Simulation of Voice over IP in Linux using SIP

Project work submitted on partial fulfillment of the requirements

for the award of the degree of

Master of Computer Applications

of

Bharathiar University

by

JOHN M.J

(Reg. No. 0038M1027)

Under the guidance of

Mr. **Sivan Arul Selvan**, MCA

Department of Computer Science and Engineering

Department of Computer Science and Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY

(Affiliated to Bharathiar University)

Coimbatore – 641 006

Department of Computer Science and Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE – 641 006.

CERTIFICATE

PROJECT REPORT 2002 - 2003

Certified that this is a bonafide report of
the project work done by

JOHN M.J

(Reg. No. 0038M1027)

K. Sivarulselvan 07/04/03	S. Thangasamy
Mr. SIVAN ARUL SELVAN, MCA.,	**Prof. S. THANGASAMY, Ph.D.,**
Project guide	Head of the Department
Computer Science & Engineering	Computer Science & Engineering

Place: Coimbatore

Date: 16-04-2003

Submitted for viva-voce examination held on

16-04-2003

16/4	16/4/03
Internal Examiner	**External Examiner**

DECLARATION

Declaration

This is to certify that this project work entitled "**SIMULATION OF VOICE OVER IP IN LINUX USING SIP**" being submitted by **JOHN M.J** (Reg. No. 0038M1027) for the award of degree of MASTER OF COMPUTER APPLICATIONS is a bonafide work carried under my guidance. The results embodied in this project work have not been submitted to any other university or institute for the award of any degree or diploma.

JOHN . M . J

K. Sivanarulselvan 07/04/03

Mr. Sivan Arul Selvan, MCA,
Dept. of Computer Science and Engineering,
Kumaraguru College of Technology,
Coimbatore – 641 006.

CERTIFICATE

SRA Systems Limited

April 03, 2003

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Mr. John M J** who is pursuing his final year M.C.A (Master of Computer Applications) at Kumaraguru College of Technology, Coimbatore has successfully completed his project work entitled **"SIMULATION OF VOIP IN LINUX USING SIP"** as part of his academic curriculum during December 2002 and March 2003.

During the project tenure his attendance was found to be regular.

We wish him all the best in his future endeavours

Anand Rajagopalan
Senior Executive - Projects

No. 100, Valluvarkottam High Road, Nungambakkam, Chennai - 600 034.
Tel: (91) (44) 825 5519, 825 5520, 825 5541 Fax: (91) (44) 825 6984
E-mail : corporate@srasys.co.in
www.srasys.co.in

✵ ACKNOWLEDGEMENT ✵

Acknowledgement

I would like to express my sincere gratitude to a few who have contributed a lot towards the completion of my project.

I express my sincere thanks to **Dr. K.K.Padmanabhan, Ph.D.,** Principal, Kumaraguru College of Technology for extending his kind cooperation.

I would like to extend my sincere thanks to **Dr.S.Thangasamy, Ph.D.,** Head of the department, Computer Science and Engineering for allowing me to take up this project at SRA Systems Limited, Nungambakkam, Chennai.

I extend my sincere thanks to **Mr. Muthukumar, MCA., M.Phil.,** Course Coordinator, Master of Computer Applications, Kumaraguru College of Technology, for his invaluable support and encouragement.

I owe a deep debt of gratitude to my guide **Mr. Sivan Arul Selvan, MCA,** Department of Computer Science and Engineering, Kumaraguru College of Technology, for his outstanding guidance throughout the project period.

I also extend my gratitude to **Mr. P.K.Shibu, Mr. Anand Rajagopalan** and **Mr. Charles Daniel** of **SRA Systems Limited** for allowing me to carry out this project at their organization and guiding me in completing it successfully.

Above all I would like to thank the **LORD Almighty** for all His invaluable blessings and guidance.

SYNOPSIS

Synopsis

The project entitled "**Simulation of Voice over Internet Protocol in Linux using SIP**" mainly deals with projecting the working of the VoIP calling system in a simulated environment setup. The simulation can be carried out in a LAN environment or within any two terminals of a single Linux system. The transmission takes place in the form of IP packets which are sent and received through specific ports. The three main protocols involved here are SIP (Session Initiation Protocol), MGCP (Media Gateway Control Protocol) and H.323. The simulation of the system on a single client is made possible with the help of a Load Generator. This simulation enables an efficient and cost effective platform for the development of VoIP systems. The steps involved are as follows

- Initializing the SETUP in both the terminals.
- Establishing a connection by dialing using the specified parameters.
- Depending on the setup either PLAY or RECORD modules are invoked.
- If one terminal is in PLAY mode then the other should be in the RECORD mode.
- After the transmission is over the TEARDOWN module is invoked and the session is terminated.
- The ANNOUNCE module posts a description of the media object identified by the request URL to the server.

CONTENTS

Contents

1. INTRODUCTION ✵

1. Introduction

1.1 Problem definition

The enormous growth of the Internet and the development of new technologies have increased the competition among IT companies to provide cost effective and quality oriented methods of communication for the people. The world of telecommunications have contributed a lot to bringing people closer and turning the world into a much smaller place to live in. But there are a few drawbacks in the existing system that resulted in the development of VoIP. One of the main drawback is the cost factor that has been reduced to a great extent in VoIP.

The implementation of VoIP in real time requires many high end machines that are very expensive. Some of the equipments needed are CISCO 5000 series VoIP enabled routers and gateways and around eight high end architecture servers. Moreover SIP enabled phones are not as freely available as the other phones.

This simulation system is developed to solve all these problems. Using this environment, the working of the VoIP system can be stimulated easily. It can be done even on a single stand alone system with RedHat Linux versions 6.2 to 7.3 and also on Sun Solaris version 8.0. It brings down the cost overhead of testing the VoIP system down by a very significant level. Moreover there is no need for all those high end equipments like routers or IP phones.

1.2 Organization Profile

S R A Systems Limited, Nungambakkam, Chennai.

SRA Systems was conceived and formed by industry professionals with the motive to deliver quality IT Services, Products, and Solutions. Since its inception in 1986, the company has been a source of Customer Delight by providing Reliable, Cost-effective, and Timely Techno-Business Solutions to its global clientele. For them, SRA represents an open access to cutting edge technology in the e-age. SRA cultivates quality processes among organizations across the globe and helps them take one step ahead of their competitors.

SRA is an ISO 9001 certified organization already assessed at CMM Level 4 and striding towards CMM Level 5 recognition. SRA adopts a matrix approach where the IT Services (Projects & IT Enabled Services) are primarily based on its IT Practices. Additionally, SRA has been specializing in a few select industry domains (Industry Practices) as a means of moving up the value chain and as a growth strategy. Operations span across USA, Europe, Japan and the Asia Pacific market. For the US Market, SRA has a fully owned subsidiary in the US. SRA has branches in

- Atlanta, Georgia
- Chicago, Illinois
- Cupertino, California
- Sydney, Australia

1.3 Existing System

The earlier versions of VoIP systems developed using SIP had to be implemented in real time using all the expensive hardware components. Error detection and recovery was a very difficult process and it involved a considerable amount of man power and time. The most important setback that the developers of VoIP faced that time was the cost factor. The system had to be executed entirely in real time and the whole process was very costly. Many organizations did not possess all of the hardware resources needed for the development of VoIP.

Due to this extensive resource consummation the cost of the VoIP system also was very high and only big corporate companies could afford it. At that time the cost of transmitting voice over the traditional PSTN (Public Switched Telephone Network) worked out to be cheaper than transmitting through a VoIP system. Moreover there were many government restrictions regarding the transmission of voice and also on the encryption of the data being transmitted.

1.4 Proposed System

The proposed system eliminates the requirements of high end routers and gateways for testing the working of the VoIP system. The simulation environment serves as an excellent platform for SIP based VoIP system developers. All the configurations and call setup can be simulated with the help of the Load generator. The system can be simulated in a small LAN or even in a stand alone computer using either RedHat Linux or Sun Solaris. In a stand alone Linux system the simulation is carried out in any of the seven terminals available. This greatly reduces the time taken to develop a VoIP system. The most important benefit of simulation is bringing down the cost factor to a considerable extent.

VoIP implementation can then be carried out in real time after all the simulations are completed and the system is found to be perfect. This makes the whole process less complicated and a time saving one. Simulations are also useful in detecting and correcting any errors present in the system.

2. System Requirements

2. System requirements

2.1 Hardware and Software requirements

Hardware requirements:

The hardware used should be Intel i86 based IBM compatibles.

The following are the recommended hardware configuration.

