KUMARAGURU COLLEGE OF TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Coimbatore – 641006

April 2003.

PC TO PC VOICE COMMUNICATION

PROJECT WORK DONE AT

D'GIPRO DESIGN AUTOMATION & MARKETING PVT.LTD

BANGALORE.

PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the Degree of **Master of Computer Applications** of Bharathiar University, Coimbatore.

SUBMITTED BY

Mr. RAGHU PRASAD.K.R
REG. NO.: 0038M1053

GUIDED BY

Internal Guide
Mrs. S. DEVAKI, B.E., MS. Asst Professor,
Department of Computer Science and Engineering

External Guide
Mr. K.Suresh Kumar, B.E,
D'Gipro Design Automation & Marketing Pvt.Ltd.

KUMARAGURU COLLEGE OF TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Coimbatore – 641006

April 2003.

CERTIFICATE

This is to certify that the project work entitled

"PC TO PC VOICE COMMUNICATION"

Done By

Raghu Prasad. K.R
Reg. No. : 0038M1053

Submitted in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications of Bharathiar University.

Professor and HOD

Internal Guide

Submitted to University Examination held on 16-04-2003

16/4
Internal Examiner

16/04/03
External Examiner

D'GIPRO
DESIGN AUTOMATION &
MARKETING PVT. LTD

Bangalore, Dated 3rd April 2003

To

Head of the Department.,
Kumaraguru College of Technology.,
Coimbatore - 641 006

Sub : Project Work

Dear Sir,

This is to certify that Mr.Raghu Prasad K.R. (0038M1053) final year M.C.A. student of Kumaraguru College of Technology, Coimbatore, has done the project entitled "PC TO PC VOICE COMMUNICATION (VoIP)" successfully in our organization for a period of four months starting from 26th Dec 2002 to 28th March'2003.

Enclosed is his Attendance and completion certificate.

We wish him all success throughout his career.

Thanking you

Your faithfully,
For D'GIPRO DESIGN AUTOMATION &
MARKETING Pvt. Ltd.,

....
K.SURESH KUMAR
DIRECTOR

DECLARATION

I here by declare that the project entitled "*PC TO PC VOICE COMMUNICATION*", submitted to Bharathiar University as the project work of Master of Computer Applications Degree, is a record of original work done by me under the supervision and guidance of Mr.K. Suresh Kumar, B.E Software Engineer, D'GIPRO Automation & marketing Pvt Ltd, Bangalore and Mrs.S.Devaki, BE.,M.S, Assistant Professor, Department of Computer Science and Engineering, Kumaraguru College of Technology. This project work has not found the basis for the award of any Degree / Diploma / Associate ship / Fellowship or similar title to any candidate of any University.

Name of Candidate	**Register Number**	**Signature**
K.R.RAGHU PRASAD	**0038M1053**	Ragu Prasad

Place: **COIMBATORE**

Date: 08-04-2003

ACKNOWLEDGEMENT

The endeavor over a long period can be successful by the advice and support of many well wishers. We avail this opportunity to express our gratitude and appreciation of all of them.

We would like to express our profound respect to our beloved Principal Dr.K.K.Padmanabhan, **B.Sc(Engg), M.Tech.,Ph.D.**, for having provided the necessary facilities to complete this project.

We are greatly indebted to our beloved Head of the Department, **Dr.S.Thangasamy,B.E(Hons),Ph.D.,** Computer Science and Engineering, Kumaraguru College of Technology, for our source of inspiration and encouragement rendered by him.

We are greatly privileged to express our deep sense of gratitude to our course coordinator **Mr.A.Muthukumar M.Sc., M.C.A., M.Phil**, Course coordinator, Master of Computer Applications, Department of Computer Science and Engineering, who has been a motivating force behind all our deeds.

We are grateful to express our gratitude and sincere thanks to our guide **Mrs.S.Devaki,B.E.,M.S Assistant Professor**, Department of Computer Science and Engineering, who has been a constant source of encouragement for our project.

I wish to extend my gratitude to **Mr.K.Suresh Kumar, B.E,** Software Engineer, D'Gipro Design Automation & Marketing Pvt Ltd., Bangalore for giving me the opportunity to do this project in their organization.

We like to express our deep sense of gratitude to our parents, friends and all others who had been directly or indirectly involved with this project for their invaluable help and consideration towards us.

SYNOPSIS

"Implementation of PC to PC voice communication", is an application developed for transmitting voice over networks based on Internet Protocol. The project is undertaken for D'Gipro Automation & Marketing Pvt Ltd.

The main objective of the project is to study the VoIP concepts and to design, develop and implement an application that transmits voice from Personal Computer to another through network based on Internet Protocol.

The system is mainly aimed at providing real time voice transmission. Implementation phase includes coding and implementing the standards available in the industry like H.323.

By introducing the gateway between the packet switched network and the public switched telephone network, this project can be extended to PC to Phone or Phone-to-Phone.

Various testing strategies are done to check the errors in the modules and in the system as a whole. Unit test, Integration test and the system testing are done to maintain reliability and efficiency of the system.

Extensive use of modern programming methodologies and careful optimization are done to archive optimum results. Care has been taken to provide easy-to-use user interface. The user does not require memorizing any details regarding the system.

Although the quality of VoIP needs to be enormously improved, its price is roughly about half of today's long distance rates and as much as ten times cheaper than comparable international rates. Therefore, VoIP technology might be new revolution to the telecommunications.

CONTENTS

1. INTRODUCTION

1.1 Problem Definition

The world of telephony has changed a lot. The major breakthroughs have been the introduction of automatic exchanges, implementation of digital telephony, use of satellites and fiber optics for communications and the coming of mobile telephony. These have all been the improvements over the basic themes of an end-to-end circuit switched connectivity.

However the coming of the Internet and the subsequent development of Internet Protocol, IP telephony in short, changed all that. IP telephony encompasses many different ways of transmitting voice, fax and related services over packet switched IP-based networks.

This project aims at developing the packet switched network that can be used to transmit voice on Internet Protocol based networks.

There are basically three types of voice calls that use internet Protocol as a medium of transmission, PC-to-PC calls, PC to Phone calls and Phone-to-Phone calls. The network that involves a phone requires gateway, to convert digital packets to equivalent Dual Tone Multiple Frequency (DTMF) tones. The gateway is high cost hardware, which is unaffordable. This project implements PC-to-PC calls.

This project implements

RTP (Real Time Transport Protocol)

The Real time Transport Protocol is used to transmit the voice packets in real time.

RTCP (Real time Transport Control Protocol)

The Real time Transport Control Protocol is used to control the RTP.RTCP also provides statistics information for monitoring the Quality of Service of the voice calls.

Gatekeeper

Gatekeepers provide call control services for H.323 endpoints, such as address translation and bandwidth management.

Multipoint Control Unit

Multipoint Control Units provide support for conferences of three or more endpoints. MCU manages conference resources and handles negotiations between endpoints for the purpose of determining the audio or video codec, which is to be used.

Encryption Algorithm (XOR)

This algorithm is used to encrypt the data that is been transmitted to the network.

G.711 CODEC

This CODEC is used to compress the voice packets before routing the voice packets in the network.

The analog voice signals generated from the microphone is converted to digital form and compressed. These compressed and encrypted voice data is packed in RTP header and sent to the network as Internet Protocols packets. At the receiving side, the Internet Protocol header. RTP header is removed and the data is decompressed and played in the speaker, where the digital voice is converted to analog signal.

While Internet Telephony has usually meant for low cost or even free voice calls, VoIP has been much more than this for the core network operators, local exchanges and corporate enterprises.

1.2 Organisation Profile

D'Gipro has been providing a variety of software development services over the last one year in the domains of Embedded application, RTOS, Device Drivers, Data Warehousing and Portals.

D'Gipro has business models supporting turnkey development, Data/Application migration, Software Testing & Validation, Manpower supplement for any of the above.

D'Gipro Systems Pvt Ltd is recognized as the leading Electronic Design Automation (EDA) design services establishment in India. D'Gipro Systems was founded in 1987 with the sole intention of revolutionizing electronic hardware design methodology in India.

D'Gipro has been the pioneer in introducing the newer concepts of Electronic CAD based design allowing reductions in time, cost of development and also increasing the reliability of design.

D'Gipro Systems started off with CAD/CAE services for Electronics Design for various clients in India. Today, D'Gipro Systems has matured to provide the best of VLSI design services with good success for our clients in India and overseas. D'gipro Systems has been providing a variety of services to cater to the needs of the Commercial, Industrial and high technology electronics industry.

1.3 Existing System

At present the circuit switched networks dominate the telecommunication world. The circuit switched network establishes a physical connection between the endpoints. The connection exists till either of the endpoints closes the connection. The cost of establishing such connections and teardown is very high.

The cost required to make long distance calls by the circuit switched networks is really high. The cost spent in communication using circuit switched network is very high, that even big corporate are badly in need of alternate system for communication.

1.4 Proposed System

The Proposed system aims at replacing the circuit switched network with packet switched network. The packet switched networks establishes connection logically. The packet switched network helps in reducing the cost required to make long distance calls. Even though there are systems already available for transmitting voice over packet switched networks, they require some cost as transaction fee. Also this system provides easy-to-use user interface, which is not present in any of the existing system. The user interface is so simple that the system does not require any end user training.

The voice over Internet Protocol does much more than just reducing voice call cost. The services include:

- One network for voice, data, and video.
- Scalable and Flexible service.
- Simplified manageability.
- Shorter time to market.
- Rich value – added service menu.
- Low maintenance and management cost.

2. ANALYSIS

2.1 System Requirement:

2.1.1 User Requirement:

Nature of work requested:

The system should be able to transmit voice packets using packet switched network namely the Internet rather than the circuit switched network and hence the cost required to make long distance calls are reduced considerably. Apart from the toll free calls, the user interface should be pleasing and ease of use. The end user should not be presented with the technical requirements. Also the management of the system should be at ease for the administrators.

