Half Duplex Radio Net Emulator

By

USAMA AHMED

HASSAN YAR KHAN

TALHA MAZHAR

AHMED QURESHI

Supervised by:

LT COL DR. HASNAT KHURSHID

Submitted to the faculty of the Department of Electrical Engineering,
Military College of Signals, National University of Sciences and Technology, Islamabad,
in partial fulfillment of the requirements of B.E Degree in Electrical (Telecom) Engineering.

June 2023

In the name of ALLAH, the Most benevolent, the Most Courteous

CERTIFICATE OF CORRECTNESS AND APPROVAL

This is to officially state that the thesis work contained in this report

"Half Duplex Radio Net Emulator"

is carried out by

USAMA AHMED

HASSAN YAR KHAN

TALHA MAZHAR

AHMED QURESHI

under my supervision and that in my judgement, it is fully ample, in scope and excellence, for the degree of Bachelor of Electrical (Telecom.) Engineering in Military College of Signals, National University of Sciences and Technology (NUST), Islamabad.

Approved by

Supervisor

LT COL DR. HASNAT KHURSHID

Department of EE, MCS

Date: ___________

DECLARATION OF ORIGINALITY

We hereby declare that no portion of the work presented in this thesis has been submitted in support of another award or qualification in either this institute or anywhere else.

ACKNOWLEDGEMENTS

Allah Subhan'Wa'Tala is the sole guidance in all domains.

Our parents, colleagues, and most of all supervisor **LT COL DR. HASNAT KHURSHID**

without your guidance.

The group members, who through all adversities worked steadfastly.

Plagiarism Certificate (Turnitin Report)

This thesis has ____ similarity index. Turnitin report endorsed by Supervisor is attached.

____

Usama Ahmed

NUST Serial no 325606

____

Hassan Yar Khan

NUST Serial no 325607

____

Talha Mazar

NUST Serial no 325036

____

Ahmed Qureshi

NUST Serial no 325061

____

Signature of Supervisor

ABSTRACT

Communication during war and peace times is paramount. Proper communication enhances one's ability to safeguard himself from enemies. Thus, communication goes hand in hand with vigilance and duty .it is only possible if recruits and officers pioneer themselves at radio and other forms of communication.

For instance, France's defeat during World War 2 is attributed directly due to lack of knowledge about equipment and thus delay in relaying the information to superiors.Nowadays, radio sets as used by Pakistan Army are very costly. Their extensive use in field makes it difficult to train young officers and recruits with the latest radio handsets. Our project emulates a half-duplex radio using python code and modules and hence aims to provide a feasible solution to this difficulty.

Contents

Chapter 1

Introduction

1.1 Overview

Half duplex radio net emulator is a virtual depiction of Harris and Motorola handsets used by Pakistan army.it used python codes and modules to run an application which is even compatible with windows 7.it also uses tinder to create a GUI that facilitates students to better comprehend the communication environment in their surroundings.

1.2 Problem statement

Traditional radio sets have a very high cost of production. This can be attributed to a number of factors such as technology, features, demand, market price etc. This puts a strain on training institutes such as Signals Training Centre Kohat to train recruits training goes hand in hand with battle readiness and equipment expertise. A thorough analysis of textbooks and training manuals is simply not enough for acquiring expertise in a particular technology. Hands-on experience is necessary in acquiring mastery over any piece of equipment.

1.3 Approach

Our project aims to deal this issue with software solution. Our project is an emulator which simulates a traditional radio set. The GUI can be modified to resemble any handset and thus this project can be worked upon to adapt it the users and students need. This project utilises socket technology to achieve a peer to peer to model. Although the technology is not the same as any radio set, however working is same and thus it achieves the purpose.

1.4 Scope

Currently, LEAs must manually retrieve calls made by a specific user. It would be impossible to do so for millions of people. When manually extracting call conversations from large databases, LEAs ran into serious problems. Authentication, security and surveillance, electronic voice eavesdropping, and identity verification all use speaker identification technology. The technology we propose will automate the entire retrieval procedure and assist LEAs in their operations.

1.5 Objectives

the proposed model will function in a peer-to-peer information sharing architecture to act as a communication platform.