480 MHz, Pentium II Intel based PC.

128 MB RAM.

1 GB Hard disk space.

10/100 Mbps NIC

Software requirements:

The following are the software requirements for the system,

- Red Hat Linux Version 6.2 or later.
- Apache Server.
- JDK 1.2.
- Java 2 Runtime Environment Plug-in (on client workstation only).
- Netscape Navigator browser 4.6 or higher (on client workstation only).
- Languages & libraries:
  - C++
  - Call processing language (CPL)
  - Java telephony API (JTAPI)

2.2 About Linux

Linux was developed in the year 1991 by Linus Torvalds. It was developed with two main considerations in mind, networking and security. It has all the speed, efficiency and flexibility of Unix, from which it has been developed. The Linux kernel has developed into the strongest one with regard to security.

The three main shells in Linux are BASH (born again shell) , CSH(C shell) and KSH(Korn shell). In Linux everything is considered as a file or directory. The kernel is a file, hard drives are files, CDROM and floppies are files. Therefore there is no need for drive letters. All file systems should be mounted before they can be accessed. All the softwares are in the form of RPMs (RedHat Package Management) which can be installed and uninstalled easily.

Features of Linux:

- **Full multitasking and 32-bit support**: Linux is a real multitasking system, allowing multiple users to run many programs on the same system at once. Linux is also a full 32-bit operating system utilizing the special protected-mode features of Intel 80386 and later processors.
- **The X Window System:** The X Window System is the de facto industry-standard graphics system for UNIX machines. The X Window System is a very powerful graphics interface, supporting many applications.

- **Virtual memory and shared libraries:** Linux can use a portion of your hard drive as virtual memory, expanding your total amount of available RAM. Linux implements shared libraries, allowing programs using standard subroutines to find the code for these subroutines in the libraries at runtime.
- **GNU software support:** Linux supports a wide range of free software written by the GNU Project, including utilities such as the GNU C and C++ compiler, gawk, groff, and so on.
- Linux is compatible with the **IEEE POSIX.1 standard**. Linux has been developed with software portability in mind, thus supporting many important features of other UNIX standards.
- Linux has built-in support for **networking and multitasking** features. In fact, Unix has implemented this "new technology" for nearly 15 years.
- Linux is **cheaper** to get than most commercially available UNIX systems because it comes under the Open Source License.

Linux distributions:

The following are the commonly available distributions of Linux;

- RedHat Linux - www.redhat.com
- Mandrake Linux - www.mandrakesoft.com
- SuSe Linux - www.suse.com
- Caldera Linux - www.calderasystems.com
- Slackware Linux - www.slackware.com
- Debian Linux - www.debian.org

3. System Environment

3. System environment

3.1 VoIP Introduction

Availability of a telephone and access to a low-cost, high-quality worldwide network is considered to be essential in modern society. Now-a-days more and more communications are in digital form and transported via packet networks such as IP, ATM cells, and Frame Relay frames. Since data traffic is growing much faster than telephone traffic, there has been considerable interest in transporting voice over data networks rather than the traditional data over voice networks. Users are seeking new types of integrated voice/data applications as well as cost benefits.

Traditional circuit-switched telephone networks were designed to carry voice traffic. However, because circuit-switched networks reserve an entire channel for each conversation (even when no-one's talking), they do not use the network's available bandwidth in the most efficient way.

VoIP can be defined as "**the ability to make telephone calls and to send facsimiles over IP-based data networks with a suitable quality of service (QoS) and a much superior cost/benefit**". Equipment producers foresee VoIP as a new opportunity to innovate and compete in today's market.

Benefits:

VoIP is particularly useful for enterprises who already operate an IP network, where huge savings can be made by using the corporate IP network to carry voice as well as data traffic. It can also be used by residential users, as it is very cheap for contacting friends and family abroad for example, where phone rates are much higher than an internet connection Since broadband services have improved and Internet bandwidth has become more reliable, so carrying voice over Internet has become smoother and of a higher standard than ever before. The benefits include

- It can lower capital and operating costs by converging separate voice and data networks into a single multi service network.
- It can increase revenues for Service Providers by raising the value of voice service, with new applications such as video calling, unified messaging and Web-enabled multimedia call centers.
- It can enhance productivity with new applications such as unified messaging which can enable employees at any location to work more effectively

It is estimated that new IP telephony and VOIP consulting and integration services can help organizations save up to 30% on network costs via lower management, bandwidth and application integration costs.

3.2 VoIP Architecture

The following is the pictorial representation of the entire VoIP architecture;

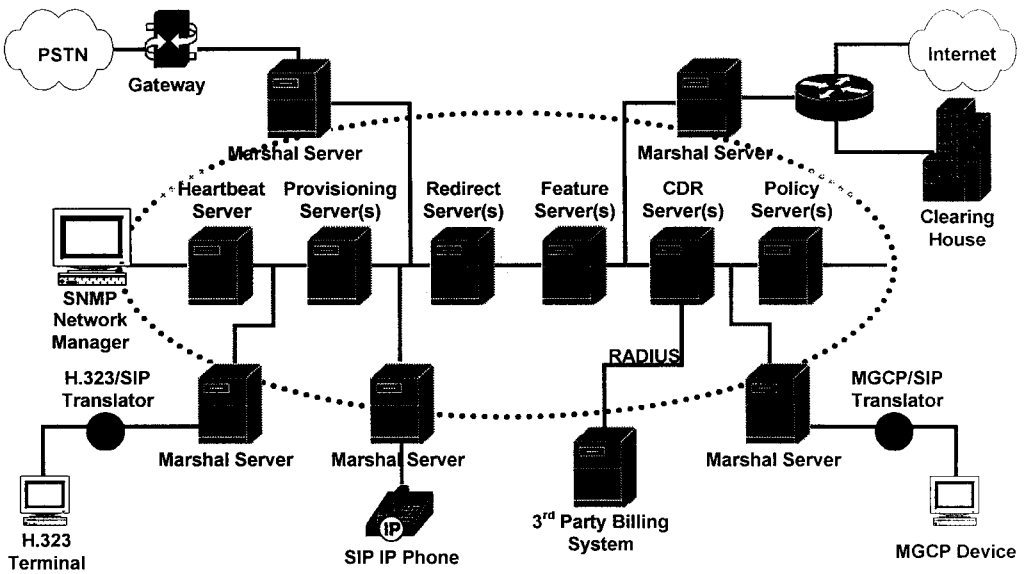


Figure: VoIP architecture

The above diagram represents the various components in the VoIP system. There are about seven servers and a couple of translators in the system. The translators are used to provide connectivity with other platforms. There is also a network manager to control and review the entire system. The system also represents the various end terminals like H.323 terminal, SIP IP phone, MGCP device being connected by a translator to the Marshall server.

3.3 VoIP Components

A VoIP network consists of following components;

1. User agent
2. Marshal server
3. Heartbeat server
4. Provisioning server
5. Redirect server
6. Feature server
7. CDR server
8. Policy server

1. User Agent:

An application that initiates, receives and terminates calls. It includes

- ❋ User Agent Clients (UAC) - An entity that initiates a call.
- ❋ User Agent Server (UAS) - An entity that receives a call.
- ❋ Both UAC and UAS can terminate a call.

A User agent supports

- ❋ call establishment
- ❋ call waiting
- ❋ call transfer
- ❋ registration with a SIP proxy server or marshal server

2. **Marshal Server:**

- They are the only points of contact for all external devices.
- Provides the logical function of the SIP proxy server and the SIP registration server.
- Performs one or more of these functions:
  - SIP message translation.
  - Authentication and security

3. **Heartbeat Server:**

The Heartbeat Server,

- Monitors the exchange of heartbeat packets from VoIP servers.
- Sends server status information to the SNMP Network Manager.

4. **Provisioning Server:**

The provisioning server

- stores data on all users and servers within the VOCAL system
- accessible from Java-based GUI via an internet browser

5. **Redirect Server:**

The Redirect Server provides these SIP services and functions:

- Registration.
- Redirection.
- Location.

The Redirect Server provides routing information to the Feature and Marshal Servers to route a call.

6. Feature Server:

Features provided to the customers are stored and managed in this server. Some of the available features are;

- ❋ Call Forward
- ❋ Call Blocking
- ❋ Caller ID Blocking
- ❋ Call Waiting
- ❋ Conferencing

7. Call Detail Record Server:

The CDR server performs the following functions

- ❋ Receives start and end times from Marshal Servers.
- ❋ Both marshal servers send start and end time to the CDR server
- ❋ Formats data into CDR data for each call.
- ❋ Forwards CDR data to 3rd party billing system using RADIUS.

8. Policy Server:

- ❋ Administers admission request for bandwidth or quality of service.
- ❋ Interacts with internetwork Marshal Servers that enforce QoS.
- ❋ Interfaces with a clearinghouse to authorize the use of a network for internetworking calls.
- ❋ Function of the Policy server as a COPS server

3.4 VoIP Protocols

There are two main protocols using which the VoIP system can be implemented. They are as follows

- SIP – Session Initiation Protocol
- The H.323 standard

Another protocol that controls the gateway mechanism is,

- MGCP – Media Gateway Control Protocol

1. Session Initiation Protocol (SIP):

SIP is an application layer signaling protocol that defines initiation, modification and termination of interactive, multimedia communication sessions between users. SIP represents a signaling protocol for establishing calls via a TCP/IP network. It offers a new degree of scalability, flexibility and integration compared to the H.323 standard.