Job Objective:

The Job Objective of the system is to reduce cost of long distance calls. The end users should also be able to operate the system with ease.

Expected benefits:

The cost required to make long distance calls are reduced considerably. This system provides one network for voice, data and video. The system also provides simplified manageability, low maintenance and management costs.

2.1.2 Voice over Internet Protocol (VoIP)

Voice over Internet Protocol is one of the hottest topics for corporations of any size, because of its promises to transmit data and business voice over data network. VoIP can reduce overall telecommunications spending, address workforce demands for flexible hours and is the most straightforward path towards true convergence.

Though the years there have been some attempts to merge voice and data networks. But it wasn't until the Internet revolution and widespread deployment of IP data networks, that the industry at large finally had the right transport mechanism to support voice and data convergence.

There are several major drives for the convergence of voice and data networks.

These include:

Flexibility: Using IP as a transport mechanism offers companies the flexibility to deliver their voice traffic to virtually any location over any type of media.

Lower cost of Ownership: Companies recognize that consolidating their networks can save tremendously on communication costs. These cost savings are associated with everything from reduced long distance charges between offices, to reduced administrative overhead.

Corporate decentralization: Companies are continuing to decentralize their operations without increasing the overhead, and draw from a larger geographically dispersed labor pool. In order to do this successfully, companies must find way to deliver a uniform level of voice and data services to remote employees in order to overcome the communications barriers imposed by time and distance.

Next-generation business applications: Competitive pressures are forcing companies to deploy next-generation business applications in order to deliver better services to customers, partners and employees.

2.1.3 H.323 Overview:

H.323 is a part of a broader family of standards developed by ITU describing how audio, Video and data communications occur between terminals in IP networks that don't provide guaranteed QoS. H.323 provides a set of standards defining real time multimedia communications and conferencing over packet based networks. These standards define how components that are built in compliance with H.323, set up calls, and exchange compressed audio and video to participate in multiunit conferences, and operate with non-H.323 endpoints.

The H.323 protocol standard:

- Defines how audio and videoconferencing systems communicate over packet switched networks that do not guarantee Quality of Services (QoS), such as Internet and Intranets.
- Addresses call control and management for both point-to-point and multipoint conferences.
- Addresses QoS issues with a centralized Gatekeeper component that LAN administrators use manage, media traffic, bandwidth, and user participation.
- Addresses gateway functionality that allows calls to connect from the Packet switched network to Public Switched Telephone Network (PSTN), as well as to other H.32x standards-based terminals.

2.1.4 H.323 Components:

The H.323 components defined by the standard include:

- Terminals
- Gateways
- Gatekeepers
- Multipoint Control Unit (MCUs)

Terminals

Terminals provide real-time communications. They must support voice communications and can optionally support video and data communications, the most common H.323 terminal can be client software running on a PC.

Gateways

H.323 gateways provide services to H.323 clients, so they can communicate with non-H.323 entities. The most common type of H.323 gateway allows communications between H.323 terminals and telephones on the circuit switched network. The gateway must provide translations between different transformation formats, communication procedures and audio codec.

Gatekeeper

Gatekeeper provides call control services for H.323 endpoints, such as address translation and bandwidth management. Gatekeepers in H.323 networks are optional. If they are present in a network, however, endpoints must use their services. The H.323 standards define mandatory services that the gatekeeper must provide and specifies other optional functionality that it can provide.

Multipoint Control Units (MCUs)

Multipoint Control Units provide support for conferences of three or more endpoints. MCU manages conference resources and handles negotiations between endpoints for the purpose of determining the audio or video codec, which is to be used.

2.2 Scope of the system

This system is used to transmit voice using packet switched network, namely Internet instead of circuit Switched network. This helps to reduce the cost required to make long distance calls.

This system also provides:

- One network for voice, data, and video.
- Scalable and Flexible service.
- Simplified manageability.
- Shorter time to market.
- Rich value – added service menu.
- Low maintenance and management cost.

This system can be used by anyone from common man to corporate houses to make voice calls.

2.3 System Environment

2.3.1 Development Environment:

Hardware Specification:

Processor Type	:	Intel Pentium III
RAM	:	128 MB
Hard Disk Space	:	20GB
Sound Card	:	Full Duplex capable
Network	:	Ethernet 10Mbps

Software Specification:

Operating System	:	Windows NT
Language used	:	Visual Studio 6.0

2.3.2 Implementation Environment :

Hardware Specification:

Processor Type	:	Intel Pentium III
RAM	:	128 MB
Hard Disk Space	:	20GB
Sound Card	:	Full Duplex capable
Network	:	Ethernet 10Mbps

Software Specification:

Operating System	:	Windows 95 and above

Description of Technology

About the tool visual C++:

1. Graphical User Interface:

   Visual C++ is one of the best available tools for developing windows based Graphical User Interface applications.

2. 32-bit compiler:

   Version 4.0 and higher visual c++ provides powerful 32- bit compiler to help us to develop win 32 applications for Windows 95, Windows NT operating systems.

3. Unicode compatibility:

   Visual C++ helps us to write compatible programs.

4. The development system:

   The Microsoft 32 –bit visual c++ compiler incorporates fully integrated windows development tools under visual interface.

5. Integrated debugger:

   The integrated debugger allows us to execute programs in single steps, view and change variable contents.

6. Integrated resource editor:

   The resource editor allows us to design and create windows resources such as bitmaps, cursors, icons, menus and dialog boxes. Resources allow us to create visually appealing user interfaces to the application. Some of the popular resource editors are:

   Dialog box editor and graphical interface editor.

7. Precompiled headers and types:

   The visual C++ places generic types, function prototypes, external references and member file declarations in special files called header files. These header files contain many of the critical definitions needed by the multiple source files that are pulled together to create the execution version of the program.

8. Library

   The SDK library features many APIs for managing Windows objects and offers a number of general purpose APIs that can be used in both MS-DOS and Windows applications.

9. Additional tools:

   The additional tools visual C++ are integrated into the compiler's package. These include spy++, MFC tracer, the test container etc.

2.4 Justification of Development Methodology

This project makes use of incremental model, which combines elements of linear sequential model with the iterative philosophy of prototyping. The incremental model applies linear sequences in a staggered fashion as calendar time progresses. Each linear sequential produces a deliverable increment of the software. It should be noted that the process flow for any increment could incorporate the prototyping paradigm.

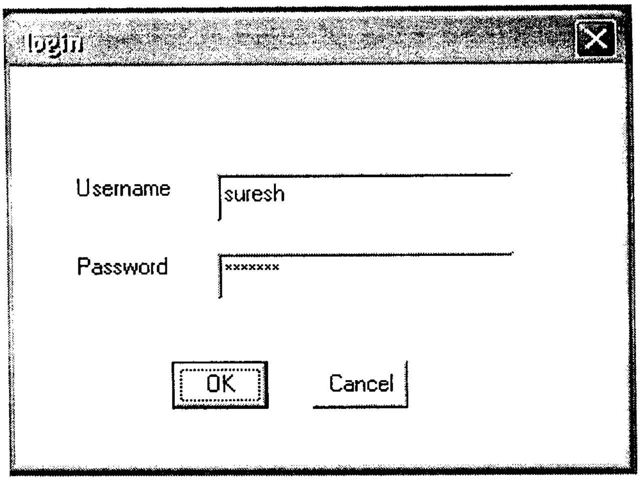
When an incremental model is used, the first increment is often the core product. That is, basic requirements are addressed, but many supplementary features remain undelivered. The plan addresses the modification of the core product to better meet the needs of the customer and the delivery of the additional features and functionality. This process is repeated following the delivery of each increment, until the complete product is produced.

The incremental process model, like prototyping and the other evolutionary approaches is iterative in nature. But unlike prototyping, the incremental model focuses on the delivery of an operational product with each increment. Early increments are stripped down versions of the final product, but they do provide capability that serves the user and also provide a platform for evolution by the user.

This method is also faster and cheaper than the modification of final software. This model has been recommended for systems that has user interface as an important part of the system being developed. This model is very effective because as users actually use the delivered parts, they start to understand better what they actually need. This leads to the changes in the requirements for further increments and revisions of the original plan. From the point of view of developer too, this is an effective model, since each increment is simpler to develop the whole system. As each increment is developed, the prototype evolves into final system.

3. DESIGN

3.1 User Interface Design

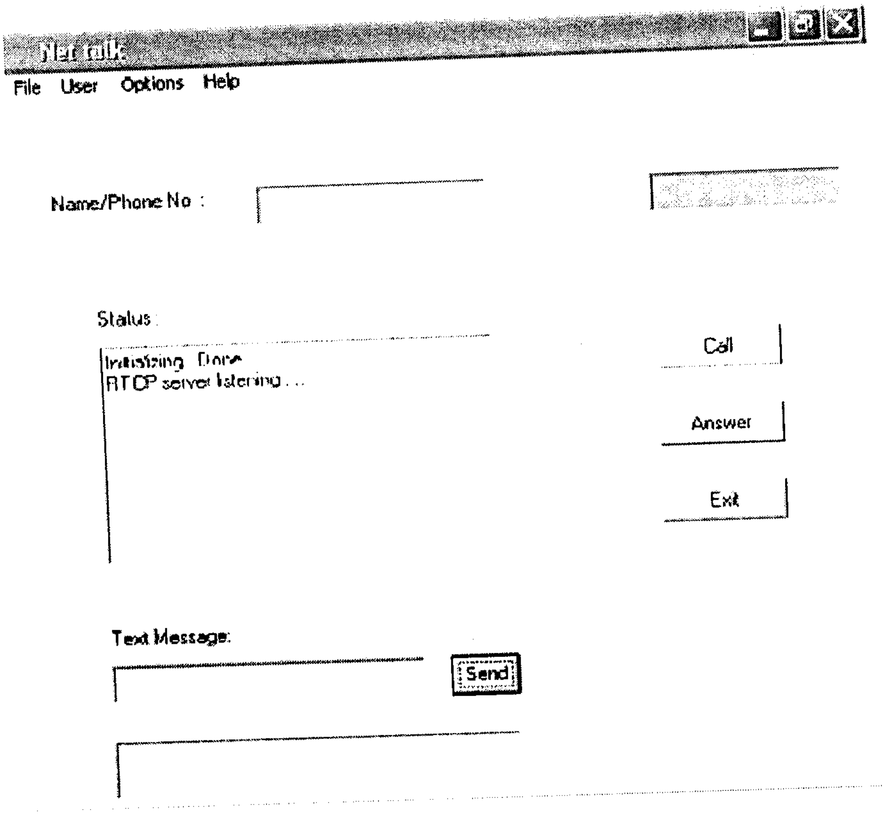


Login Screen

The login screen is used to authenticate that only valid users are allowed to use the service. The username and password are sent to Gatekeeper where the username and password are validated. Only valid users are allowed to make use of the options available in the main screen.