1.6 Deliverables

Sr. No Tasks

1 Literature Review

2 Requirements Specification

3 Detailed Design 4 Implementation 5 Training

6 Testing

7 Deployment

Deliverables Literature Survey

Software Requirements Specification document (SRS)

Software Design Specification document (SDS) Project demonstration

Deployment plan Evaluation plan

Complete desktop application with the necessary

documentation

1.7 Justification for Selection of Topic

For training purposes this will assist training institutions to impart technical and communication skills to the officer and recruit cadre. training and specializing in good technical skills enhance battle radiness and vigilance. This project can not only be used for the military but also for civilian use and training purposes such as security agencies. What makes this model unique is its ability to adapt to any network or application and its GUIs can be modified to resemble any equipment whereas, the working of the project remains same.

1.8 Overview of the Document

This document explains how our application AACRS works in detail. It begins with a literature review, which highlights previous work in a similar field, a system requirement analysis, a system architecture that highlights the software modules and represents the system in the form of a component diagram, a Use Case Diagram, a Sequence Diagram, and the overall design of the system. The discussion will then move on to a full description of all of the components involved. The system's dependencies, its relationship with other goods, and its ability to be reused will also be explored. Finally, test scenarios and a recommendation for future work were provided.

1.9 Document Conventions

Headings are numbered in order of priority. Font used is Times New Roman. All the main headings are of size 16 and bold. All the second level sub-headings are of size 14 and bold. All the further sub-headings are of size 12 and bold. Where necessary references are provided in this document. However, where references are not provided, the meaning is self-explanatory.

1.10. Intended Audience This document is intended for:

1. Developers: (Project Group)

To ensure that they are constructing the suitable enterprise that meets the requirements outlined in this report.

2. Testers: (Project Group, Supervisor)

To have a detailed list of the features and capabilities that must respond to requirements. 3. Users:

To be familiar with the task's potential and how to use/react in let down situations, as well as to suggest additional features that would make it significantly more valuable.

4. Documentation writers: (Project Group)

Recognize which characteristics and how they should be clarified. What innovations are required, how will the framework react in each client's activity, what potential framework disappointments may emerge, and what are the solutions to each of those disappointments, and so on.

5. Project Supervisors: (Dr. Hasnat Khurshid)

The project supervisor will utilise this document to ensure that all needs have been understood and, in the end, that the requirements have been implemented correctly and thoroughly.

Chapter 2

Literature Review

2.1 What is voice procedure training?

Speech training is a type of training that teaches a person to communicate effectively over the radio or over the phone. This type of training is often used in military, aviation, maritime and emergency response environments where clear and concise communication is critical to safety and performance.

Speech training usually includes topics such as pronunciation, phonetics, sentence structure and communication. The training also emphasizes the importance of clear, concise, and open communication to minimize misunderstandings and mistakes.

Through speaking education, individuals develop the skills necessary to communicate effectively under pressure and ensure that important messages are communicated clearly and effectively.

2.2 Half duplex Radio

Traditionally, half duplex radios are used in field. This is because bandwidth is an important commodity and hence it must be confirmed. however, the unidirectional function of a half-duplex radio might perplex a novice who has no prior hands-on experience.

halfduplex can be used for media sharing by more than two nodes, while fullduplex is generally not. In a shared environment such as a coaxial cable with many connections, all nodes can share the channel because each node knows that it has to check that the channel is free before sending.

2.3 How does the emulator work?

This emulator works through a peer-to-peer mechanism.in a P2P environment all the users share an equal responsibility to share and transmit data. This reduces the processing power drawback of severs .it saves cost and reduces the overall clutter on the network.

2.4 Previous work

We worked to adapt our project closest to handsets used by Pakistan Army Developers have created apps like Mumble to impart voice procedure training. However, no work has been done in the military domain.

Chapter 3

Design

A few parameters have been kept in mind while designing the GUIs. these parameters emulate the actual radio set. Some of these parameters include frequency, NIS, call sign, squelch(noise).

The image above shows an RF 5800 H set .it has many parameters which are to be set and put in place before tactical communication can take place. Although the functionality of the software allows a work out of the box ability but it has been modified to ask user for radio parameters which are essential for communication.

