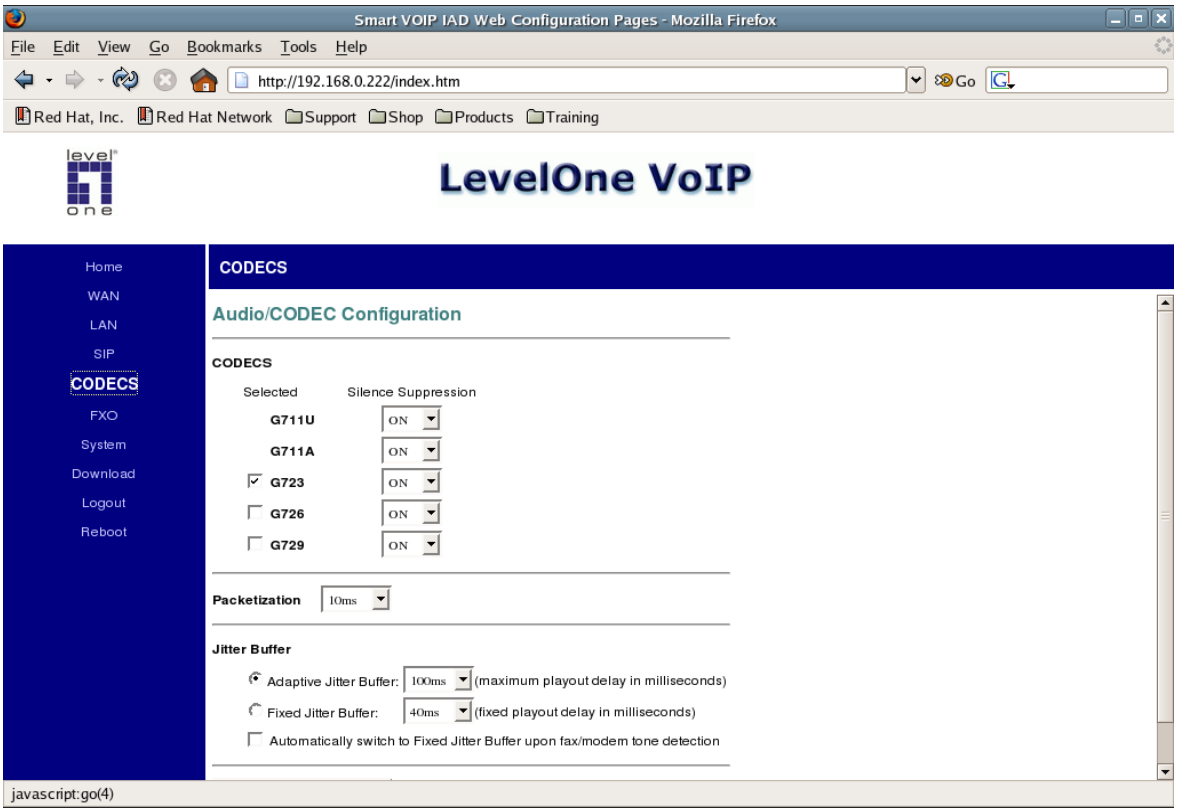


Figure 16.21: The CODECS tab of LevelOne VoIP Administration Panel

Now click on codecs. This option allows you to determine which voice compression method or codec that can be activated. Usually, it is better to activate all of them so you will have flexibility in communicating with a variety of softphones or IP phones, just in case a codec does not work properly and you have to switch to different one.

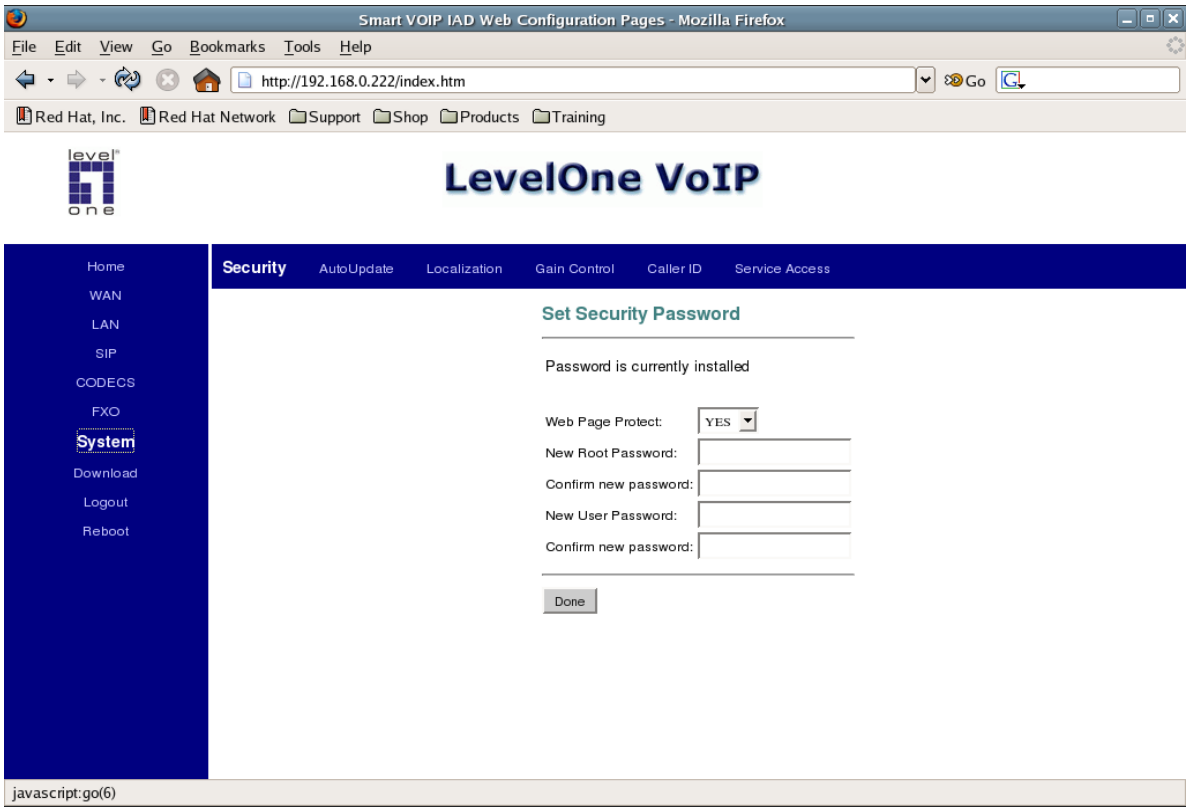


Figure 16.22: The Security tab of LevelOne VoIP Administration Panel

Now click on System, then to security. Under this tab, we can change the web administrator password needed to access LevelOne VOI-2100 web menu.

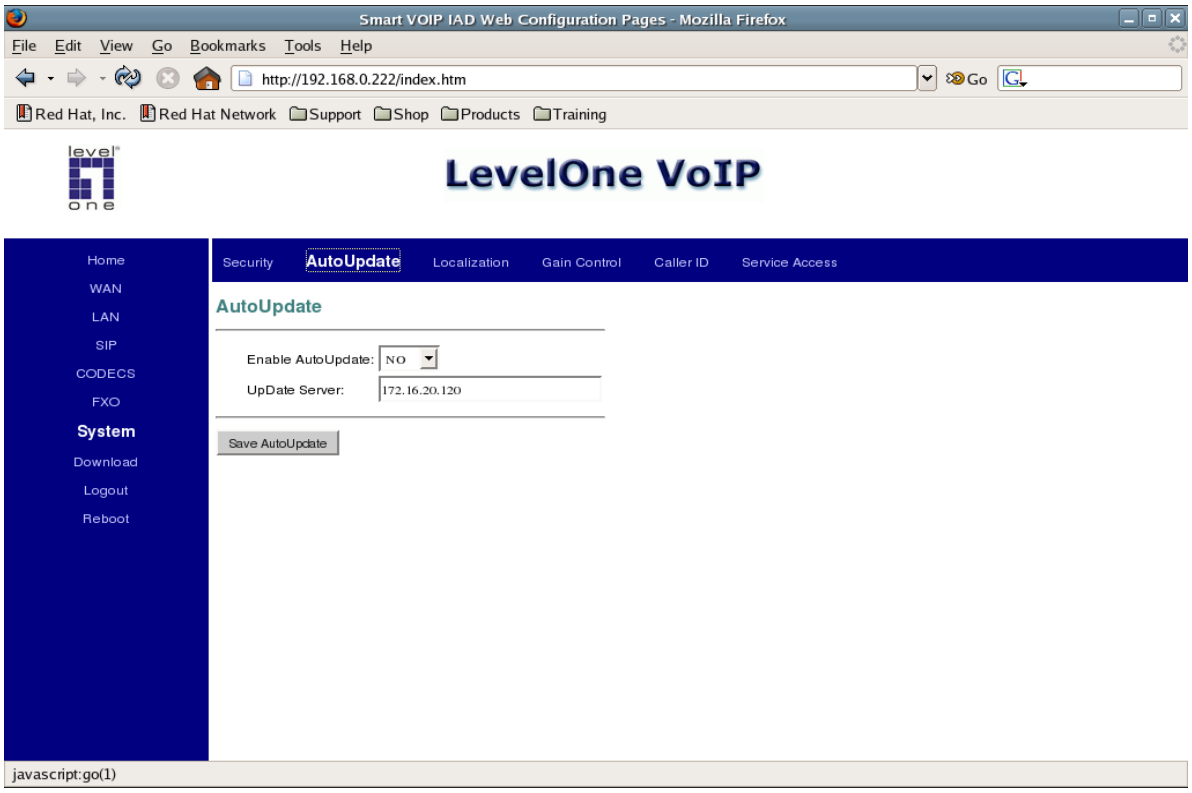


Figure 16.23: The AutoUpdate tab of LevelOne VoIP Administration Panel

Now click the AutoUpdate tab, next to Security. The submenu under this tab allows you to update firmware of LevelOne VOI-2100 automatically through the Internet.

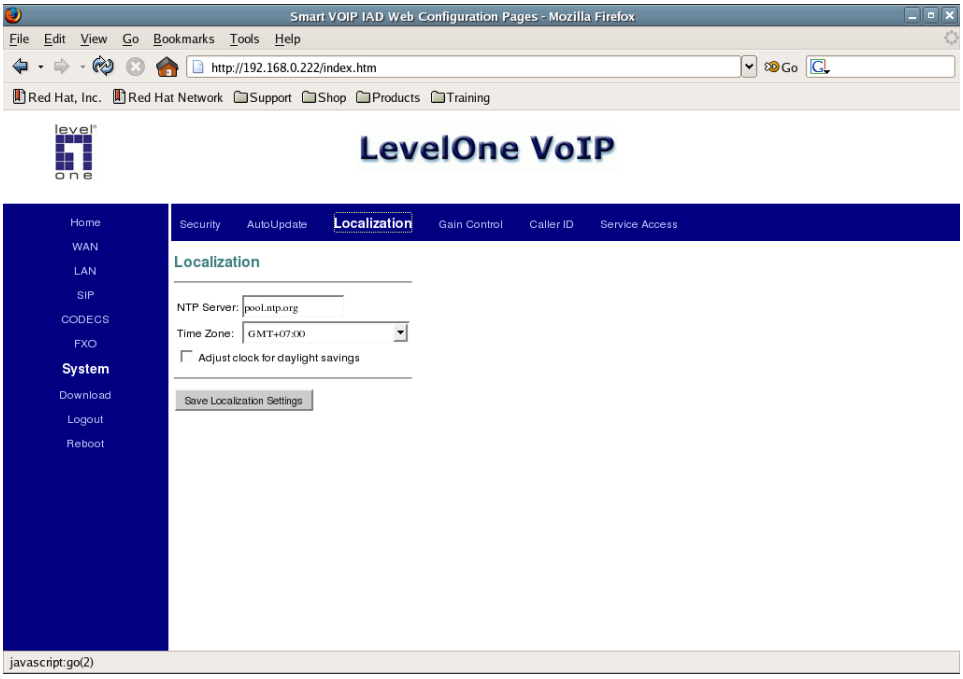


Figure 16.24: The Localization tab of LevelOne VoIP Administration Panel

Now click Localization. Set the time to synchronize our time to the server's in the internet and also set our location to the time zone for our location.

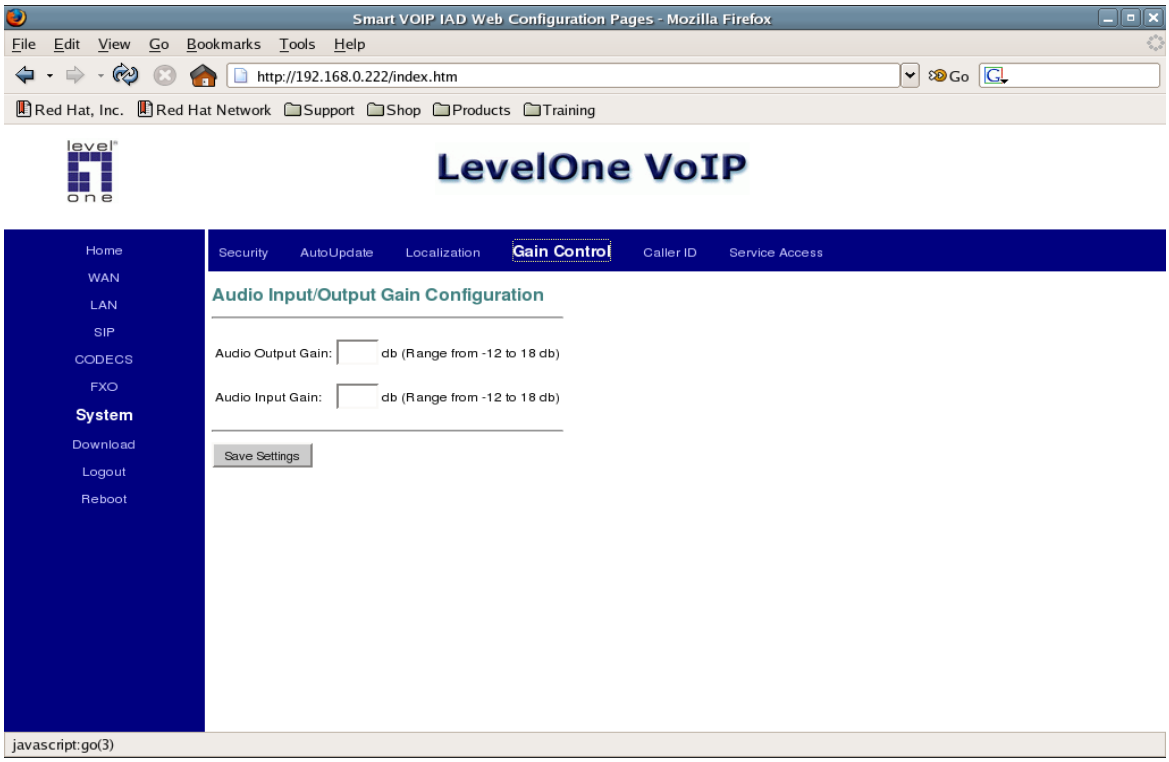


Figure 16.25: The Gain Control tab of LevelOne VoIP Administration Panel

Now click Gain Control. This tab allows us to adjust the volume of both audio output and audio input. This is measured in decibels. To decrease the volume, we need to enter negative audio gain values, such as -2 dB and so on. To increase the volume, put some positive integers.

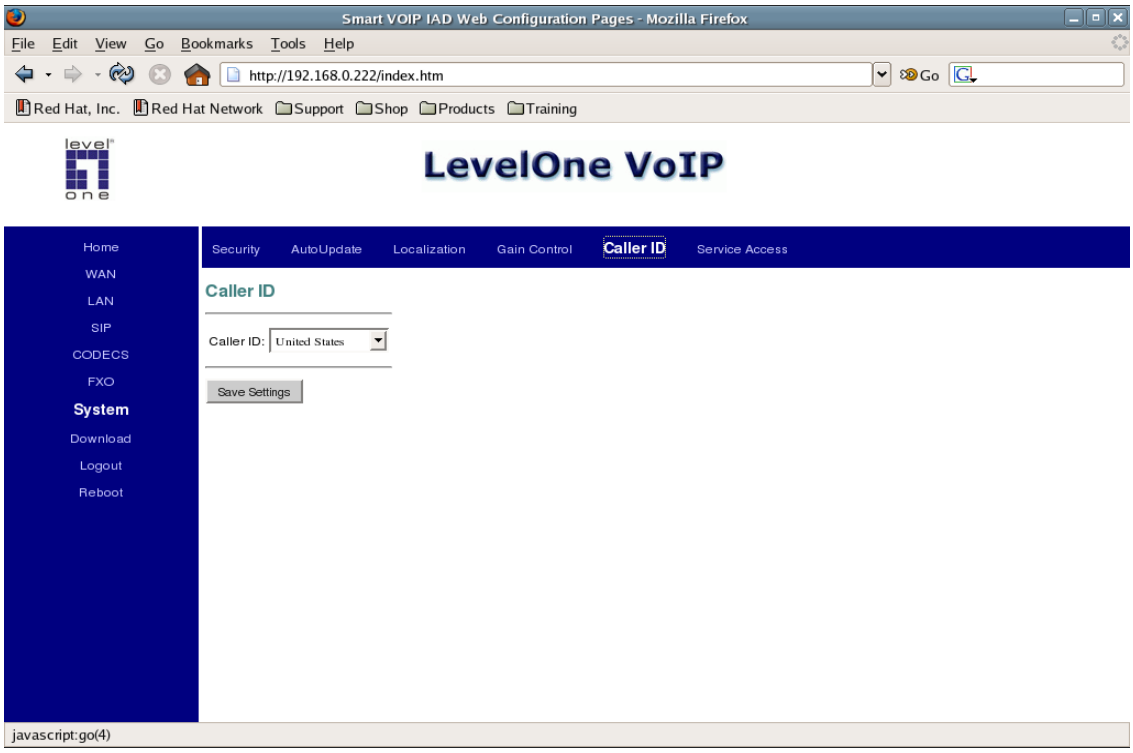


Figure 16.26: The Caller ID tab of LevelOne VoIP Administration Panel

Now click Caller ID. Choose the sort of caller ID you want to use.

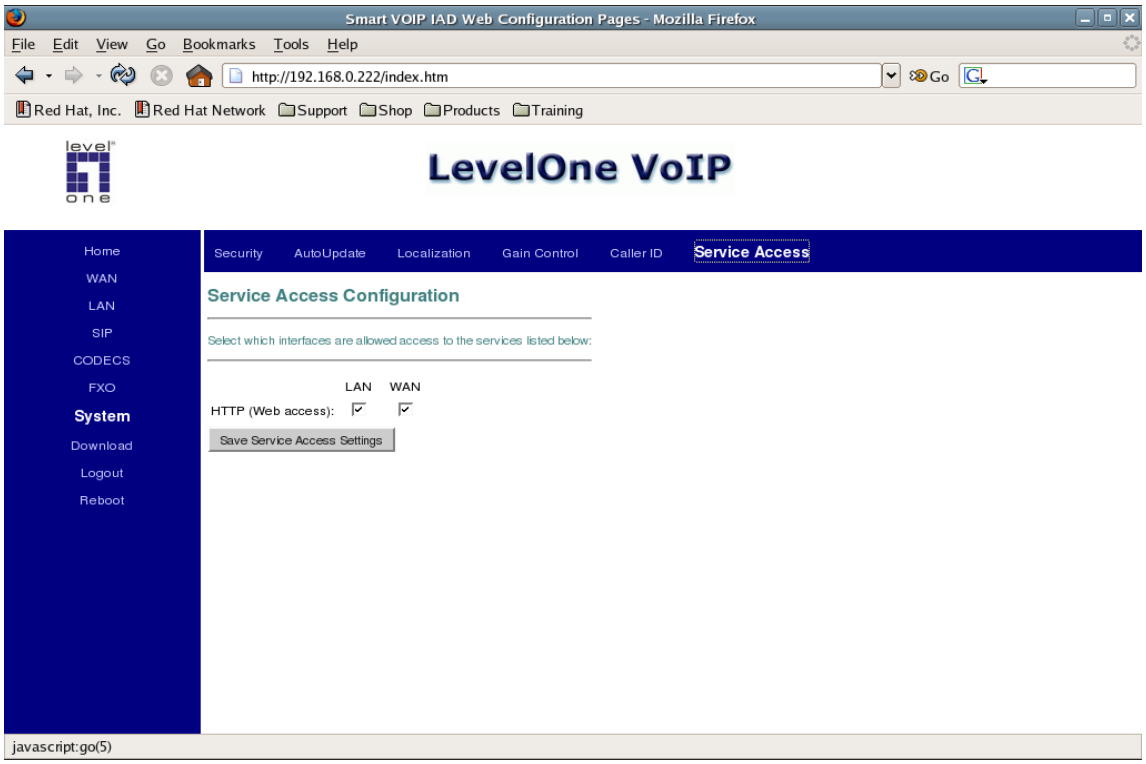


Figure 16.27: Service Access tab of LevelOne VoIP Administration Panel

Click Service Access, the last tab under System submenu. Service Access allows us to determine which interface will be accessible throough LevelOne VOI-2100 administration. The default configuration allows web administration access through LAN and WAN.

Linksys SPA400 with four FXOs

If you need many connections to PSTN, one of possible attractive alternatives to be used is SPA400 which has four FXOs which can be connected directly to PSTN. In addition, SPA400 also has a USB storage to store voicemail.

Figure 16.28: Linksys SPA400 with Four FXOs

Using the SPA400 with Asterisk

Steps that need to be carried out to link SPA400 to Asterisk are as follows:

Configuring the SPA400 IP address

- Configuring SPA400 IP address
- Configuring Asterisk account in SPA400
- Configuring sip.conf in Asterisk to have it registered to SPA400
- Conguring extensions.conf in Asterisk so it dial out using SPA400

Making all these configurations is not difficult and can be done through the web. The default username is Admin (Case sensitive) without a password.

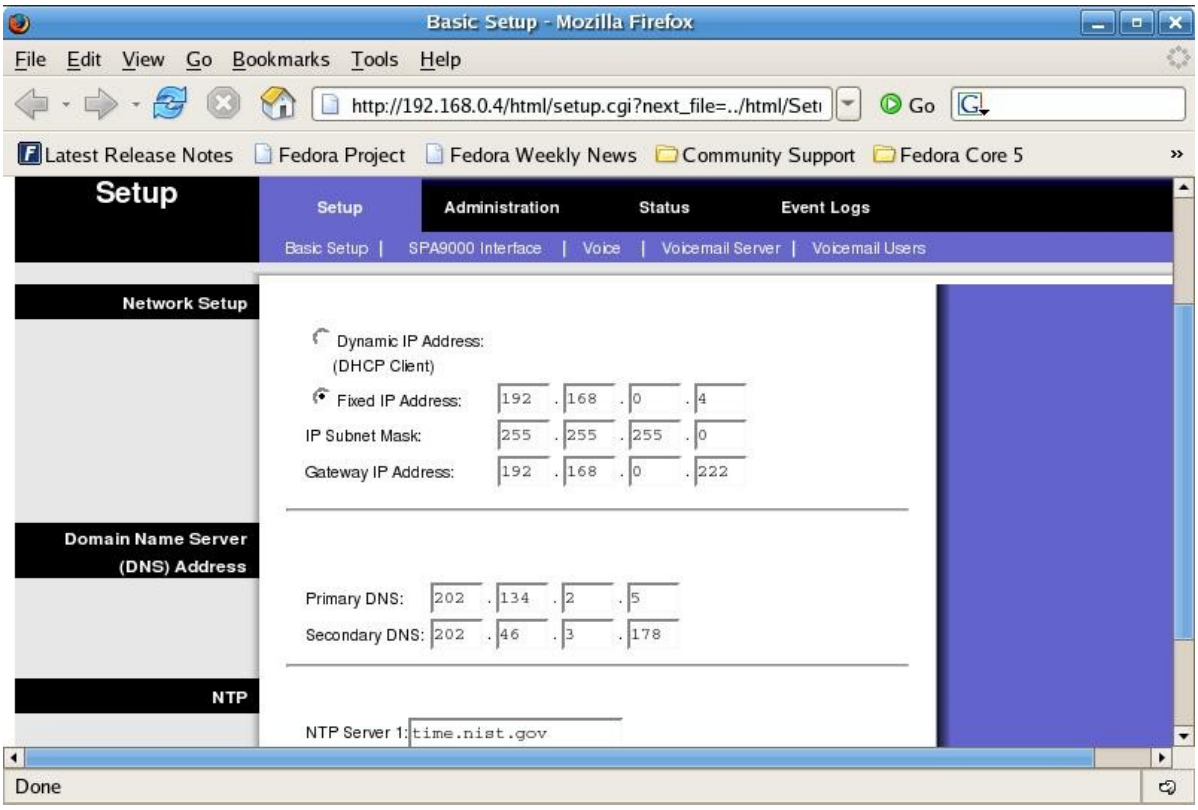


Figure 16.29: The Setup tab of SPA400 VoIP Administration Panel

To configure the IP address of SPA 400, click Setup and under this tab, click Basic Setup. Do not use Dynamic IP Address, since Asterisk needs to seek SPA400 and register itself to SPA400. Instead, choose Fixed IP Address. If necessary, we can also set the DNS and NTP server we often use. Obtain the information on DNS server from your internet service provider. For NTP server, type in time.nist.gov or pool.ntp.org. After all configuration is completed, click Save Settings.

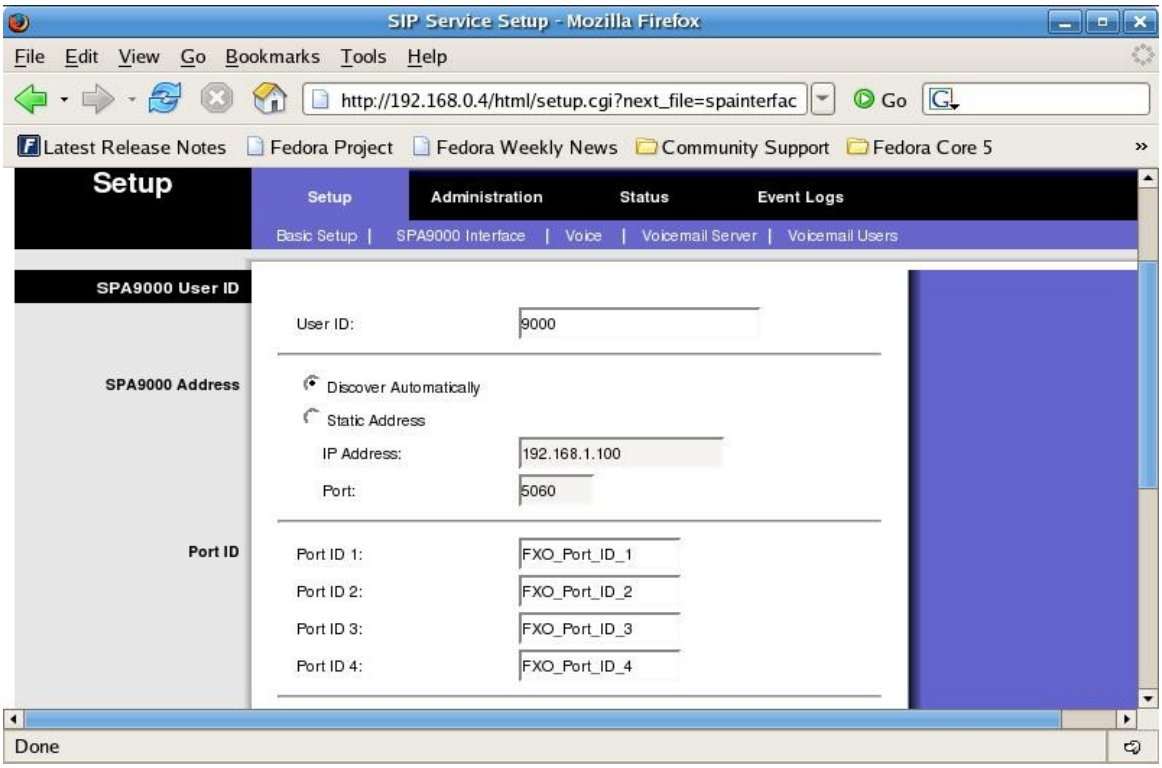


Figure 16.30: The Setup tab of SPA400 VoIP Administration Panel

Next, configure the account so either Asterisk or SPA9000 will be able to log into SPA400. The way to do this is to go into the Setup menu and click SPA9000 Interface. Change the user ID to the username we use to log in. Here we use "9000" as an example. Then choose Discover Automatically. Provided this setting works properly, you may want to change this setting in order to have a more secure connectivity, by setting the values of Asterisk server to match those of SPA400 server. Once everything is completed, click Save Settings to save the configuration.

Configure Asterisk to talk to Linksys SPA400

On Asterisk /etc/asterisk/sip.conf, you need to configure the account exactly similar to User ID of SPA400

The entries in sip.conf to enable Asterisk register to SPA400 are as follow:

```
[general]
register => 9000@192.168.0.6/9000
```

Replace 9000 with the value you entered in the User ID of SPA400, and replace 192.168.0.2 with the IP address of the SPA400.

Create a SIP entry for SPA400, with the following information:

```
901     user: User ID of SPA400
902     host: IP address of SPA400
903     context: the context that will be used to handle inbound calls from SPA400
```

SIP entry to receive calls from SPA400 are as the following:

```
[9000]
type=friend
user=9000
host=192.168.0.6
dtmfmode=rfc2833
canreinvite=no
context=from-trunk
insecure=very
```

To see whether you are registered to Asterisk or not, you can carry out the following command:

```
localhost*CLI> sip show registry
Host                Username        Refresh State
192.168.0.6:5060      9000          105 Registered
```

In Extension.conf file we can configure the routing for dial-out using SPA400. An example of a generic configuration for dial-out route by pressing 9 and enter SPA400 FXO trunk is as follows:

```
[general]
Trunk=SIP/9000
TRUNKMSD = 1

[trunkint]
;
; International long distance through trunk
;
exten => _9011.,1,Macro(dundi-e164,${EXTEN:4})
```

```
exten => _9011.,n,Dial(${TRUNK}/${EXTEN:${TRUNKMSD}})

[trunkld]
;
; Long distance context accessed through trunk
;
exten => _91NXXNXXXXXX,1,Macro(dundi-e164,${EXTEN:1})
exten => _91NXXNXXXXXX,n,Dial(${TRUNK}/${EXTEN:${TRUNKMSD}})

[trunklocal]
;
; Local seven-digit dialing accessed through trunk interface
;
exten => _9NXXXXXX,1,Dial(${TRUNK}/${EXTEN:${TRUNKMSD}})

[trunktollfree]
;
; Long distance context accessed through trunk interface
;
exten => _91800NXXXXXX,1,Dial(${TRUNK}/${EXTEN:${TRUNKMSD}})
exten => _91888NXXXXXX,1,Dial(${TRUNK}/${EXTEN:${TRUNKMSD}})
exten => _91877NXXXXXX,1,Dial(${TRUNK}/${EXTEN:${TRUNKMSD}})
exten => _91866NXXXXXX,1,Dial(${TRUNK}/${EXTEN:${TRUNKMSD}})
```

Note that the SPA400's account number in Asterisk is 9000, the number we are using as an example.

Incoming call routing is more complex. If we assume the incoming call will be connected to extension 200, then the configuration is approximately as follows:

```
[from-trunk]
include => from-pstn
...

[from-pstn]
include=> from-pstn-custom
...

[from-pstn-custom]
exten =>9000,1,Goto(ext-local,200,1)
```

Connect PSTN using Linksys SPA9000 and Linksys SPA400

Connecting PBX softswitch like Linksys SPA9000 to the PSTN can be carried out in several ways. One of them is to use the Linksys SPA400 as mediator to PSTN. What you have to do is to make SPA400's IP address to be fixed, enable User ID to register itself to SPA400, enable SPA9000 to register itself to SPA400, and enable SPA9000 to use SPA400's trunk for PSTN calls. SPA400 configuration can be done through the web, using the given default username without a password.

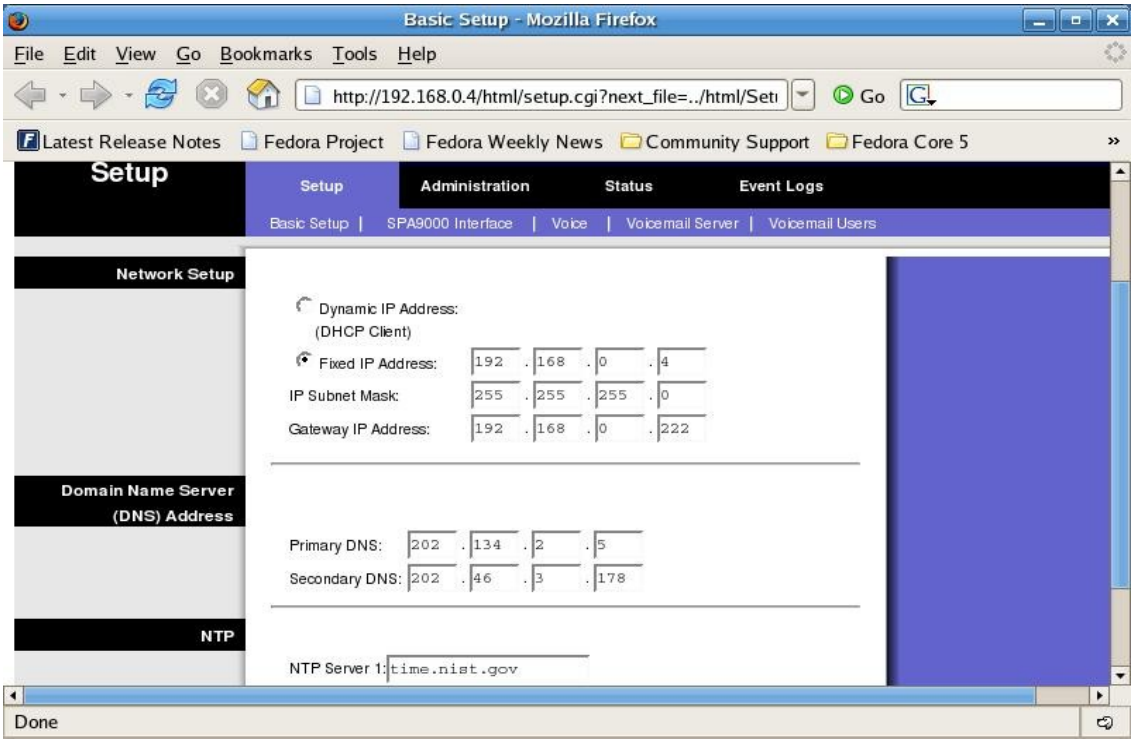


Figure 16.31: The Setup tab of SPA400

Through the Setup menu, click Basic Setup. Here we have configure fixed IP address, IP subnet mask, and gateway IP address, information on Domain Name Server (DNS) Address and NTP. For our example, we use 202.134.2.5, 202.46.3.178, and time.nist.gov for Primary DNS, Secondary DNS and NTP Server, respectively.