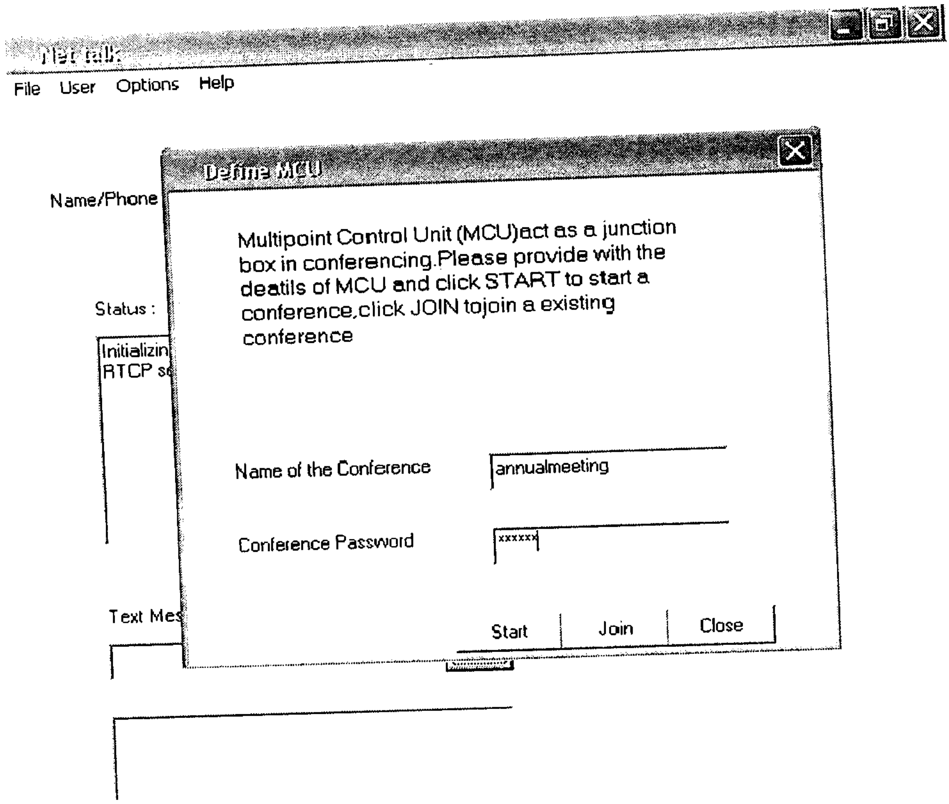
The user can click "cancel" and can proceed to the main screen, but cannot access the options available in the main screen. This option is to show the demo of the system to the user.

This is part of the client software "NetTalk". This also provides the present IP address of the client to the Gatekeeper where it is stored. Any other user who needs to communicate with others has to contact the Gatekeeper to obtain the IP address of the called and establish session.



Main Screen

The main screen has a log box where errors and user activities are displayed. The user types the username or the virtual number of the registered users and has to click "call to setup a call. Once the connection is established, the call button changes "hung up". The call can be broken down by clicking the "hung up" button. Once the call is broken down, the button again changes to "Call". The application can be terminated by clicking exit button. While exiting, the application sends "BYE" packet to the gatekeeper to inform that it no longer is available in the network for receiving calls.

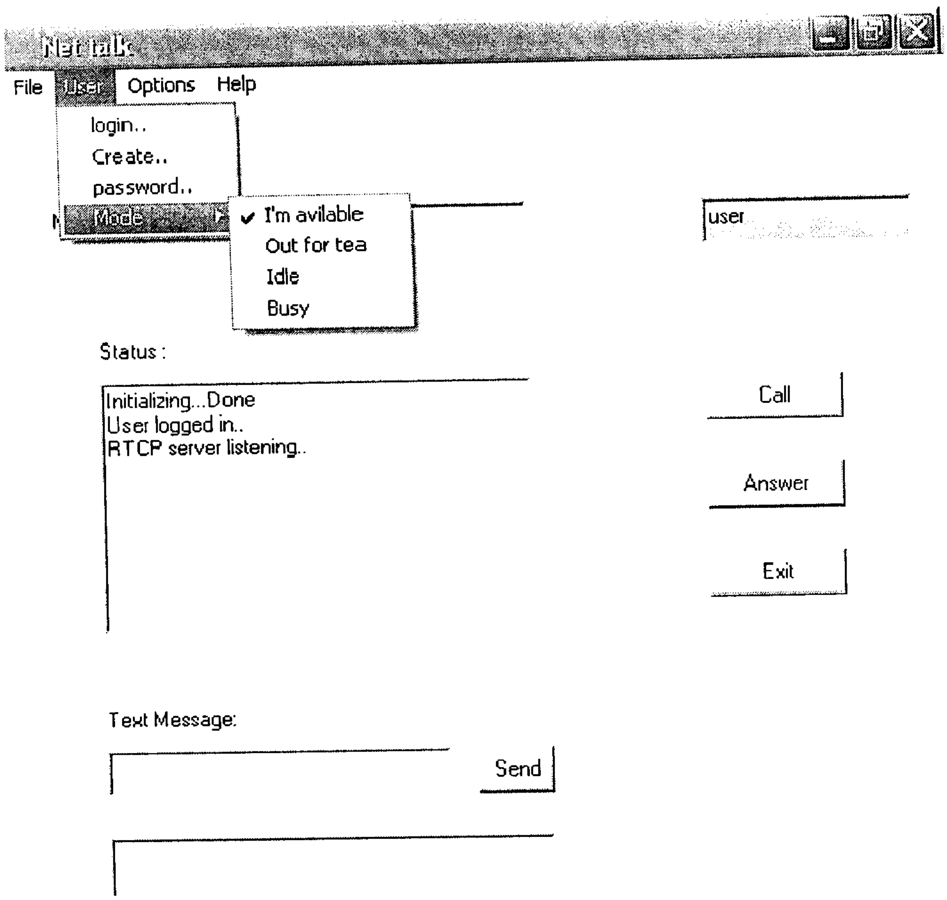


Call Conference Screen

This is the dialog box, which enables for making conference. The starter has to supply both the conference name, which is unique among the ongoing conferences. The password is also supplied to verify the integrity of the user.

The starter of the conference has to click "start" button to start conference. The conference name and the password are supplied by other means. The starter has to be present for the whole duration of the conference. If he exits before other users exit due to some unavoidable reason, the conference is closed.

Other users those who wish to join the conference can supply the conference name and password to join the conference. The conference can be closed by clicking the "Close" button.



User Menu Screen

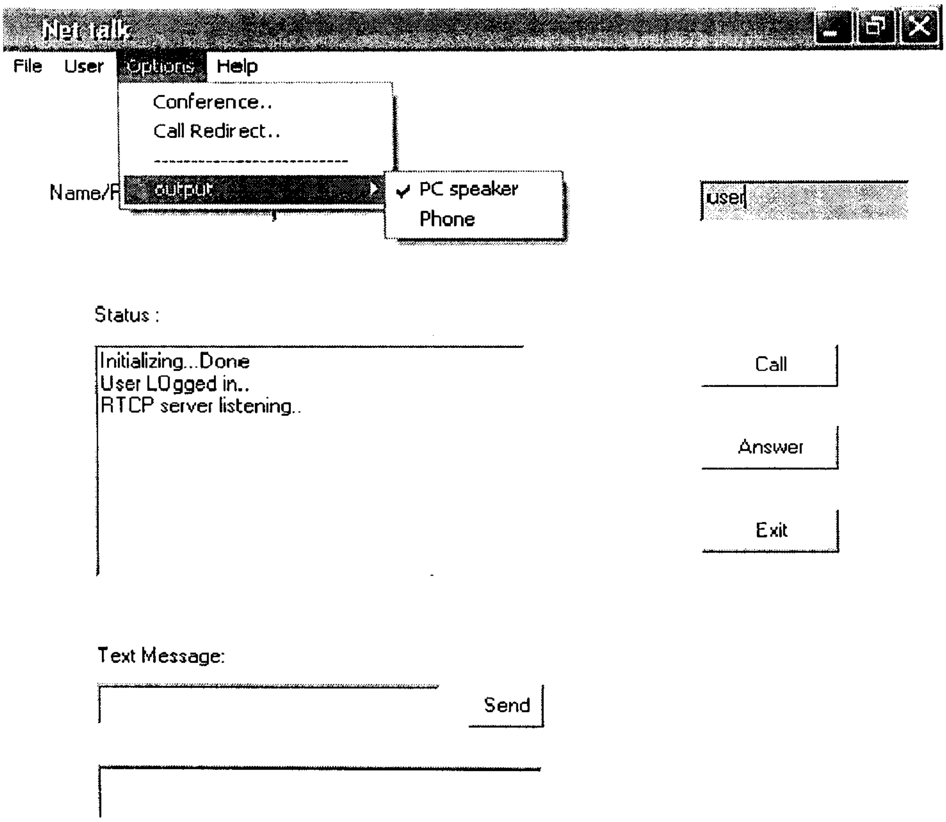
This screen shows option menu of the "NetTalk". The LOGIN option is used to login to the gatekeeper to register his/her presence and to inform the gatekeeper about the present IP address.