Frequency

Radio frequency (RF) is a unit of measurement used to describe how quickly electromagnetic radio waves oscillate between frequencies as high as 300 gigahertz (GHz) and as low as 9 kilohertz (kHz). An RF field can be used for many forms of wireless broadcasting and communications with the use of antennas and transmitter

Callsign

In radio communication, a callsign is a distinctive string of letters and digits that designates a certain radio station or operator. It is a crucial component of radio communication protocols and is used to establish communication between various radio stations.

Callsigns are used in aviation to distinguish between specific aircraft and to aid with air traffic control. Callsigns are a means of identifying and coordinating communication between military units, such as squadrons or platoons, during military operations.

Callsigns are used in amateur radio transmission to pinpoint certain operators' whereabouts. Radio operators are given callsigns in accordance with national regulations, and many choose for memorable, personalised callsigns.

Squelch

Squelch is a radio communication characteristic that mutes or suppresses a receiver's audio output when there isn't a strong signal. It is made to improve audio clarity when a strong signal is received and to reduce noise or static that can be heard when a weak signal is obtained.

Setting a threshold level below which the receiver's audio output is muffled or suppressed is how the squelch function operates. The audio output is permitted to pass through the receiver and can be heard by the user when a signal is received that is powerful enough to surpass the threshold level.

In two-way radio transmission, squelch is especially helpful since it reduces unneeded background noise.

GUI

Hardware

The hardware of the project is comprised of a radio handset, a telephone, and a headset.

All the hardware sets have same programming functionality. The main components of each set include:

1. Microcontroller (Arduino micro)
2. Microphones
3. Bread board
4. Resistor
5. Jumper wires
6. Connectors
7. Push button
8. Cable

The microcontroller in the hardware sets is programmed such, on the click of push button, it will operate as left mouse click. The push button is then directly related to the main software python code of the project where we have used pyAudio libraries for audio interface of the project.

By pressing the push button, the user voice will be transferred to the other corresponding system or unit where the code is running through the microphones used in each hardware set.

The connectors used in all three sets are functioned to connect the set with the system and has two wires with a connector ahead, one for connecting microphone and other for speaker.

Resistor is connected with Arduino and the push button through wires to avoid the overflow or circuit break.

The speaker and microphones in each set are connected to the system through connectors, once the set is connected with the system and software code is running, we can easily transmit our voice to the other system by pressing the push button and after transmission the button is released by the user.

A cable is attached with the Arduino micro and connected to the system. Once the cable is attached with the system the Arduino will start its function, i.e., will perform the left mouse click operation each time the button will be pressed and hence the audio will be transmitted

Chapter 4

Software Requirement Engineering

4.1 Introduction

The Half Duplex Radio Net Emulator for Voice Procedure Lab is an innovative software project that aims to create a virtual platform for conducting voice communication experiments in a controlled environment. This project comes under the domain of software requirement engineering, which deals with the systematic approach of identifying, documenting, validating, and managing the requirements of software systems.

The project's main objective is to develop a software system that can simulate radio communication scenarios, enabling researchers to perform voice communication experiments using a variety of protocols, algorithms, and models. The system will enable users to emulate different network configurations and scenarios, including point-to-point (P2P), point-to-multipoint, and multipoint-to-multipoint communication.

The software system will be based on the concept of half-duplex communication, where only one user can transmit at a time, while others must wait for their turn. This approach is commonly used in radio communication systems, such as walkie-talkies and two-way radios, where users must take turns to speak and listen. The project will utilize a P2P approach, which is a communication model where two users are connected to each other to communicate. This approach is particularly suitable for voice communication experiments where only two users are involved.

The Half Duplex Radio Net Emulator for Voice Procedure Lab project will also include the development of a user-friendly graphical user interface (GUI) that will allow users to set up and configure different scenarios easily. The GUI will provide users with real-time feedback and status updates, enabling them to monitor and analyse the performance of the communication system.

The project's potential applications are extensive, ranging from academic research to military and emergency services training. The project's expected outcome is a comprehensive platform that will provide researchers with a comprehensive environment for conducting voice communication experiments, contributing to the advancement of communication technology.