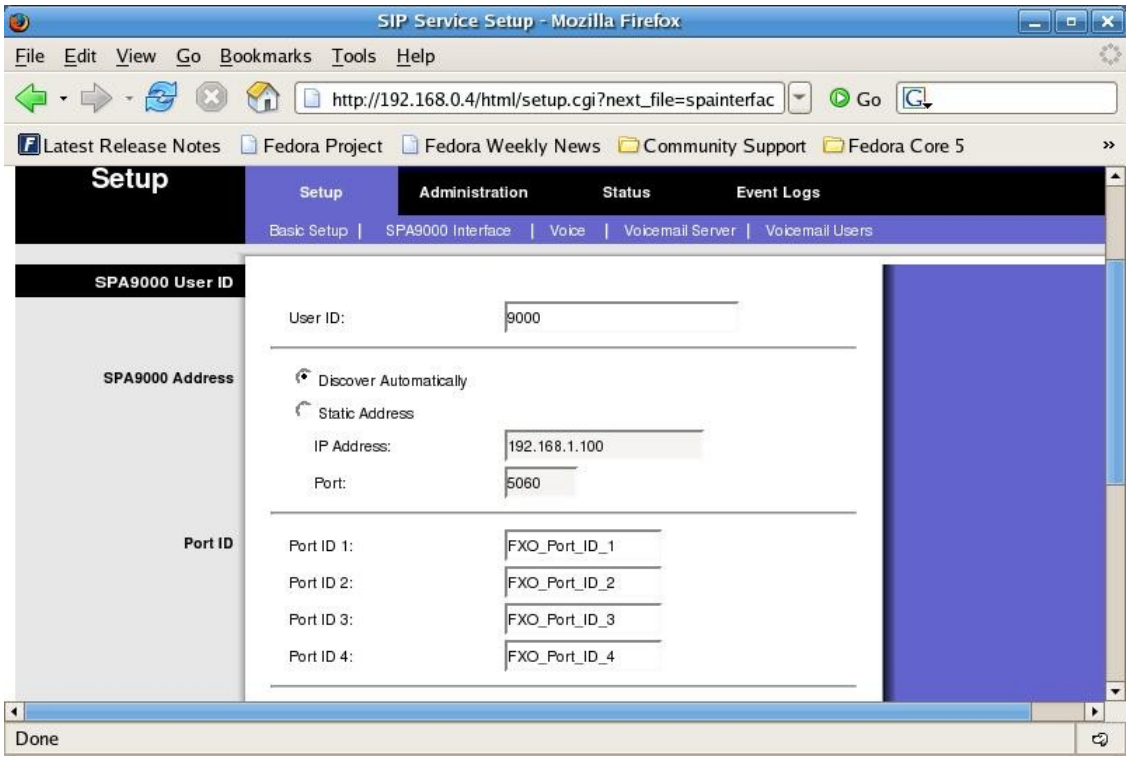


Figure 16.32: The Setup tab of SPA400

Through the Setup menu, click SPA9000 Interface. The configuration you have to do is as the following:

Create an account on SPA400 such that SPA9000 softswitch can register.

- Create an account in SPA400 so a softswitch like SPA9000 can be registered.
- Configure SPA400 so it will know the IP address and port of the softswitch/SPA9000. It is recommended that you choose “Discover Automatically”

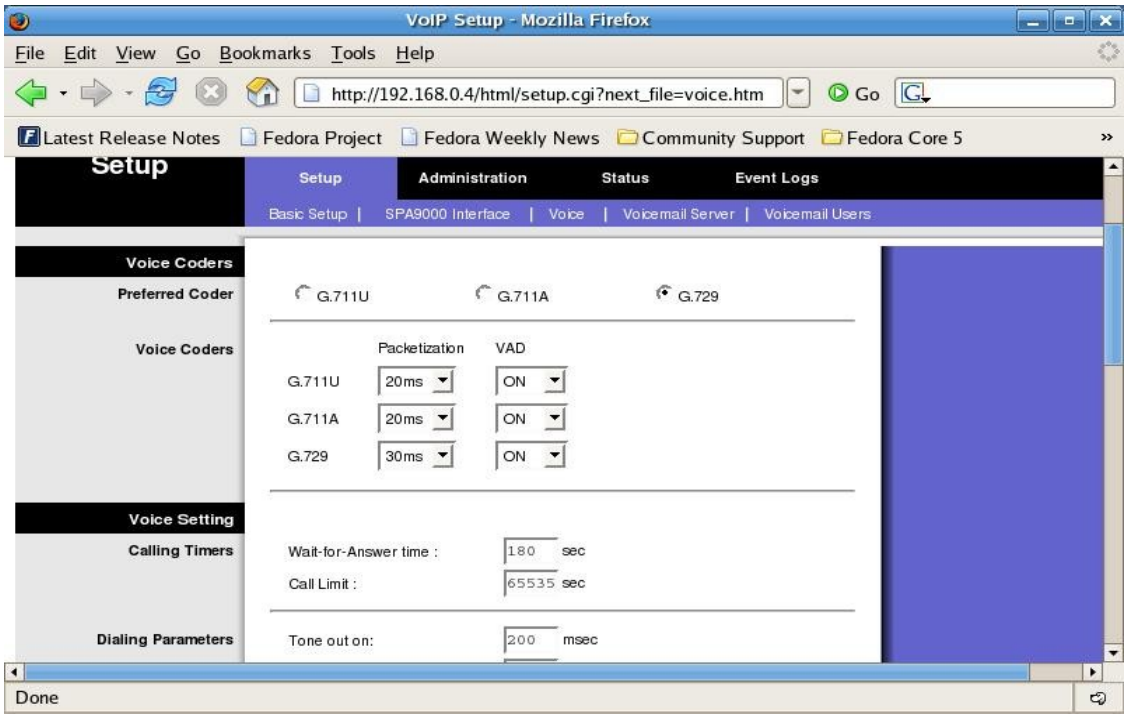


Figure 16.33: Voice Menu in SPA400

On the Setup menu, click Voice. Here we can configure the sort of codec we want to use and other setting parameters such as Wait-for-Answer time.

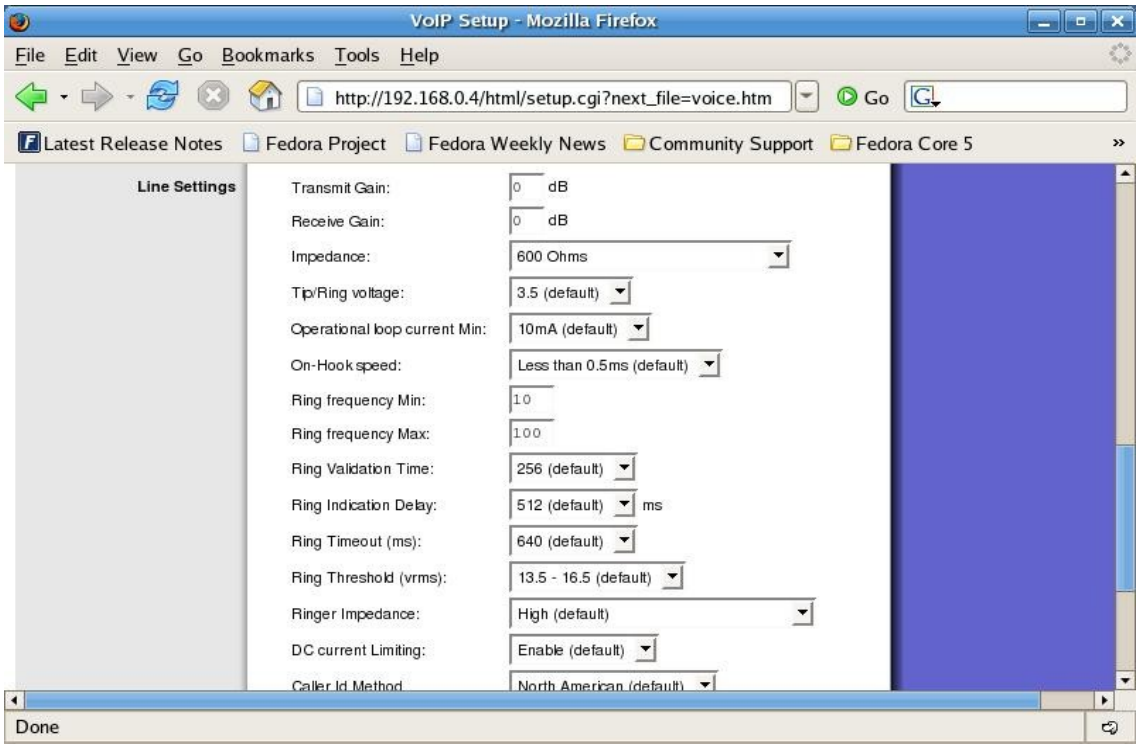


Figure 16.34: Menu Line Settings in Linksys SPA400

On the Setup menu, click Voice. In the Line Settings tab, you can configure the transmit gain, receive gain, impedance, ring voltage, on-hook speed, etc.. Basically, we do not need to change these values of these parameters, and simply use the default values.

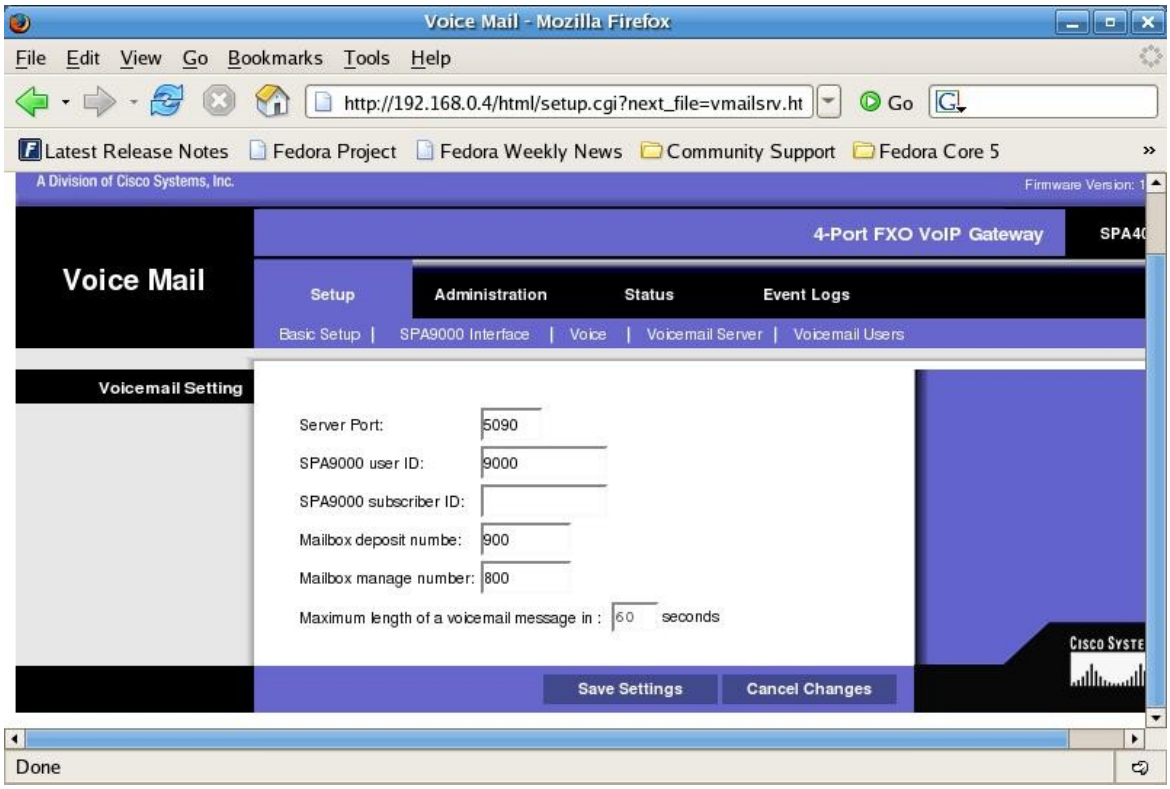


Figure 16.35: Voicemail Menu in SPA400

Through the Setup menu, click the Voicemail Server tab. Here we can set some important parameters of the Voicemail server, such as:

- Server Port – default 5090
- SPA9000 User ID – user ID for the SIP Proxy to register to Linksys SPA400
- SPA9000 subscriber ID – subscriber ID for the SIP Proxy to obtain Voicemail
- Mailbox Deposit Number – the number needed to put voicemail. The default value is 900.
- Mailbox manage number – the number used to manage Voicemail. The default value is 800.

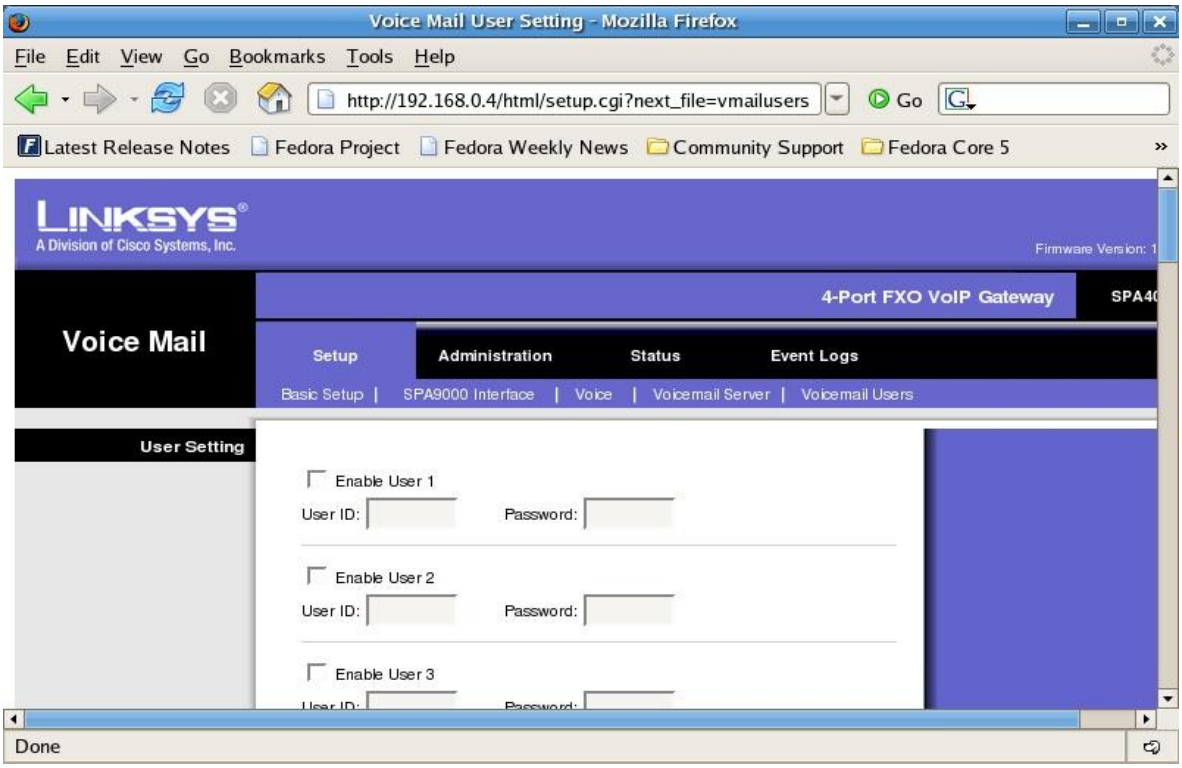


Figure 16.36: Voicemail User Menu in SPA400

One of many great things about SPA400 is its capability to store Voicemail data in USB storage. Through the Setup menu and Voicemail Users, we configure the user and password of our Voicemail. By clicking on the "enable" box of any user, we can now activate that user account.

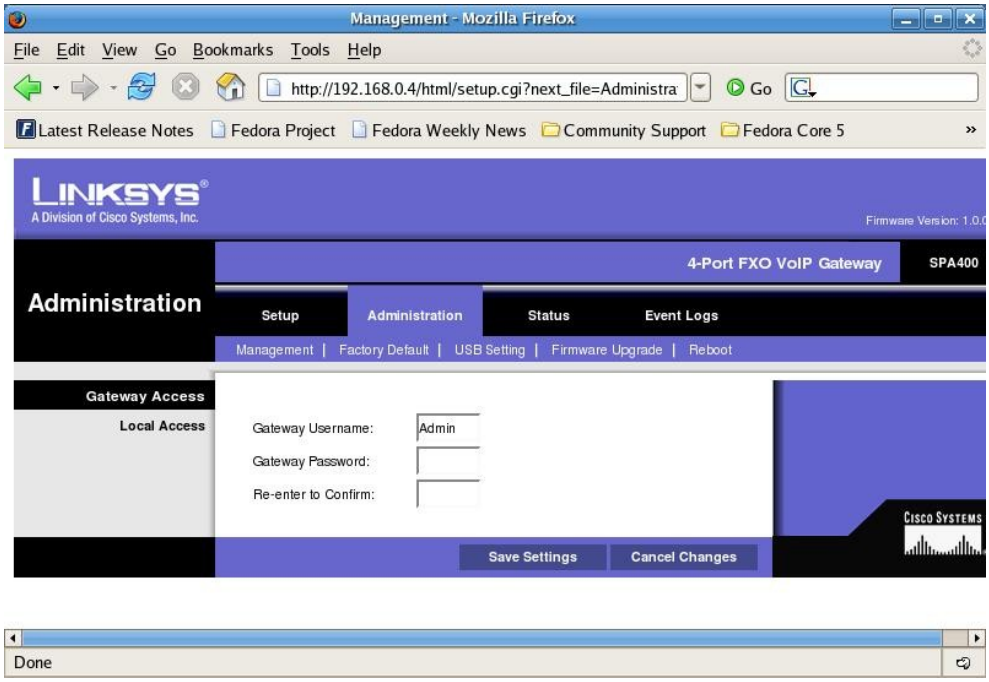


Figure 16.37: The Administration tab of Linksys SPA400

Go to Administration, then click Management. Here we can configure our username and password from Gateway Administrator. The default username is “Admin”, without a password.

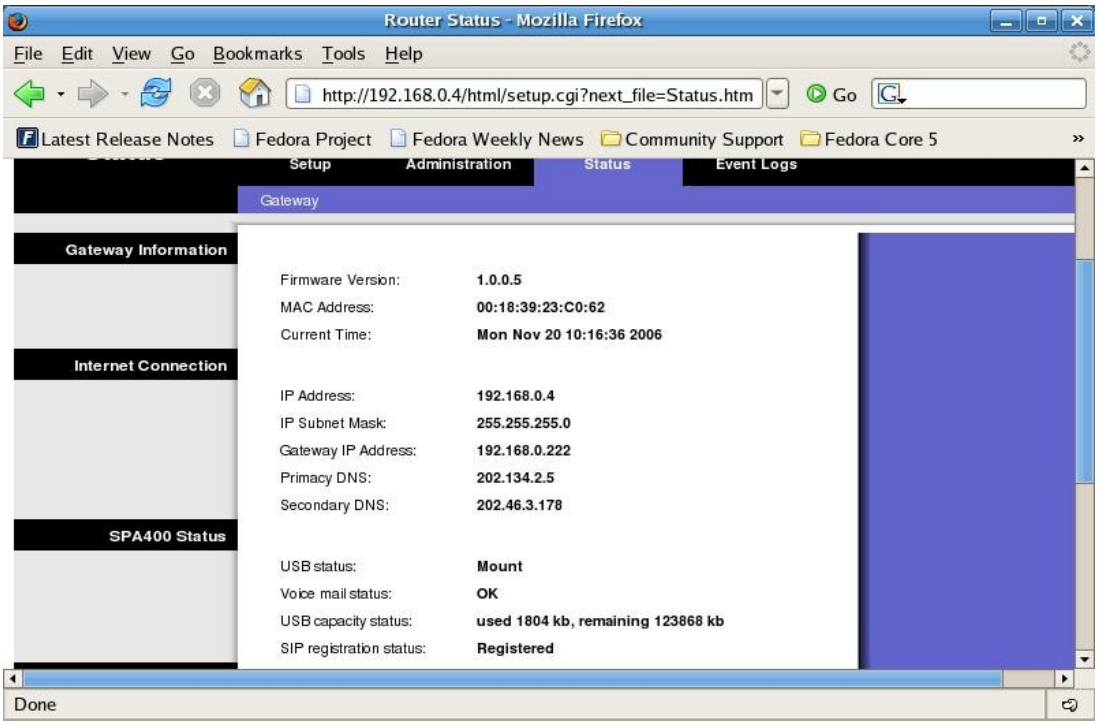


Figure 16.38: The Status tab of Linksys SPA400

Through the status menu, you will find more information on SPA400 operation, such as, time, IP address, subnet, gateway, DNS, USB disk etc. The most important parameter is the SIP Registration Status. It indicates the condition of the SIP Proxy we use: whether SPA9000 or Asterisk is successfully registered with SPA400 or not. If the SIP Proxy is successfully registered with SPA400, you can use SPA400 as an Analog Telephony Adapter to call to Telkom.

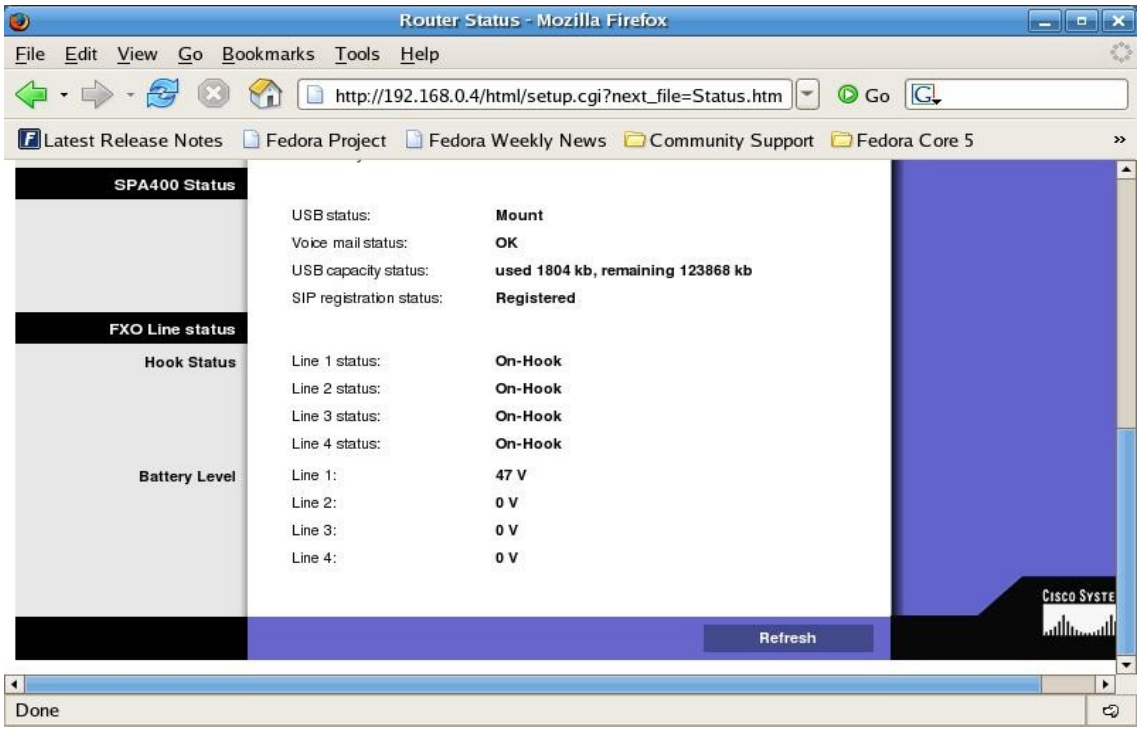


Figure 16.39: The Status tab of SPA400

While you are still in status pane, notice that there is also the status for Battery Level. This shows the voltage available at RJ-11 port of SPA400. If it is connected to your PSTN provider, then the voltage normally is about 45 V. If it is connected to PABX, the voltage experienced by the RJ-11 port is around 24 V. If all ports have their voltage level 0 V, then SPA400, when we attempt to place a call, will get a busy signal.

Configure Linksys SPA9000 to talk to Linksys SPA400

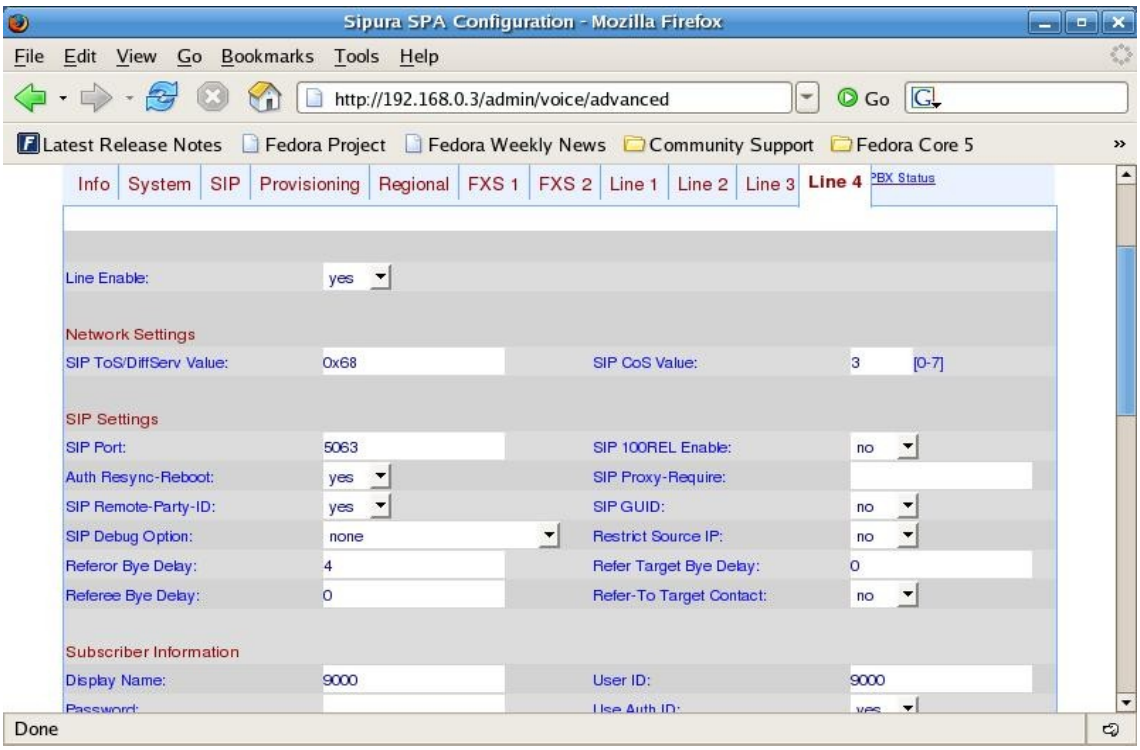


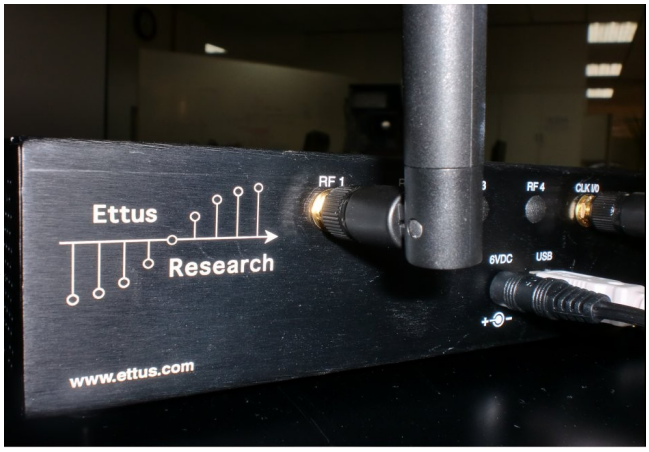
Figure 16.40: The Line 4 tab of SPA400

To have SPA9000 capable of communicating with SPA400, you need to register the user ID you have set in SPA9000 to SPA400. Go to admin menu, choose Advanced and choose one of the four lines. Under the Subscriber Information, you need to set the following parameters:

Display Name - according to SPA400.

- Display Name – according to SPA400
- User ID – according to SPA400
- Password – just leave it blank
- User Auth ID – should be No
- Proxy – SPA400 IP address
- Outbound Proxy – SPA400 IP address

CHAPTER 17: OpenBTS



OpenBTS (Open Base Transceiver Station) is a software based GSM BTS that allows GSM mobile phone to operate without using cellullar operator infrastructure. OpenBTS is known to be the first open source implementation of industrial standard GSM protocol stack.

The heart of OpenBTS is an Universal Software Radio Peripheral (USRP). USRP may be obtained from Ettus http://www.ettus.com/products. The estimate cost of the USRP hardware needed for OpenBTS is around US$1500-2500 much lower than the normal commercial GSM equipments.

Open GSM Infrastructure

OpenBTS replaces traditional GSM infrastructure, especially behind the Base Transceiver Station (BTS). Traditionally, traffic will pass into Mobile Switching Center (MSC). In OpenBTS, traffic terminated on the same Box and forward the data to Asterisk IP PBX through SIP and Voice-over-IP (VoIP).

The reference Air Interface (Um) uses software-defined radio (SDR) on Universal Software Radio Peripheral (USRP) USB board.

History

The project is started by Harvind Samra http://www.linkedin.com/in/harvindsamra and David A. Burgess http://ecommconf.com/2009/speakers/davidburgess/. The aim of the project is to reduce the GSM cost in rural and developing countries to be under US$1 / month / subscriber.

Field Test

Field test is done in Nevada and North California, US. Temporary radio license for a short period is obtained through Kestrel Signal Processing (KSP) – used to be a consulting frim of the developer of OpenBTS.

Niue

In 2010, an OpenBTS is permanently installed in Nieu and it is for the first time a cellular BTS connected to the local telecommunication operator. Niue is a small country with 1700 in habitant amd not so much attracting mobile operator. Cost structure of OpenBTS fits to Niue which unable to procure the conventional GSM Base Station.

GNURadio

Reference http://gnuradio.org/redmine/wiki/gnuradio/UbuntuInstall. The needed development tools are:

- g++
- subversion
- make
- autoconf, automake, libtool
- sdcc
- guile
- ccache

The needed library for runtime and compilation processes are

- python-dev
- FFTW 3.X (fftw3, fftw3-dev)
- cppunit (libcppunit and libcppunit-dev)
- Boost 1.35 (or later)
- libusb and libusb-dev
- wxWidgets (wx-common) and wxPython (python-wxgtk2.8)
- python-numpy (via python-numpy-ext) (for SVN on or after 2007-May-28)
- ALSA (alsa-base, libasound2 and libasound2-dev)
- Qt (libqt3-mt-dev for versions earlier than 8.04; version 4 works for 8.04 and later)
- SDL (libsdl-dev)
- GSL GNU Scientific Library (libgsl0-dev >= 1.10 required for SVN trunk, not in binary repositories for 7.10 and earlier)

Library Installation

Update

```
sudo apt-get update
```

For Maverick (Ubuntu 10.10) we can use the following command

```
sudo apt-get -y install libfontconfig1-dev libxrender-dev libpulse-dev swig g++ automake \
libtool python-dev libfftw3-dev libcppunit-dev libboost-all-dev libusb-dev fort77 sdcc \
sdcc-libraries libsdl1.2-dev python-wxgtk2.8 subversion git-core guile-1.8-dev \
libqt4-dev python-numpy ccache python-opengl libgsl0-dev python-cheetah python-lxml \
doxygen qt4-dev-tools libqwt5-qt4-dev libqwtplot3d-qt4-dev pyqt4-dev-tools \
libpcre3 libpcre3-dbg libpcre3-dev libpcrecpp0
```

WxWidget Installation

Although this seems to be not critical. Those who wish to install the latest WxWidget can follow the following command.