SIP comes under the Open source policy and therefore the development communities are able to build a variety of applications based on it. SIP is an IETF (Internet Engineering Task Force) application layer protocol defined in RFC 2543. It was designed independent of lower layer protocols and therefore can use TCP or UDP.

Example of SIP URL:

sip:ranka@192.168.10.2

sip:geekz@geks.com

SIP Methods:

- **INVITE** – Initiates a call by inviting user to participate in session.
- **ACK** - Confirms that the client has received a final response to an INVITE request.
- **BYE** - Indicates termination of the call.
- **CANCEL** - Cancels a pending request.
- **REGISTER** – Registers the user agent.
- **OPTIONS** – Used to query the capabilities of a server.
- **INFO** – Used to carry out-of-bound information, such as DTMF.

SIP Responses:

- **1xx** - Informational Messages.
- **2xx** - Successful Responses.
- **3xx** - Redirection Responses.
- **4xx** - Request Failure Responses.
- **5xx** - Server Failure Responses.
- **6xx** - Global Failures Responses.

Response code examples:

100 – Continue

180 – Ringing

200 – OK

401 – Unauthorized

403 – Forbidden

408 – Request time-out

2. The H.323 standard:

The H.323 standard describes terminals and other entities that provide multimedia communications services over Packet Based Networks (PBN) which may not provide a guaranteed Quality of Service. H.323 entities may provide real-time audio, video and/or data communications. This is the ITU's standard that vendors should comply while providing VoIP services. This standard was originally developed for use on a single LAN. The concept of zones was introduced in the next version of H.323 that enabled its scalability.

The H.323 standard defines four major components;

- Terminals – represents a LAN endpoint
- Gateways – provides a translation service
- Gatekeepers – controls access to the network
- Multipoint control units – supports conferences between three or more endpoints by functioning as a bridge.

The audio, video and registration packets use the unreliable UDP while the data and control application packets use the reliable as the transport protocol. H.323 also defines several protocol exchanges between terminals, gateways and gatekeepers that must occur prior to establishing an audio connection between two terminals.

Media Gateway Control Protocol(MGCP):

The Media Gateway Control Protocol (MGCP) is a protocol for controlling telephony gateways from external call control elements called media gateway controllers or call agents. MGCP is defined in RFC 2705. It is transported via UDP and represents an application residing at layer 5 in the TCP/IP protocol stack. MGCP is a sort of master/slave protocol used between call agents and media gateways. Although it differs from SIP and H.323, which are peer-to-peer protocols, MGCP fully interoperates with both of them.

MGCP performs its operations by issuing a sequence of ASCII commands to the endpoints. Each command consists of a verb that defines an action to be performed by the selected endpoint. Some of the MGCP commands are as follows;

- NotificationRequest
- CreateConnection
- ModifyConnection
- DeleteConnection
- Notify
- DeleteConnection
- RestartInProgress

Each of these commands has a header which contains a session description that sets up and endpoint to both recognize and generate an applicable media format.

Comparison between SIP and H.323:

The following table gives the differences between the H.323 standard and SIP.

H.323	SIP
1. H.323 was designed for multimedia communication over IP networks	1. SIP was designed basically to setup a session between two endpoints.
2. H.323 has defined a number of features for error handling	2. SIP has no defined procedures for handling device failures.
3. It has only a limited extensibility	3. It can be extended to a great extent.
4. It has a flexible addressing mechanism.	4. SIP only understands URL based addresses.
5. H.323 is very complex in nature	5. SIP is simple compared to H.323
6. H.323 requires a multicast controller for conferencing	6. SIP has built-in support for conference calls.
7. H.323 is limited to TCP only.	7. SIP can operate under X.25 also.
8. Clients are Intelligent H.323 terminals	8. Clients are Intelligent User Agents
9. H.323 is governed by ITU	9. SIP is governed by IETF
10. H.323 messages are encoded in a compact binary format.	10. SIP messages are encoded in ASCII text format.

3.5 Load generator

The Load generator (LG) is a very simple proxy that is useful in SIP-based VoIP installations where there are multiple ingress proxy servers. The load generator should start sending and also receive media streams. The following are the main functions of the LG application

- Support for Sending SDP message in Invite to receiver
- Support for Sending RTSP message to the RTSP server
- Support for listening media streams on various ports from the RTSP server.
- Support for reflecting media messages from RTSP server to the LG sender.

Currently the program can support only signaling via SIP. The program is capable of generating 10 calls / second which can be customized according to the needs of the user.

LG supports three running modes: -

- Originating
- Registration
- Terminating

The LG and the RTSP server can be configured on the same machine or can reside in different machines.

4. Process Modeling ✵

4. Process modeling

4.1 Context diagram:

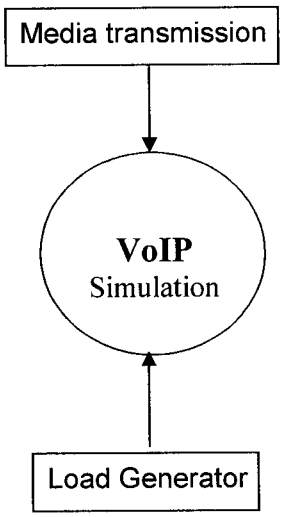


Figure: Context diagram for the VoIP simulation system

From the above diagram it is evident that there are two main modules for the simulation of the VoIP system, namely,

- Media transmission
  - Used to send and receive the media streams between any two clients.
- Load generator
  - Used in a LAN environment to generate artificial calls to provide a simulation.

4.2 Level 1 – Media transmission:

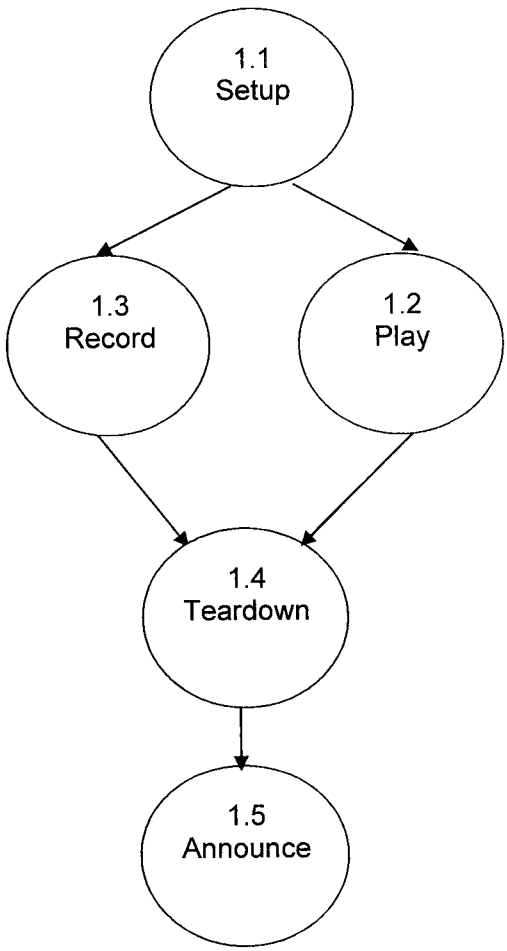


Figure: Level -1 diagram for the media transmission

The above diagram represents the five different modules that make up the media transmission. The SETUP is for either PLAY or RECORD and both cannot be invoked simultaneously at the same terminal.