The "Create user" option is used to create a new user where the information regarding the user such as username, password and virtual or nickname are stored in Gatekeeper.

The password option is used to change the existing password with a new one.

The modes available are

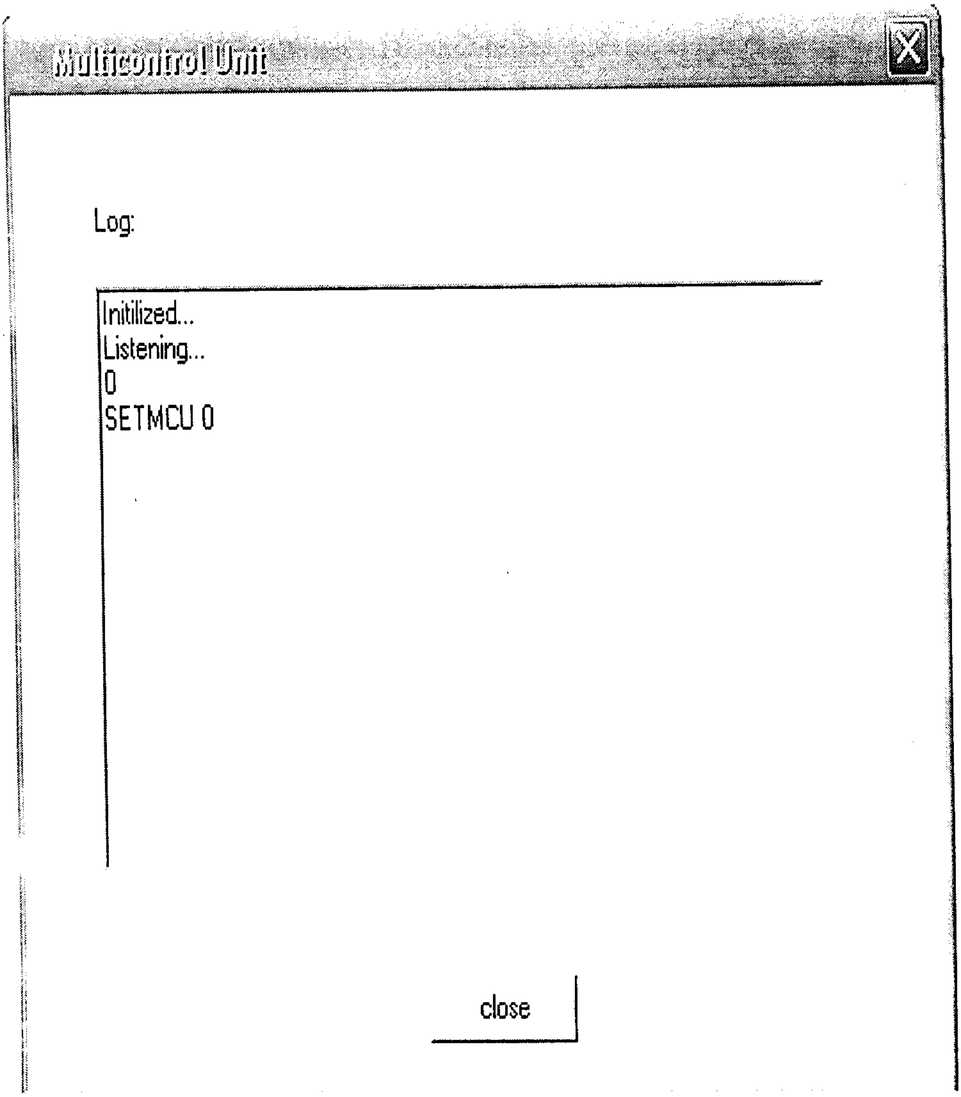
I am available	- ready to receive calls
Out for tea	- the user would be back in 10 minutes
Out for lunch	- the user would be back in 30 minutes
Busy	- not ready to accept calls



Option Menu Screen

The options menu of the application has the following

Conference:	This option is to Start or Join a Conference.
Call redirect:	This option is supplementary service that is used to redirect a call to some other users.
Output:	This is to redirect the output to Phone or to a Speaker.



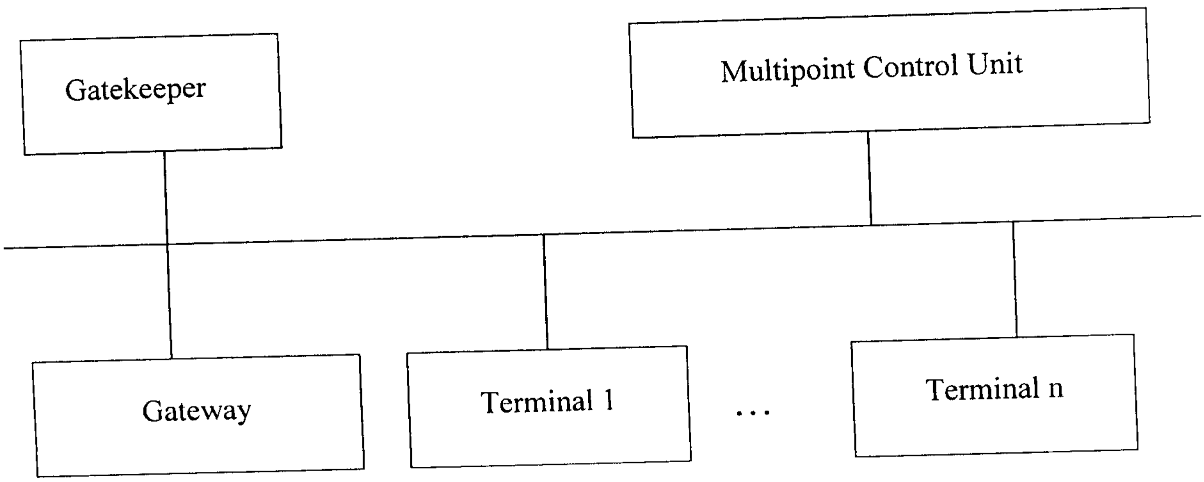
MCU Log Screen

This screen shows *Multipoint Control Unit* (MCU), which is used to control all the conferences.

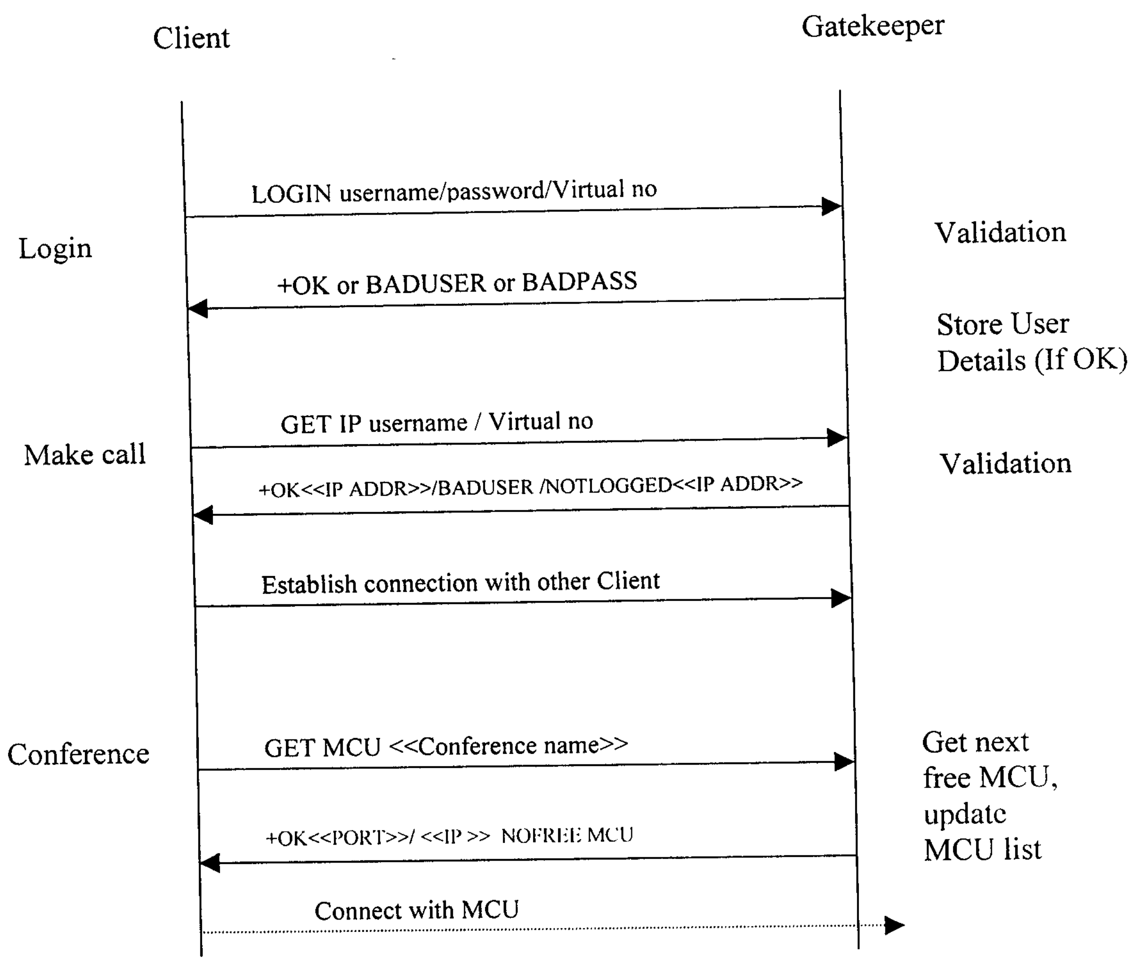
3.2 Design Diagrams and Charts:

3.2.1 Architecture Diagram:

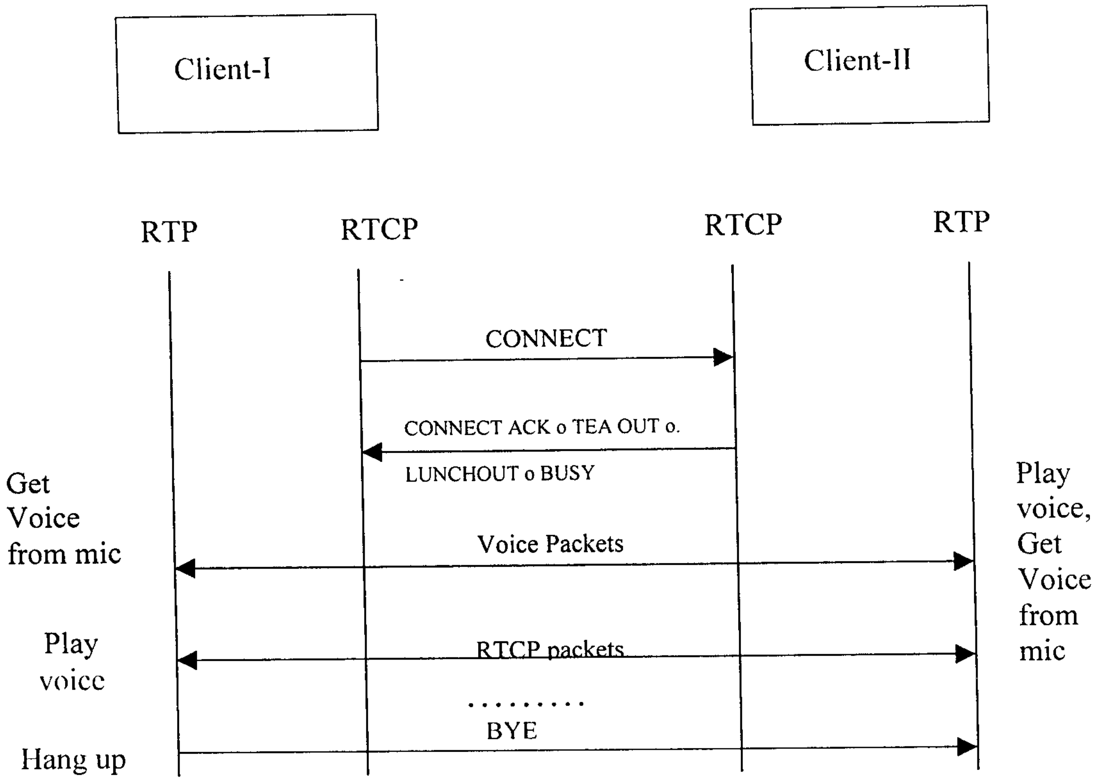
H.323 Zone



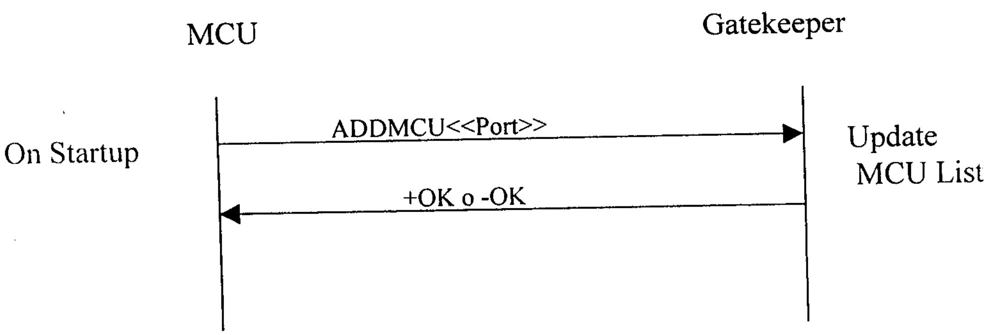
3.2.2 Communication between Client and Gatekeeper



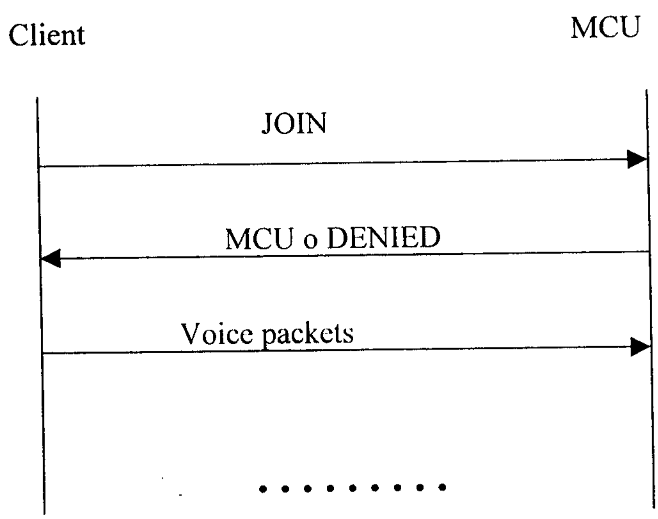
3.2.3 Communication between Clients



3.2.4 Communication between MCU and Gatekeeper

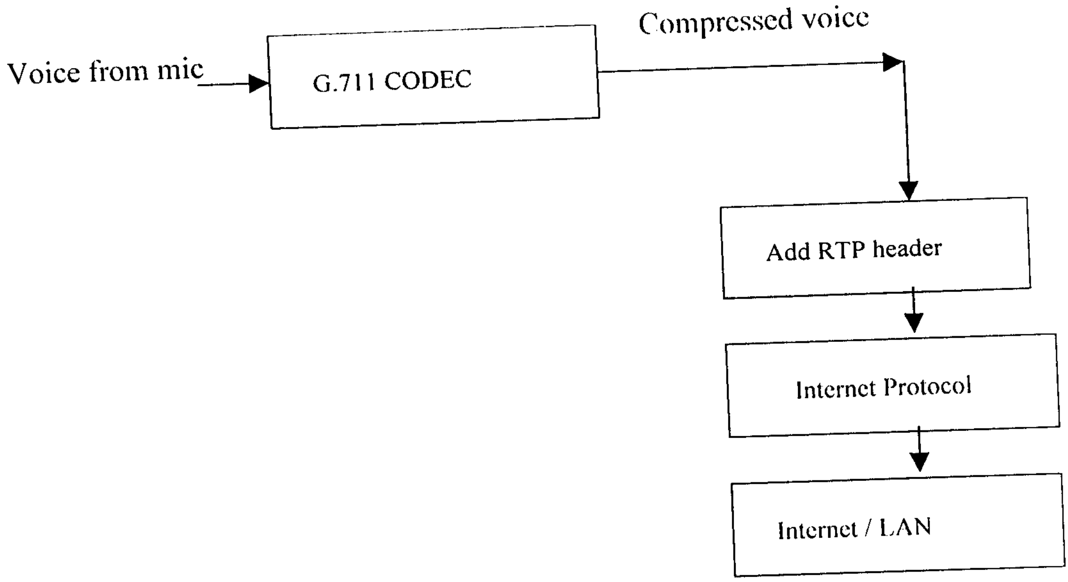


3.2.5 Communication between MCU and Client

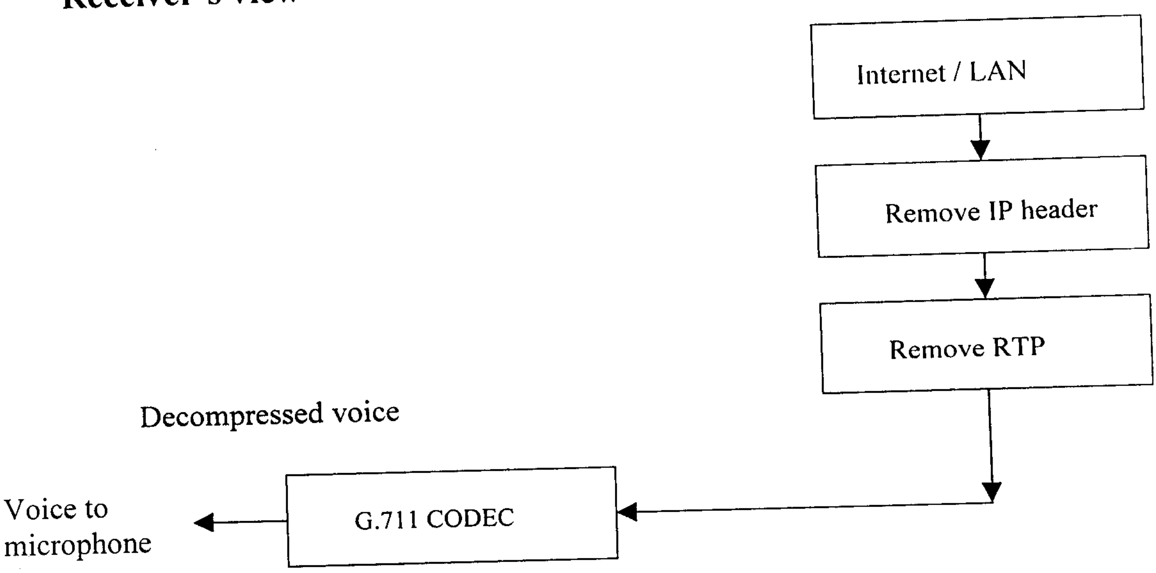


3.2.6 System Flow Diagram

Sender's View



Receiver's view



3.2.7 RTP header

RTP fixed header fields:

V=2	P	X	CC	M	PT	Sequence number
Time stamp						
Synchronization source (SSRC) identifier						
Contributing source (CSRC) identifier						

Version (V): Identifies version of RTP.