Edit /etc/apt/sources.list

```
# wxWidgets/wxPython repository at apt.wxwidgets.org
deb http://apt.wxwidgets.org/ DIST-wx main
deb-src http://apt.wxwidgets.org/ DIST-wx main
```

An example for gutsy

```
# wxWidgets/wxPython repository at apt.wxwidgets.org
deb http://apt.wxwidgets.org/ gutsy-wx main
deb-src http://apt.wxwidgets.org/ gutsy-wx main
```

Do update

```
sudo apt-get update
```

Install

```
sudo apt-get install python-wxgtk2.8 python-wxtools wx2.8-i18n
sudo apt-get install python-wxgtk2.8 python-wxtools wx2.8-i18n libwxgtk2.8-dev libgtk2.0-dev
```

SWIG Installation

To manually install SWIG, we need to download the source code from
http://sourceforge.net/projects/swig/files/swig/

Then do the followings

```
cp swig-2.0.1.tar.gz /usr/local/src/
cd /usr/local/src/
tar zxvf swig-2.0.1.tar.gz
cd /usr/local/src/swig-2.0.1/
./configure
make
make install
```

QWT Installation

To manually install QWT, we need to download the source code from
http://sourceforge.net/projects/qwt/files/

Then do the followings

```
cp qwt-5.2.1.tar.bz2 /usr/local/src/
cd /usr/local/src/
tar jxvf qwt-5.2.1.tar.bz2
cd /usr/local/src/qwt-5.2.1/
qmake
make
make install
```

For those who brave may use the beta version such as

```
cp qwt-6.0.0-rc5.tar.bz2 /usr/local/src/
cd /usr/local/src/
```

```
tar jxvf qwt-6.0.0-rc5.tar.bz2
cd /usr/local/src/qwt-6.0.0-rc5
qmake
make
make install
```

GNURadio Installation

Download the source code from

http://gnuradio.org/redmine/wiki/gnuradio/Download

compile the source code

```
cp gnuradio-3.3.0.tar.gz /usr/local/src/
cd /usr/local/src/
tar zxvf gnuradio-3.3.0.tar.gz
cd /usr/local/src/gnuradio-3.3.0/
./configure
make
make check
make install
```

USRP Handling

Ubuntu uses udev to handle hotplug devices, and by default give no access to non-root to USRP. The following script will give access to user to handel USRP via USB for either live or hot-plug.

```
sudo addgroup usrp
sudo usermod -G usrp -a <YOUR_USERNAME>
echo 'ACTION=="add", BUS=="usb", SYSFS{idVendor}=="fffe",
SYSFS{idProduct}=="0002", GROUP:="usrp", MODE:="0660"' > tmpfile
sudo chown root.root tmpfile
sudo mv tmpfile /etc/udev/rules.d/10-usrp.rules
```

At this point, Ubuntu has been configured to know what it should do when detecting USRP in the USB. "udev" must be reload rules to load our new rules. The followings may do the trick without booting the

computer.

```
sudo udevadm control --reload-rules
```

or

```
sudo /etc/init.d/udev stop
sudo /etc/init.d/udev start
```

or

```
sudo killall -HUP udevd
```

We may check if USRP has been recongized by monitoring /dev/bus/usb after USRP is plugged using the following command

```
ls -lR /dev/bus/usb | grep usrp
```

we should see something like

```
crw-rw---- 1 root usrp 189, 1 2010-12-09 17:38 002
```

Everytime USRP is plugged it will be registered in group 'usrp' and mode 'crw-rw----'.

USRP Verification

Next we need to verify wether GNURadio can work properly with USRP. At this point we need to connect USRP to computer.

Check the USB speed to USRP

```
cd /usr/local/src/gnuradio-3.3.0/gnuradio-examples/python/usrp
./usrp_benchmark_usb.py
```

We will see something like

```
Testing 2MB/sec... usb_throughput = 2M
ntotal    = 1000000
nright    = 999918
runlength = 999918
delta     = 82
OK
```

```
Testing 4MB/sec... usb_throughput = 4M
ntotal    = 2000000
nright    = 1999492
runlength = 1999492
delta     = 508
OK
Testing 8MB/sec... usb_throughput = 8M
ntotal    = 4000000
nright    = 3998860
runlength = 3998860
delta     = 1140
OK
Testing 16MB/sec... usb_throughput = 16M
ntotal    = 8000000
nright    = 7997680
runlength = 7997680
delta     = 2320
OK
Testing 32MB/sec... usb_throughput = 32M
ntotal    = 16000000
nright    = 15995986
runlength = 15995986
delta     = 4014
OK
Max USB/USRP throughput = 32MB/sec
```

C++ interface to USRP, provide estimate maximum throughput between PC and USRP

```
cd /usr/local/src/gnuradio-3.3.0/usrp/host/apps
./test_usrp_standard_tx
./test_usrp_standard_rx
```

Typical result from USRP_standard_tx test

```
which:    0
interp:   16
rf_freq:  -1
amp:      10000.000000
nsamples: 3.2e+07
```

```
Subdevice name is Flex 900 Tx MIMO B
Subdevice freq range: (7.5e+08, 1.05e+09)
mux: 0x000098
baseband rate: 8e+06
target_freq:     900000000.000000
ok:              true
r.baseband_freq: 904000000.000000
r.dxc_freq:      -4000000.000000
r.residual_freq: 0.000000
r.inverted:      0
tx_underrun
tx_underrun
tx_underrun
tx_underrun
tx_underrun
tx_underrun
tx_underrun
tx_underrun
tx_underrun
xfered 3.2e+07 bytes in 1.01 seconds.  3.154e+07 bytes/sec.  cpu time = 0.16
9 underruns
```

Typical result from USRP standard RX test

```
xfered 1.34e+08 bytes in 4.19 seconds.  3.2e+07 bytes/sec.  cpu time = 0.8681
noverruns = 0
```

If needed, we can upgrade the whole system

```
sudo apt-get -y upgrade
```

Then reboot and upgrade the distro

```
sudo apt-get -y dist-upgrade
```

OpenBTS Installation

Before we do OpenBTS installation, we need to compile and install GNURadio. Without GNURadio installed, OpenBTS may not be installed. We need to install additional library

```
apt-get install libosip2-4 libosip2-dev libortp8 libortp-dev
```

Download the source code from

```
http://www.openbts.org
http://sourceforge.net/projects/openbts/
```

Then do the followings

```
cp openbts-2.6.0Mamou.tar.gz /usr/local/src/
cd /usr/local/src/
tar zxvf openbts-2.6.0Mamou.tar.gz
cd /usr/local/src/openbts-2.6.0Mamou/
./configure
make
make all
make install
```

OpenBTS is compiled and installed. To enable SMS facility in OpenBTS, we need to compile the smqueue separately. For strange reason, we need to install g++-4.3 to compile smqueue

```
apt-get install g++-4.3
```

Edit Makefile.standalone file of smqueue

```
vi /usr/local/src/openbts-2.6.0Mamou/smqueue/Makefile.standalone
```

Replace the g++

```
g++ -o smqueue $(CPPFLAGS) $(INCLUDES) smqueue.cpp smnet.cpp smcommands.cpp
../HLR/HLR.cpp $(LIBS)
```

to

```
g++-4.3 -o smqueue $(CPPFLAGS) $(INCLUDES) smqueue.cpp smnet.cpp
smcommands.cpp ../HLR/HLR.cpp $(LIBS)
```

Compile smqueue

```
cd /usr/local/src/openbts-2.6.0Mamou/smqueue/
make -f Makefile.standalone
```

If we use g++ 4.4 we will see the following error

```
smnet.cpp:423: error: invalid conversion from ‘const char*’ to ‘char*’
make: *** [smqueue] Error 1
```

Compilation of smqueue of OpenBTS is done.

A Glimpse on OpenBTS Configuration

Some specification of OpenBTS configuration

AsteriskConfig	Asterisk configuration files for use with OpenBTS.
CommonLib	Common-use libraries, mostly C++ wrappers for basic facilities.
Control	Control-layer functions for the protocols of GSM 04.08 and SIP.
GSM	The GSM stack.
SIP	Components of the SIP state machines ued by the control layer.
SMS	The SMS stack.
TRXManager	The interface between the GSM stack and the radio.
Transceiver	The software transceiver and specific installation tests.
apps	OpenBTS application binaries.
doc	Project documentation.
tests	Test fixtures for subsets of OpenBTS components.
smqueue	RFC-3428 store-and-forward server for SMS

OpenBTS assume the following UDP port

5060 -- Asterisk SIP interface
5061 -- local SIP softphone
5062 -- OpenBTS SIP interface

5063 -- smqueue SIP interface
5700-range -- OpenBTS-transceiver interface

These ports can set via configuration file apps/OpenBTS.config. For those who do OpenBTS installtion for the first time, need to copy OpenBTS.config file

```
cd /usr/local/src/openbts-2.6.0Mamou/apps
cp OpenBTS.config.example OpenBTS.config
```

If needed, we can edit the configuration file

```
vi /usr/local/src/openbts-2.6.0Mamou/apps/OpenBTS.config
```

Most of the default parameter may be used as it is. Sometimes, we need to change the network information such as

```
# Network and cell identity.

# Network Color Code, 0-7
# Also set GSM.NCCsPermitted later in this file.
GSM.NCC 0
# Basesation Color Code, 0-7
GSM.BCC 2
# Mobile Country Code, 3 digits.
# MCC MUST BE 3 DIGITS.  Prefix with 0s if needed.
# Test code is 001.
GSM.MCC 001
# Mobile Network Code, 2 or 3 digits.
# Test code is 01.
GSM.MNC 01
# Location Area Code, 0-65535
GSM.LAC 1000
# Cell ID, 0-65535
GSM.CI 10
```

smqueue Configuration

Disable IPv6 by editing /etc/default/grub change

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
```

into

```
GRUB_CMDLINE_LINUX_DEFAULT="ipv6.disable=1 quiet splash"
```

After save and exit, update grub using

```
sudo update-grub
```

Edit smqueue configuration, copy smqueue.config.example to smqueue.config

```
cd /usr/local/src/openbts-2.6.0Mamou/smqueue/
cp smqueue.config.example smqueue.config
```

smqueue config file is in ./smqueue/smqueue.config.

```
vi /usr/local/src/openbts-2.6.0Mamou/smqueue/smqueue.config
```

add to the config file the following command to limit alarm for SMS registration

```
Log.Alarms.Max 10
```

create savedqueue.txt in ./smqueue directory

```
touch /usr/local/src/openbts-2.6.0Mamou/smqueue/savedqueue.txt
```

Run smqueue

```
sudo su
cd /usr/local/src/openbts-2.6.0Mamou/smqueue/
./smqueue &
```

If it runs correctly, we will see something like

```
1296968828.6030 INFO 3077809872 smnet.cpp:319:listen_on_port: Listening at address '0.0.0.0:5063'.
1296968828.6045 INFO 3077809872 smqueue.cpp:2222:main: My own IP address is configured as 127.0.0.1
1296968828.6045 INFO 3077809872 smqueue.cpp:2223:main: The HLR registry is at 127.0.0.1:5060
1296968828.6046 INFO 3077809872 smqueue.cpp:2110:read_queue_from_file: === Read 0 messages total, 0 bad ones.
1296968828.6047 INFO 3077809872 smqueue.cpp:2230:main: Queue contains 0 msgs.
1296968828.6048 INFO 3077809872 smqueue.cpp:1852:main_loop: === Feb  6 12:07:08 0 queued; waiting.
```

Small Kludge in smqueue :(

We still have alot of problem in the SMS. To make things "easier" it might be helpfull in the registration process if we could edit the subscriber table in smqueue.cpp such as

```
cd /usr/local/src/openbts-2.6.0Mamou/smqueue
vi smqueue.cpp
```

such that

```
/* ==== FIXME KLUDGE ====
 * Table of IMSIs and phone numbers, for translation.
 * This is only for test-bench use.  Real life uses the Home Location
* Register (../HLR), currently implemented via Asterisk.
*/
static
struct imsi_phone { char imsi[4+15+1]; char phone[1+15+1]; } imsi_phone[] = {
	{"IMSI666410186585295", "+17074700741"},	/* Nokia 8890 */
	{"IMSI777100223456161", "+17074700746"},	/* Palm Treo */
	{"IMSI510110301694405", "2101"},	/* Bob */
	{"IMSI238209700014858", "2102"},	/* SB */
	{"IMSI310260254136340", "2103"},	/* Steve */
	{"IMSI520189606386106", "2104"},
	{{0}, {0}}
};
```

Asterisk Configuration to work with OpenBTS

Integration OpenBTS to Asterisk is principally very simple. Every SIM card may be configured as SIP user using IMSI as username. Process to add OpenBTS subscriber through the following two steps:

1. Get IMSI from the SIM. This is implemented by sending SMS as mobile phone connect to

OpenBTS.

2. Edit sip.conf and extensions.conf to support the new SIP user.

Thus, in principal, there is not much to configure Asterisk to be able to talk to OpenBTS. We need to edit

```
/etc/asterisk/sip.conf
/etc/asterisk/extensions.conf
```

Example of Asterisk Configuration can be found in

```
/usr/local/src/openbts-2.6.0Mamou/AsteriskConfig
```

Example of /etc/asterisk/sip.conf is as follows

```
[IMSI510110301694405]
callerid=2101
canreinvite=no
type=friend
callerid=2101
; context=sip-external
allow=gsm
host=dynamic

[IMSI520010104743577]
callerid=21011
canreinvite=no
type=friend
allow=gsm
context=sip-external
host=dynamic
```

Example of /etc/asterisk/extensions.conf is as follows

```
exten => 2101,1,Dial(SIP/IMSI510110301694405,60,rt)
exten => 2102,1,Dial(SIP/IMSI238209700014858,60,rt)
exten => 2103,1,Dial(SIP/IMSI310260254136340,60,rt)
```

IMSI520154100006647 is obtained from the SMS received by the OpenBTS user.

Automatic SIM Registration

As mention at http://gnuradio.org/redmine/wiki/gnuradio/OpenBTSThe_use_of_autocreatepeer=yes we may add some parameters in /etc/asterisk/sip.conf to enable automatic SIM registration

```
[general]
allowoverlap=no   ; Disable overlap dialing support. (Default is yes)
bindport=5060     ; UDP Port to bind to (SIP standard port is 5060)
bindaddr=0.0.0.0  ; IP address to bind to (0.0.0.0 binds to all)
srvlookup=yes     ; Enable DNS SRV lookups on outbound calls

; line untuk automatic sim registration
autocreatepeer=yes
canreinvite=no
call-limit=1
type=friend
allow=gsm
context=sip-internal
host=127.0.0.1   ; assuming OpenBts and Asterisk run on the same machine
```

We can expand the capability of asterisk to recognize numbers using country code like +62XXX using ENUM.

OpenBTS Operation

To operate OpenBTS we can follow the following steps.

Chek the connection between OpenBTS and USRP. This can be done using USRPping as follows

```
cd /usr/local/src/openbts-2.6.0Mamou/Transceiver
./USRPping
```

Assuming Asterisk is correctly configure, we can run it via

```
asterisk
```

or

```
/etc/init.d/asterisk restart
```

Run smqueue

```
sudo su
cd /usr/local/src/openbts-2.6.0Mamou/smqueue/
./smqueue &
```

Run OpenBTS

```
cd /usr/local/src/openbts-2.6.0Mamou/apps
./OpenBTS
```

We need to copy OpenBTS.config.example to OpenBTS.config if we run it for the first time before run OpenBTS.