4.3 Level 1 – Load generator:

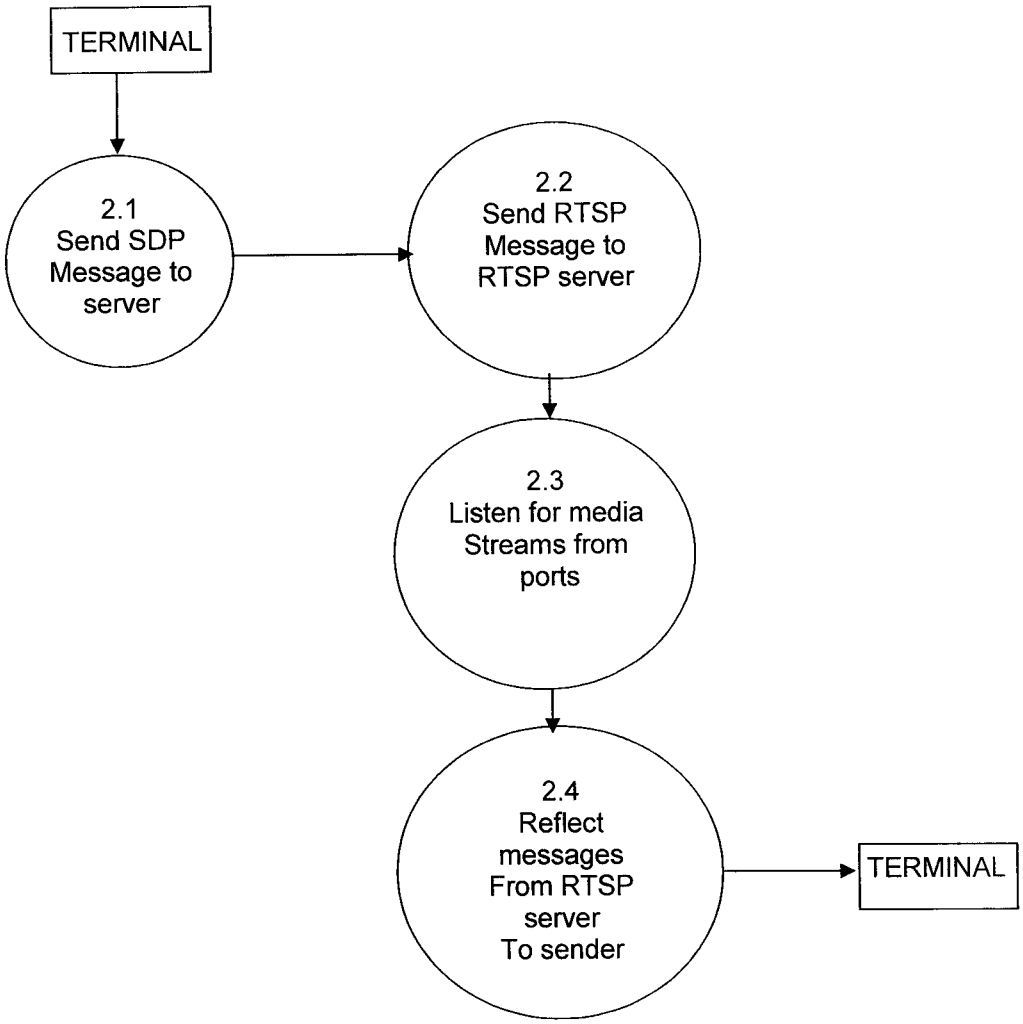


Figure: Level -1 diagram for the Load generator

The Session Description Protocol (SDP) and the Real Time Service Protocol (RTSP) provide the medium of connectivity between the two terminals in a network.

4.4 Level 1.1 – Setup:

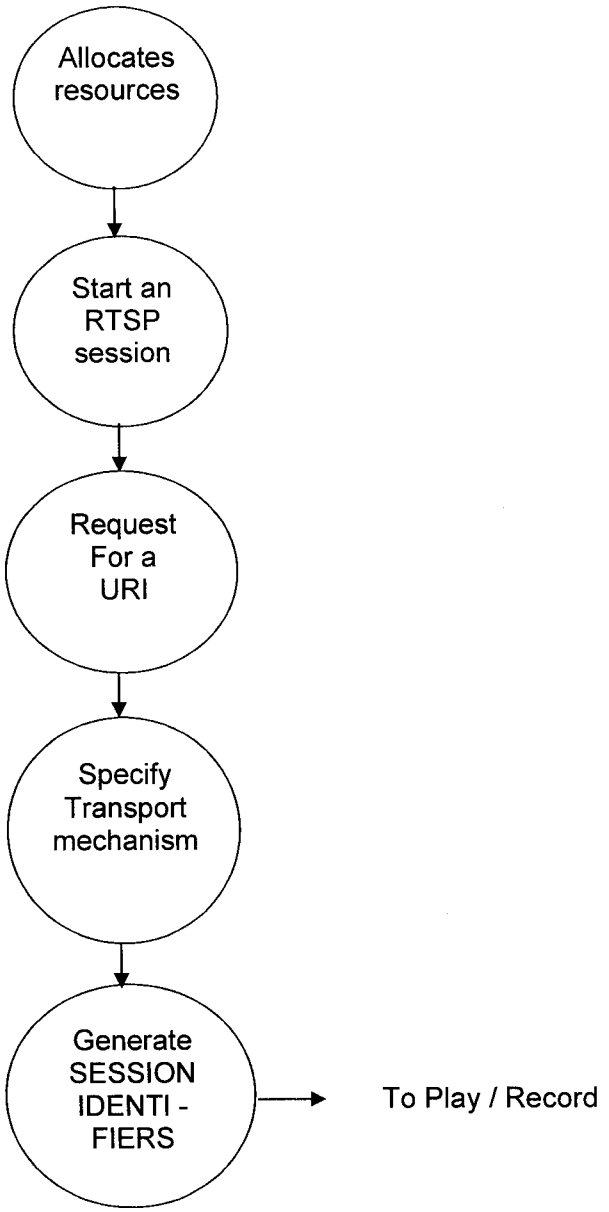


Figure: Level -1.1 diagram for the SETUP module

The transport parameters of both the PLAY and RECORD modules are configured in this module.

4.5 Level 1.2 – Play:

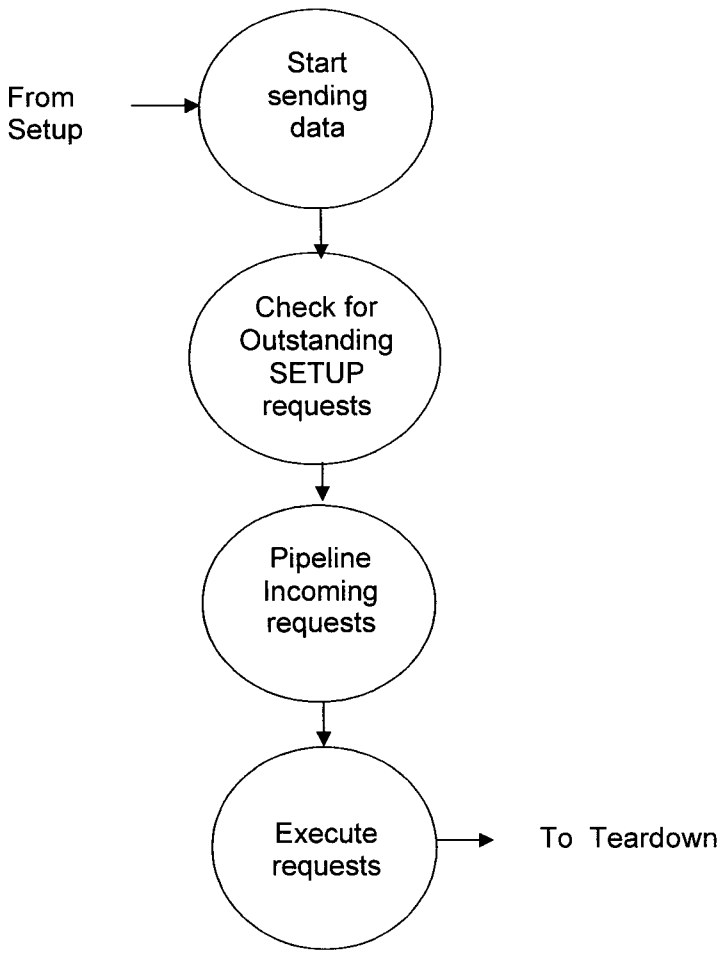


Figure: Level -1.2 diagram for the PLAY module

Based on the specifications done in the SETUP module, the parameters of the PLAY module are initialized to play the specific media stream. After completion the control is passed on to the TEARDOWN module.

4.6 Level 1.3 – Record:

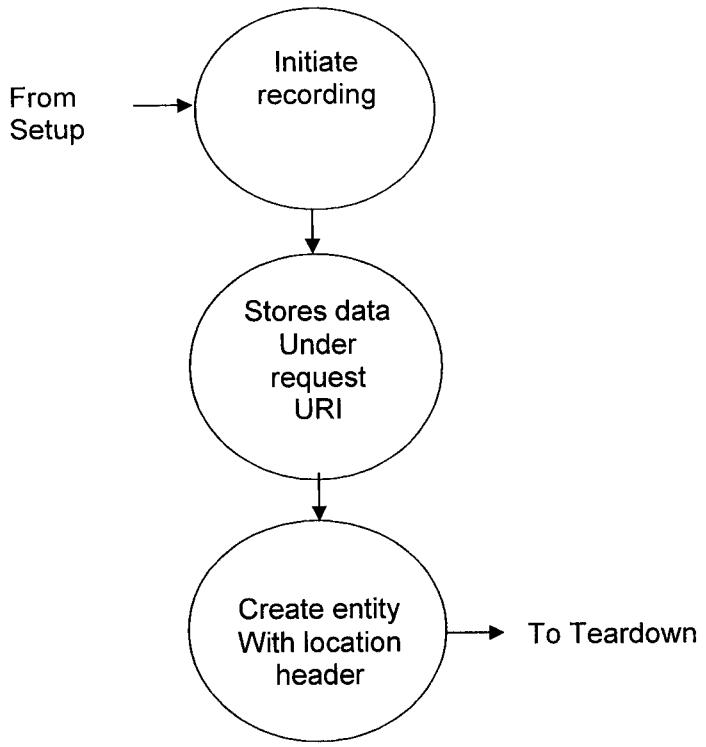


Figure: Level -1.3 diagram for the RECORD module

All the parameters and mode of communication for the record process are configured in the SETUP module. It also specifies where to store the recorded stream. After successful completion of the record process, the control is passed on to the TEARDOWN module.