4.1 Purpose

The project is being initiated at the Signal Training Centre (STC) in Kohat to provide a comprehensive platform for the training and evaluation of military personnel on voice communication procedures. The project aims to develop a software system that can simulate radio communication scenarios, enabling soldiers to perform voice communication experiments using a variety of protocols, algorithms, and models.

The purpose of this project is to enhance the capabilities of the STC in providing advanced training to military personnel on radio communication procedures. The STC is responsible for providing training on various communication equipment and systems to the Pakistan Army. The proposed system will provide a cost-effective and efficient alternative to traditional training methods, which often require expensive hardware and live field exercises.

The software system will be based on the concept of half-duplex communication, which is commonly used in military radio communication systems. This approach will enable trainees to practice communication protocols in a safe and controlled environment, enabling them to gain valuable experience and expertise before being deployed in the field.

The project's expected outcome is a comprehensive platform that will enable military personnel to practice various communication procedures and protocols, including call signs, signal phrases, and message formatting. The system will enable trainers to evaluate the trainees' performance, identify areas for improvement, and provide personalized feedback to enhance their communication skills.

Furthermore, the project will provide an opportunity for the STC to collaborate with academic institutions and research organizations, contributing to the advancement of communication technology. The project's potential applications are extensive, ranging from military training to emergency services and disaster management.

Overall, the project's purpose is to provide an advanced and cost-effective training platform for military personnel on voice communication procedures, contributing to the enhancement of their skills and capabilities.

4.2 Overall Description

The project is a software system that will enable users to simulate radio communication scenarios for voice communication experiments. The system will provide a range of functions that will enable users to create and configure various network scenarios, including point-to-point (P2P) communication, point-to-multipoint, and multipoint-to-multipoint communication.

The primary function of the software system is to provide users with a platform to perform voice communication experiments using a variety of protocols, algorithms, and models. The system will enable users to set up different scenarios and evaluate their performance using various parameters, such as signal quality, latency, and packet loss.

The system will also enable users to configure different network topologies and protocols, such as simplex, half-duplex, and full-duplex communication. Users will be able to specify the transmission power, bandwidth, and frequency of the emulated radio devices, enabling them to simulate different real-world scenarios.

Another important function of the system is the graphical user interface (GUI) that will allow users to set up and configure different scenarios easily. The GUI will provide users with real-time feedback and status updates, enabling them to monitor and analyse the performance of the communication system.

The system will also provide the functionality to record and playback voice communications, enabling users to analyse and evaluate their performance. The recorded communication can be analysed in real-time or later using various metrics, such as signal quality, clarity, and intelligibility.

The software system will also provide the functionality to generate and send pre-recorded voice messages to emulate real-world communication scenarios. Users will be able to customize the content, tone, and frequency of the messages, enabling them to create realistic communication scenarios for training and evaluation purposes.

Overall, the product will provide a comprehensive platform for conducting voice communication experiments in a controlled environment. The system will provide a range of functions, including scenario creation, configuration, and evaluation, enabling users to gain valuable experience and expertise in radio communication procedures.

4.3 User Classes and Characteristics

The project is designed to cater to a wide range of users, including military personnel, trainers, and researchers. The users can be broadly categorized into the following classes:

4.3.1 Military Personnel

Military personnel are the primary users of the system, and the system is designed to cater to their specific needs. The military personnel will be able to use the system to practice and evaluate their communication skills using various protocols and scenarios. They will be able to experiment with different network topologies, transmission power, and frequencies to simulate real-world scenarios.

The military personnel will be the primary users of the system, and the system is designed to cater to their specific needs. The system must be easy to use and navigate, as military personnel may have limited technical knowledge. They require a reliable and stable system that can simulate different scenarios and provide real-time feedback on their performance.

4.3.2 Trainers

Trainers are another important class of users who will be able to use the system to evaluate the trainees' performance and provide feedback. Trainers will be able to create and configure different scenarios and parameters to evaluate the trainees' communication skills. They will be able to monitor and analyse the trainees' performance in real-time, providing personalized feedback to enhance their communication skills.