```
cd /usr/local/src/openbts-2.6.0Mamou/apps
cp OpenBTS.config.example OpenBTS.config
./OpenBTS
```

Using all default values, with no modification, we can operate OpenBTS.

CHAPTER 18: Peering Among Providers

There are actually plenty of free services like VoIPRakyat across the world. Most of these services are interconnected through a broker facility run by a SIP Broker http://www.sipbroker.com. SIP Broker functions as an SIP internet exchange where all SIP account providers share their traffic. SIP Broker provides area code for each of them, allowing them to be connected to each other. In short, those who have account in VoIP Rakyat can also communicate with others who own account in different servers.

A complete list of more than a thousand of a variety of SIP account providers worldwide can be found in the SIP Broker's white pages from the following URL:

http://sipbroker.com/sipbroker/action/providerWhitePages.

Notice that there are some important information on the providers, such as, Provider Name, some with their URL; SIP Proxy, the SIP server in use; Area code, or better known in SIP Broker terminology as SIP-Code. Also listed in such a list is VoIP Rakyat's area code, which is *536. This means that other SIP providers willing to connect to VoIP Rakyat have to add the *536 code as a prefix preceeding VoIP Rakyat's SIP account number. For example, if a SIP provider wishes to call someone in VoIP Rakyat whose SIP account number is 20123, then user of such provider should dial *53620123 instead of 20123.

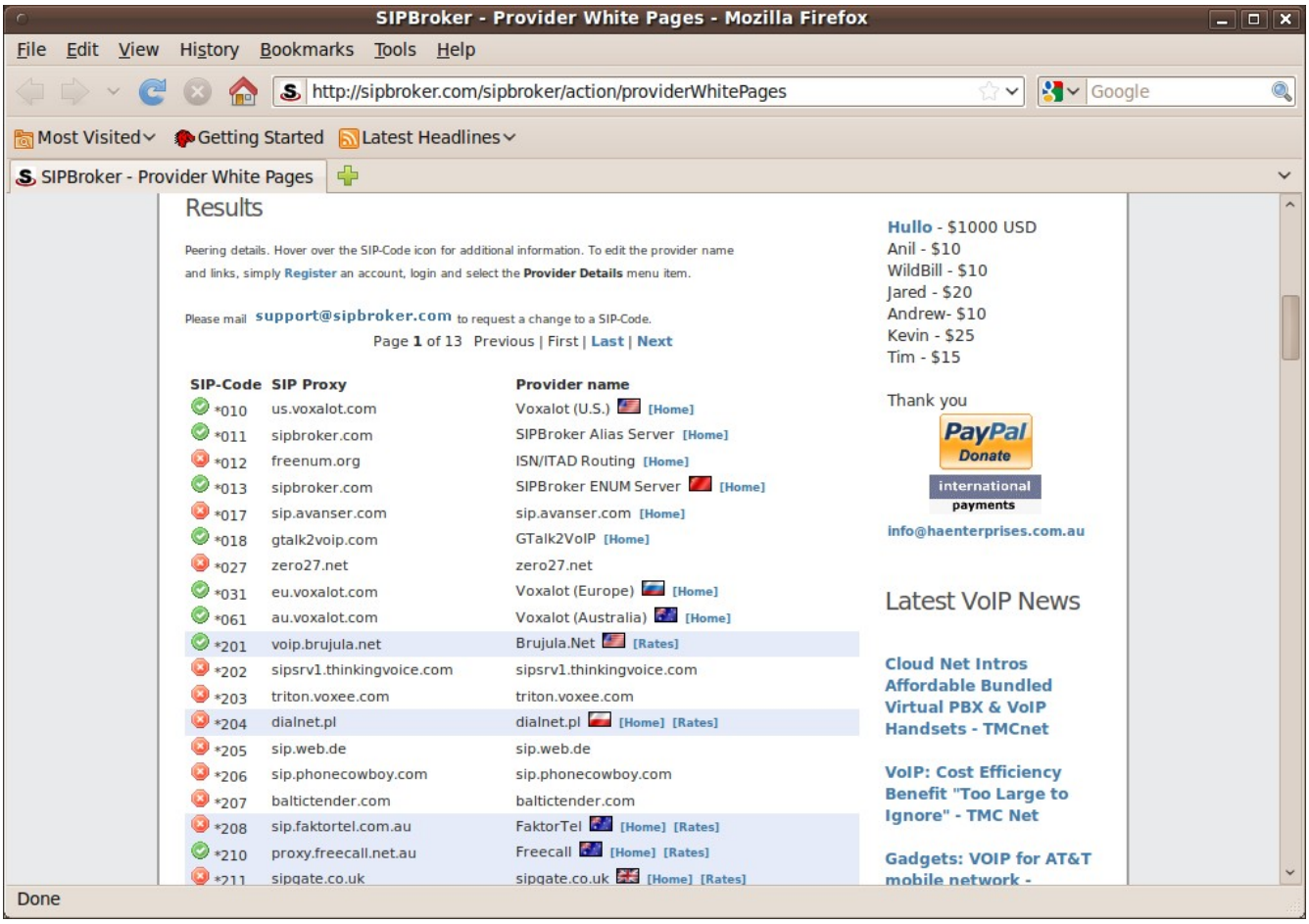


Figure 18.1: Through SIPbroker.com you can find a number of SIP providers with their respective proxy. The site also indicates which provider is active and is not active

Free SIP Proxy Servers

There are also other free SIP Proxy providers we can connect to through SIP Broker. Some of which are, such as:

Area Code	SIP Proxy	URL for registration
*201	voip.brujula.net	http://voip.brujula.net/english/
*208	sip.faktortel.com.au	http://www.faktortel.com.au/
*211	sipgate.co.uk	http://www.sipgate.co.uk/user/index.php
*234	sip.pennytel.com	http://www.pennytel.com/
*247	sip.freshtel.net	http://www.freshtel.net/
*258	voiptalk.org	http://www.voiptalk.org/products/index.php
*269	sip03.astrasip.com.au	http://www.astratel.com.au/
*272	sip2.bbpglobal.com	http://www.bbpglobal.com/global/
*320	sip.sipme.com.au	http://www.sipme.com.au/

Becoming a Peer in SIP Network

If you're operating your own softswitch using the public IP address, you can register with SIP Broker for free in order to obtain the area code of the SIP Broker so that you can be called by other VoIP network. The procedure you have to do is as the following: create an account in SIP Broker at http://www.sipbroker.com/sipbroker/action/memberRegister, and register your SIP Proxy through SIP Broker at http://www.sipbroker.com/sipbroker/action/providerWhitePages. Enter your SIP Proxy in the provided blanks and click the Save button.

The process of creating an account at SIP Broker is not difficult. All you have to do is to select the menu for member registration.

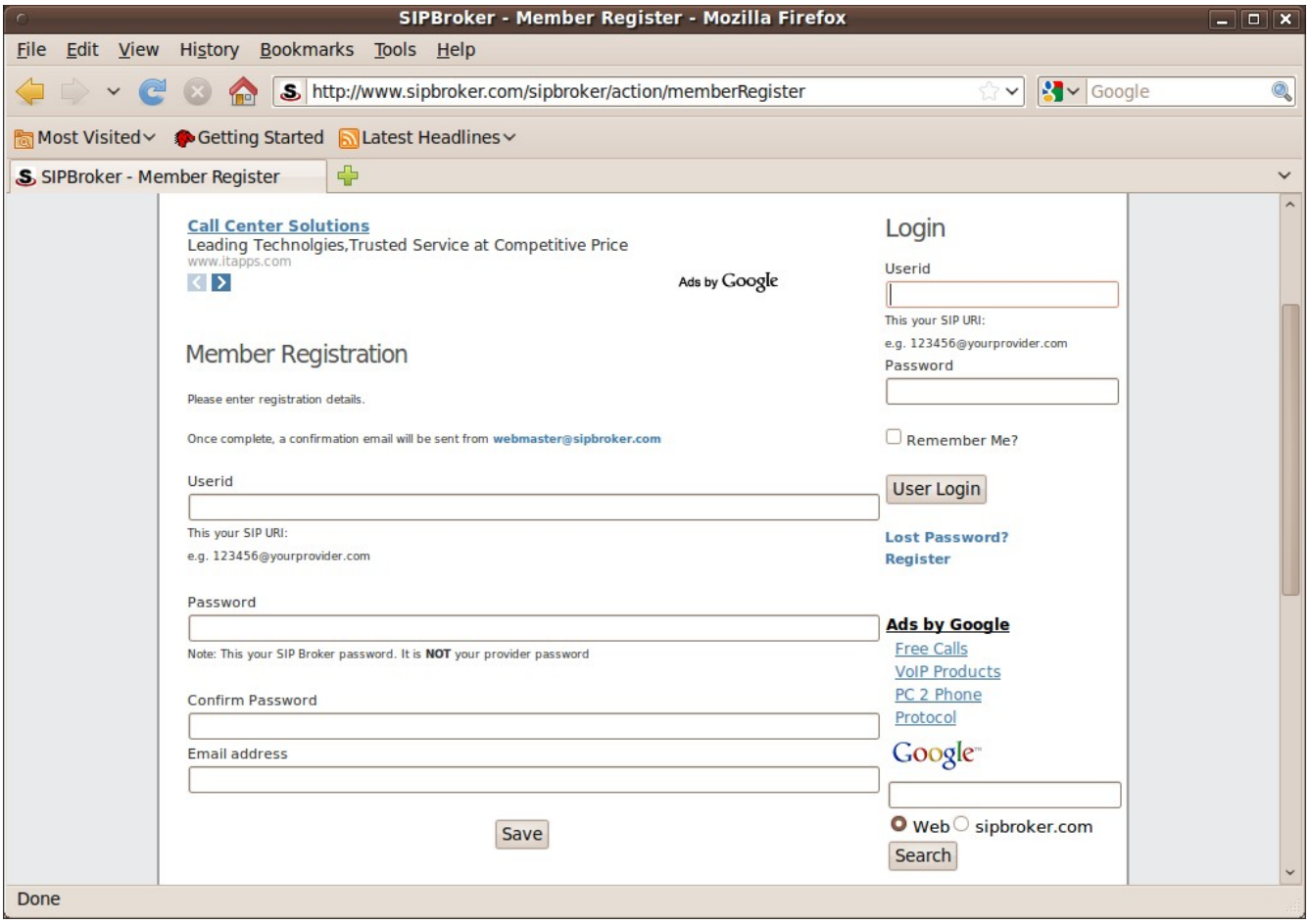


Figure 18.2: Through SIPbroker.com you can find a number of SIP providers with their respective proxy. The site also indicates which provider is active and is not

In the member registration menu, you only have to enter the information pertaining to the SIP URL (such as 20123@voiprakyat.or.id), the password, e-mail address to register yourself to the SIP Broker through http://www.sipbroker.com/sipbroker/action/memberRegister.

CHAPTER 19: Internet Telephony Bandwidth

In VoIP, Voice quality is important, as it will determine how clear the voice in a given phone conversation. Such quality is dependent on a number of things: the compression technique being used (Codec), the amount of packet loss occurring in a given network, and bandwidth availability. So there is a number of things you need to take into account when planning your VoIP network. With these information, you will know whether the bandwidth you have is sufficient for your communication.

Coding Decoding (CODEC)

If you want to know how a voice produced by a codec sounds like, you can listen to a 8 Kbps mono WAV, which can be downloaded from http://www.signalogic.com/index.pl?page=codec_samples. Question often asked when using this service is how much bandwidth is cunsumed. When idle, the system does not need large amount of bandwidth, but when you communicate, the bandwidth needed could be as big as 32 kbps up and 32 kbps down per channel (1 communication session), depending which codec is being used.

There is a number of codecs used:

Codec	**Bandwidth consumption**
GIPS	13.3 Kbps or higher
GSM	13 Kbps (full rate), 20 ms frame size
iLBC	15 Kbps, 20 ms frame size, 13.3 Kbps, 30 ms frame size
ITU G.711	64 Kbps, sample-based is also known as alaw/ulaw
ITU G.722	48/56/64 Kbps
ITU G.723.1	5.3/6.3 Kbps, 30ms frame size
ITU G.726	16/24/32/40 Kbps
ITU G.728	16 Kbps
ITU G.729	8 Kbps, 10 ms frame size
Speex	2.15 to 44.2 Kbps

LPC10	2.5 Kbps
DoD CELF	4.8 Kbps

The next question would be which codec is the most suitable for a provider? The answer depends on the amount of bandwidth you have. If you have a maximum bandwidth of 32 Kbps both up and down for a VoIP traffic, it is recommended that you use GSM or iLBC as your codec. On the other hand, if the amount of bandwidth is higher, say, higher than 128 Kbps, you can use G711u (PCMU), which will increase the voice quality in a communication session, with clearer voice and lower delay. Other codec that could produce optimal result is the G.729 Codec. Unfortunately, it is a proprietary codec which is not favourable for those who use open source platform.

The most commonly used codecs are the G.729, GSM, and G.711. Of these, the G.711 is favorable as it delivers good quality in LAN network, GSM is preferred by open source users as GSM is not copyrighted, while many VoIP devices use G.729 for their codec, the one which is copyrighted.

Mean Opinion Score (MOS)

To find out how good your voice quality is, you can measure it in terms of Mean Opinion Score (MOS) and R Factor, both of which are the unit measurement derived from the users perception on the voice they hear. MOS can be found at http://www.voiptroubleshooter.com/diagnosis/emodel.html. Below we have provided you with examples of MOS measurements based on a number of Codecs.

User's opinion	**R Factor**	**MOS Score**
Maximum values obtained by G.711	93	4.4
Highly satisfactory	90-100	4.3-5.0
Satisfactory	80-90	4.0-4.3
Good	70-80	3.6-4.0
Unasatisfactory	60-70	3.1-3.6
Poor	50-60	2.6-3.1
Not recommended	0 – 50	1.0-2.6

MOS Score and R Factor are measured
based on users' experience on a communication session

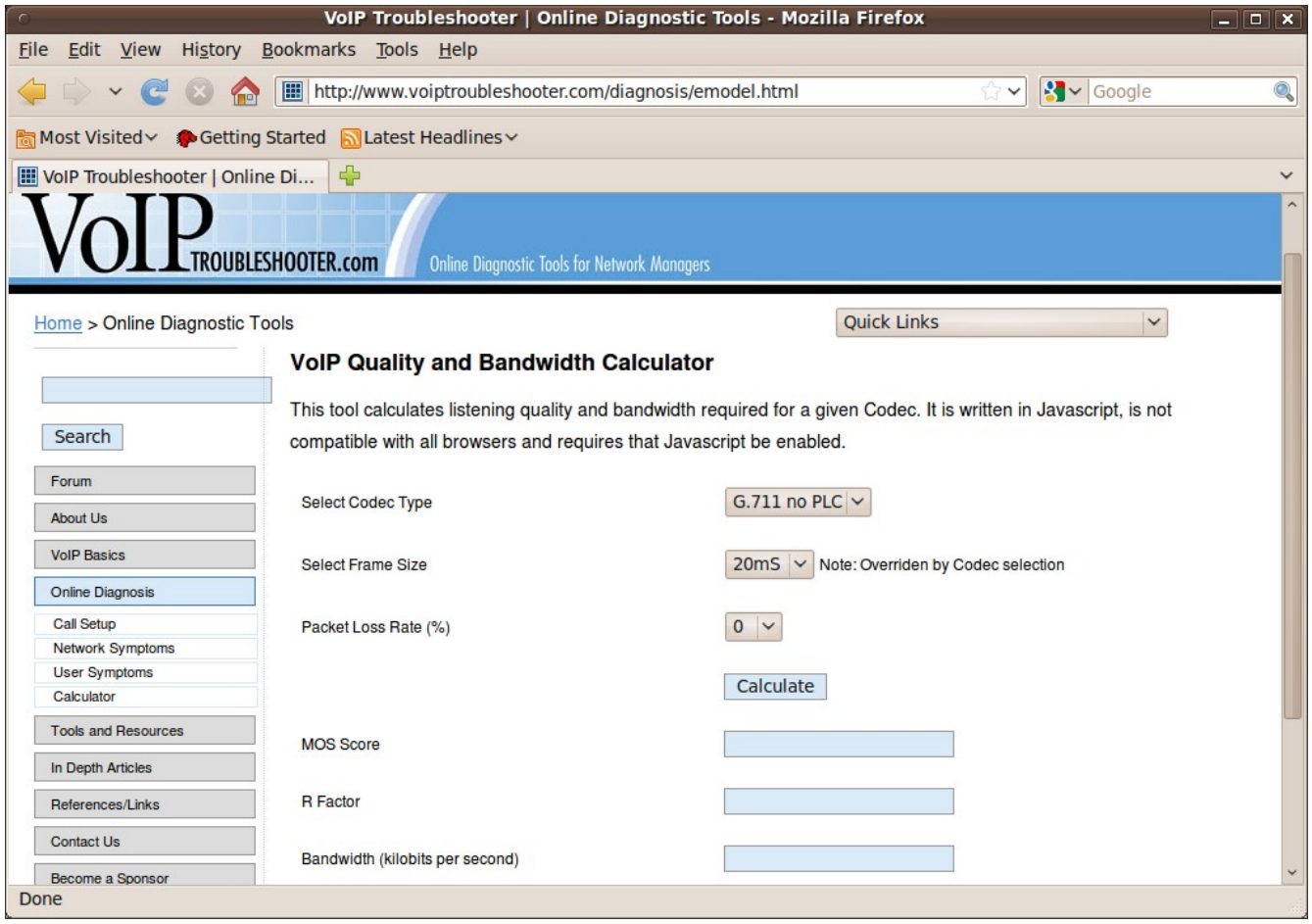


Figure 19.1: VoIP Quality and Bandwidth Calculator.

Entering other types of codec and values pertaining to our bandwidth for every codec, we obtained the following MOS and R Factor calculation:

Codec	Frame	Packet Loss	MOS	R Factor	Kbps
G.711	20ms	0%	4.4	93	80.8
G.723 5kbps	20ms	0%	3.8	74	16.5
G.723 6kbps	20ms	0%	4.0	78	17.5
G.729	20ms	0%	4.1	83	24.8

MOS and R Factor values for G.711, G.723, and G.729

The effect caused by packet loss occurring in a given network will make the MOS and R Factor value decreasing, as shown on table above. The higher the packet loss, the lower the value of MOS and R Factor.

Codec	Frame	Packet Loss	MOS	R Factor	Kbps
G.729	20ms	0%	4.1	83	24.8
G.729	20ms	5%	3.3	64	24.8
G.729	20ms	10%	2.7	52	24.8
G.729	20ms	20%	1.9	37	24.8

Typically, the frame size used for measurement is 20 ms, with the bandwidth around 25 Kbps. The longer the length of payload or voice frame size, the less bandwidth is needed because the overhead protocol is smaller.

Codec	Frame	Packet Loss	MOS	R Factor	Kbps
G.729	2.5ms	0%	4.1	83	41.6
G.729	5ms	0%	4.1	83	41.6
G.729	10ms	0%	4.1	83	41.6
G.729	20ms	0%	4.1	83	24.8
G.729	30ms	0%	4.1	83	19.2

Calculating The Required Bandwidth

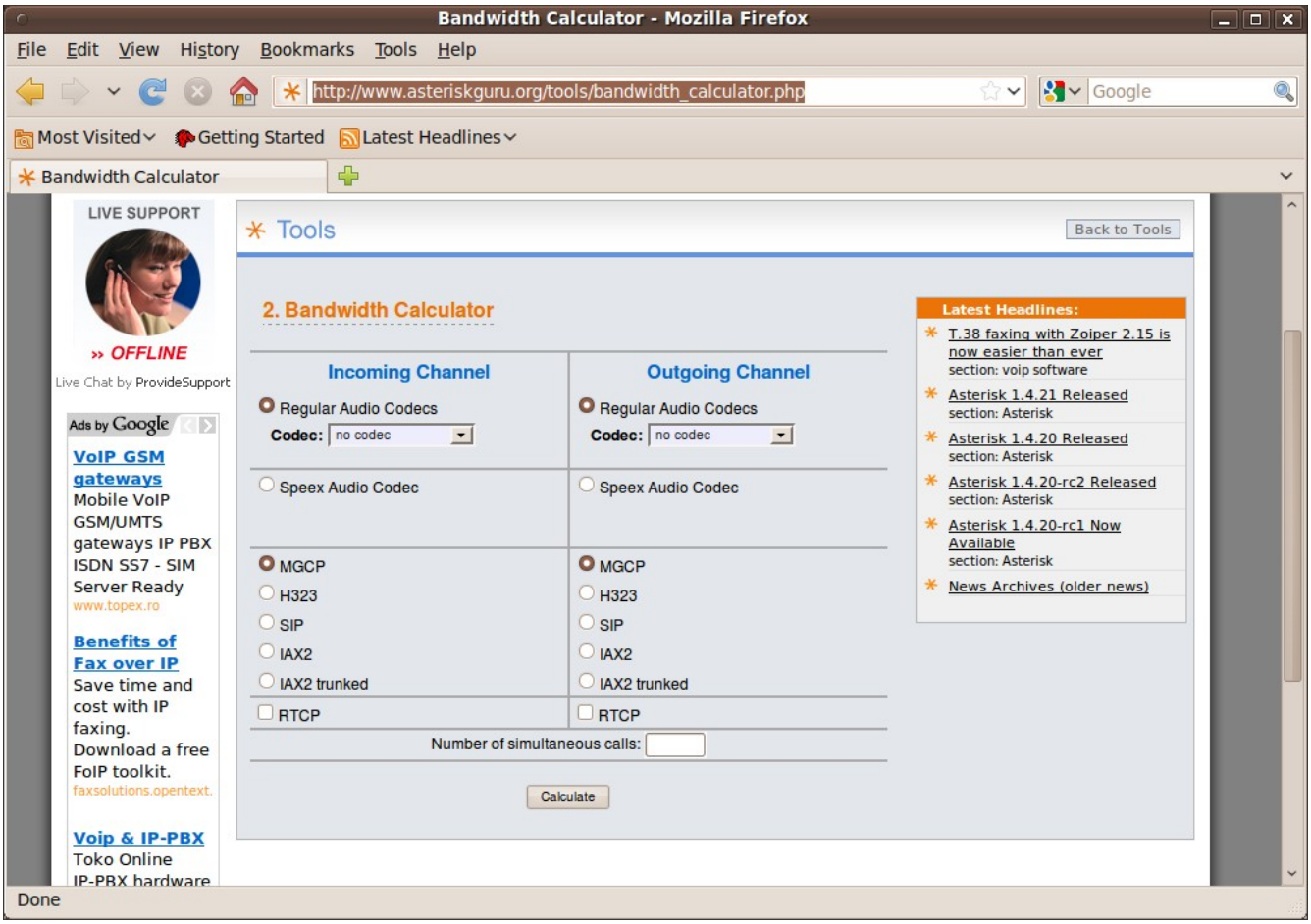


Figure 19.2: Bandwidth Calculator at Asterisk Guru.

To estimate the amount of bandwidth consumed by a Codec, use Asterisk Guru's tools, which is available at http://www.asteriskguru.org/tools/bandwidth_calculator.php.This tool enables us to calculate the amount needed by a variety of Codecs in respect to a certain number of calls taking place simultaneously. The resulting calculated values will be shown as Incoming and Outgoing bandwidth.

An example of calculated required bandwidth for GSM and G.729 Codec is shown in the following table

Number of calls	GSM		G.729	
	Incoming (Kbps)	Outgoing (Kbps)	Incoming (Kbps)	Outgoing (Kbps)
1	28.63	28.63	23.63	23.63
2	57.25	57.25	47.25	47.25
3	85.88	85.88	70.88	70.88
4	114.50	114.50	94.50	94.50
5	143.13	143.13	118.13	118.13
6	171.75	171.75	141.75	141.75

So suppose your bandwidth capacity is 64 Kbps and the Codec you use is G.729. Then the maximum number of VoIP calls you can have for optimal voice quality is two. This is of course assuming that the Internet is not being used for other traffic, such as, e-mail, browsing, chatting or downloading.

A much more detailed calculation of a VoIP packet can be looked at http://www.packetizer.com/voip/diagnostics/bandcalc.html. By using this tool, we can see the bandwidth, packet rate, delay and even performance. The parameters used to derive the calculation are Payload (Codec), type of protocol and whether we want to use Silence Suppression. If you tick Silence Suppression, the average of bandwidth or packet delivered will become smaller.

From all of these calculations, we can deduce at least two things:

- The smaller the bandwidth, the larger the Mega Instruction Per Second (MIPS) required to operate in a processor. Today this might not be a problem, given that high-speed processors are now largely available at affordable price.
- The smaller the bandwidth, the higher the delay process. This is related to the needs to process high voice compression.

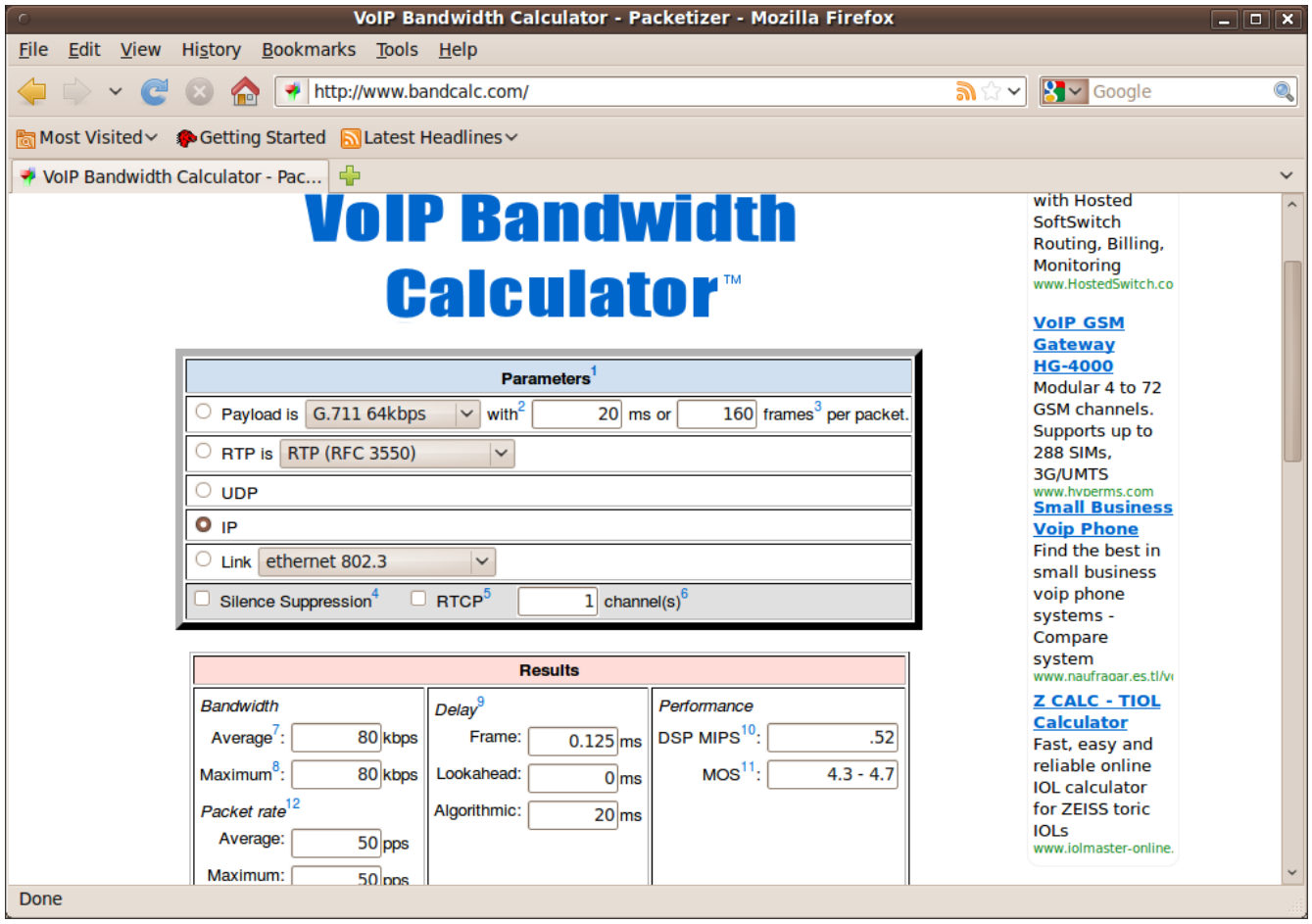


Figure 19.3: More Detailed Bandwidth Calculator at Packetizer.

There are more interesting things we can conclude from these calculation. We recommend that you spend more time using the tool so that you will understand a variety of effects occurring in VoIP communication session.

Calculation for Call Center

Unlike a network serving small number of VoIP calls, a call center is often used to accommodate a high number of calls served by a given network, thus requiring a much more sophisticated calculation and planning tools. A Call Center can typically be a bank, travel agent, ticket reservation or even a mobile operator. So if you intend to establish a call center, please visit http://www.erlang.com/calculator/call/ to calculate how much bandwidth needed.

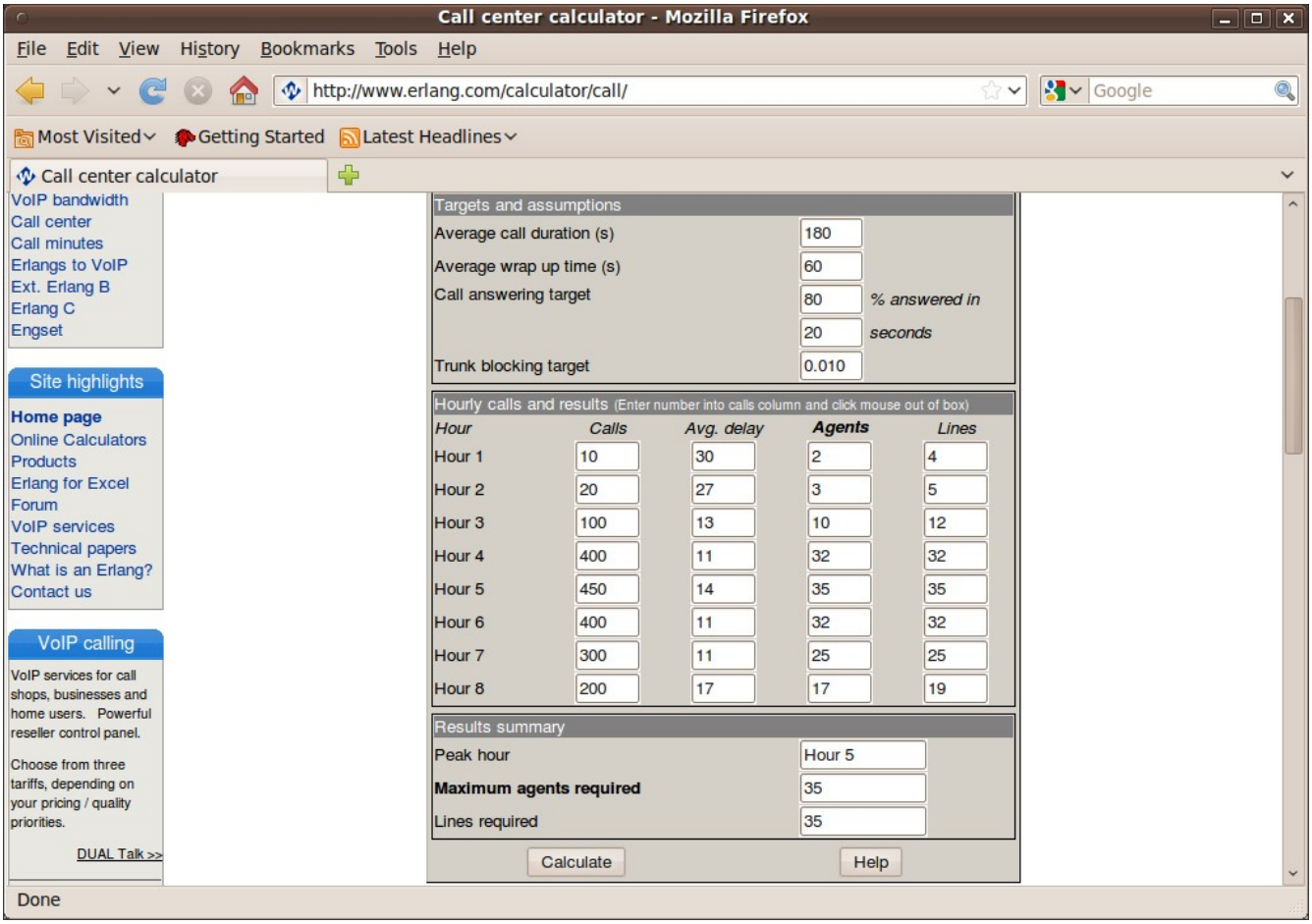


Figure 19.4: Call Center Calculator on Erlang.com

What we have to fill in as parameter are the length of calls (seconds), resolving time (seconds), percentage of calls answered within seconds, and percentage of calls that will be blocked. Blocked calls will receive busy tone, a sign indicating that all lines are busy. After all parameters are filled in, we

need to enter the number of calls coming in an hour. Based on the values we entered, we will obtain simulation result showing how many agents and lines are needed to respond to the calls.

The next step is to estimate how much bandwidth is needed for a specific number of lines or a number of minutes, in a given percentage of calls that will be blocked and receiving busy tone. This is often measured using Erlang Traffic Model in terms of Erlangs, the unit representing usage of a voice channel. Erlangs Traffic Model measurement is very important to help you being a telecommunication network engineer to understand the traffic pattern and network topology necessary to determine the size of trunk group. In addition, the measurement can also be used to determine the number of lines needed between a telephone network or system and telephone center, or between network locations.

In practice Erlangs can be used to give an overall picture of traffic volume in an hour. For example, a user group makes 30 calls in an hour, and every call has an average talking duration of 5 minutes, then the Erlangs resulting from the traffic that takes place is:

Traffic minutes in a given hour	=	Number of call x duration
Traffic minutes in a given hour	=	30 x 5
Traffic minutes in a given hour	=	150
Traffic hours in a given time	=	150 / 60
Traffic hours in a given time	=	2.5
Traffic size	=	2.5 Erlangs

The size of traffic is also called Busy Hour Traffic (BHT). The Busy Hour Factor parameter is a percentage of daily minutes of calls made during the most busy hour in a given day.

In addition to Erlangs, there is also blocking. Blocking represents unsuccessful calls because of insufficient number of lines. In other words, the caller will receive a busy tone from the center as all lines/trunks are being used. 0.01 blocking implies that 1% of calls made will be blocked. This decimal number is usually used in traffic telecommunication engineering. In a number of applications, blocking up to 0.03 (3%) is still tolerable. So this number should ideally be as small as possible.

It appears that the more we tolerate the number of blocked calls, the more minutes per day we will have. However, the more calls take place during peak hours or when busy hour factor increases, the less number of minutes per day we will have.

Now that you know what Erlang Traffic Model measurement is, you can plan your connectivity capacity for your call center, using a number of measurement tools available at

http://www.voipcalculator.com/calculator/.

VoIP Capacity Planning

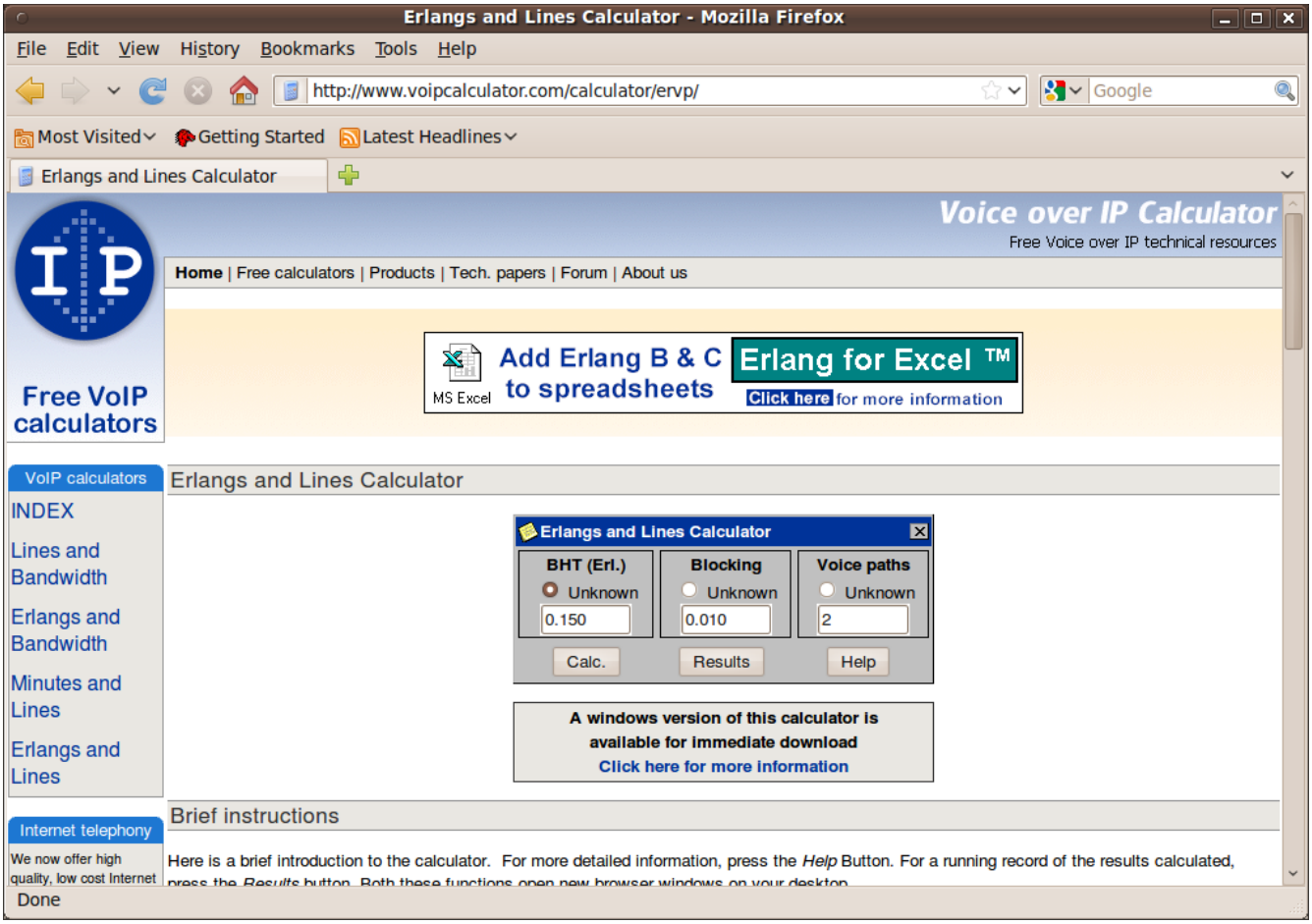


Figure 19.5: Erlangs and Lines Calculator.

By using Erlangs and Lines Calculator, you can obtain the number of Busy Hour Traffic (BHT) of a number of lines. In our example, we simulate 2, 4 and 8 voice paths facilitated by 64 Kbps connectivity with some Blocking values. The calculation result can be seen at the following table:

Voice Path	Blocking	BHT (Erlangs)
2	0.01	0.15

2	0.03	0.25
2	0.10	0.55
4	0.01	0.85
8	0.01	3.10

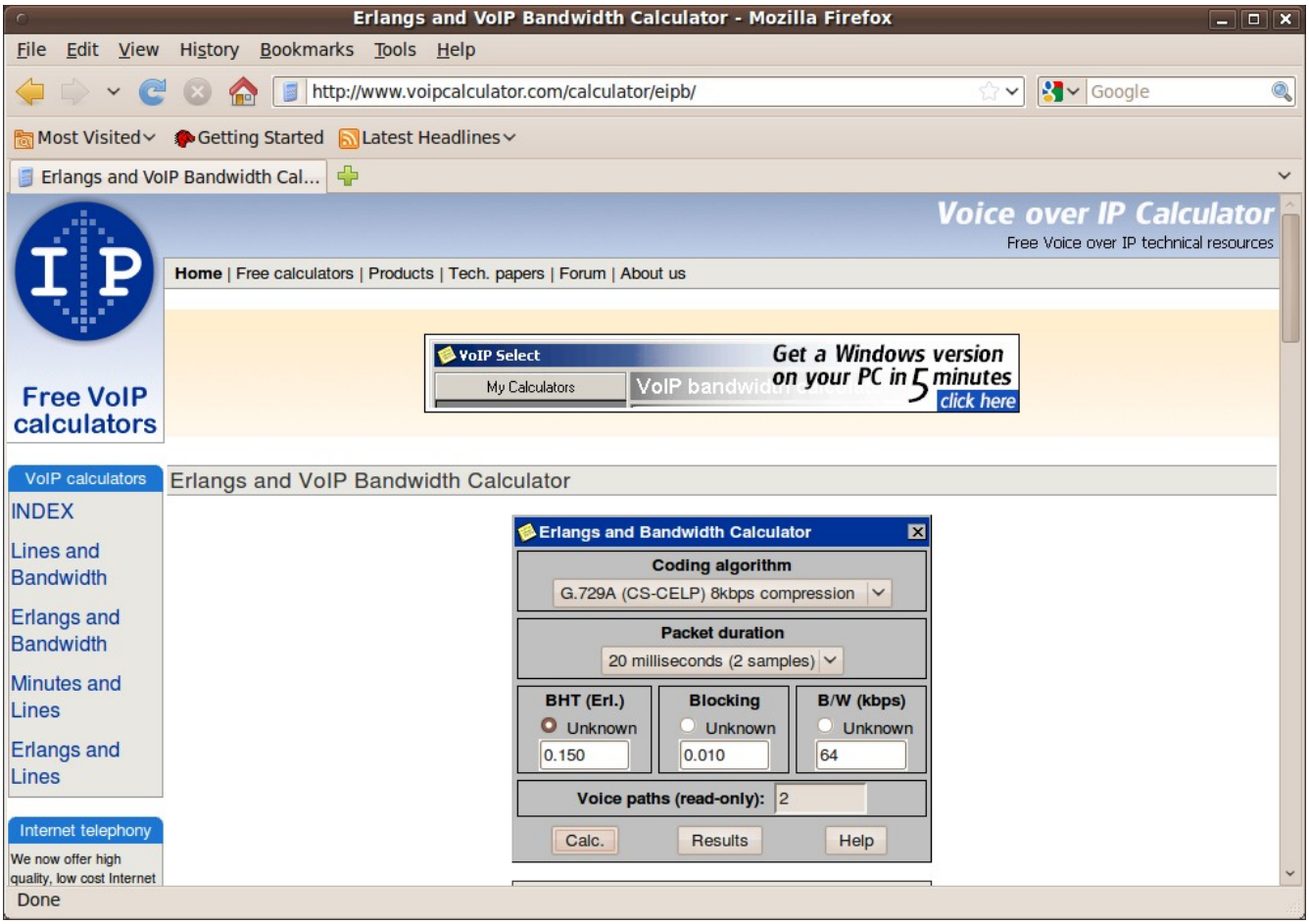


Figure 19.6: Erlangs and Bandwidth Calculator.

This calculator can be used to estimate the amount of bandwidth required to deliver the traffic, when the Busy Hour Traffic is known. By using Erlangs and bandwidth calculator, we can obtain the number of Busy Hour Traffic (BHT) of a given Codec. Suppose we run the simulation using a variety of bandwidth values and G.729 Codec; we will obtain the following result:

Bandwidth	Blocking	Voice Path	BHT

(kbps)			(Erlangs)
64	0.01	2	0.15
64	0.03	2	0.25
64	0.10	2	0.55
128	0.01	5	1.35
256	0.01	10	4.45
512	0.01	21	12.80

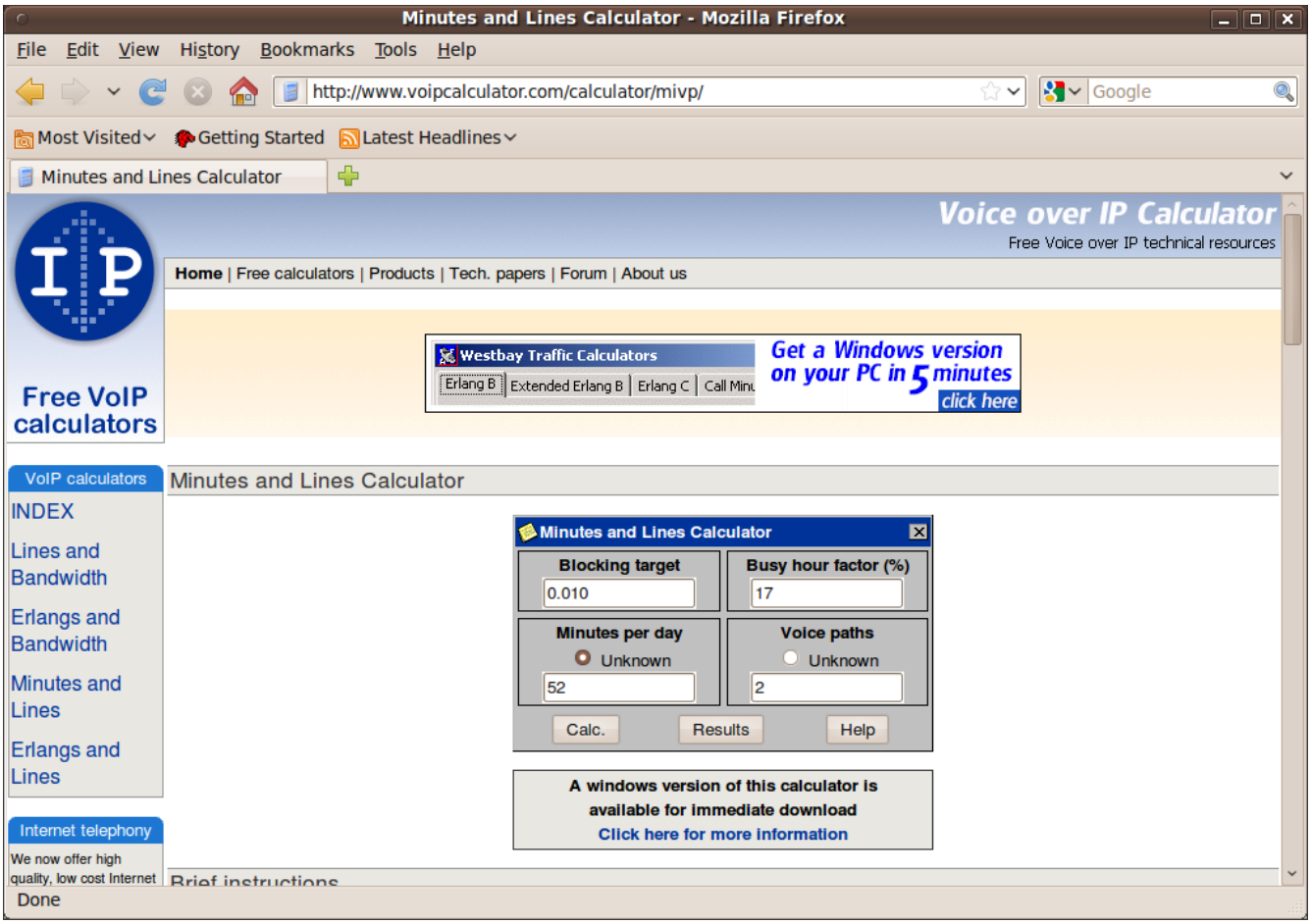


Figure 19.7: Minutes and Lines Calculator.

The Minutes and Lines Calculator allows us to estimate how many voice channels are needed for a given minutes of calls on the network. A network planner must make sure that the network has sufficient bandwidth to accommodate communication session at peak hours. As shown in the Figure, The Busy Hour Factor parameter is a percentage of daily minutes of calls made during the most busy

hour in a given day. The default value 17% is an acceptable percentage for an office operating 8 hours per day. The percentage is normally higher for an office operating in less number of hours, or an office that often places calls in a different time zone.
The following is the result of calculation for a ADSL channel capable of facilitating only two channels:

Voice Channel	**Blocking**	**Busy Hour Factor**	**Minutes / Day**
2	0.01	17%	52
2	0.03	17%	88
2	0.10	17%	194
2	0.03	20%	45
2	0.03	30%	30
2	0.03	40%	22

It appears that the more we tolerate the number of blocked calls, the more minutes per day we will have. However, the more calls take place during peak hours, or when busy hour factor increases, the less number of minutes per day we will have.

CHAPTER 20: VoIP Evaluation

In this chapter, we will discuss how to evaluate a VoIP system. Two (2) application software, i.e.,

- VQManager Software
- SIPp

Will be used.

Evaluate VoIP Performance using VQManager

VQManager Installation

Download VQManager from
http://www.manageengine.com/products/vqmanager/91408665/ManageEngine_VQManager.bin

Do the following on shell

```
sudo su -
cp ManageEngine_VQManager.bin /usr/local/src/
cd /usr/local/src
chmod a+x ManageEngine_VQManager.bin
./ManageEngine_VQManager.bin -console
```

We need to press <ENTER> several times. Normally, VQManager will be installed in

/root/ManageEngine/VQManager

All important script of VQManager is located at

/root/ManageEngine/VQManager/bin/

Some of the Important Scripts of VQManager

To start VQManager

```
sudo su -
cd /root/ManageEngine/VQManager/bin/
/root/ManageEngine/VQManager/bin/run.sh
```

To start as background process,

```
sudo su -
cd /root/ManageEngine/VQManager/bin/
/root/ManageEngine/VQManager/bin/run.sh &
```

To stop VQManager

```
sudo su -
cd /root/ManageEngine/VQManager/bin/
/root/ManageEngine/VQManager/bin/shutdown.sh
```

To reinitialized the Database in case we have a corrupt data,

```
sudo su -
cd /root/ManageEngine/VQManager/bin/
/root/ManageEngine/VQManager/bin/reinitializeDB.sh
```

Activate VQManager Web Service

When we run VQManager Web Service for the first time, we need to activate the Web Service. Firstly, we need to start VQManager as background process, such as,