4.7 Level 1.4 – Teardown:

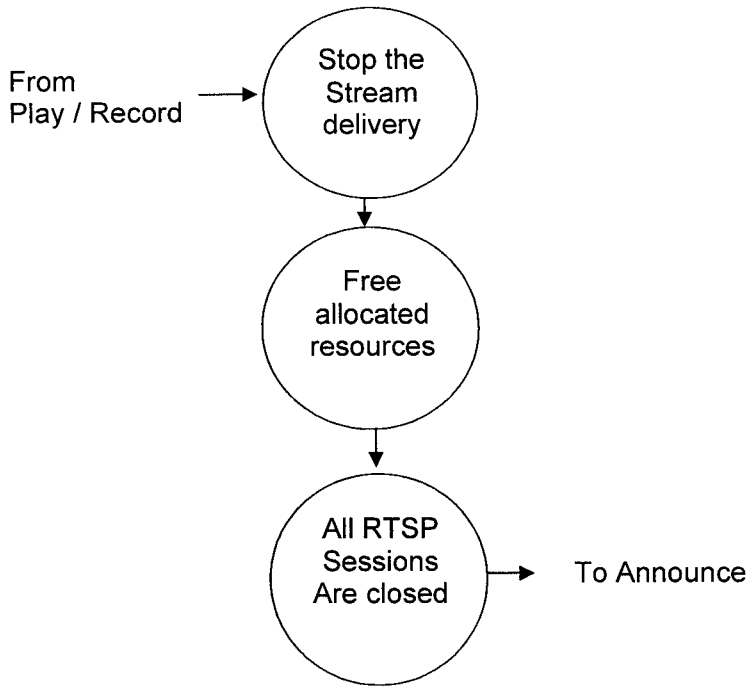


Figure: Level -1.4 diagram for the TEARDOWN module

The input to this module comes from both the PLAY and the RECORD modules. When this module is invoked it stops the delivery of the media streams. All the resources are unallocated and the RTSP sessions are closed thus signaling the session termination.

4.8 Level 1.5 – Announce:

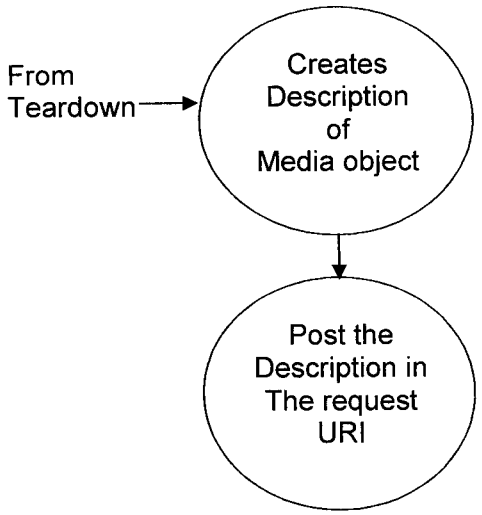


Figure: Level -1.5 diagram for the ANNOUNCE module

The ANNOUNCE module is used to post a message containing the description of a media object identified by the request URL to a server. The description consists of the path of the file, type of media, host IP address, content length, content type,...etc.

5. Function Modules ✵

5. Function modules

5.1 Setup:

The SETUP module is used to initialize the simulation system by specifying the configuration parameters to the other modules. This should be done to ensure a standard in sending and receiving media streams. The following are the responsibilities of this module;

- Causes the server to allocate resources for a stream
- Start an RTSP session
- Request for a URI
- Specify the transport mechanism to be used for the streamed media
- The server generates SESSION IDENTIFIERS in response to SETUP requests

For the benefit of any intervening firewalls, a client must indicate the transport parameters even if it has no influence over these parameters. The transport header specifies the transport parameters acceptable to the client for data transmission. The response will contain the transport parameter selected by the server.

5.2 Play:

This main function of this module is to play the selected file according to the parameters specified in the SETUP module. This message tells the server to start sending data which will be received at the other end by the terminal running the RECORD module.

There is one important condition to be checked before a file can be played. A client must not issue a PLAY request until any outstanding SETUP requests have been acknowledged as successful. This is because the transmitting parameters should be accepted at both the terminals before the actual transmission starts.

The PLAY requests may be pipelined if there many requests arrive simultaneously. A server must queue all the incoming PLAY requests and they should be executed in order. That is, a PLAY request arriving while a previous PLAY request is still active is delayed until the first has been completed. A new PLAY session ID is created for all the requests that contain all the details about the file.

After the PLAY request is over, a status code is returned to the server. If the response is 200 OK then the PLAY request was successful, else an error log is created and the request becomes a failure.

5.3 Record:

This main function of this module is to record the file being played by the PLAY module. The configuration parameters of the RECORD module are defined in the SETUP module.

This method initiates recording a range of media data according to the presentation description in the SETUP request. The information in the configuration consists of the name of the record file, sequence counter, client port. Additionally the location where the file is to be stored is also specified in the SETUP module.

The server decides whether to store the recorded data under the request-uri or another uri. If the server does not use the request-uri, the response should be 201 (created) and contain an entity which describes the status of the request and refers to the new resource and a location header.

After the RECORD request is over, a status code is returned to the server. If the response is 200 OK then the RECORD request was successful, else an error log is created and the request becomes a failure.

5.4 Teardown:

The TEARDOWN module can be invoked from either the PLAY or the RECORD module. It signifies the end of the transmission by closing the session and socket connections. The following are the functions of this module;

- It stops the stream delivery for the given URI.
- Releases the resources associated with it.
- It signals the termination of the PLAY_SESSION_ID and the RECORD_SESSION_ID.

5.5 Announce:

The ANNOUNCE module is invoked by the TEARDOWN module to post a description of the media object identified by the request uri to the server. The description consists of the following details,

- path of the file
- type of media
- host IP address
- content length
- content type

6. System Implementation

6. Implementation

6.1 Installation and verification

Before beginning the installation of the VoIP system the following tests have to be performed,

- ❋ Check loopback and multicast addresses
- ❋ Check for proper LAN connectivity
- ❋ Verify the Domain Name System (DNS)
- ❋ Verify the host name

All the above tests should be done prior to installation. There should not be any error else the VoIP system will not get installed properly. Installation can be carried out in the following two ways,

1. Installing from the RPM , and
2. Installing from the source

❋ **Installing from the RPM**

Installing the VoIP system from the RPM package is a very simple and easy process. RPM stands for RedHat Package Management which is the standard format in which all the RedHat packages are available.

 # rpm –U <name of the rpm>

This installs the complete VoIP system into the */usr/local/vocal* directory.

Only the root user can perform this task.

❋ **Installing from the source**

To install from source perform the following operations as root after extracting the contents of the tarball.

 # ./configure

 # make

 # make install

After the script executes, many questions will be asked regarding the configuration. If everything goes well then we get a success message.

Verification

The installation of the VoIP system should be verified and checked for its integrity. This is to ensure that all the packages are installed correctly. The installation can be verified by the following command,

 # /usr/local/vocal/bin/allinoneconfigure/verifyinstall

This will return any errors if there are any else the status of the installation process. This command should be run as root only.

6.2 Configuration

The VoIP system should be configured before being able to make or receive any calls. The following command should be typed as root,

 # /usr/ local/vocal/bin/allinoneconfigure/ allinoneconfigure

The configuration script generates a few questions with a default option. We can configure the host and destination addresses to make the VoIP system work on a LAN environment. The directories where the HTML files and the CGI script can be specified. After the configuration is completed a success message is returned.

The Apache web server daemon should be restarted. This can be done by typing the following command,

 # service httpd restart

After the service is started the web-based provisioning can by invoked by pointing a web browser to the following location,

http://<hostname>/vocal/

The features available in the web-based provisioning system are

- Add user
- Delete user
- Change user
- Change gateway
- System status

6.3 Basic call setup

The SIP UA (User Agent) should be executed simultaneously in any two of the terminals (xterm) on the machine where the VoIP system is installed. The UA will be in the */usr/local/vocal/bin/* directory. There are two users that are configured by default. They are 1000 and 1001, each with separate configurations.