Padding (P): If set, holds one or more additional padding octets at the end which are not part of the payload. Needed by some encryption algorithm.

Extension (x): If set, the fixed header is followed by exactly one header extension.

CSRC Count (CC): Contains number of CSRC identifiers that follow the fixed header

Marker (M): Intended to allow significant events such as frame boundaries to be marked in the packet stream.

Payload type (PT): Identifies the format of the RTP payload and determines its interpretation by the application.

Sequence number: Increments by one for each RTP data packet sent.

Timestamp: Reflects the sampling instant of the first octet in the RTP data packet.

SSRC: Identifies the synchronization source.

CSRC: Identifies the contributing source.

RTP header Extension:

Defined by profile	Length
Header extension	

An extension mechanism is provided to allow individual implementations to experiment with new payload-format-independent function that require additional information to be carried in the RTP data packet header.

3.2.8 RTCP Header

Sender Report RTCP Packet

<table>
<tr><td>V=2</td><td>P</td><td>R C</td><td>PT=SR=200</td><td colspan="2">Length</td></tr>
<tr><td colspan="6">SSRC of Sender</td></tr>
<tr><td colspan="6">Time stamp, most significant word</td></tr>
<tr><td colspan="6">Time stamp, least significant word</td></tr>
<tr><td colspan="6">Sender's packet count</td></tr>
<tr><td colspan="6">Sender's octet count</td></tr>
<tr><td colspan="6">SSRC</td></tr>
<tr><td colspan="3">Fraction lost</td><td colspan="3">cumulative numbers of packet lost</td></tr>
<tr><td colspan="6">Highest sequence Number received</td></tr>
<tr><td colspan="6">Last SR (LSR)</td></tr>
</table>

Version (V): Identifies the version of RTP, which is same in RTCP packets as in RTP data packets.

Padding (P): If set, this RTCP packet contains some additional padding octets at the end which are not part of the control information.

Reception report count (RC): The number of reception report blocks contained in this packet.

Packet type (PT): Contains the constant 200 to identify this as an RTCP SR packet.

SSRC: The synchronization source identifier for the originator of this SR packet.

Sender's packet count: The total number of RTP data packets transmitted by the sender since starting transmission.

Sender's octet count: The total number of payload octets transmitted in RTP data packets by the sender since starting transmission.

Fraction lost: The fraction of RTP data packets from source SSRC lost since the previous SR or RR packet was sent.

Cumulative number of packet lost: The total number of RTP data packets form source SSRC that have been lost since the beginning of reception.

Receiver Report RTCP Packet

<table>
<tr><td>V=2</td><td>P</td><td>RC</td><td>PT=RR=201</td><td colspan="2">Length</td></tr>
<tr><td colspan="6">SSRC of sender</td></tr>
<tr><td colspan="6">SSRC</td></tr>
<tr><td colspan="3">Fraction lost</td><td colspan="3">cumulative no of packet lost</td></tr>
<tr><td colspan="6">Highest sequence number received</td></tr>
<tr><td colspan="6">Last SR (LSR)</td></tr>
</table>

The format of the receiver report (RR) packet is the same as that of the SR packet except that the packet type field contains the constant 201. The remaining field has the same meaning as the SR packet.

An empty RR packet (RC-0) is put at the head of a RTCP packet when there is no data transmission.

4. IMPLEMENTATION

4.1 Testing and test Plan

Testing is a process of executing a program with the intent of finding errors. One should not start testing with the intent of showing that the program works, but the intent should be to show that the program does not work.

During testing the program to be tested is executed with set of test cases and the output of the program for test cases is evaluated to determine the program is performing as expected.

Testing a large system is a complex activity, and like any complex activity it has to be broken into smaller activities. Due to this, for a project, incremental testing is generally performed in which components and subsystem of the system tested separately before integrating them to form the system for system testing. This form of testing, though necessary to ensure the quality of the system, introduces new issues of how to select components of testing and how to combine them to form the subsystem and the systems.

Generally, parts of the program are tested before testing the entire program. Besides partitioning the problem of testing, another reason for testing parts separately is that if the test case deducts an error in a large program, it will be extremely difficult to pinpoint the source of the error. If a huge program did not work, determining which module has errors can be a formidable task. In many cases it is even difficult to construct test cases so that all the modules will be executed. This increases the chances of module's errors going undetected. Hence it is clear that for a large system, we should first test different parts of the system independently, before testing the entire system.

In incremental testing, some parts of the system are first tested independently, and then these parts are combined to form a sub system, which is then tested independently. There are two common ways modules can be combined, as they are tested, to form a working program: top – down and bottom- up.

In top down approach the testing is started from the top of the hierarchy, and modules are added incrementally that it calls and the new combined system is tested.

In this approach, a module cannot be tested in isolation because they invoke some other modules. To allow the modules to be tested before their subordinates have been coding, stubs simulate the behavior of the subordinates.

The bottom – up approach starts from the bottom of the hierarchy. First the modules at the very bottom, which have no subordinates, are tested. Then these modules are combined with higher- level modules for testing. At any stage of testing all the subordinated modules are exist and have been tested earlier.

It is often best to select testing method to conform to the development method. The development mentioned here are actual development and not the design method. This project uses top – down testing strategy.

4.2 Testing Methods

Having test cases that are good at revealing the presence of faults is central to successful testing. Ideally a set of test cases is to be determined such that successful execution of all of them implies that there are no errors in the program. This ideal goal cannot usually be achieved due to practical and theoretical constraints. Each test case needs more effort, machine time to evaluate the results. While selecting the test cases the primary objective is to ensure that if there is an error or fault in the program, it is exercised by one of the test cases. One possible ideal set of test cases is one that includes all the possible inputs to the program this is impractical and infeasible, as for even small programs, the number of elements in the input domain can be extremely large. Hence a realistic goal for testing is to select a set of test cases that is close to ideal.

For unit testing, structural testing based on the branch coverage criterion is used. The goal is to achieve branch coverage of more than 95%. System testing is largely functional in nature. The focus is on invalid and valid cases, boundary values and special cases.

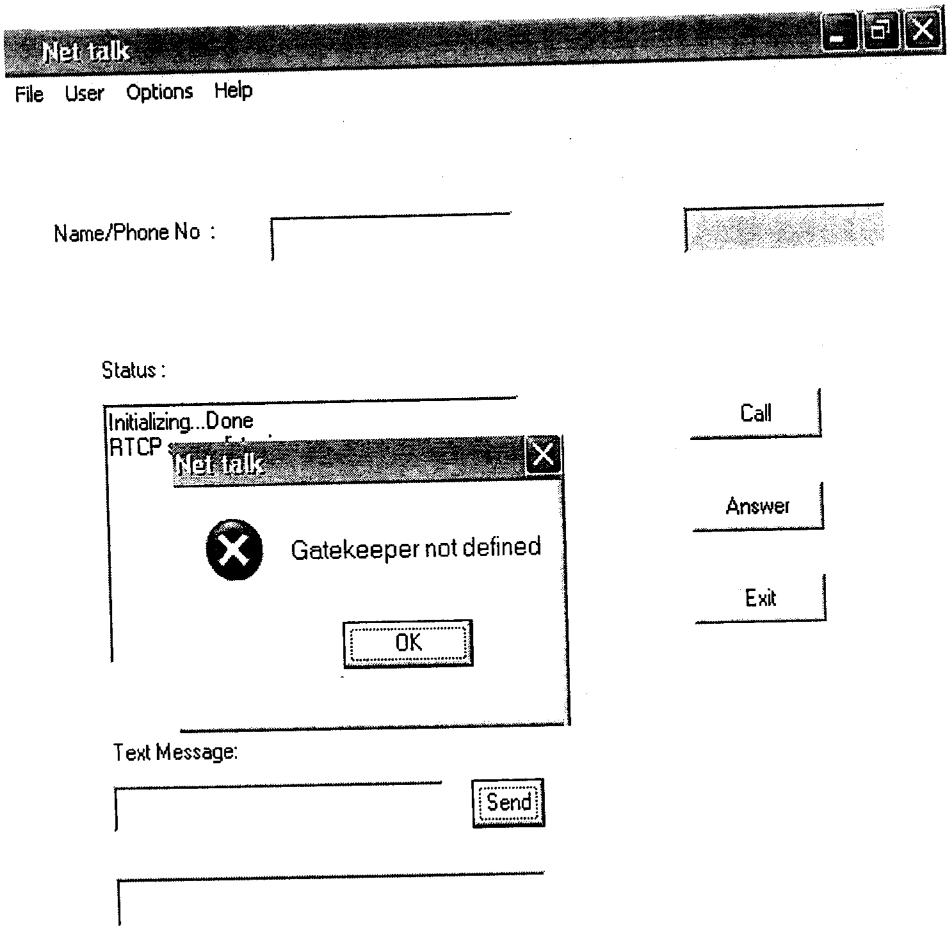
4.2.1 Unit Testing

All the modules in the system are tested in isolation. The user inputs are validated for empty fields, invalid characters and wrong data. The messages passed through the network have control characters such as '/', which are not allowed in user inputs. The user name is restricted to have only alpha numerals. Any special character in the user name is considered error and the message box is displayed promptly stating the error.