```
sudo su -
cd /root/ManageEngine/VQManager/bin/
/root/ManageEngine/VQManager/bin/run.sh &
```

Access via Web to http://localhost:8647 with default username & password admin & admin.

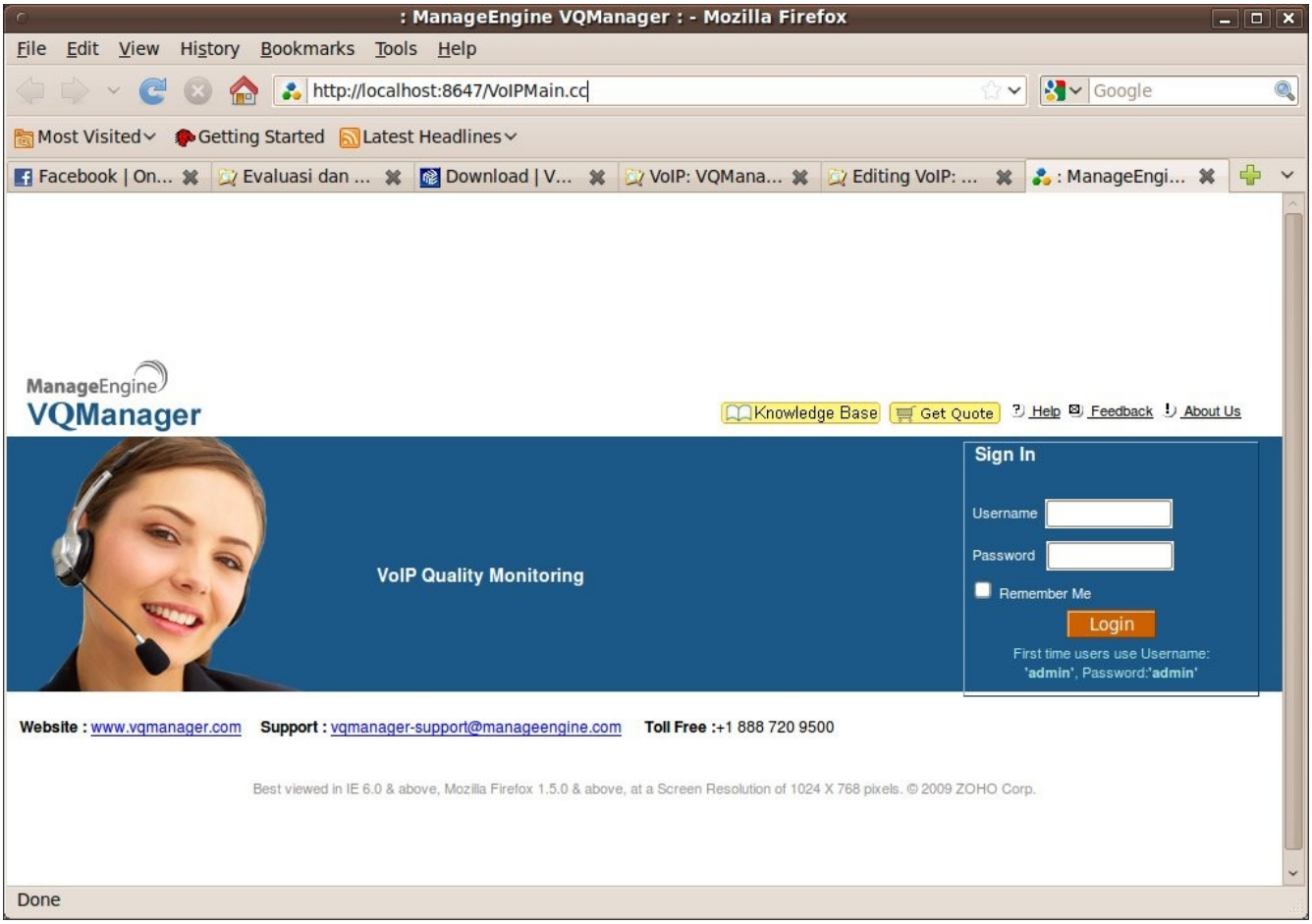


Figure 20.1 Login Menu in VQManager.

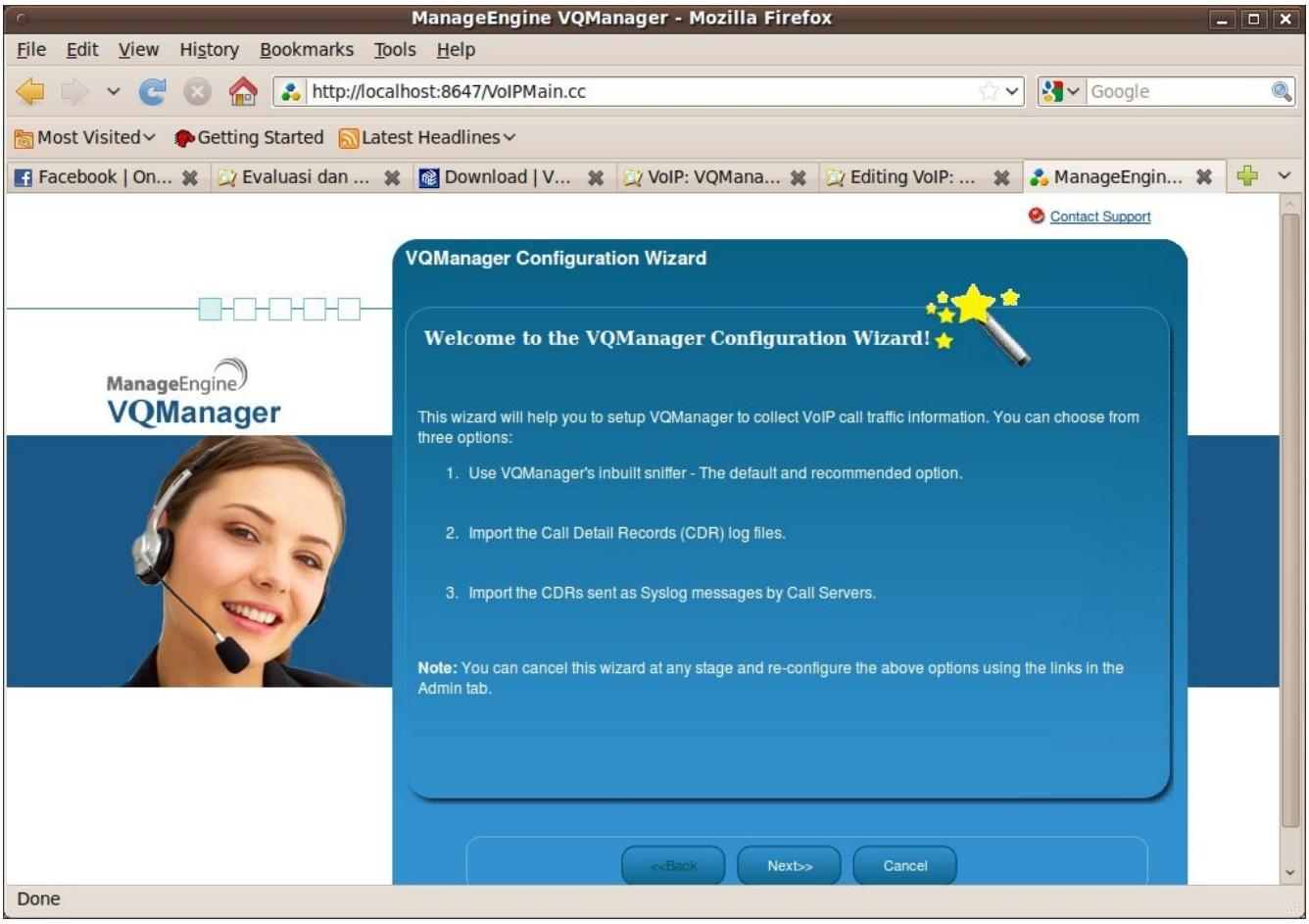


Figure 20.2. Welcome Message of VQManager.

As we access the VQManager Web for the first time, it will tell us that we can choose whether,

- Use VQManager builtin sniffer.
- Import the Call Detailed Records (CDR) log files.
- Import the CDR sent as Syslog messafes by the Call Servers.

Click next to continue.

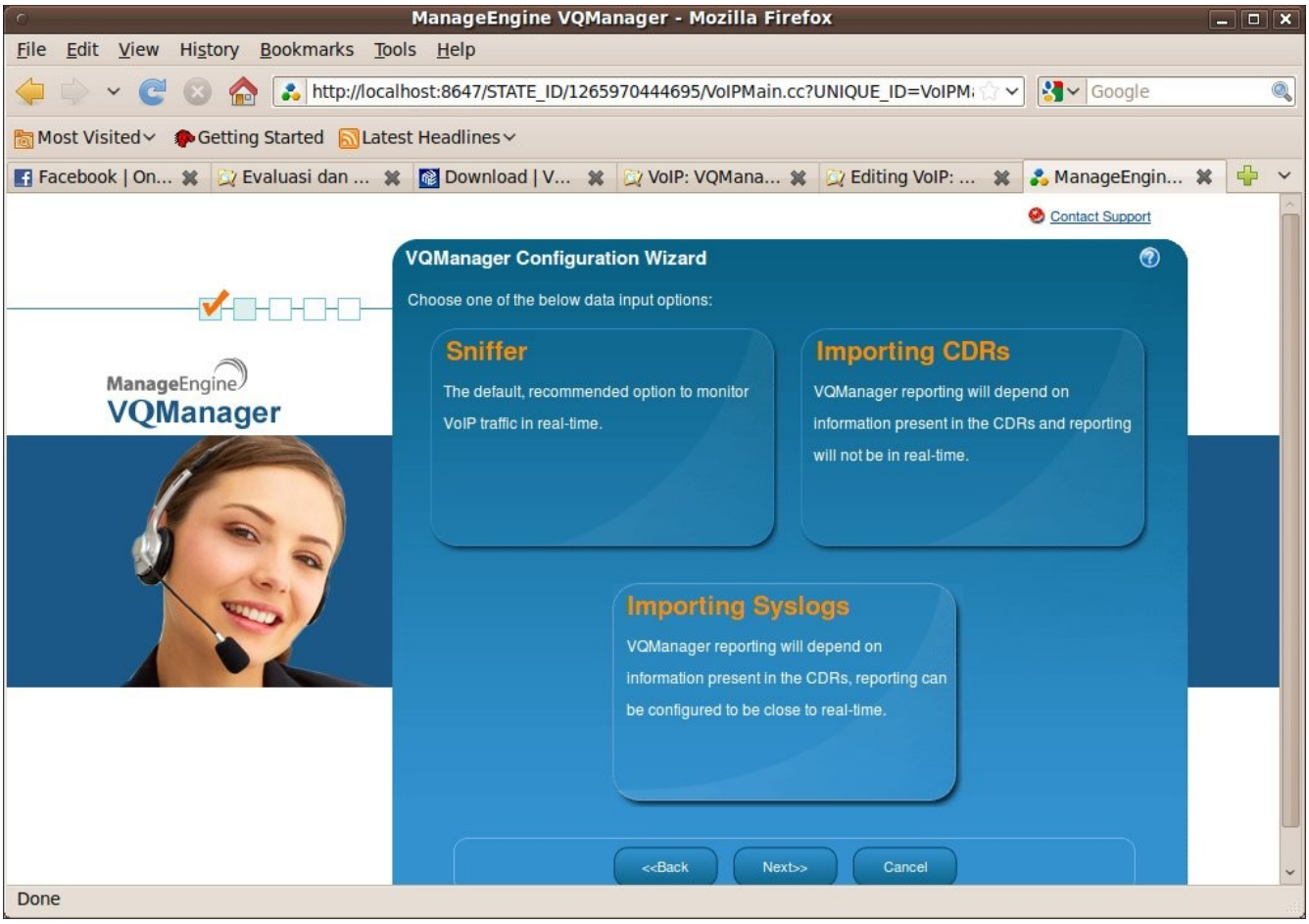


Figure 20.3 Option to monitor VoIP network.

Click on one of the option. The easiest may be “Sniffer”.
Click Next; after we choose the method.

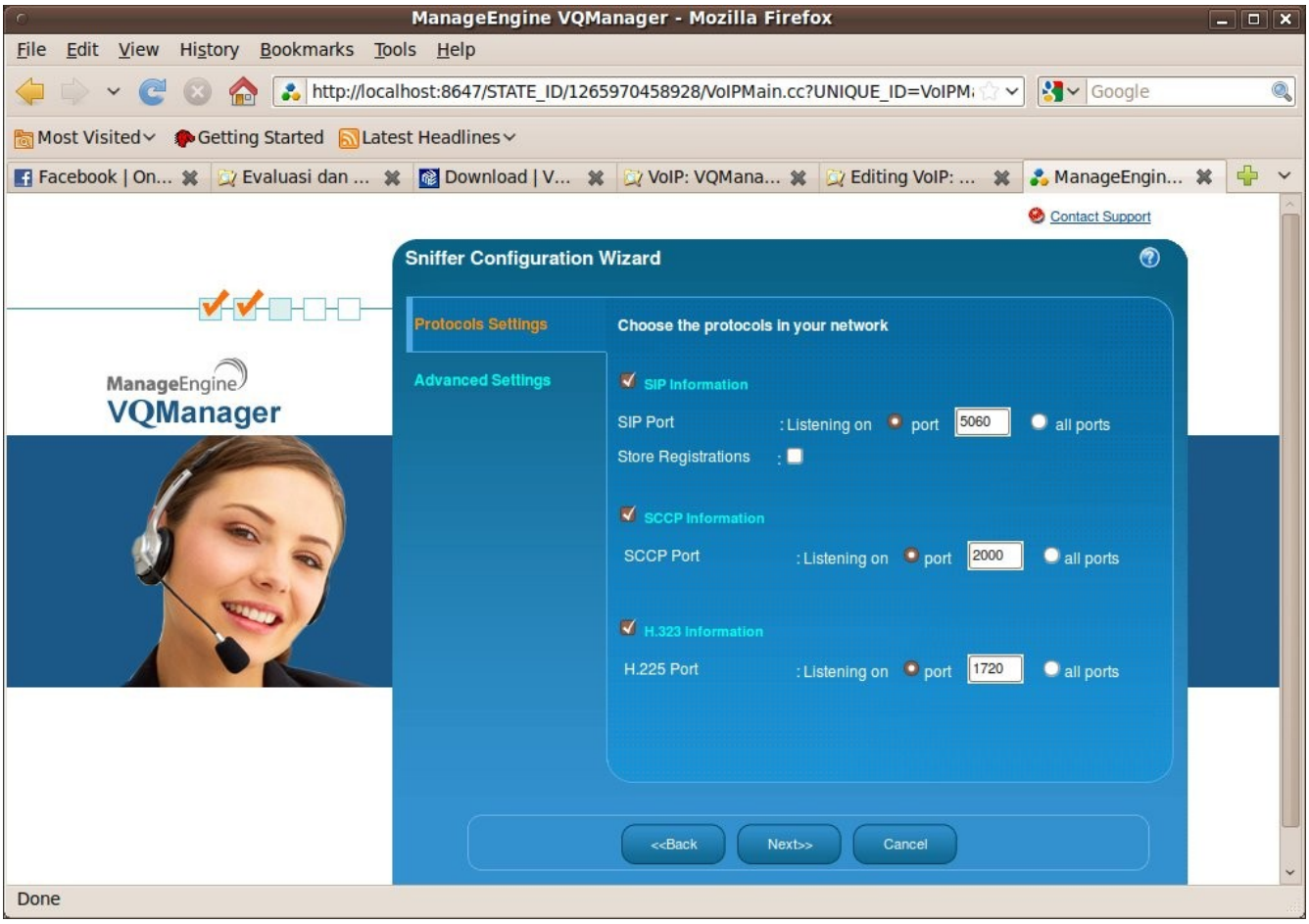


Figure 20.4 Protocol Settings in VQManager.

In the the following menu, we can choose the protocols to be monitored in the network. In the normal VoIP network, we don't have to change the values.

Click Next to continue.

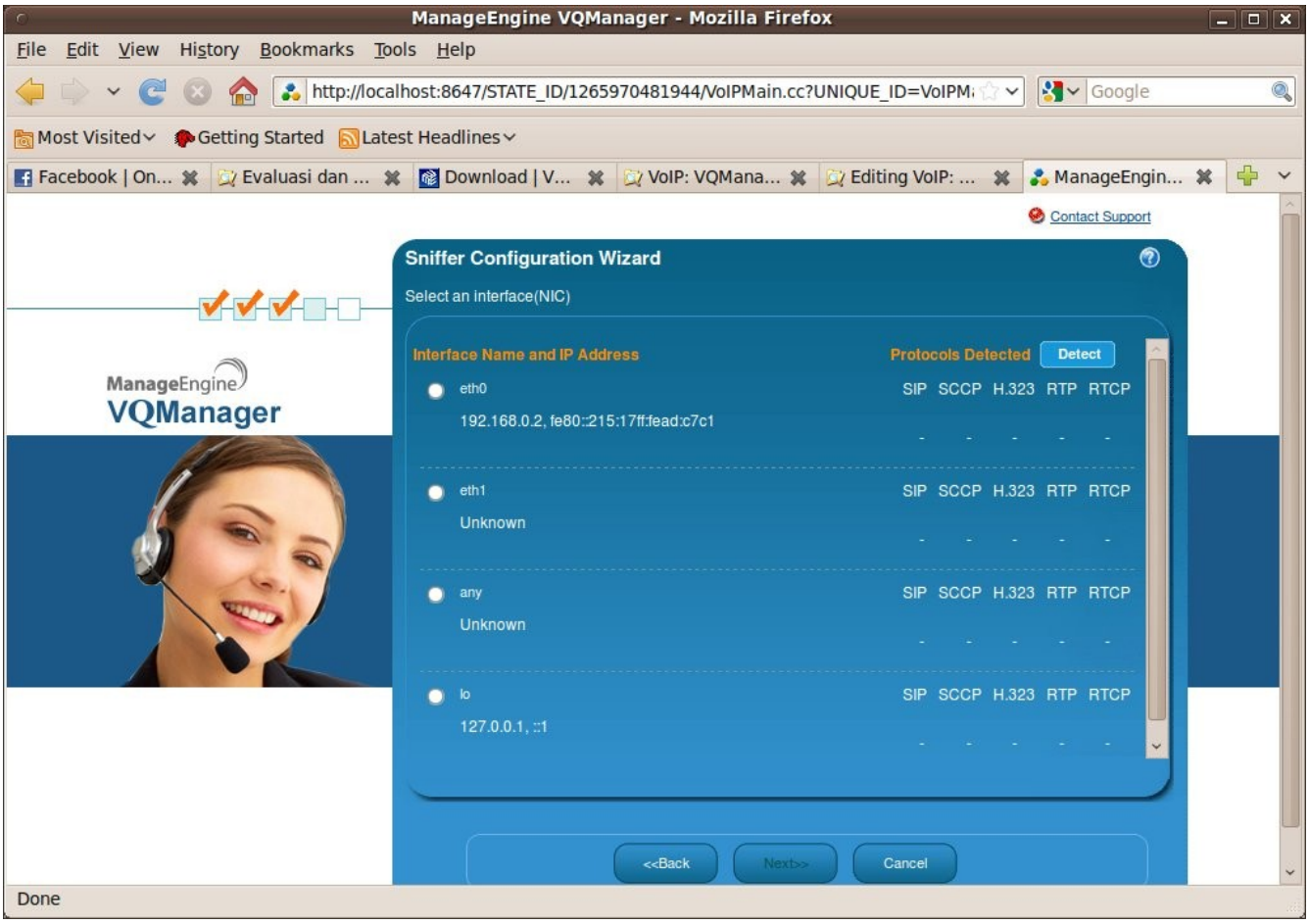


Figure 20.5 Select the Interface to be monitored.

This is the important part. We need to select, the interface to be sniffed.

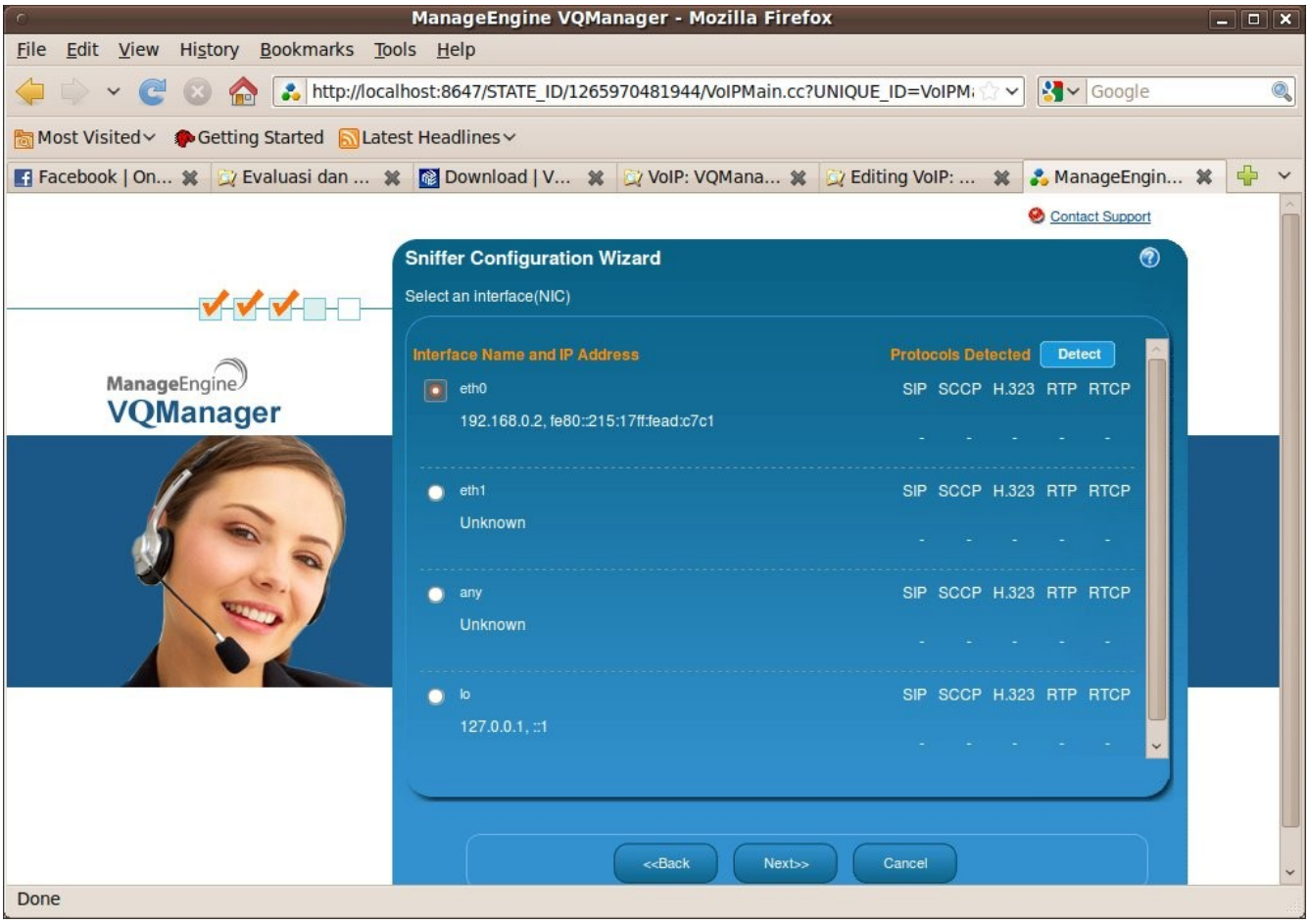


Figure 20.6 Interface Selected.

In the above figure, we choose interface eth0 to be sniffed.

Click Next to continue.

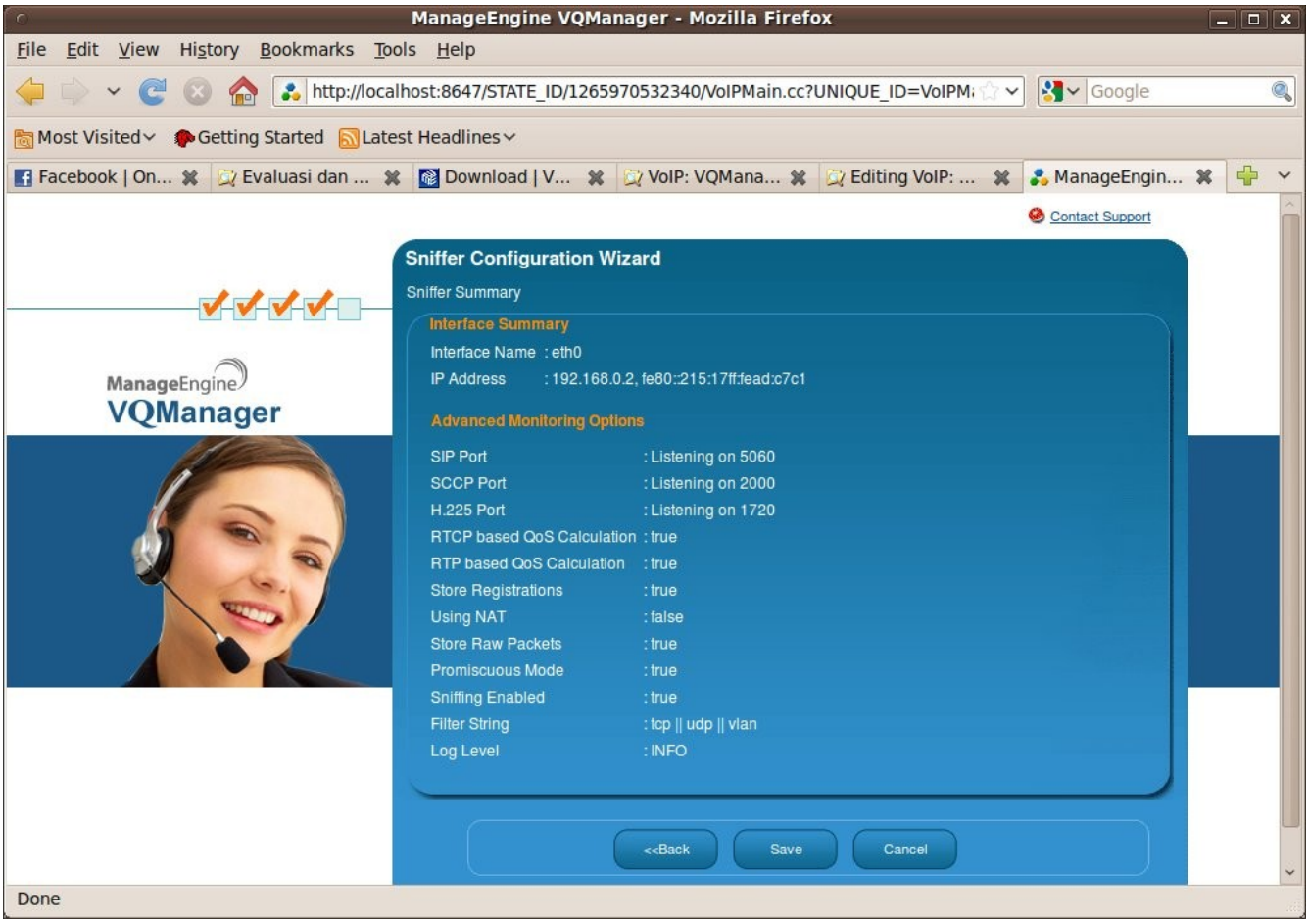


Figure 20.7 Configuration Summary.

Finally, VQManager will show the summary of our VoIP Monitoring platform.

Click Next to continue.

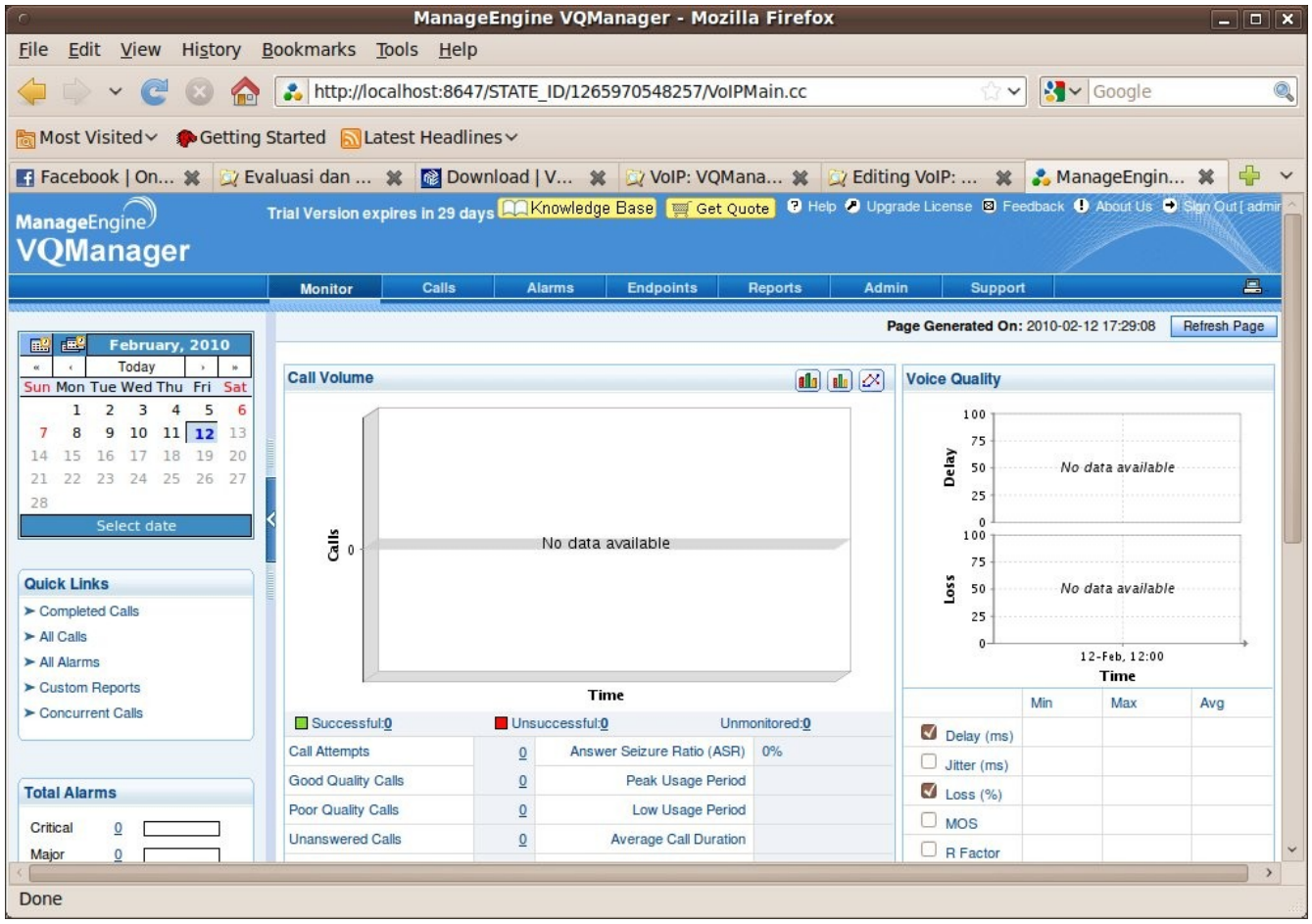


Figure 20.8 VQManager Web Console.

Finish configure VQManager.

The above figure shows the VQManager monitoring display. It shows a lot of important information regarding the monitored VoIP infrastructure.

Changing the Monitored Interface

In some cases, we need to monitor different interface. In this case, we need to login to the web at

http://localhost:8647
username admin
password admin

Click on the following sequence

Admin -> Sniffer -> Protocol Settings -> Next -> " select interface" -> Next -> Save

If we need to change the monitored interface, we need to click on the following sequence.

Admin -> Sniffer -> Reconfigure -> Protocol Settings -> Next -> "interface-nya" -> Next -> Save

Inserting new Interface

In some cases, we have inserted a new interface into the server and need to monitor this particular new interface. To do so, we need to reset the database,

sudo su -
/root/ManageEngine/VQManager/bin/reinitializeDB.sh

then activate the interface through VQManager Web again

Monitor VoIP Performance

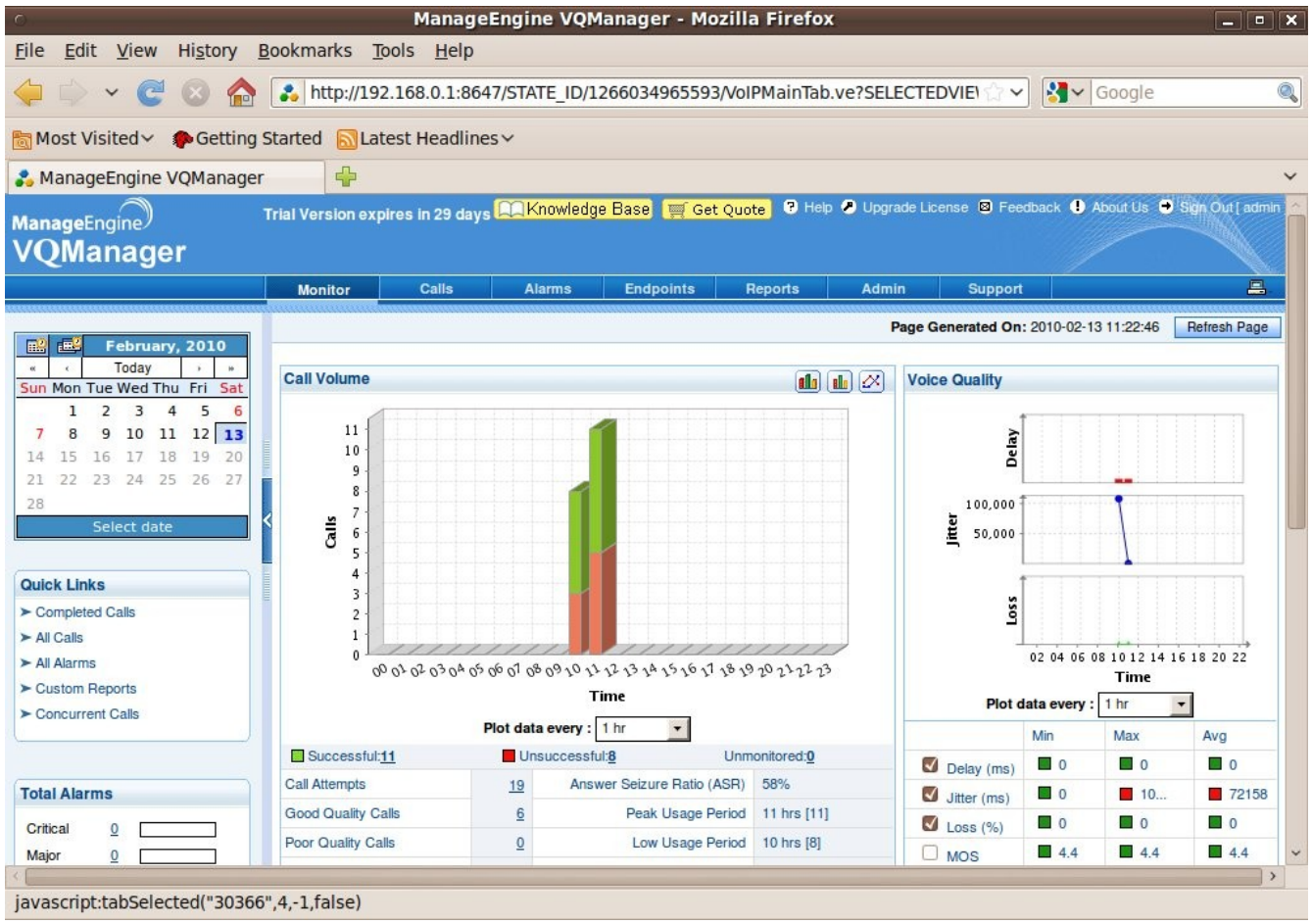


Figure 20.9 Frontend Web Console of VQManager.

The Web front end of VQManager. We can easily see the call volume, including the success call, the failed calls. On the right, we can see the quality of calls in general, including, its delay, jitter, packet loss, MOS, R-Factor.

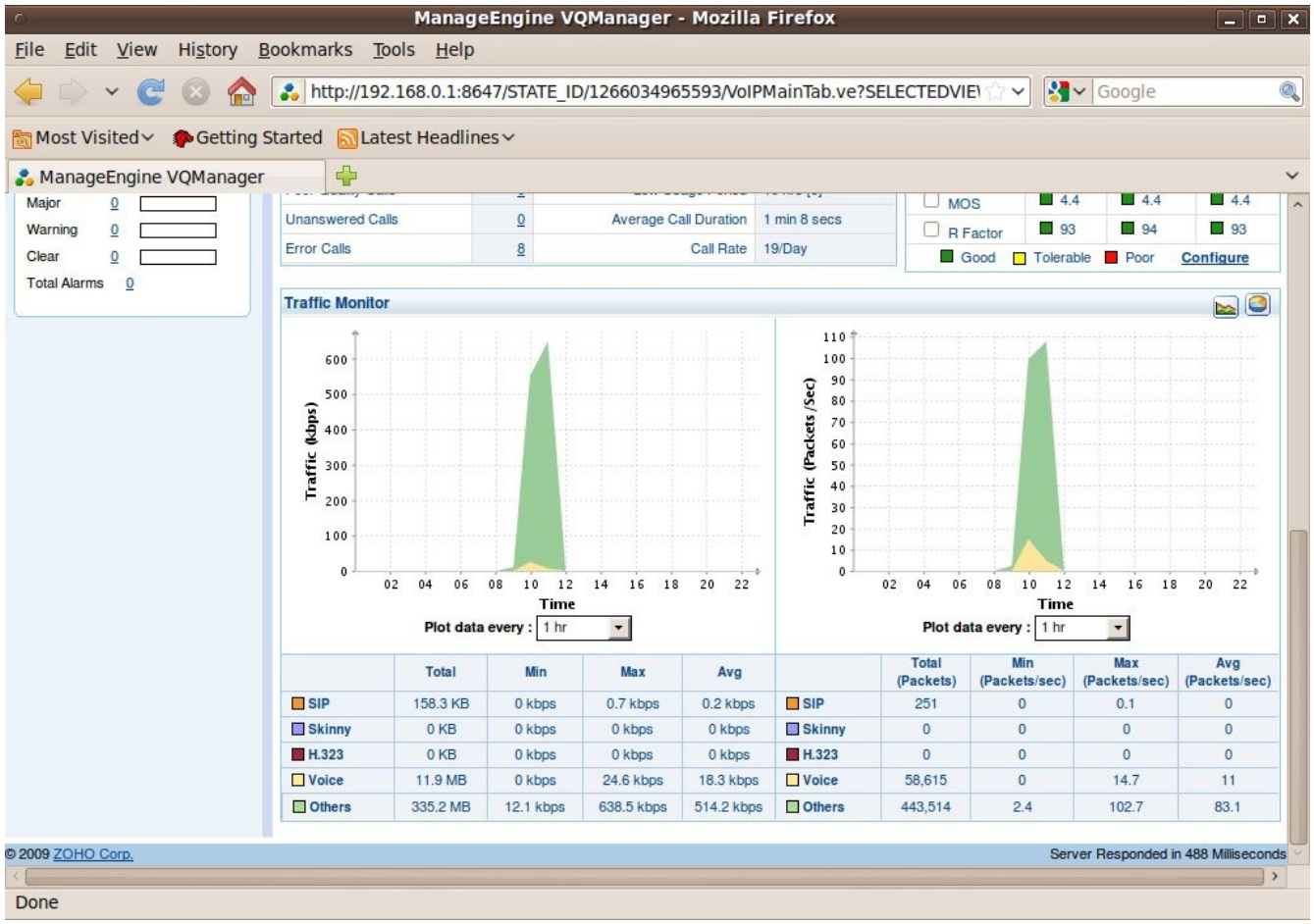


Figure 20.10 Web Front End

At the end, of the Web Front end. On the left, we can see traffic passing in Kbps. On the right, we can see traffic passing in packet per second. In addition, we can see the type of traffic passing through our system.

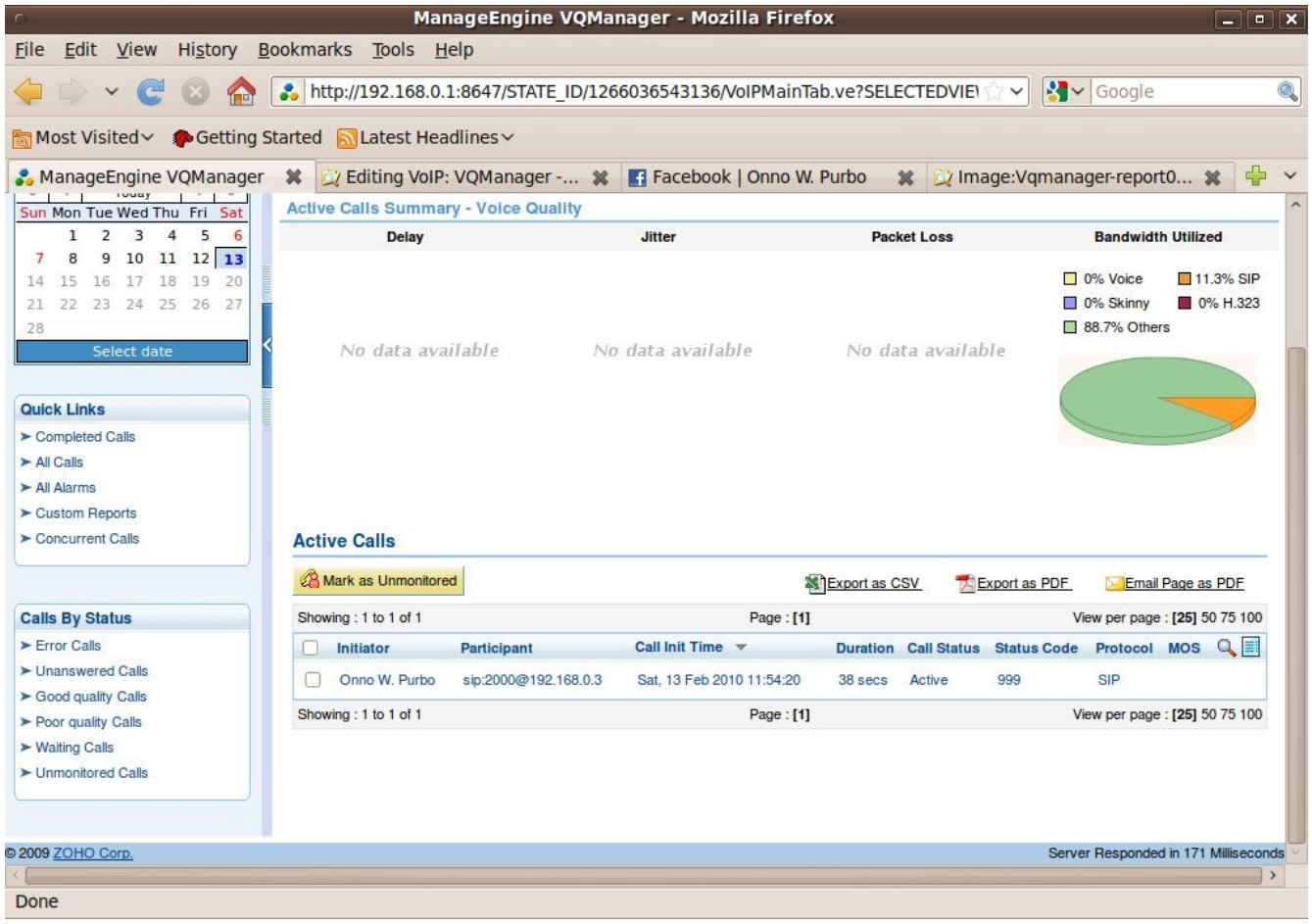


Figure 20.11 Call Menu VQManager.

In VQManager call menu, we can see in summery of calls, including, the usage profile. In addition, we can see the active call at the moment.

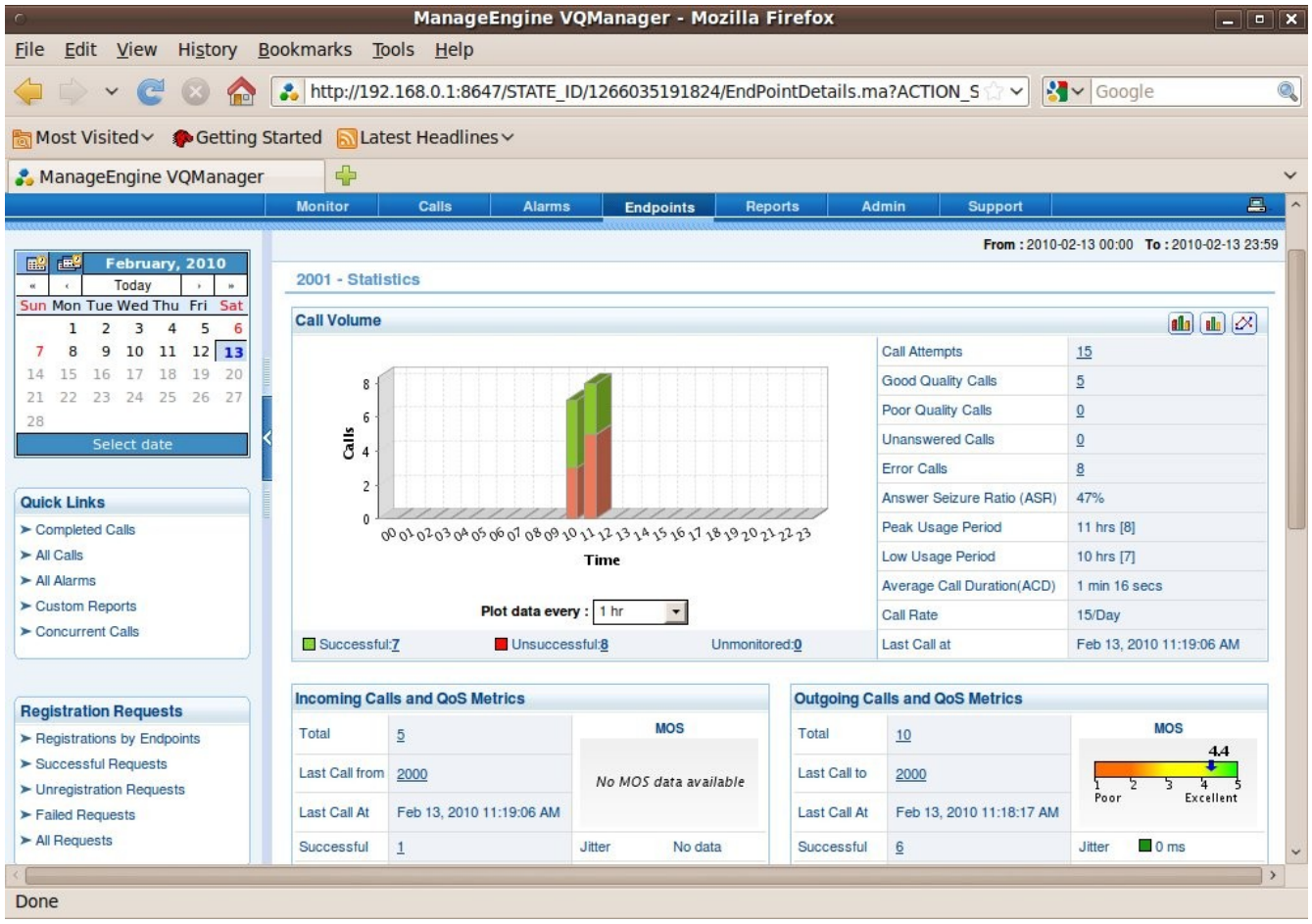


Figure 20.12 Endpoint Menu.

In the Endpoint menu, we can see a more detailed information of particular endpoint, such as, its activities, performance and usage. On the left, we can see the usage activities. On the right, we can see more detailed on the statistics and voice quality. Below it, we can see more detailed on the incoming and outgoing call QoS.

ManageEngine VQManager - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://192.168.0.1:8647/STATE_ID/1266035191824/EndPointDetails.ma?ACTION_S

Most Visited Getting Started Latest Headlines

ManageEngine VQManager

Traffic Metrics

In Octets	Out Octets	In Packets	Out Packets
3.8 MB	3.8 MB	24,728	24,748

Calls for 2001

Export as CSV Export as PDF Email Page as PDF

Showing : 1 to 10 of 15 Page : [1] 2 | View per page : [10] 20 30 40 50

Initiator	Participant	Call Init Time	Duration	Call Status	Status Code	Protocol	MOS
Onno W. Purbo	sip:2001@192.168.0.3	Sat, 13 Feb 2010 11:19:06	4 secs	Completed	1000	SIP	
Onno W. Purbo	sip:2000@192.168.0.3	Sat, 13 Feb 2010 11:18:17	10 secs	Completed	1000	SIP	
Onno W. Purbo	sip:2000@192.168.0.3	Sat, 13 Feb 2010 11:14:01	NA	Not Found	404	SIP	
Onno W. Purbo	sip:600@192.168.0.3	Sat, 13 Feb 2010 11:13:44	1 sec	Completed	1000	SIP	4.4
Onno W. Purbo	sip:2000@192.168.0.3	Sat, 13 Feb 2010 11:12:18	NA	Not Found	404	SIP	
Onno W. Purbo	sip:2001@192.168.0.3	Sat, 13 Feb 2010 11:11:55	NA	Not Found	404	SIP	
Onno W. Purbo	sip:2001@192.168.0.3	Sat, 13 Feb 2010 11:11:28	NA	Not Found	404	SIP	
Onno W. Purbo	sip:2001@192.168.0.3	Sat, 13 Feb 2010 11:09:48	NA	Not Found	404	SIP	
Onno W. Purbo	sip:600@192.168.0.3	Sat, 13 Feb 2010 10:59:54	2 secs	Completed	1000	SIP	4.4
Onno W. Purbo	sip:2000@192.168.0.3	Sat, 13 Feb 2010 10:59:19	NA	Not Found	404	SIP	

Showing : 1 to 10 of 15 Page : [1] 2 | View per page : [10] 20 30 40 50

© 2009 ZOHO Corp. Server Responded in 222 Milliseconds

Done

Figure 20.13 Endpoint Detailed Calls.

At the bottom of endpoint menu, we can see detailed calls performed by the particular endpoint.

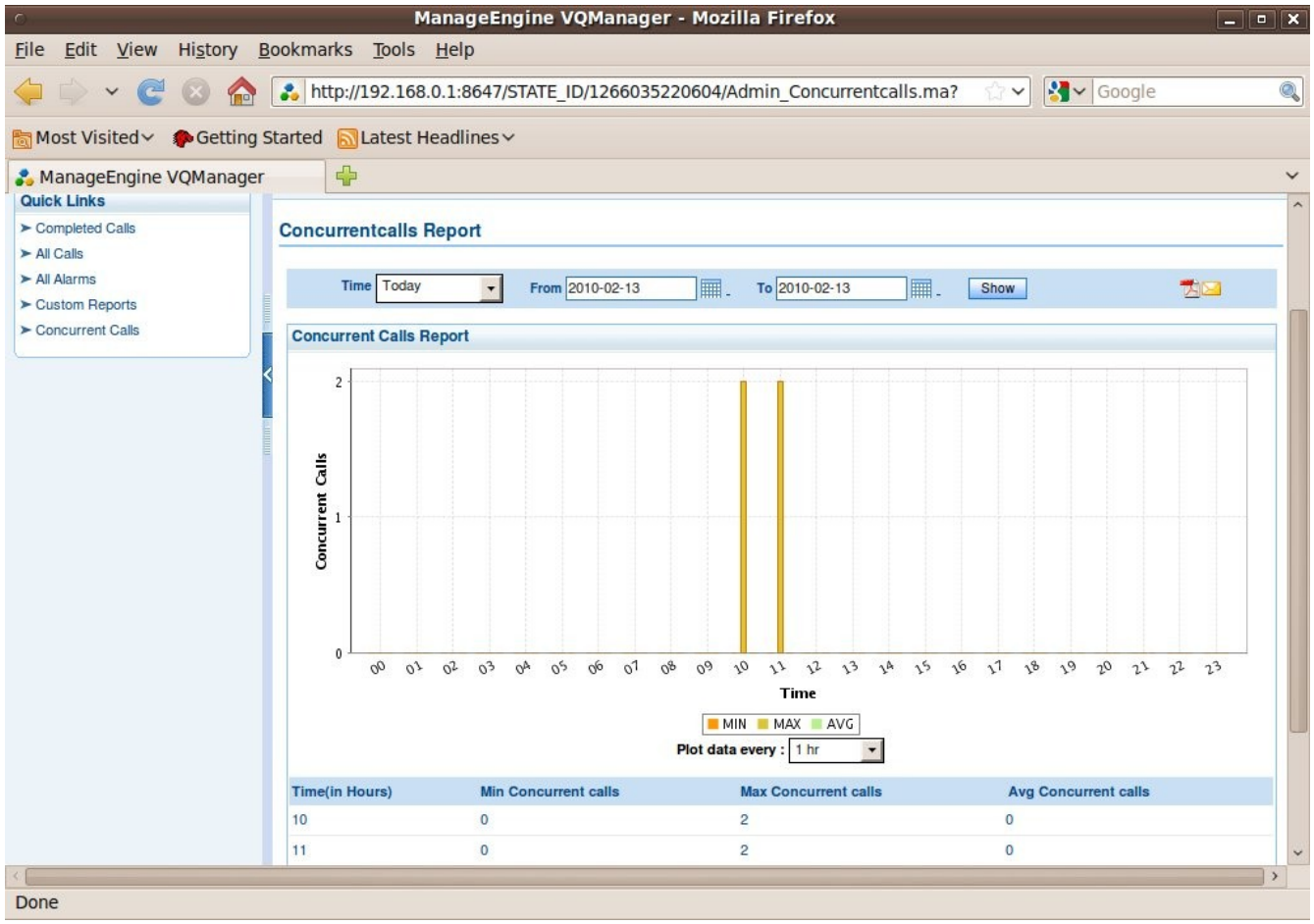


Figure 20.14 Concurrent Call.

In Concurrent Call Report menu, we can easily see how many concurrent call is handled by the softswitch. In addition, we can also see the peak hours of the traffic and its total and average concurrent calls.

ManageEngine VQManager - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://192.168.0.1:8647/STATE_ID/1266035244788/ShowViewInContentAreaAction

Most Visited Getting Started Latest Headlines

ManageEngine VQManager

Quick Links
- Completed Calls
- All Calls
- All Alarms
- Custom Reports
- Concurrent Calls

Custom Reports » CustomReports

CustomReports

Calls Summary for GoodQualityCallsReport

Export as CSV Export as PDF Email Page as PDF

Showing : 1 to 6 of 6 Page : [1] View per page : [25] 50 75 100

Initiator	Participant	Call Init Time	Duration	Call Status	Status Code	Protocol	MOS
Onno W. Purbo	sip:600@192.168.0.3	Sat, 13 Feb 2010 11:13:44	1 sec	Completed	1000	SIP	4.4
Onno W. Purbo	sip:192.168.0.3	Sat, 13 Feb 2010 11:09:50	1 min 50 secs	Completed	1000	SIP	4.4
Onno W. Purbo	sip:600@192.168.0.3	Sat, 13 Feb 2010 10:59:54	2 secs	Completed	1000	SIP	4.4
Onno W. Purbo	sip:600@192.168.0.3	Sat, 13 Feb 2010 10:52:29	6 mins 43 secs	Completed	1000	SIP	4.4
Onno W. Purbo	sip:600@192.168.0.3	Sat, 13 Feb 2010 10:12:44	1 min 45 secs	Completed	1000	SIP	4.4
Onno W. Purbo	sip:600@192.168.0.3	Sat, 13 Feb 2010 10:11:25	11 secs	Completed	1000	SIP	4.4

Showing : 1 to 6 of 6 Page : [1] View per page : [25] 50 75 100

© 2009 ZOHO Corp. Server Responded in 166 Milliseconds

Done

Figure 20.15 Good Quality Calls Report

Through Report Menu -> GoodQualityCallsReport, we can easily see the good quality call made through our softswitch.

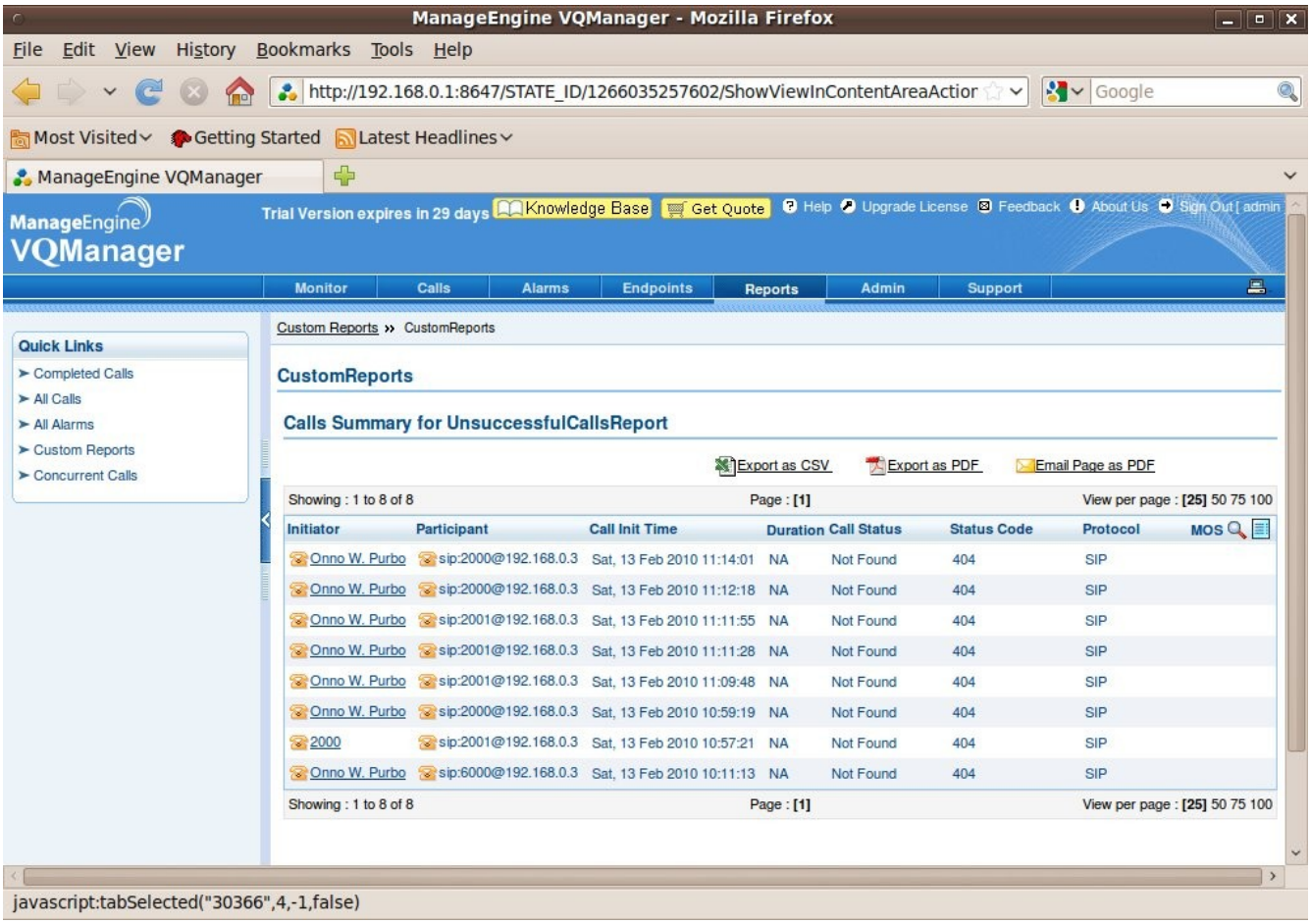


Figure 20.16 Unsuccessful Calls Report

Through Report Menu -> Unsuccessful Calls Report, we can the unsuccessfull calls. We may furthjer analyze the failure reasons.

ManageEngine VQManager - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://192.168.0.1:8647/STATE_ID/1266035271812/ShowViewInContentAreaAction

Most Visited Getting Started Latest Headlines

ManageEngine VQManager

- All Alarms
- Custom Reports
- Concurrent Calls

Calls Summary for SuccessfulCallsReport

Export as CSV Export as PDF Email Page as PDF

Showing : 1 to 11 of 11 Page : [1] View per page : [25] 50 75 100

Initiator	Participant	Call Init Time	Duration	Call Status	Status Code	Protocol	MOS
Onno W. Purbo	sip:2001@192.168.0.3	Sat, 13 Feb 2010 11:19:06	4 secs	Completed	1000	SIP	
Onno W. Purbo	sip:2000@192.168.0.3	Sat, 13 Feb 2010 11:18:17	10 secs	Completed	1000	SIP	
Onno W. Purbo	sip:600@192.168.0.3	Sat, 13 Feb 2010 11:13:44	1 sec	Completed	1000	SIP	4.4
Onno W. Purbo	sip:600@192.168.0.3	Sat, 13 Feb 2010 11:10:59	54 secs	Media Transmission S ...	8100	SIP	
Onno W. Purbo	sip:192.168.0.3	Sat, 13 Feb 2010 11:09:50	1 min 50 secs	Completed	1000	SIP	4.4
2000	sip:600@192.168.0.3	Sat, 13 Feb 2010 11:00:15	39 secs	Media Transmission S ...	8100	SIP	
Onno W. Purbo	sip:600@192.168.0.3	Sat, 13 Feb 2010 10:59:54	2 secs	Completed	1000	SIP	4.4
2000	sip:600@192.168.0.3	Sat, 13 Feb 2010 10:57:49	13 secs	Completed	1000	SIP	
Onno W. Purbo	sip:600@192.168.0.3	Sat, 13 Feb 2010 10:52:29	6 mins 43 secs	Completed	1000	SIP	4.4
Onno W. Purbo	sip:600@192.168.0.3	Sat, 13 Feb 2010 10:12:44	1 min 45 secs	Completed	1000	SIP	4.4
Onno W. Purbo	sip:600@192.168.0.3	Sat, 13 Feb 2010 10:11:25	11 secs	Completed	1000	SIP	4.4

Done

Figure 20.17 Successful Calls Report

Through Report Menu -> SuccessfulCallsReport, we can see the report on the successful call through our softswitch.

Evaluate VoIP Performance using SIPp

Installation of SIPp

To install SIPp, we can use