The following sequence of steps should be followed in order to initialize call setup,

From */usr/local/vocal/bin*, run the following;

./ua -r -f /usr/local/vocal/etc/ua1000.cfg in one xterm and run

./ua -r -f /usr/local/vocal/etc/ua1001.cfg in another xterm.

A message displaying "Registration OK: Ready" indicates a successful setup and that the system is ready to make or receive calls.

Dialling:

The following steps should be followed to dial from one xterm to the other,

Press 'a' for offhook on each UA terminal.

To call one terminal from the other, type:

a 1 0 0 0 # or a 1 0 0 1 #

Press 'z' to hangup and log out of the system by typing "logout"

6.4 Load generator setup

Load generator is used to generate simulated calls and can be configured according to the LAN configuration. The following are the steps necessary to configure the load generator;

Calling load generator:

1. Edit the ua.cfg file for the calling load generator.
2. Set Proxy_Server field to the <ip:port> of the receiving side.
3. Set LoadGen_On to True.
4. Set CallUrl to sip:<id>@<host of receiving>
5. Set MonitorMsgOn to True.
6. Set RtpGenOn to False.

Receiving load generator:

1. Edit the ua.cfg file for the receiving load generator.
2. Set Proxy_Server field to the <ip:port> of the calling side.
3. Set LoadGen_On to True.
4. Set RunMode to Receiving.
5. Set MonitorMsgOn to True.
6. Set RtpGenOn to False.

To run the load generator:

1. Run the receiving load generator first

   ./ua -f ua.cfg

2. Next run the calling load generator.

   ./ua -f ua.cfg

7. Future Enhancements ✵

7. Future enhancements

The scope of VoIP in Linux is increasing in leaps and bounds everyday. Newer enhancements are developed at an unimaginable rate since its source is open to all. In due course simulations can be performed even by a newbie because all developments are focused mainly on user friendliness. Some of the areas that will be enhanced in future are,

- Performance improvements (calls per second per box)
- User friendly GUI simulation
- Drag and drop features
- Digest authentication fixes

The extension of VoIP technology to mobiles is one of the many areas where developments are done at a rapid pace. The advent of wireless technologies like Bluetooth and WAP increases the scope of the VoIP systems.

8. Conclusion

8. Conclusion

Simulation for VoIP using SIP in Linux was developed keeping in mind the cost and other resources involved in testing the system. With this simulation the entire VoIP functionality can be demonstrated without the need for expensive and high end components. This helps in bringing the whole system within a single stand alone computer. The system is demonstrated by simultaneously running the simulation in any two terminals of a single Linux client.

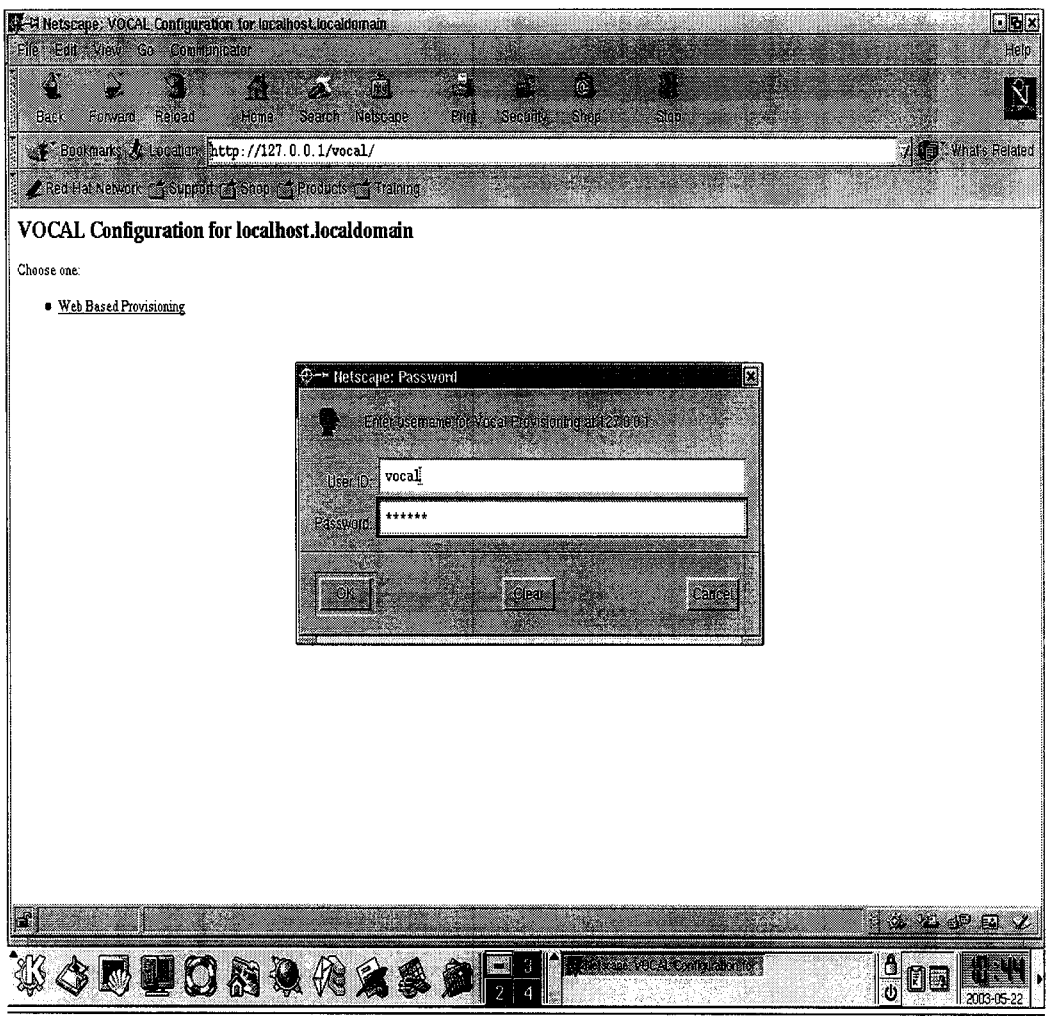
This simulation serves as a great boon to VoIP developers since it serves as a platform wherein they can have a hands on experience as if they are working in real time. It also helps a great deal in bringing down the cost of the VoIP systems. The system can be simulated in a LAN environment with the help of a Load generator.