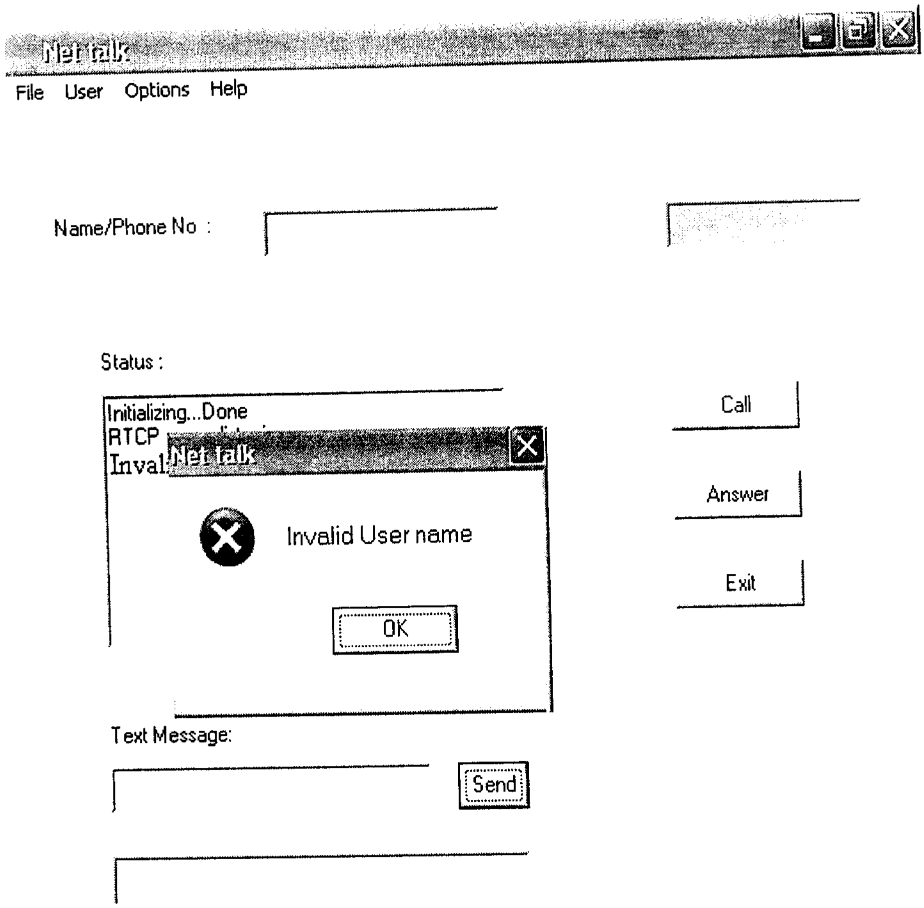
The zero length input, invalid character are tested such as number of input parameters match number of arguments and parameter attributes match, number of attributes and order of arguments to built – in function's correctness are also considered.

For the modules that perform Input/Output operations, the testing such as correctness of file attributes, file open statement ending up with file close statement, match of buffer size with that of record size, handling of end – of – file condition, I/O error handling are done.

All the modules are tested for improper or inconsistent typing, erroneous initialization, misspelled variable names, underflow, overflow and addressing exceptions.



The client should connect to gatekeeper to advertise its presence. The gatekeeper should have static IP address, so that client defines the address of gatekeeper to establish connection. If gatekeeper address is not defined or invalid address is defined this error message instructs user to define valid gatekeeper address.



The username and the password are verified in the gatekeeper. The gatekeeper returns BADUSER or BADPASS message in case if the username or password supplied is wrong.

4.2.2 Integration testing

Though the unit testing is preformed on all modules, errors may creep in when the units are integrated, sub functions, when combined, may not produce desired outputs. The strategy followed for integration testing is incremental model. The program is constructed and tested in small segments, where errors are easier to isolate and correct. This system is tested using top - down integration testing. Modules are integrated by moving downward through the control hierarchy, beginning with the main module.

The main control module is used as a test driver, and the stubs are substituted for all modules directly subordinate to the main control. The subordinate stubs are replaced one at a time with actual modules and tests are conducted as each module is integrated. Each time a new module is added as a part of the integration testing, the software changes, new data flow paths are established, new I/O may occur, and new control logic is invoked. These changes may cause problems with functions that previously worked flawlessly. So some subset of tests are re – executed to ensure that changes have not propagated unintended side effects.

4.2.3 Test cases for system testing

System testing is largely functional in nature. The focus is on invalid and valid cases, boundary value and special cases.

Seq. No.	Test case	Conditions being checked	Expected output
1	User data file empty	User login	Print message and stop
2	MCU data file empty	MCU registration	Create a new file and continue
3	Microphone off	Try to make call	Print message and stop
4	Network disconnected	Try to make call/login	Print message and stop
5	Invalid called person	Try calling invalid user	Print message and stop
6	Invalid GK definition	Try making call/Login	Print message and stop
7	Conference starter exit	Conference starter try to exit before joiners exit	Print message and stop
8	Exit without logout	User tries to exit without logging out	Automatically logout and exit
9	Speaker off condition	On receiving call, with speaker in off condition	Warn the user and wait
10	No disk space	User or MCU Information could not be stored due to disk out of space	Print message and stop

4.2.4 Platform relevant testing

Testing graphical user Interface

A series of standards are suggested for testing GUIs. Testing is been performed for the following:

- Proper opening and closing of windows
- Proper moving and scrolling of windows
- Proper regeneration of windows when it is overwritten and recalled
- All the relevant pull down menus, dialog boxes, buttons, icons, and other controls are properly displayed for the window
- Highlighting of active window
- Appropriate menu bars displayed in appropriate context
- Proper working of pull down operations
- Based on the context of current operations within the window, the menu items are highlighted or grayed
- Self explanatory menu items
- Proper addressing of menu items by mouse

Testing client server environment

The nature of client server systems poses a set of unique problems. Testing of client server environment includes client GUI consideration and target environment.

The client software is tested in disconnected mode. The network is manually disconnected and the client is run to test whether proper error message is displayed.

Next the client and the associated server applications are tested in concert, but network operations are not exercised explicitly. Finally the complete client server architecture is tested including network operation and performance is tested.

Apart from these tests, a series of tests are conducted to ensure that each class of transactions is processed according to requirements. Also tests are conducted to verify the communication among the nodes of network occurs correctly and the message passing, transactions, and related network traffic occur without error.

The software is allowed to run in a network environment of 20 clients where all the clients copy a large avi file-using FTP. The system performs well with some increased latency.

4.3 Installation of the system

The installation of the system does not require any additional components. The client software NetTalk uses only the existing components of the Windows operating system. It is associated with a file called gatekpr.def, which holds the address of the gatekeeper. This file is automatically created if intentionally or unknowingly deleted. The system is started when the system is turned on, but can be closed at any time. When the application is closed the user might not be able to receive calls. The application can be made to run in background.

The gatekeeper and *Multipoint control unit* are "start and run" type programs. The programs are started and left. The gatekeeper should have static IP address.

4.4 Errors and failures during system development and solutions

During the development of the project, some problems were faced. The memory allocated to pointer variables could not be released manually. Windows takes care of releasing memory when the process ceases to exist. This unnecessarily keeps memory till the process exit. So the use of pointers is avoided as much as possible.

One more error is uncovered while showing the demo. One of the constraints specified in the specification is that the user cannot make self-calls. The supplementary service “call redirect”, stores the name of the user in the IP address field, to whom the call is to be redirected. Whenever call comes in for the user, the IP address field is checked for any username. If present the IP address of that user is retrieved and the call is made. Say if user1 redirects the calls to user2 and user2 in turn redirects the calls to user1. This forms a cycle, which indirectly allows a user to make self-calls. This error is rectified by introducing a constraint that a call cannot be redirected to a user, who already redirected the calls to some other user.

5. LIMITATIONS AND FUTURE ENHANCEMENTS

Limitations:

- The quality of VoIP needs to be enormously improved.
- Since VoIP is in developing stage, the standard might become obsolete when a new technique with some superiority is found.
- Although the sound quality of VoIP is tolerable, it cannot match with PSTN.

Future Enhancements:

- By introducing the gateway, this project can be extended to provide PC to Phone and Phone-to-Phone services.
- VoIP is a single network for voice, data and video. Thus this project can be extended to provide video conferencing.
- This project can also be extended to provide Fax over Internet Protocol (FoIP).
- Since this project uses Internet Protocol to transmit data, anything over Internet Protocol can be implemented.

6. CONCLUSION

NetTalk provides a way for communicating through the network by means of voice. Voice data is transmitted through the network by appropriate compression and encryption techniques. This project can be extended to provide video conferencing and Fax over IP by implementing appropriate protocols and integrating with this project.

By placing the gateway between this system and the circuit switched network, this project can be extended to provide PC to Phone service.

The application is inter operable with many of the existing systems as it uses H.323 protocol, a standard for transmitting voice over the packet switched network.