```
sudo apt-get install sip-tester
```

Installation of SIPp Webfrontend

Download SIPp Webfrontend from

http://sourceforge.net/projects/sipp/files/sipp/3.1/

Copy & Extract

```
mkdir /var/www/sipp
cp sipp_webfrontend_v1.2.tgz /var/www/sipp/
cd /var/www/sipp/
tar zxvf sipp_webfrontend_v1.2.tgz
mv /var/www/sipp/src/* /var/www/sipp/
```

Create database

```
mysql -u root -p
password:
CREATE DATABASE SIPpDB;
USE SIPpDB;
\. /var/www/sipp/tables.sql
quit
```

Edit config.ini.php

```
vi /var/www/sipp/config.ini.php
```

Such that

```
[EXECUTABLES]
3.0 = "/usr/bin/sipp"

[CONFIG]
db_host = "localhost"
db_user = "root"
db_pwd = "123456"
db_name = "SIPpDB"
admin_pwd = ""
```

To make it easier for accessing the web, empty the admin_pwd field. SIPp Webfrontend can be accessed via

http://localhost/sipp

using

```
username admin
password <no password>
```

Transaction Oriented Test using SIPp

In this example, we assume the IP address of the softswitch is 192.168.0.3.

Firstly, we need to setup the configuration file /usr/local/etc/opensips/cfg-test-uas.cfg at the server side. The list of cfg-test-uas.cfg is in the Appendix. Test the opensips configuration file, it can be done via,

```
# opensips -c -f /usr/local/etc/opensips/cfg-test-uas.cfg
```

If no error, we can run the server using

```
# opensips -f /usr/local/etc/opensips/cfg-test-uas.cfg
```

Run SIPp at the client side, using

```
$ sipp -sn uac 192.168.0.3
```

Or using a more complex command such as,

```
$ sipp -sn uac 192.168.0.3:5060 -m 200000 -r 10000 -d 1 -l 70
```

Example of stress testing with 1000 call per second and 10000 concurrent call using

```
$ sipp -sn uac 192.168.0.3 -r 1000 -l 10000 -d 10000
```

```
File  Edit  View  Terminal  Help

------------------------------ Scenario Screen -------- [1-9]: Change Screen --
  Call-rate(length)   Port   Total-time  Total-calls  Remote-host
1000.0(10000 ms)/1.000s   5060      72.12 s        72122  192.168.0.3:5060(UDP)

  1001 new calls during 1.001 s period   0 ms scheduler resolution
  10006 calls (limit 400000)             Peak was 10039 calls, after 50 s
  0 Running, 43007 Paused, 2462 Woken up
  0 dead call msg (discarded)            0 out-of-call msg (discarded)
  3 open sockets

                                 Messages  Retrans   Timeout   Unexpected-Msg
      INVITE ---------->         72122     5         0
         100 <----------         0         0         0         0
         180 <----------         0         0         0         0
         183 <----------         0         0         0         0
         200 <----------  E-RTD1 72122     0         0         0
         ACK ---------->         72122     0
       Pause [    10.0s]         72122                           0
         BYE ---------->         62117     8         0
         200 <----------         62116     0         0         0

------ [+|-|*|/]: Adjust rate ---- [q]: Soft exit ---- [p]: Pause traffic -----
```

Figure 20.18 SIPp Stress Test with 1000 call per second and 10000 concurrent call

```
File  Edit  View  Terminal  Help
  Start Time              | 2010-02-14   10:38:22:169    1266118702.169399