9. APPENDIX

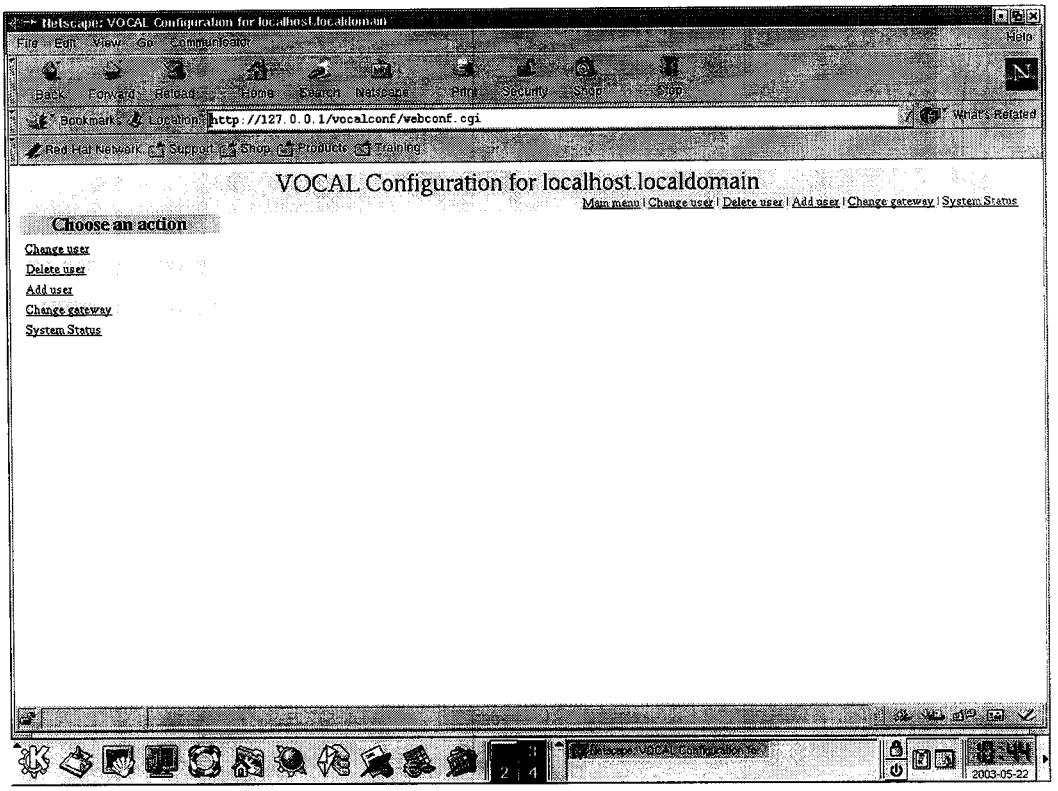
Appendix

Screenshots

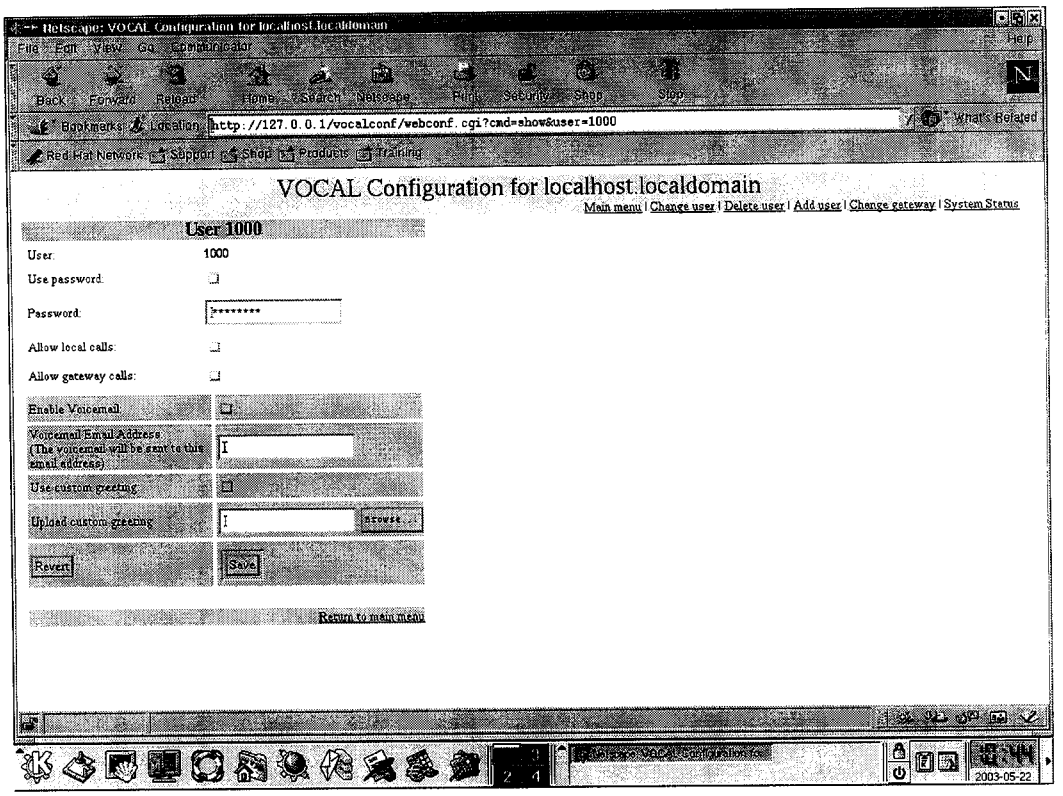
Login screen



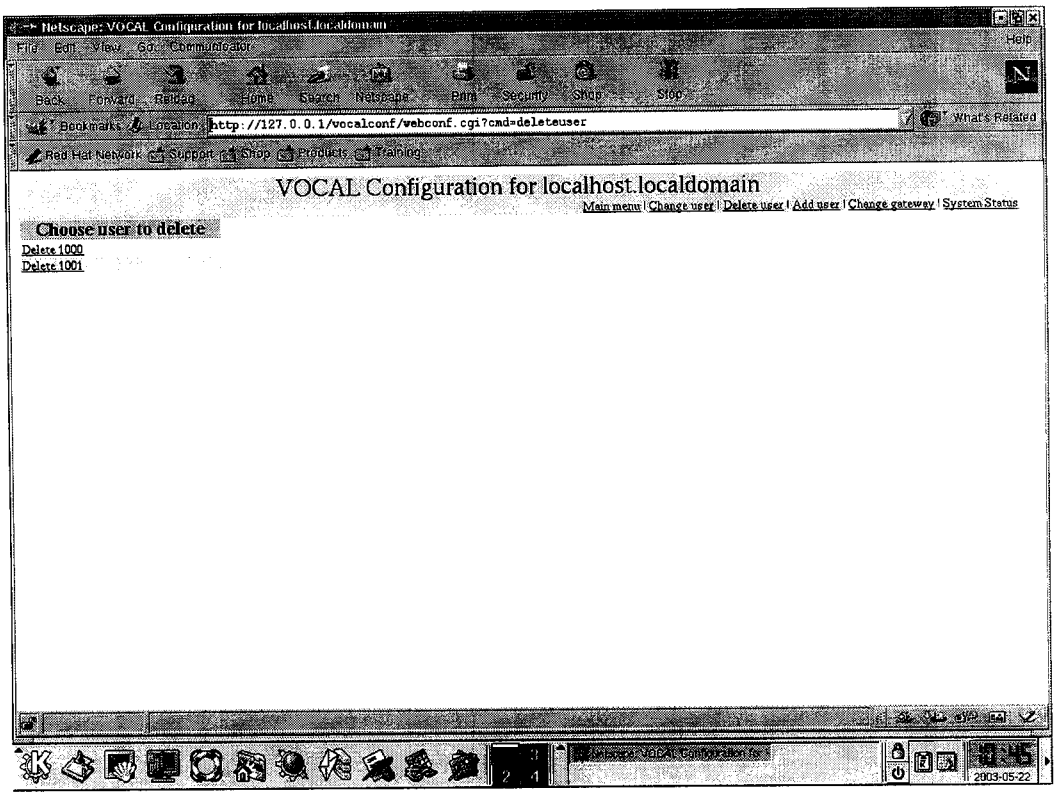
Main menu


Netscape: VOCAL Configuration for localhost.localdomain
File Edit View Go Communicator
Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://127.0.0.1/vocalconf/webconf.cgi
What's Related
Red Hat Network Support Shop Products Training
VOCAL Configuration for localhost.localdomain
Main menu | Change user | Delete user | Add user | Change gateway | System Status
Choose an action
Change user
Delete user
Add user
Change gateway
System Status
2003-05-22

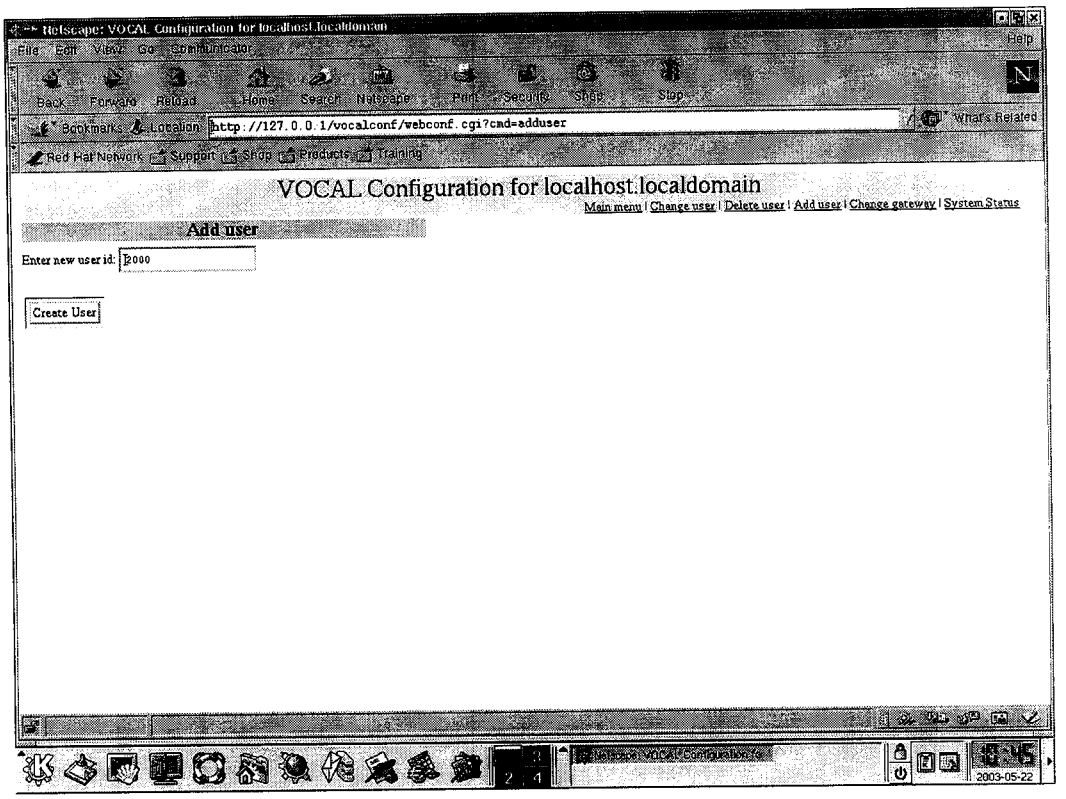
Change user


Netscape: VOCAL Configuration for localhost.localdomain
http://127.0.0.1/vocalconf/webconf.cgi?cmd=show&user=1000
VOCAL Configuration for localhost.localdomain
Main menu | Change user | Delete user | Add user | Change gateway | System Status
User 1000
User:
1000
Use password:
Password:
Allow local calls:
Allow gateway calls:
Enable Voicemail:
Voicemail Email Address (The voicemail will be sent to this email address):
Use custom greeting:
Upload custom greeting:
Browse...
Revert
Save
Return to main menu