7. REFERENCES

Books:

1. Buck Graham, "TCP/IP Addressing"
2. Charles Petzold, "Programming Windows", Microsoft Press
3. James, Keith "Computer Networking"
4. Jerry D. Gibson, "Multimedia communication"
5. Larry Hughes, "Introduction to data communication"
6. Lewis Napper, "Winsock 2.0", Comdex computer publishing
7. Ulysses Black, "Internet architecture – An introduction to IP protocols"

Web sites:

1. www.cisco.com
2. www.cs.columbia.edu
3. ftp://fife.speech.cs.cmu.edu/comp.speech
4. www.freetel.com
5. www.ietf.org
6. www.iLocus.com
7. www.iptelephony.org
8. www.itu.int
9. www.microsoft.com
10. www.netspeak.com
11. www.openH323.org
12. www.phonelan.org
13. www.speakfreely.org
14. http://techlibrary.networkcomputing.about.com
15. www.telesoft.com

RFCs:

1. RFC 1889

APPENDIX

Project Dictionary

The details maintained by Gatekeeper

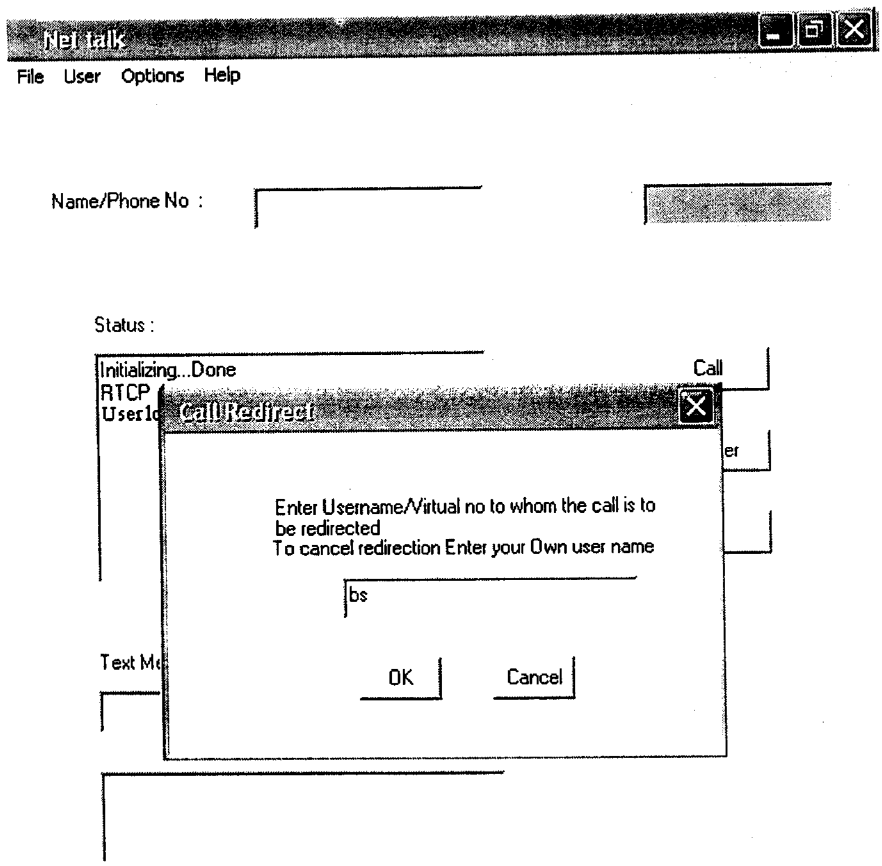
User information:

- Username
- Virtual number/Nick name
- Password
- IP address of client
- IsLogged

Multipoint Control Unit Information:

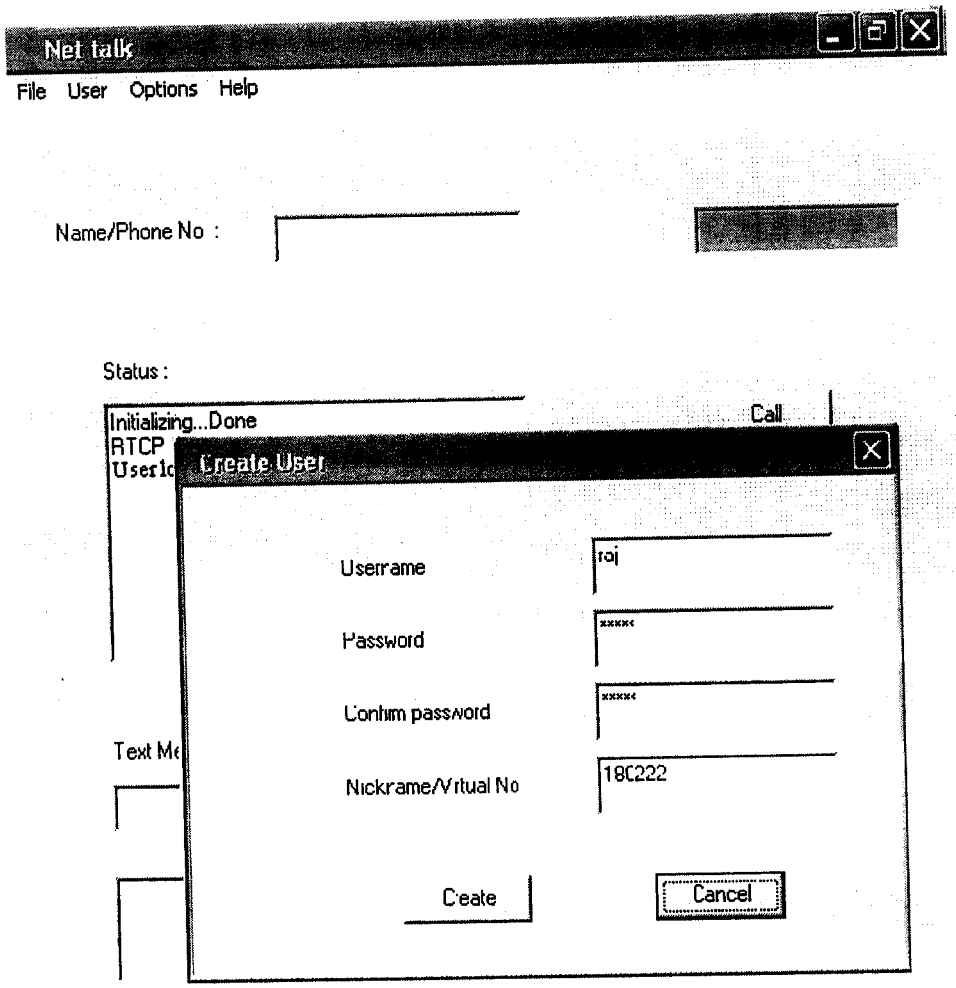
- Conference name
- IP address of MCU
- Port number
- IsFree

Call redirection (supplementary service) screen shot



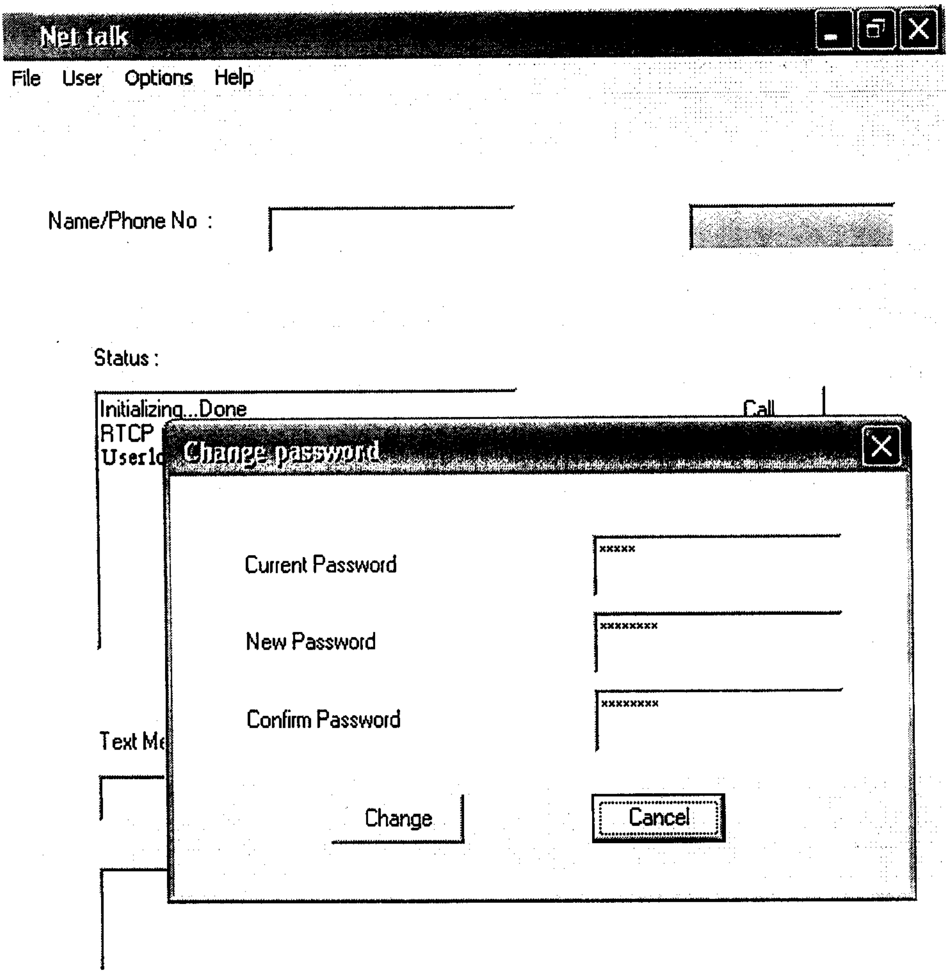
This screen shows the supplementary services, call redirect, offered by NetTalk. This service helps to redirect any incoming call to another account. When the user is away and could not attend the calls, the incoming calls may be redirected to another user who could attend the calls. The user just needs to type the username to whom the call redirection is intended. The redirection is cancelled by providing own account name.

Create user



The create user dialog box is used to create and register use account with Gatekeeper. The Gatekeeper maintains the user account details. The client needs to communicate with the gatekeeper to login. A user can be identified by account name or virtual number/ nickname. The details are sent to gatekeeper where the account is registered.

Change password



The user can change the password at any time. The user should first login, and should supply current password and new password (twice). The password details are sent to gatekeeper where the password is validated and if found valid the password is changed to new one.