  Last Reset Time         | 2010-02-14   10:39:50:334    1266118790.334685

  Current Time            | 2010-02-14   10:39:51:336    1266118791.336647

--------------------------+---------------------------+---------------------------
  Counter Name            | Periodic value            | Cumulative value
--------------------------+---------------------------+---------------------------
  Elapsed Time            | 00:00:01:001              | 00:01:29:167
  Call Rate               |  999.001 cps              |  999.821 cps
--------------------------+---------------------------+---------------------------
  Incoming call created   |        0                  |        0
  OutGoing call created   |     1000                  |    89151
  Total Call created      |                           |    89151
  Current Call            |    10010                  |
--------------------------+---------------------------+---------------------------
  Successful call         |      996                  |    79141
  Failed call             |        0                  |        0
--------------------------+---------------------------+---------------------------
  Response Time 1         | 00:00:00:000              | 00:00:00:002
  Call Length             | 00:00:10:007              | 00:00:10:007
------ [+|-|*|/]: Adjust rate ---- [q]: Soft exit ---- [p]: Pause traffic -----
```

Figure 20.19 SIPp Stress Test page 2

The complete list of SIPp switch command is listed in the Appendix. For some, it seems very difficult to do a stress test in text mode. We can use the SIPp Webfrontend for a more user-friendly graphical interface.

Access to the SIPp Webfrontend

SIPp Webfrontend can be accessed via

http://localhost/sipp
username admin
password <no password>

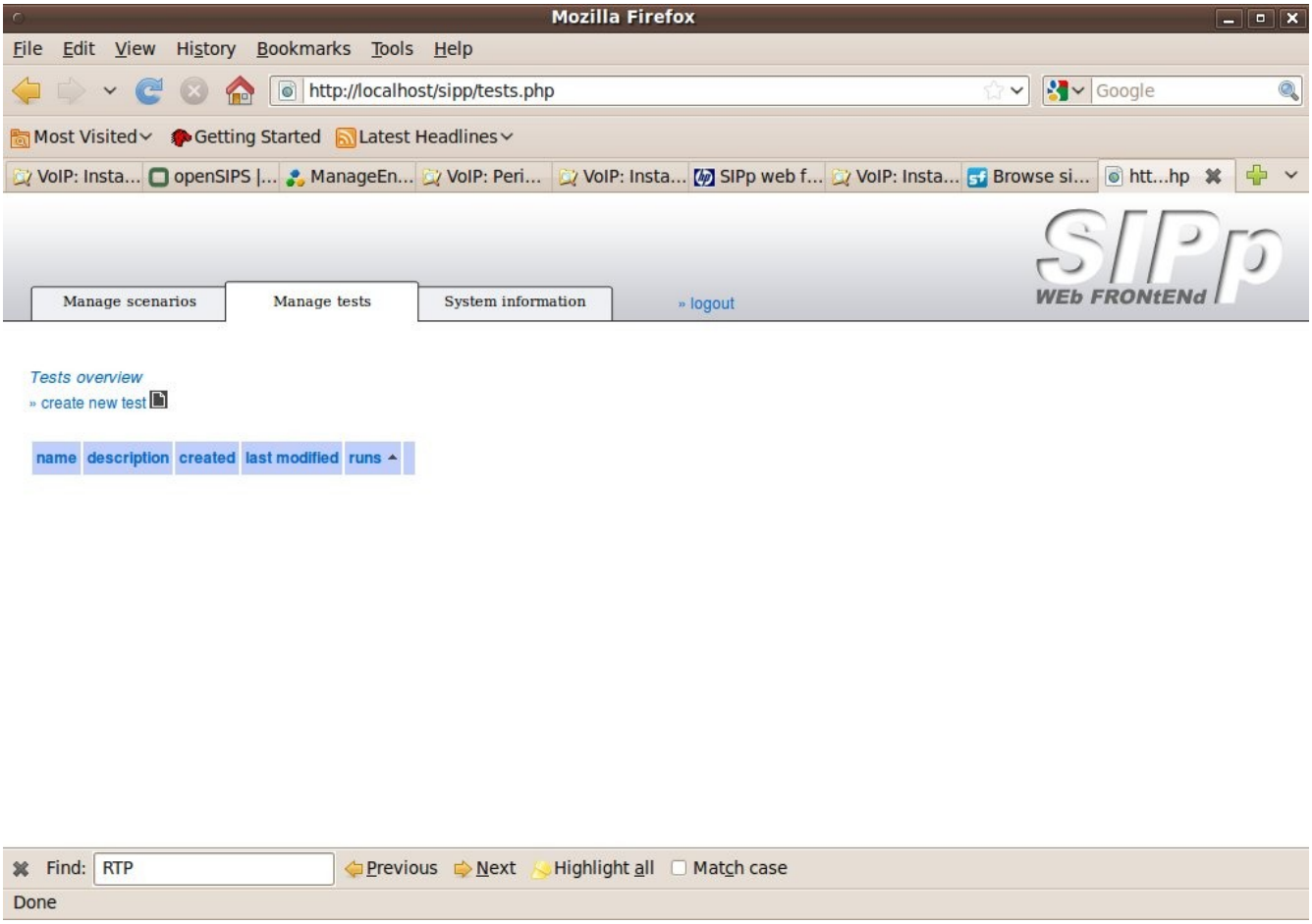


Figure 20.20 Managed Test Menu in Web Frontend

Shown in the Figure is the Web frontend menu for managing the test. For simple & common test, we basically need to access this menu only.

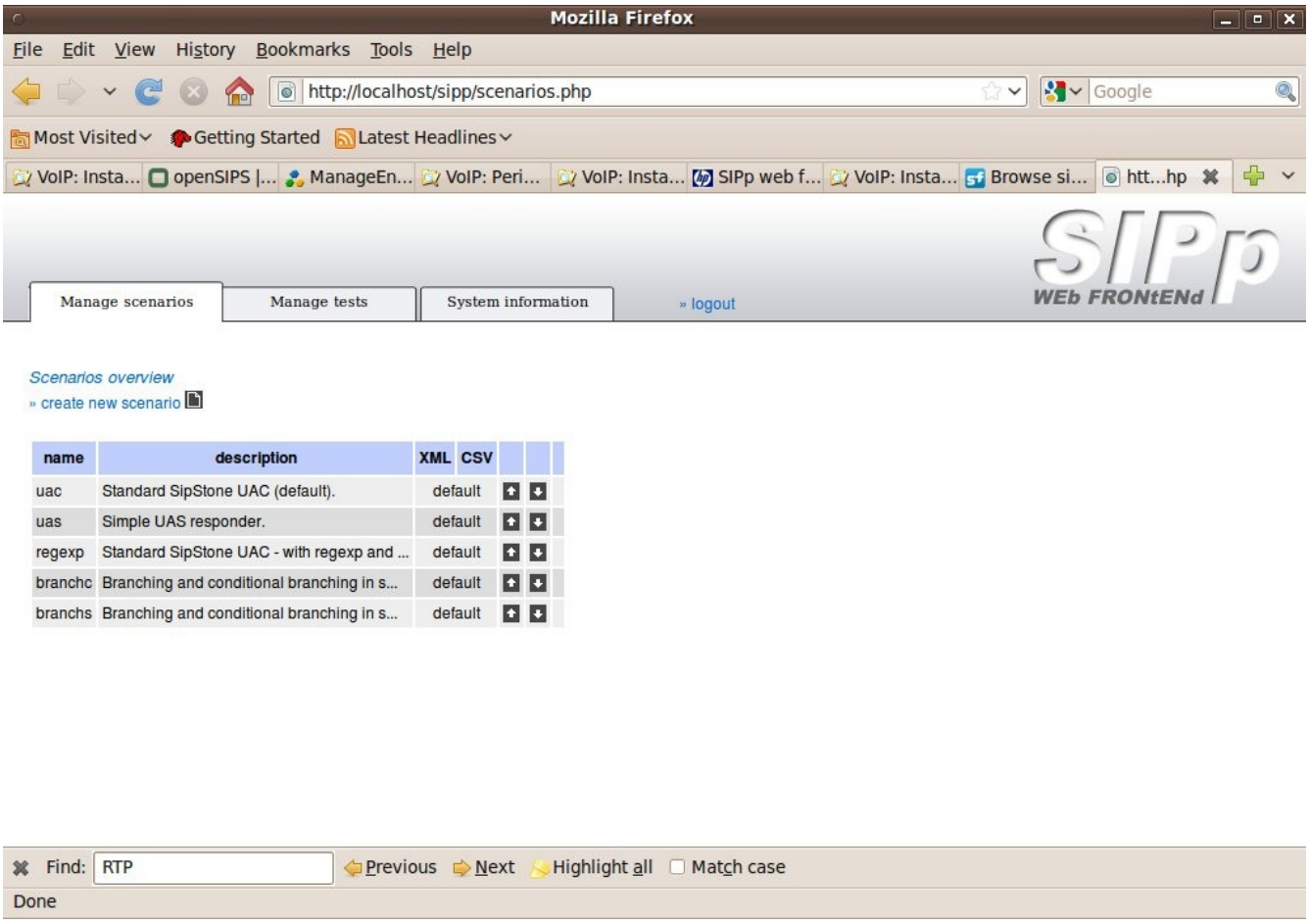


Figure 20.21 Managed Scenario Menu

Several scenario has been builtin in SIPp can be seen in the Managed Scenario menu. We can always add more scenario if you like.

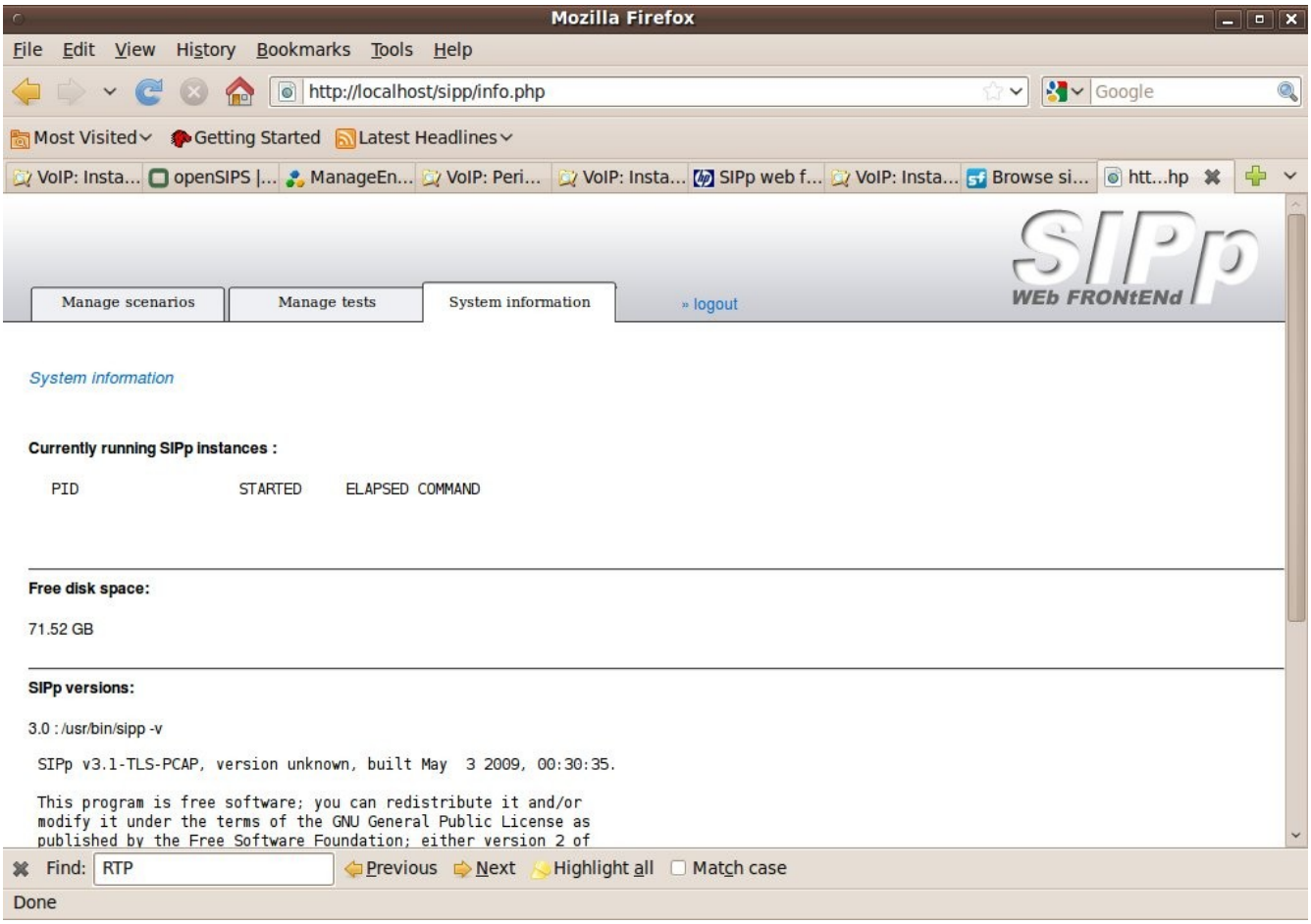


Figure 20.22 System Information Menu

In the System Information Menu, we can see some information on the system, such as, any running test, free harddisk space, SIPp version etc.

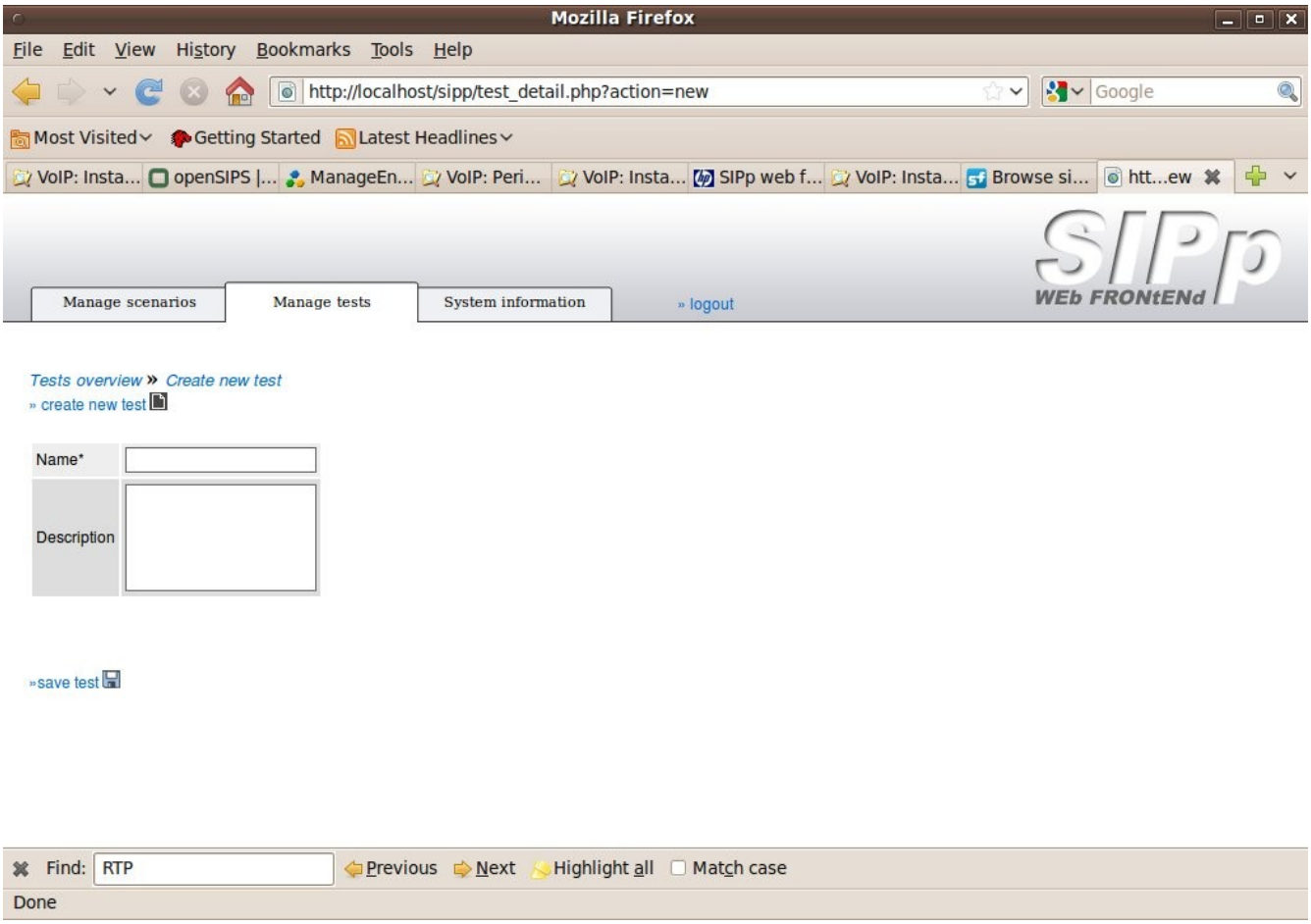


Figure 20.23 Create New Test Menu

To create a new test, in the Managed Test Menu, select Create New Test. We can type in the name of test and its description. Don't forger to press the "Save test" button to save the test.

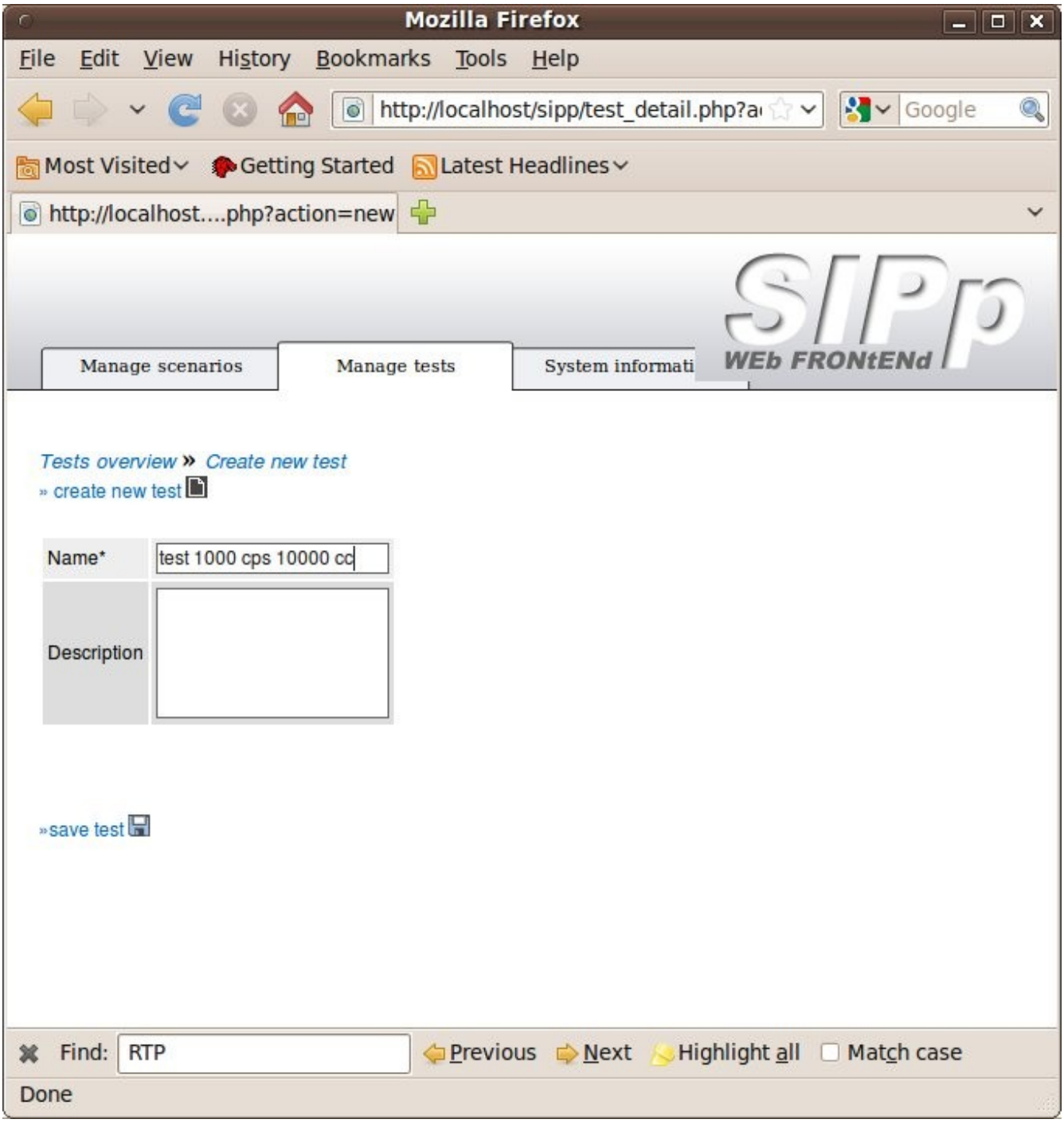


Figure 20.24 Create New Test

In the example, we use “test 1000 cps 10000 cc” for a test to create 1000 call per second and 10.000

concurrent call. Title can be anything, as long as it is informative.

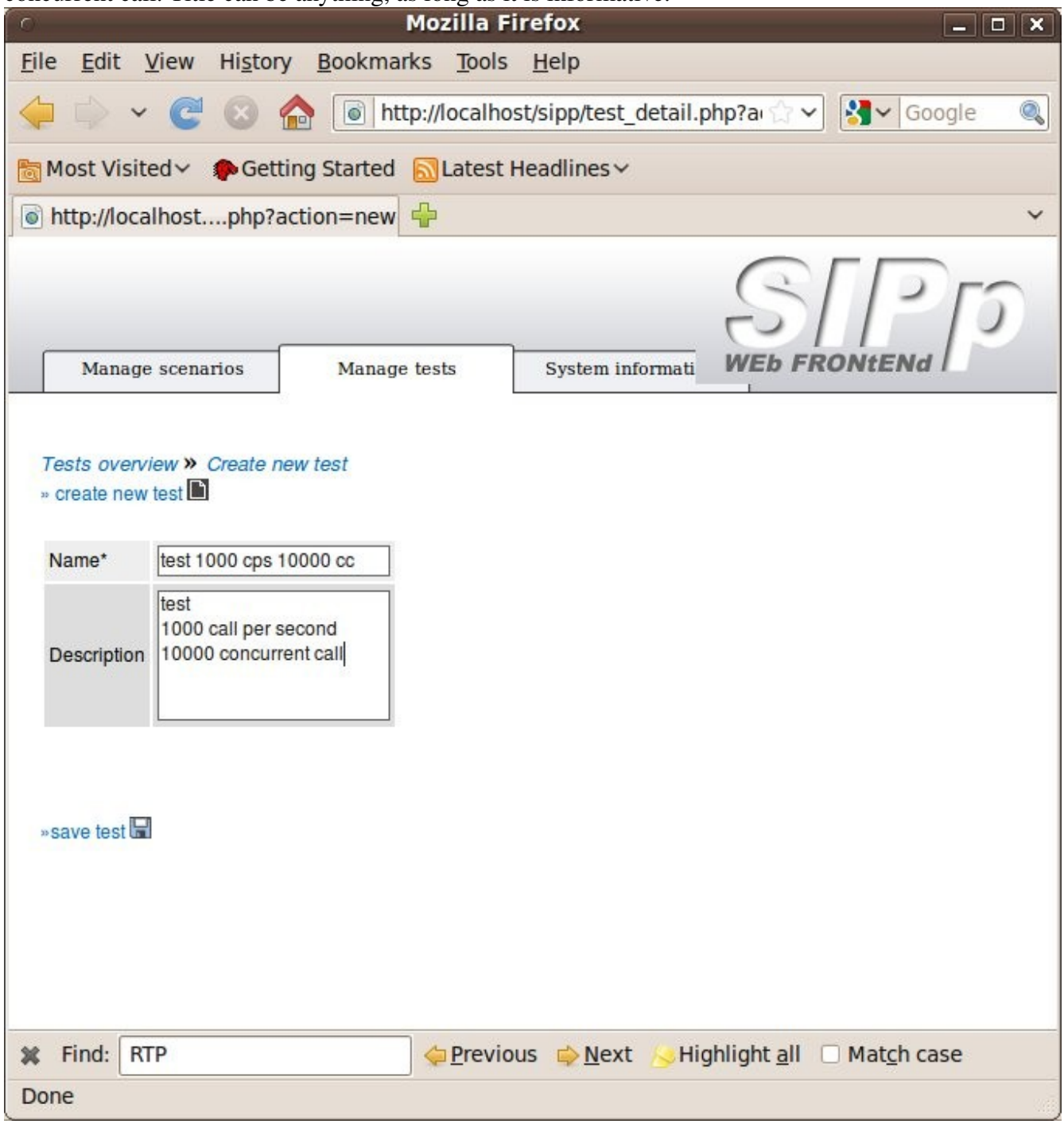


Figure 20.25 Create New Test

We can then complete the form by putting some info in the description.

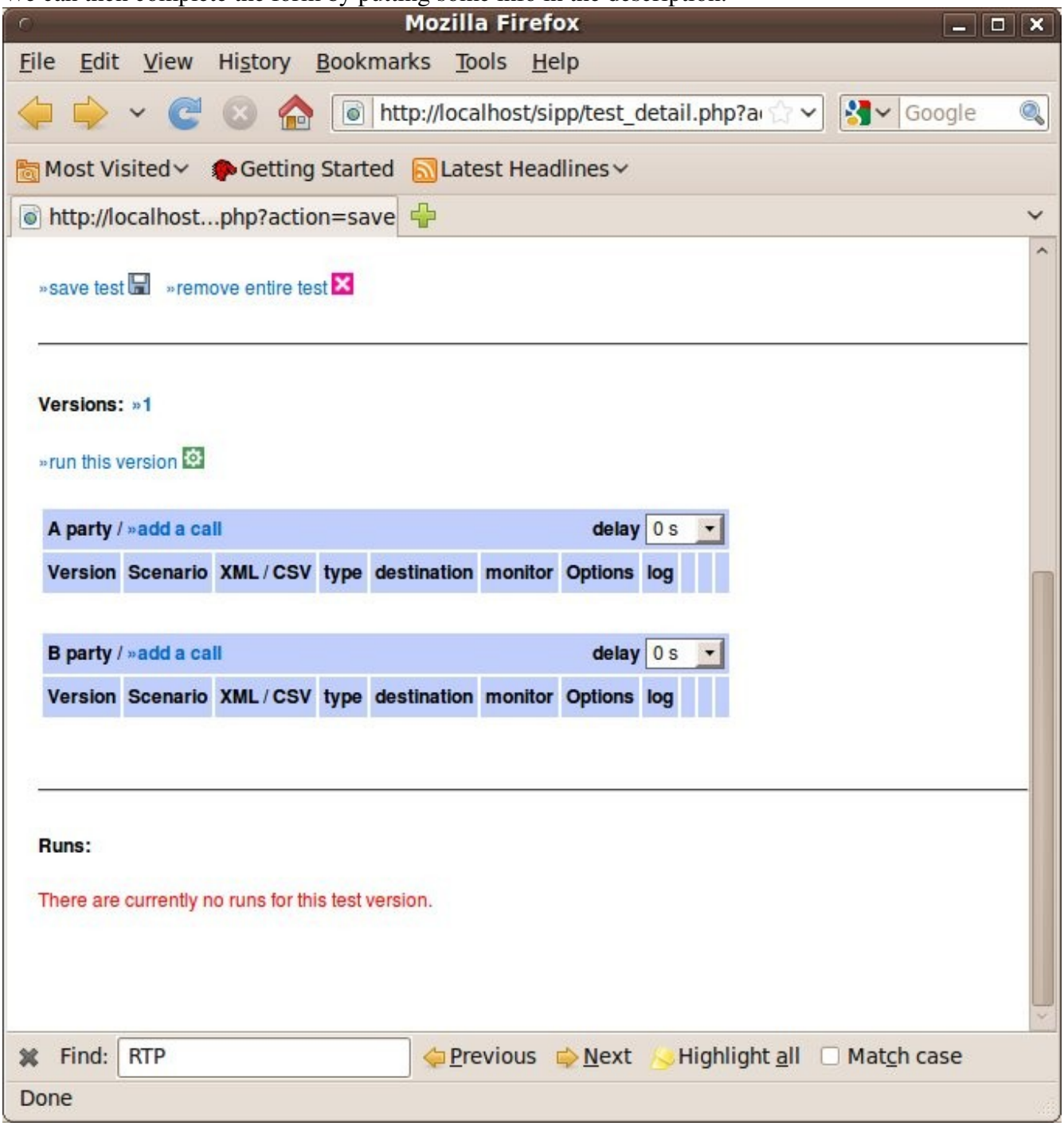


Figure 20.26 Create New Test

Below the "Save test" button will appear menu to configure the test in a more detail. It is interesting to note that we can simultaneously configure two (2) devices, namely, Party A and Party B.

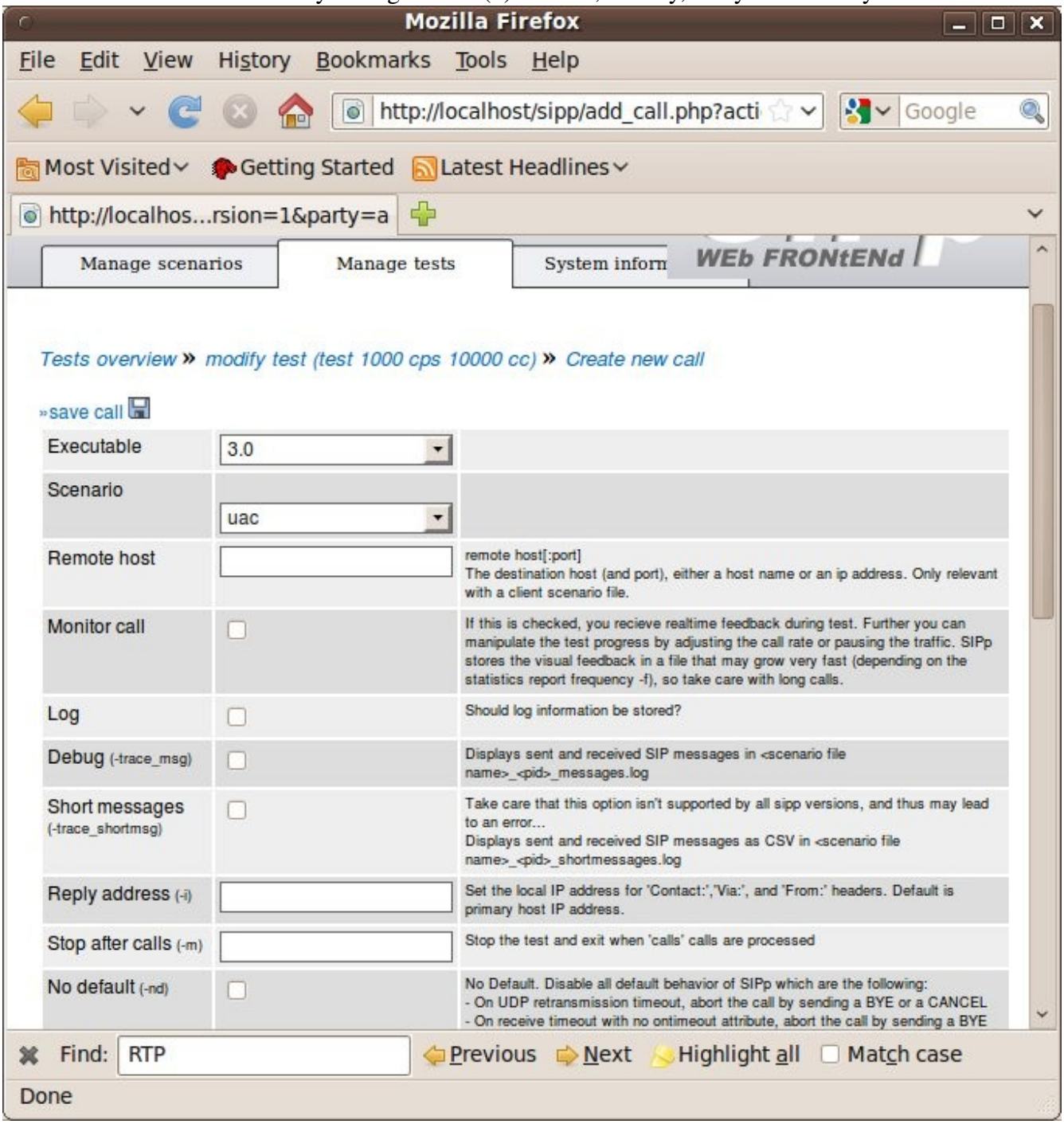
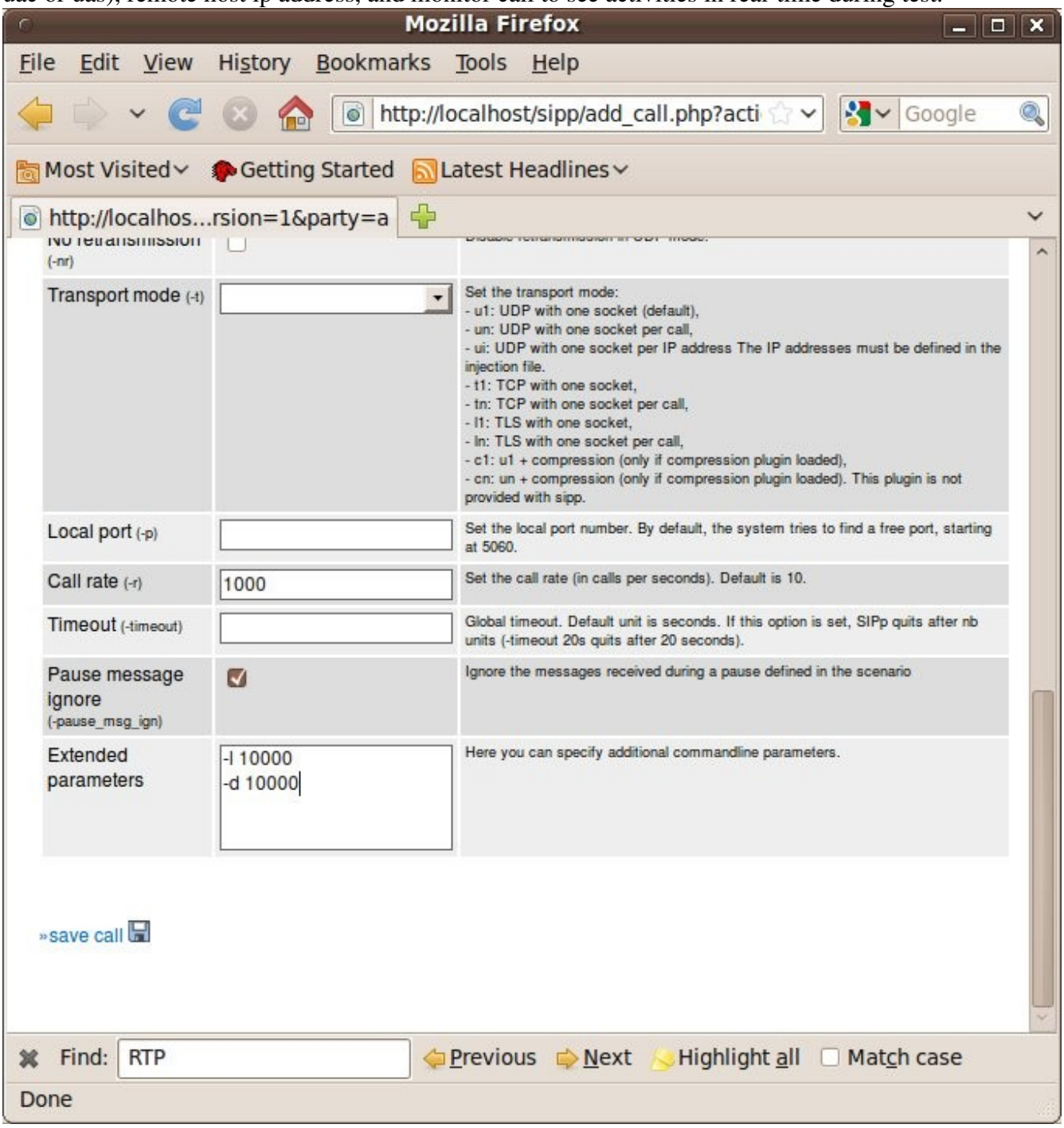


Figure 20.27 First Section of the Add Call in Create New Test

When we click add call to one of the party, some of the parameters to be configured are, scenario (as uac or uas), remote host ip address, and monitor call to see activities in real-time during test.



At the end of configuration call menu, we can set call rate and extended parameter. Press “Save call” button after we complete the configuration.

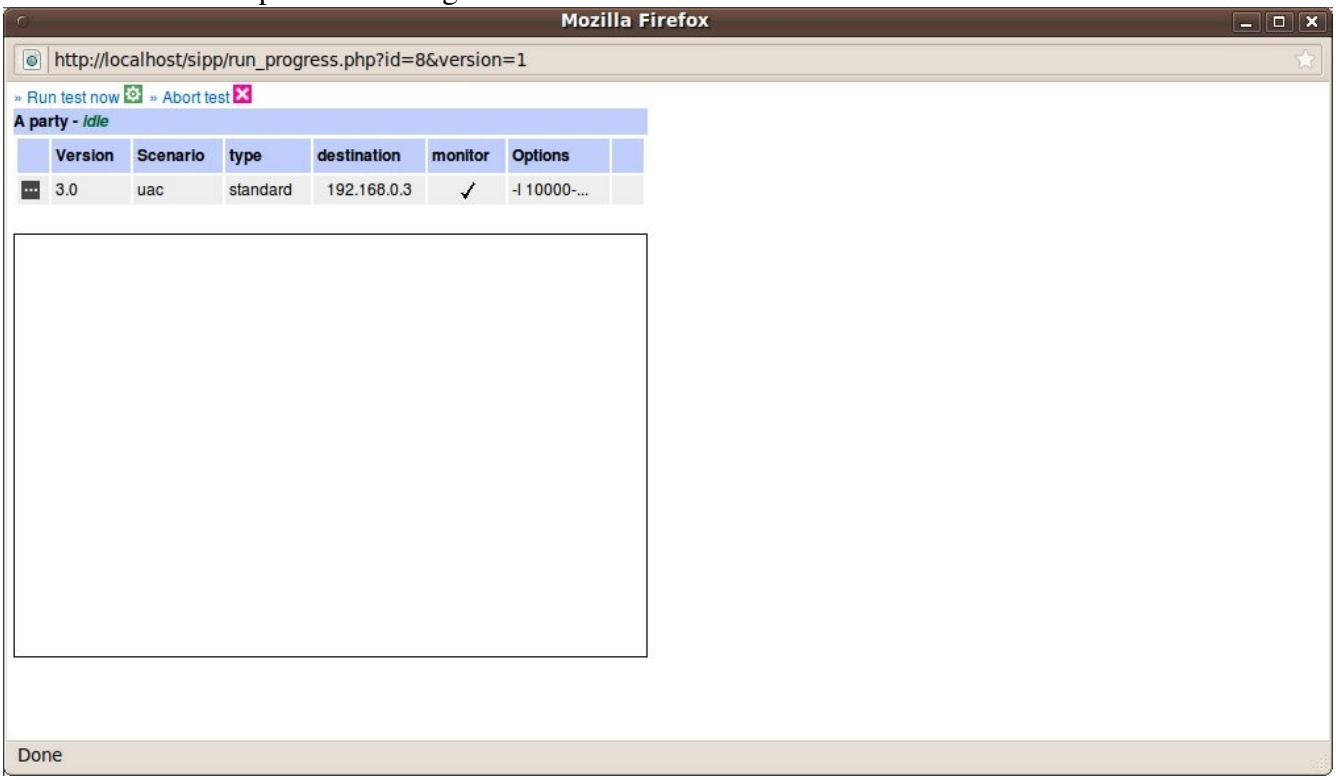


Figure 20.29 Run test menu

After all parameters completely set, we can click on “Run test” button. To do the actual run, we need to do another click on "Run test now".

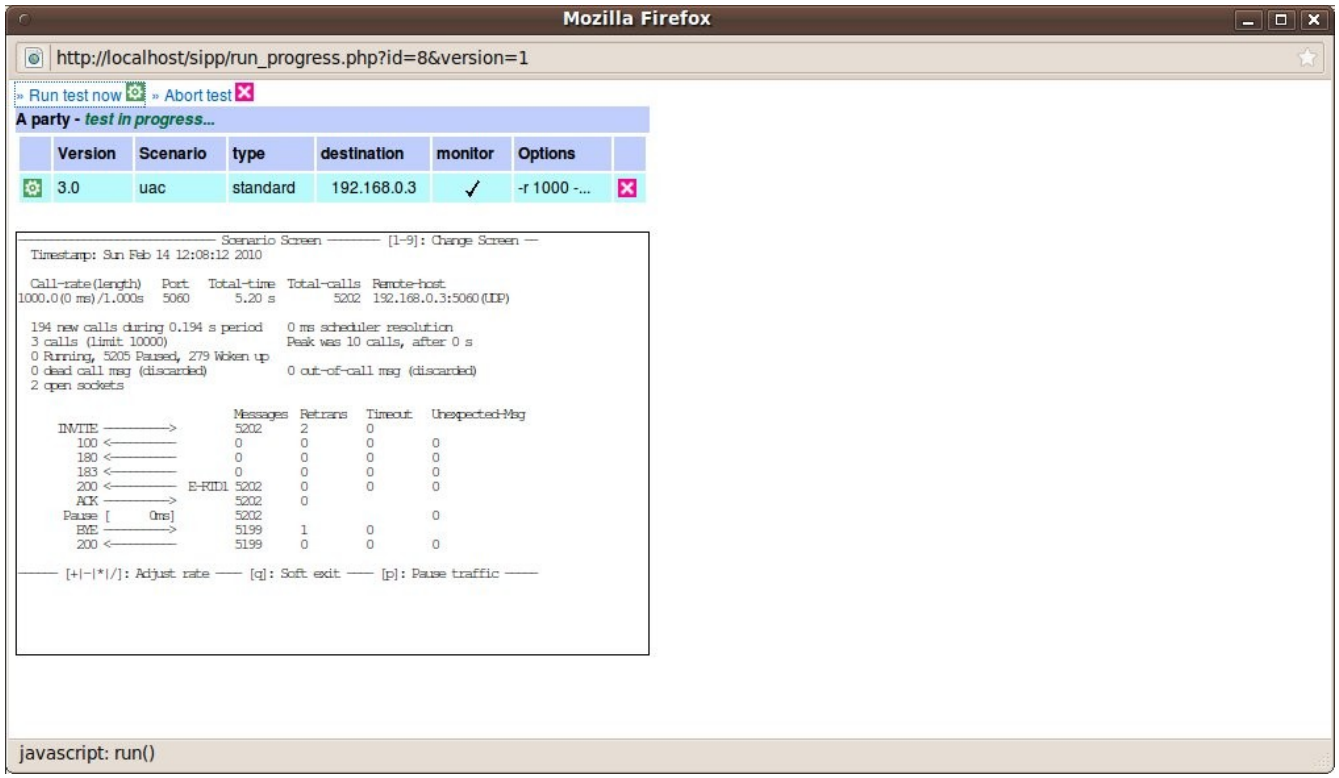


Figure 20.30 Run the Test

If the monitor call active, we can see the actual test run by SIPp on the screen. Press 1, 2, 3 key to see more information on test test.

CHAPTER 21: VoIP Troubleshooting

In general, a VoIP communication will include receive the analog voice from telephone handset, digitized, compression, packetized, sent it over the network, decode, and reconstructed into voice at the other end.

Packet network used for the job can be based on IP, ATM, Frame Relay, thus, it is logical to use the term “Voice over” includingVoIP, VoATM, VoDSL, VoCable, VoP etc. We can use all the term for VoIP.

IP Phone is usually used in the process of digitalization, compresion, and packetization internally within the phone and the produced packet is sent over the Ethernet LAN.

IP Gateway is an interface for analog telephones or digital telephones or TDM trunk to convert its audio signal to VoIP Packet. An IP PBX or Enterprise Gateway may be used by an enterprise to connect IP Phone to conventional telephone network. A trunk gateway usually used by a company to connect analog phone lines to VoIP.

CODEC and Vocoder

The term CODEC (Coder Decoder) and Vocoder may be interchangeable and normally refer to the subroutine facilities inside a softphone needed by the IP Phone or Gateway in the digitalization process. There are several CODEC that may be used, such as,

- G.711 - a PCM standard, code audio into 8 bit sample at 8000 sample per second, produces 64kbps digital audio.
- G.729 / G.729A – the 8 kbps standard.
- CODEC that aims for lower speed will normally experience distortion in the coded audio.

The published MOS of low speed CODE will normally have a relatively good quality. In reality, low speed CODEC will experience a distortion in the coded audio. In addition, it was speculated that low speed CODEC such as G.729A creates stress for continuous usage in call center.

Preparing A VoIP Ready Network

VoIP is very sensitive on any problem in IP network. Thus, it would be beneficial to prepare a clean network prior to VoIP operation. In general, a good network can be easily deployed if we know how to design a good network. We need to understand that many problem in the network that do not affect applications, such as, Web, e-mal, may create a huge problem on VoIP applications.

Minimal requirement / configuration

- Use 100 Mbps LAN switch hub with dedicated segment.
- Use Gigabit Ethernet to switch / router that connected to the server.
- Use a good IP Phone, it may be beneficial to use a switch that can prioritized voice traffic.
- Talk to your ISP, make sure we receive enough bandwidth. It would be beneficial if the ISP can prioritize Real Time Protocol (RTP) traffic.
- Make sure the router may prioritize RTP traffic.
- Make sure the firewall is configure to pass VoIP traffic.
- Design a management and monitoring infrastructure to help quick problem detection and solving.
- Do test prior to system operation.

Test prior to operation of the system

Prior to the operation of the system, it would be beneficial to systematically test the system performance to support VoIP operation. Basically, the test is merely sampling the system performance to see if it fits to the design requirement. It may increase our confidence as well as to see any possible problems.

In general, there are several type of testing prior to the operation of the system, namely,

- Test between sites. To see any possible problems due to congestion in the access network or on the wide area network. It would be better if the test can be performed in the long period of time, say one (1) month. Test process can be done by using a simple tools to simulate RTP traffic as router should handle differently as compared to ICMP.

- Pilot trial. Limited test of the planned IP telephony system to see if there would be any problem in full fledge deployment.

- Desktop testing. Our LAN may experiencing a lot of problems, we need to develop a mechanism to test each segment to see how severe the collisions and the packet loss. Use VoIP analyzer to see packet loss and jitter during installation.

Some Useful References For VoIP Troubleshooting

- http://www.voiptroubleshooter.com/
- http://www.voiptroubleshooter.com/tools/index.html
- http://www.telchemy.com/
- http://www.voiptroubleshooter.com/basics/mosr.html

References

- http://www.asterisk.org
- http://www.voip-info.org/
- http://www.voiptroubleshooter.com/
- http://www.voiprakyat.or.id
- http://www.asteriskguru.com
- http://www.e164.org
- http://www.telchemy.com/
- http://sipbroker.com/

VoIP Hardware

- http://www.digiumcards.com/
- http://www.voipon.co.uk/
- http://www.thevoipconnection.com/
- http://www.level1.com/
- http://www.linksysbycisco.com/
- https://www.digium.com/en/supportcenter/documentation/viewdocs/TDM2400P
- https://www.digium.com/en/supportcenter/documentation/viewdocs/TDM400P
- https://www.digium.com/en/supportcenter/documentation/viewdocs/TDM410

VoIP Softswitch

- http://www.asterisk.org
- http://www.briker.org
- http://www.opensips.org
- http://www.asterisk.org/asterisknow

VoIP Client Software

- http://www.counterpath.com/x-lite-download.html
- http://www.virbiage.com/cubix.php
- http://www.asteriskguru.com/idefisk/free/
- http://www.sjlabs.com/sjp.html
- http://www.xten.com/index.php?menu=download
- http://ekiga.org

Testing Software

- http://sipp.sourceforge.net/
- http://sourceforge.net/projects/sipp

- http://www.manageengine.com/products/vqmanager/

APPENDIX A: Example of /etc/sip.conf