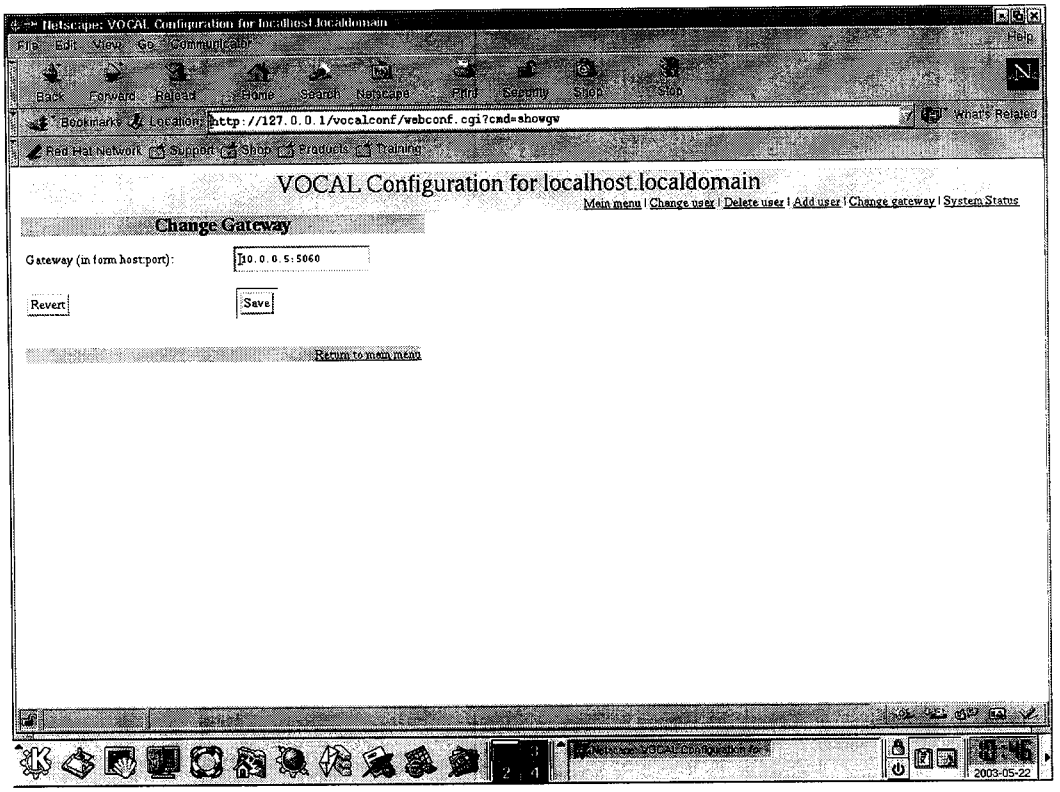
Delete user:


Netscape: VOCAL Configuration for localhost.localdomain
http://127.0.0.1/vocalconf/webconf.cgi?cmd=deleteuser
VOCAL Configuration for localhost.localdomain
Main menu | Change user | Delete user | Add user | Change gateway | System Status
Choose user to delete
Delete 1000
Delete 1001

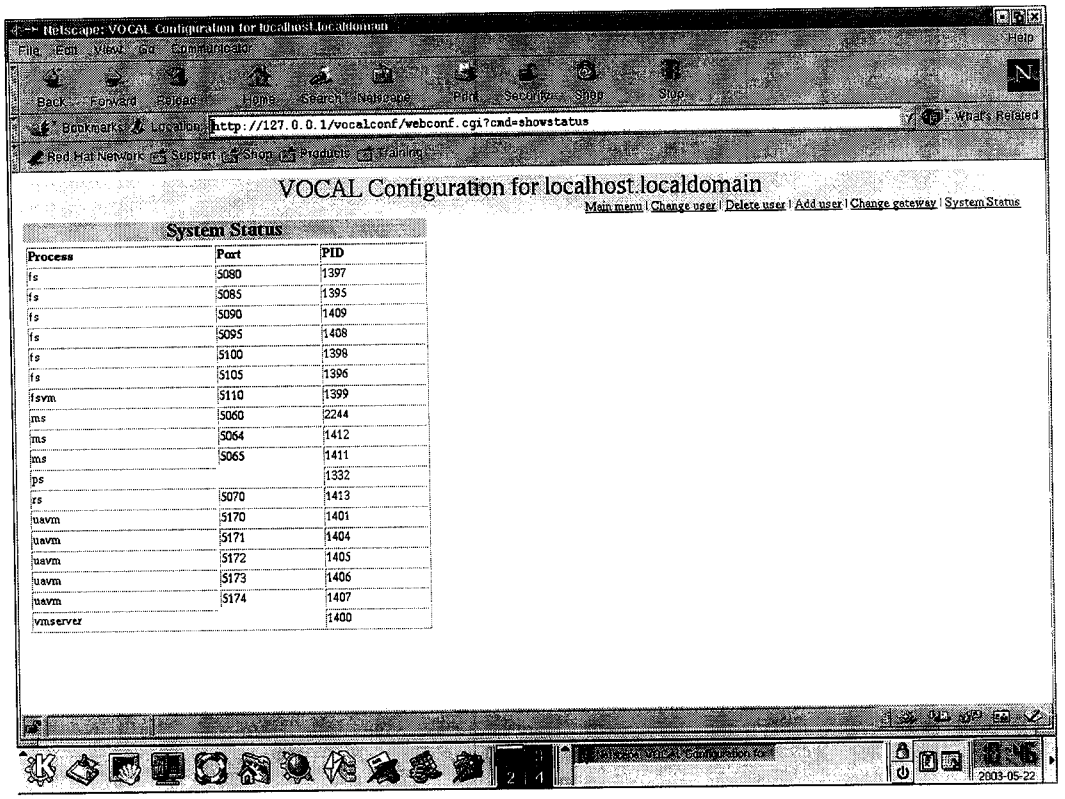
Add user:


Netscape: VOCAL Configuration for localhost.localdomain
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://127.0.0.1/vocalconf/webconf.cgi?cmd=adduser
What's Related
Red Hat Network Support Shop Products Training
VOCAL Configuration for localhost.localdomain
Main menu | Change user | Delete user | Add user | Change gateway | System Status
Add user
Enter new user id: 2000
Create User
2003-05-22

Change gateway



System status



10. REFERENCES ✵

10. References

Books:

1. Douglas.E.Comer, "*Computer Networks and Internets*" second edition.
2. Gil Held, "*Voice and Data Internetworking*" third edition.
3. Keith.W.Ross, James.F.Kurose, "*Computer Networking*"

RFC:

rfc 2327 – SDP (Session Description Protocol) April 1998

rfc 2543 – SIP (Session Initiation Protocol) March 2001

rfc 2705 – MGCP (Media Gateway Control Protocol) March 2001

Links:

www.protocols.com

www.linuxtel.com

www.sipcenter.com

www.networkcomputing.com

www.itpapers.com

www.cisco.com

www.iptelephony.org

11. GLOSSARY

11. Glossary

AAA – Authentication, Authorization and Accounting

ARP – Address Resolution Protocol

CHAP – Challenge Handshake Authentication Protocol

COPS – Common Open Policy Service

DNS – Domain Name Service

DTMF – Dual Tone Multi-Frequency

GSM – Global System for Mobile communications

IETF – Internet Engineering Task Force

ISP – Internet Service Provider

ITU – International Telecommunications Union

MGCP – Media Gateway Control Protocol

NIC – Network Interface Card

OSPF – Open Shortest Path First

POTS – Plain Old Telephone System

PSTN – Public Switched Telephone Network

QoS – Quality of Service

RTCP – Real Time Control Protocol

RTP – Real-time Transport Protocol

SDP – Session Description Protocol

SIP – Session Initiation Protocol

SNMP – Simple Network Management Protocol

TCP /IP – Transmission Control Protocol / Internet Protocol

UDP – User Datagram Protocol

UAS – User Agent Server

UAC – User Agent Client

URI – Uniform Resource Identifier

VLAN – Virtual Local Area Network