```
;
; SIP configuration for Asterisk
;
[general]
disallow=all
allow=ulaw
port=5060                    ; the SIP port which has to be bind (attach).
bindaddr=0.0.0.0             ; Address in the SIP bind
externip=xxx.xxx.xxx.xxx     ; if we know the public IP address that we use
localnet=192.168.0.0
localmask=255.255.255.0
context=inbound-sip          ; Default context for incoming calls
maxexpirey=180
defaultexpirey=160
tos=reliability
srvlookup=yes


; IMPORTANT! Registration to SIP Server
register => 2012345:abcdef@voiprakyat.or.id/2012345      ; username 2012345 password abcdef
register => 2055555:123456@voiprakyat.or.id/2055555      ; username 2055555 passwd 123456

; We need to establish a SIP account in our place in order to receive calls from
; voiprakyat through our PABX
;
[fwd1]
type=friend
secret=secret
username=2055555
fromuser=2055555
fromdomain=voiprakyat.or.id
host=voiprakyat.or.id
dtmfmode=inband
nat=yes
canreinvite=no
```

```
[fwd2]
type=friend
secret=secret
username=2012345
fromuser=2012345
fromdomain=voiprakyat.or.id
host=voiprakyat.or.id
dtmfmode=inband
nat=yes
canreinvite=no


; The following is a SIP account for IP phone in house / office
;
[phone17]
disallow=all
allow=ulaw
type=friend
host=dynamic
defaultip=192.168.0.17
dtmfmode=inband
secret=voip17
mailbox=2206
context=home
callerid="Bill Mandra" <2206>
nat=no

[phone18]
disallow=all
allow=ulaw
type=friend
host=dynamic
defaultip=192.168.0.18
dtmfmode=inband
secret=voip18
mailbox=2204
context=home
callerid="Kitchen" <2204>
```

```
nat=no
extensions.conf

;
; Static extension file configuration used by pbx_config module
; In this module we configure all incoming calls and outgoing calls in Asterisk
;
[general]
static=yes
writeprotect=no

[globals]
DIALOUTANALOG=Zap/1
MAINPHONE=Zap/2
JESSICA=Zap/3
CHRISTOPHER=Zap/4
PORCH=Zap/5
KITCHEN=SIP/phone18
BILL=SIP/phone17
;
; For this example the card used is ZAPTEL
; We can replace Zap/1, Zap/2, Zap/3 s/d Zap/5
; with SIP account in Asterisk for IP Phone or WiFi Phone
; for example SIP/2000 to IP Phone with extension 2000 etc.

FWDUSERID1=2012345
FWD1USERNAME=William Mandra
FWDUSERID2=2055555
FWD2USERNAME=Bujubuneng Mandra
FWDPREFIX=*

HOMENUMBER=4208888

BILLCELLPHONE=0811888888
MOMCELLPHONE=0811999999
JESSCELLPHONE=0813222222

;
; Macro for Asterisk Extension
```

```
;
[macro-fastbusy]
exten => s,1,Answer
exten => s,2,Wait,1
exten => s,3,Playback(ss-noservice)
exten => s,4,Wait(30)
exten => s,5,Hangup

[macro-dialoutsip]
exten => s,1,SetCallerID(${FWDUSERID2})
exten => s,2,SetCIDName(${FWD2USERNAME})
exten => s,3,Dial(SIP/${FWDPREFIX}${ARG1}@fwd1,70)
exten => s,4,Macro(fastbusy)
exten => s,5,Hangup
exten => s,104,Macro(fastbusy)
exten => s,105,Wait,3
exten => s,106,Playtones(congestion)
exten => s,107,Wait,30
exten => s,108,Hangup

[macro-billcellfwdoutsip2]
exten => s,1,SetCallerID(${ARG2})
exten => s,2,Dial(SIP/${FWDPREFIX}${ARG1}@fwd2,20)
exten => s,3,Goto(local,2206,4)
exten => s,102,Goto(local,2206,4)

;
; Outbound
;
;
[operator]
exten => 0,1,Dial(${DIALOUTANALOG}/${EXTEN},70)
exten => 0,2,Macro(fastbusy)
exten => 0,102,Playback(ss-noservice)
exten => 0,103,Macro(fastbusy)

[e911]
exten => 911,1,Dial(${DIALOUTANALOG}/${EXTEN})
exten => 911,2,Macro(fastbusy)
```

```
exten => 911,102,Playback(ss-noservice)
exten => 911,103,Macro(fastbusy)

[forced-analog]
exten => _9.,1,Dial(${DIALOUTANALOG}/${EXTEN:1},70)
exten => _9.,2,Macro(fastbusy)
exten => _9.,102,Macro(fastbusy)

[fwd1-out]
exten => _8.,1,SetCallerID(${FWDUSERID2})
exten => _8.,2,SetCIDName(${FWD2USERNAME})
exten => _8.,3,Dial(SIP/${EXTEN:1}@fwd1,70)
exten => _8.,4,Macro(fastbusy)
exten => _8.,5,Hangup

[fwd2-out-pvt]
exten => _7.,1,SetCallerID(${FWDUSERID1})
exten => _7.,2,SetCIDName(${FWD1USERNAME})
exten => _7.,3,Dial(SIP/${EXTEN:1}@fwd2,70)
exten => _7.,4,Macro(fastbusy)
exten => _7.,5,Hangup

[information]
exten => 108,1,Dial(${DIALOUTANALOG}/${EXTEN},70)
exten => 108,2,Macro(fastbusy)
exten => 108,102,Playback(ss-noservice)
exten => 108,103,Macro(fastbusy)

; Local PSTN
;
[pstn-local]
exten => _021.,1,Dial(${DIALOUTANALOG}/${EXTEN:3})
exten => _021.,2,Macro(fastbusy)
exten => _021.,102,Macro(dialoutsip,${EXTEN})

[toll-free]
exten => _0800.,1,Dial(${DIALOUTANALOG}/${EXTEN})
exten => _0800.,2,Macro(fastbusy)
exten => _0800.,102,Macro(dialoutsip,${EXTEN})
```

```
[long-distance]
exten => _0XXXXXXXXXX,1,Macro(dialoutsip,${EXTEN})
exten => _0XXXXXXXXXX,2,Macro(fastbusy)
exten => _0XXXXXXXXXX,102,Dial(${DIALOUTANALOG}/${EXTEN})
exten => _0XXXXXXXXXX,103,Macro(fastbusy)

[home]
include => operator
include => e911
include => forced-analog
include => fwd1-out
include => fwd2-out-pvt
include => information
include => local
include => pstn-local
include => toll-free
include => long-distance

;
; Inbound
; analog line

[nighttime-analog]
exten => s,1,Wait(2)
exten => s,2,Background(nighttime)
exten => 1,1,Goto(daytime-analog,s,1)
exten => 2,1,Voicemail(u2201)
exten => 3,1,Voicemail(u2206)
exten => 4,1,Voicemail(u2202)
exten => 9,1,Playback(transfer)
exten => 9,2,Ringing(1)
exten => 9,3,Goto(local,2206,1)

[daytime-analog]
exten => s,1,Zapateller(answer|nocallerid)
exten => s,2,PrivacyManager
exten => s,3,Ringing(1)
exten => s,4,Dial(${MAINPHONE}&${KITCHEN},15)
```

```
exten => s,5,Dial(${JESSICA},6)
exten => s,6,Dial(${BILL},6)
exten => s,7,Voicemail(u2201)
exten => s,8,Hangup

[inbound-analog]
include => daytime-analog|9:00-21:00|*|*
include => nighttime-analog|21:00-09:00|*|*


; sip lines
;
[nighttime-fwd1]
exten => s,1,Wait(2)
exten => s,2,Background(nighttime)
exten => 1,1,Goto(daytime-sip1,s,1)
exten => 2,1,Voicemail(u2201)
exten => 3,1,Voicemail(u2206)
exten => 4,1,Voicemail(u2202)
exten => 9,1,Playback(transfer)
exten => 9,2,Goto(local,2206,1)

[daytime-fwd1]
exten => s,1,Dial(${MAINPHONE}&${KITCHEN},15)
exten => s,2,Dial(${JESSICA},6)
exten => s,3,Dial(${BILL},6)
exten => s,4,Voicemail(u2201)
exten => s,5,Hangup

[inbound-fwd1]
include => daytime-fwd1|9:00-21:00|*|*
include => nighttime-fwd1|21:00-9:00|*|*

[inbound-sip]
exten => 2055555,1,Goto(inbound-fwd1,s,1)
exten => 2012345,1,Goto(local,2206,1)

;
; Internal Extension
```

```
;
[local]
exten => 2201,1,Dial(${MAINPHONE},20,Tt)
exten => 2201,2,Voicemail(u2201)
exten => 2201,3,Hangup
exten => 2201,102,Voicemail(b2201)
exten => 2201,103,Hangup

exten => 2202,1,Dial(${JESSICA},20,Tt)
exten => 2202,2,Voicemail(u2202)
exten => 2202,3,Hangup
exten => 2202,102,Voicemail(b2202)
exten => 2202,103,Hangup

exten => 2203,1,Dial(${CHRISTOPHER},20,Tt)
exten => 2203,2,Playback(vm-nobodyavail)
exten => 2203,3,Hangup

exten => 2204,1,Dial(${KITCHEN},20,Tt)
exten => 2204,2,Playback(vm-nobodyavail)
exten => 2204,3,Hangup

exten => 2205,1,Dial(${PORCH},20,Tt)
exten => 2205,2,Playback(vm-nobodyavail)
exten => 2205,3,Hangup

exten => 2206,1,Dial(${BILL},20,Tt)
exten => 2206,2,Playback(transfer)
exten => 2206,3,Macro(billcellfwdoutsip2,${BILLCELLPHONE},${CALLERIDNUM})
exten => 2206,4,Voicemail(u2206)
exten => 2206,5,Hangup
exten => 2206,102,Voicemail(b2206)
exten => 2206,103,Hangup

exten => 2500,1,Wait,2
exten => 2500,2,VoicemailMain
exten => 2500,3,Hangup
;
; A variety of facilities that can be used for testing
```

```
;

exten => 2001,1,Answer
exten => 2001,2,Playback(demo-echotest)
exten => 2001,3,Echo
exten => 2001,4,Playback(demo-echodone)
exten => 2001,5,Hangup

exten => 2002,1,Answer
exten => 2002,2,WaitMusicOnHold(30)
exten => 2002,3,Hangup

exten => 2003,1,Answer
exten => 2003,2,Wait(1)
exten => 2003,3,SayUnixTime( | | k)
exten => 2003,4,SayUnixTime( | | M)
exten => 2003,5,Playback(vm-and)
exten => 2003,6,SayUnixTime( | | S)
exten => 2003,7,Wait(2)
exten => 2003,8,Hangup

exten => 2004,1,Answer
exten => 2004,2,Wait(1)
exten => 2004,3,Playback(vm-extension)
exten => 2004,4,SayDigits(${CALLERIDNUM})
exten => 2004,5,Wait(2)
exten => 2004,6,Hangup

exten => 2005,1,Goto(nighttime-analog,s,1)
;exten => 2005,2,Playback(ss-noservice)
;exten => 2005,3,Playback(vm-nobodyavail)
;exten => 2005,4,Playback(agent-incorrect)
;exten => 2005,5,Playback(agent-user)
;exten => 2005,6,Playback(pbx-invalid)
;exten => 2005,7,Playback(tt-somethingwrong)
;exten => 2005,8,Playback(vm-extension)
;exten => 2005,9,Playback(vm-isunavail)
;exten => 2005,10,Playback(vm-isonphone)
;exten => 2005,11,Playback(vm-sorry)
```

```
exten => 2005,2,Hangup
```

APPENDIX B: SIPp COMMANDS

Usage:

```
 sipp remote_host[:remote_port] [options]
```

Available options:

```
  -v              : Display version and copyright information.
  -aa             : Enable automatic 200 OK answer for INFO, UPDATE and NOTIFY messages.
  -auth_uri       : Force the value of the URI for authentication.
                    By default, the URI is composed of remote_ip:remote_port.
  -base_cseq      : Start value of [cseq] for each call.
  -bg             : Launch SIPp in background mode.
  -bind_local     : Bind socket to local IP address, i.e. the local IP address is used as the source IP
                     address.  If SIPp runs in server mode it will only listen on the local IP address
                     instead of all IP addresses.
  -buff_size      : Set the send and receive buffer size.
  -calldebug_file : Set the name of the call debug file.
  -calldebug_overwrite: Overwrite the call debug file (default true).
  -cid_str        : Call ID string (default %u-%p@%s).  %u=call_number,
                    %s=ip_address, %p=process_number, %%=% (in any order).
  -ci             : Set the local control IP address
  -cp             : Set the local control port number. Default is 8888.
  -d              : Controls the length of calls. More precisely, this controls the duration of 'pause'
                    instructions in the scenario, if they do not have a 'milliseconds' section.
                    Default value is 0 and default unit is milliseconds.
  -deadcall_wait : How long the Call-ID and final status of calls should be kept to improve message
                    and error logs (default unit is ms).
  -default_behaviors: Set the default behaviors that SIPp will use.  Possbile values are:
                              - all     Use all default behaviors
                              - none  Use no default behaviors
                              - bye    Send byes for aborted calls
                              - abortunexp    Abort calls on unexpected messages
                              - pingreply      Reply to ping requests
                    If a behavior is prefaced with a -, then it is turnedoff.  Example: all,-bye
  -error_file     : Set the name of the error log file.
  -error_overwrite : Overwrite the error log file (default true).
```

```
-f            : Set the statistics report frequency on screen. Default is 1 and default unit is seconds.
-fd           : Set the statistics dump log report frequency. Default is 60 and default unit is seconds.
-i            : Set the local IP address for 'Contact:','Via:', and 'From:' headers.
                Default is primary host IP address.
-inf          : Inject values from an external CSV file during calls into the scenarios.
                First line of this file say whether the data is to be read in sequence (SEQUENTIAL),
                random (RANDOM), or user (USER) order.
                Each line corresponds to one call and has one or more ';' delimited data fields.
                Those fields can be referred as [field0], [field1], ... in the xml scenario file.
                Several CSV files can be used simultaneously (syntax: -inf f1.csv -inf f2.csv ...)
-infindex     : file field
                Create an index of file using field.  For example -inf users.csv -infindex users.csv 0
                creates an index on the first key.
-ip_field     : Set which field from the injection file contains the IP address from which the client
                will send its messages. If this option is omitted and the '-t ui' option is present,
                then field 0 is assumed. Use this option together with '-t ui'
-l            : Set the maximum number of simultaneous calls. Once this limit is reached,
                traffic is decreased until the number of open calls goes down.
                Default: (3 * call_duration (s) * rate).
-log_file     : Set the name of the log actions log file.
-log_overwrite : Overwrite the log actions log file (default true).
-lost         : Set the number of packets to lose by default
                (scenario specifications override this value).
-rtcheck      : Select the retransmisison detection method: full (default) or loose.
-m            : Stop the test and exit when 'calls' calls are processed
-mi           : Set the local media IP address (default: local primary host IP address)
-master       : 3pcc extended mode: indicates the master number
-max_recv_loops  : Set the maximum number of messages received read per cycle.
                Increase this value for high traffic level.  The default value is 1000.
-max_sched_loops : Set the maximum number of calsl run per event loop.
                Increase this value for high traffic level.  The default value is 1000.
-max_reconnect  : Set the the maximum number of reconnection.
-max_retrans : Maximum number of UDP retransmissions before call ends on timeout.
                Default is 5 for INVITE transactions and 7 for others.
-max_invite_retrans: Maximum number of UDP retransmissions for invite transactions
                before call ends on timeout.
-max_non_invite_retrans: Maximum number of UDP retransmissions for non-invite transactions
                before call ends on timeout.
```

-max_log_size : What is the limit for error and message log file sizes.
-max_socket : Set the max number of sockets to open simultaneously. This option is significant
if you use one socket per call. Once this limit is reached,
traffic is distributed over the sockets already opened. Default value is 50000
-mb : Set the RTP echo buffer size (default: 2048).
-message_file : Set the name of the message log file.
-message_overwrite: Overwrite the message log file (default true).
-mp : Set the local RTP echo port number. Default is 6000.
-nd : No Default. Disable all default behavior of SIPp which are the following:

- On UDP retransmission timeout, abort the call by sending a BYE or a CANCEL
- On receive timeout with no ontimeout attribute, abort the call by sending a BYE or a CANCEL
- On unexpected BYE send a 200 OK and close the call
- On unexpected CANCEL send a 200 OK and close the call
- On unexpected PING send a 200 OK and continue the call
- On any other unexpected message, abort the call by sending a BYE or a CANCEL

-nr : Disable retransmission in UDP mode.
-nostdin : Disable stdin.
-p : Set the local port number. Default is a random free port chosen by the system.
-pause_msg_ign : Ignore the messages received during a pause defined in the scenario
-periodic_rtd : Reset response time partition counters each logging interval.
-plugin : Load a plugin.
-r : Set the call rate (in calls per seconds). This value can bechanged during test
by pressing '+','_','*' or '/'. Default is 10.
pressing '+' key to increase call rate by 1 * rate_scale,
pressing '-' key to decrease call rate by 1 * rate_scale,
pressing '*' key to increase call rate by 10 * rate_scale,
pressing '/' key to decrease call rate by 10 * rate_scale.
If the -rp option is used, the call rate is calculated with the period in ms
given by the user.
-rp : Specify the rate period for the call rate. Default is 1 second and default unit
is milliseconds. This allows you to have n calls every m milliseconds
(by using -r n -rp m).
Example: -r 7 -rp 2000 ==> 7 calls every 2 seconds.
-r 10 -rp 5s => 10 calls every 5 seconds.
-rate_scale : Control the units for the '+', '-', '*', and '/' keys.
-rate_increase : Specify the rate increase every -fd units (default is seconds).
This allows you to increase the load for each independent logging period.

```
                Example: -rate_increase 10 -fd 10s ==> increase calls by 10 every 10 seconds.
-rate_max     : If -rate_increase is set, then quit after the rate reaches this value.
                Example: -rate_increase 10 -rate_max 100 ==> increase calls by 10 until 100 cps is hit.
-no_rate_quit : If -rate_increase is set, do not quit after the rate reaches -rate_max.
-recv_timeout : Global receive timeout. Default unit is milliseconds. If the expected message is not
                received, the call times out and is aborted.
-send_timeout : Global send timeout. Default unit is milliseconds. If a message is not sent
                (due to congestion), the call times out and is aborted.
-sleep        : How long to sleep for at startup. Default unit is seconds.
-reconnect_close : Should calls be closed on reconnect?
-reconnect_sleep : How long (in milliseconds) to sleep between the close and reconnect?
-ringbuffer_files: How many error/message files should be kept after rotation?
-ringbuffer_size : How large should error/message files be before they get rotated?
-rsa          : Set the remote sending address to host:port for sending the messages.
-rtp_echo     : Enable RTP echo. RTP/UDP packets received on port defined
                by -mp are echoed to their sender. RTP/UDP packets coming on
                this port + 2 are also echoed to their sender (used for sound and video echo).
-rtt_freq     : freq is mandatory. Dump response times every freq calls in the log file defined
                by -trace_rtt. Default value is 200.
-s            : Set the username part of the resquest URI. Default is 'service'.
-sd           : Dumps a default scenario (embeded in the sipp executable)
-sf           : Loads an alternate xml scenario file.  To learn more about XML scenario syntax,
                use the -sd option to dump embedded scenarios. They contain all the necessary help.
-shortmessage_file: Set the name of the short message log file.
-shortmessage_overwrite: Overwrite the short message log file (default true).
-oocsf        : Load out-of-call scenario.
-oocsn        : Load out-of-call scenario.
-skip_rlimit  : Do not perform rlimit tuning of file descriptor limits.
                Default: false.
-slave        : 3pcc extended mode: indicates the slave number
-slave_cfg    : 3pcc extended mode: indicates the file where the master and slave addresses are stored
-sn           : Use a default scenario (embedded in the sipp executable).
                If this option is omitted, the Standard SipStone UAC scenario is loaded.
                Available values in this version:
                        - 'uac'     : Standard SipStone UAC (default).
                        - 'uas'     : Simple UAS responder.
                        - 'regexp'  : Standard SipStone UAC - with regexp and variables.
                        - 'branchc' : Branching and conditional branching in scenarios - client.
                        - 'branchs' : Branching and conditional branching in scenarios - server.
```

```
                    Default 3pcc scenarios (see -3pcc option):
                            - '3pcc-C-A' : Controller A side (must be started after all other 3pcc scenarios)
                            - '3pcc-C-B' : Controller B side.
                            - '3pcc-A'   : A side.
                            - '3pcc-B'   : B side.
-stat_delimiter : Set the delimiter for the statistics file
-stf            : Set the file name to use to dump statistics
-t              : Set the transport mode:
                      - u1: UDP with one socket (default),
                      - un: UDP with one socket per call,
                      - ui: UDP with one socket per IP address.
                          The IP addresses must be defined in the injection file.
                      - t1: TCP with one socket,
                      - tn: TCP with one socket per call,
                      - l1: TLS with one socket,
                      - ln: TLS with one socket per call,
                      - c1: u1 + compression (only if compression plugin loaded),
                      - cn: un + compression (only if compression plugin loaded).
                     This plugin is not provided with sipp.
-timeout        : Global timeout. Default unit is seconds.  If this option is set, SIPp quits after
                  nb units (-timeout 20s quits after 20 seconds).
-timeout_error : SIPp fails if the global timeout is reached is set (-timeout option required).
-timer_resol : Set the timer resolution. Default unit is milliseconds.  This option has an impact
                  on timers precision. Small values allow more precise scheduling but impacts CPU
                  usage.If the compression is on, the value is set to 50ms. The default value is 10ms.
-sendbuffer_warn : Produce warnings instead of errors on SendBuffer failures.
-trace_msg    : Displays sent and received SIP messages in <scenario file name>_<pid>_messages.log
-trace_shortmsg : Displays sent and received SIP messages as CSV
                  in <scenario file name>_<pid>_shortmessages.log
-trace_screen: Dump statistic screens in the <scenario_name>_<pid>_cenaris.log file
                  when quitting SIPp. Useful to get a final status report in background mode (-bg option).
-trace_err      : Trace all unexpected messages in <scenario file name>_<pid>_errors.log.
-trace_calldebug : Dumps debugging information about aborted calls to
                  <scenario_name>_<pid>_calldebug.log file.
-trace_stat     : Dumps all statistics in <scenario_name>_<pid>.csv file.
                  Use the '-h stat' option for a detailed description of the statistics file content.
-trace_counts : Dumps individual message counts in a CSV file.
-trace_rtt      : Allow tracing of all response times in <scenario file name>_<pid>_rtt.csv.
-trace_logs     : Allow tracing of <log> actions in <scenario file name>_<pid>_logs.log.
```

```
-users          : Instead of starting calls at a fixed rate, begin 'users' calls at startup, and
                  keep the number of calls constant.
-watchdog_interval: Set gap between watchdog timer firings.  Default is 400.
-watchdog_reset  : If the watchdog timer has not fired in more than this time period,
                  then reset the max triggers counters. Default is 10 minutes.
-watchdog_minor_threshold: If it has been longer than this period between watchdog
                  executions count a minor trip.  Default is 500.
-watchdog_major_threshold: If it has been longer than this period between watchdog
                  executions count a major trip.  Default is 3000.
-watchdog_major_maxtriggers: How many times the major watchdog timer can be tripped
                  before the test is terminated.  Default is 10.
-watchdog_minor_maxtriggers: How many times the minor watchdog timer can be tripped
                  before the test is terminated.  Default is 120.
-ap             : Set the password for authentication challenges. Default is 'password
-tls_cert       : Set the name for TLS Certificate file. Default is 'cacert.pem
-tls_key        : Set the name for TLS Private Key file. Default is 'cakey.pem'
-tls_crl        : Set the name for Certificate Revocation List file.
                  If not specified, X509 CRL is not activated.
-3pcc           : Launch the tool in 3pcc mode ("Third Party call control").
                  The passed ip address is depending on the 3PCC role.
                 - When the first twin command is 'sendCmd' then this is
                   the address of the remote twin socket.  SIPp will try to
                   connect to this address:port to send the twin command
                   (This instance must be started after all other 3PCC scenario).
                   Example: 3PCC-C-A scenario.
                 - When the first twin command is 'recvCmd' then this is
                   the address of the local twin socket. SIPp will open
                   this address:port to listen for twin command.
                   Example: 3PCC-C-B scenario.
-tdmmap         : Generate and handle a table of TDM circuits.
                  A circuit must be available for the call to be placed.
                  Format: -tdmmap {0-3}{99}{5-8}{1-31}
-key            : keyword value
                  Set the generic parameter named "keyword" to "value".
-set            : variable value
                  Set the global variable parameter named "variable" to "value".
-dynamicStart : variable value
                  Set the start offset of dynamic_id varaiable
-dynamicMax : variable value. Set the maximum of dynamic_id variable
```

-dynamicStep : variable value. Set the increment of dynamic_id variable

Signal handling:

SIPp can be controlled using posix signals. The following signals are handled:

USR1: Similar to press 'q' keyboard key. It triggers a soft exit of SIPp.
No more new calls are placed and all ongoing calls are finished before SIPp exits.
Example: kill -SIGUSR1 732

USR2: Triggers a dump of all statistics screens in <scenario_name>_<pid>_screens.log file.
Especially useful in background mode to know what the current status is.
Example: kill -SIGUSR2 732

Exit code:

Upon exit (on fatal error or when the number of asked calls (-m option) is reached,
sipp exits with one of the following exit code:

0: All calls were successful
1: At least one call failed
97: exit on internal command. Calls may have been processed
99: Normal exit without calls processed
-1: Fatal error

Example:

Run sipp with embedded server (uas) scenario:

```
./sipp -sn uas
```

On the same host, run sipp with embedded client (uac) scenario

```
./sipp -sn uac 127.0.0.1
```

APPENDIX C: File /usr/local/etc/opensips/cfg-test-uas.cfg

```
# ----------- global configuration parameters ------------------------

debug=3           # debug level (cmd line: -dddddddddd)
fork=yes
log_stderror=no    # (cmd line: -E)
children=8

disable_tcp=yes
disable_dns_blacklist=yes
disable_dns_failover=yes

# Uncomment these lines to enter debugging mode
#fork=no
#log_stderror=yes

#listen=udp:192.168.2.102:5070
#listen=192.168.0.2:5060

#------------------ module loading ----------------------------------

# set module path
# mpath="/usr/local/lib/openser/modules/"
# mpath="/usr/lib/opensips/modules/"
mpath="/usr/local/lib/opensips/modules/"

loadmodule "sl.so"

# -------------------------  request routing logic -------------------

# main routing logic

route{
      sl_send_reply("200", "OK");
}
```