Trainers require a system that is easy to use and navigate, enabling them to create and configure different scenarios quickly. They require a system that can monitor and analyse the trainees' performance in real-time, providing personalized feedback to enhance their communication skills.

4.3.3 Researchers

Researchers are another class of users who will be able to use the system to conduct research and experiments on communication protocols and algorithms. The system will provide them with a comprehensive platform to simulate different scenarios and evaluate their performance using various metrics.

Researchers require a system that is flexible and can support various communication protocols and algorithms. They require a system that can simulate different scenarios and evaluate their performance using various metrics, such as signal quality, latency, and packet loss.

4.4 Operating Environment

4.4.1 Hardware

Half Duplex Radio Net Emulator is composed of following hardware.

- Microcontroller
- Audio codec
- Radio module
- Power supply
- UI components

4.4.2 Software

Half Duplex Radio Net Emulator will have the following software specifications:

- Python
- Socket Programming
- Threading
- Arduino

Python is used for coding using some libraries including “PyAudio and PyInput”.

4.5 Design and implementation Constraints

The design and implementation of a half-duplex radio net emulator for a voice procedure lab may be subject to several constraints. Here are some possible examples:

1. Equipment limits: The gadget might have equipment limits that influence its plan and execution, for example, restricted handling power, memory, or capacity.

2. Cost requirements: The device's selection of components and features may be influenced by the need to design and implement it within a certain budget.

3. Both portability and size: The device may need to be small and portable to make it easy to transport and set up, depending on its intended use.

4. Requirements for power: The device's selection of components and features may be affected by the device's need to run on battery power or a limited power supply.

5. Compatibility: It may be necessary for the device to be compatible with the training environment's existing communication equipment and protocols.

6. Security guidelines: Electrical and radio equipment-related safety regulations and standards may need to be met by the device.

7. Interface for use: It's possible that the device needs to have an intuitive user interface that makes it simple to set up and use.

8. Robustness: In a training environment, the device may need to be designed and implemented to withstand daily use.

9. Controlling the interference: To reduce interference from other radio devices and sources of electromagnetic noise, the device may need to include features.

10. Execution and unwavering quality: The device should be made and used in a way that minimizes downtime and errors in the training environment while performing well and reliably.

4.6 User Documentation

The users will be given a user handbook with following instructions on how to run Half Duplex Radio Emulator.

A user manual is a thorough manual that describes the device's operation, features, and functions.

Guide to Getting Started: A short tutorial that explains how to set up and use the gadget step by step.

Technical specs: A document outlining the device's technical specifications, such as its hardware components, power needs, and radio network characteristics.

Safety and compliance information: A document that contains instructions for safety, regulatory compliance information, and any other important legal or safety information.

A troubleshooting guide is a document that gives troubleshooting hints and guidance for typical problems that users may face.

Frequently Asked Questions (FAQs): A document that answers typical questions regarding the device that users may have, such as how to utilise certain capabilities or how to handle specific difficulties.

A glossary is a document that defines technical words and jargon used in documentation.

Contact information: A document including contact information for the manufacturer or support team, as well as any additional resources or references that users may find beneficial.

4.7 Assumptions and Dependencies

Assumptions:

- The device assumes that users have a basic understanding of voice procedures and radio communications.

- The device assumes that users have access to a radio network to transmit and receive messages.

- The device assumes that users have a basic understanding of electronics and are capable of assembling and operating the device safely.

Dependencies:

- Constant electricity supply.

- Audio codec to convert analogue signals to digital signals and vice versa.

- Radio module to transmit and receive radio signals over the network.

- User-interface components such as buttons, switches etc to operate.

4.8 External Interface Requirements

4.8.1 User Interfaces

Display: To display relevant data like the device's current status, battery level, and signal strength, a display may be required.

Control buttons: To initiate transmissions, adjust volume, change channels, and perform other actions, the device may require control buttons.

LED indicators: The gadget might require Drove pointers to give visual input about the gadget's status, like power, transmission, gathering, and battery level.

Output and input of audio: To enable users to communicate over the radio network, the device may require audio input and output components, such as a microphone and speaker.

Control of volume: The gadget might require a volume control handle or fasten to change the volume of the sound result.

Chooser of a channel: The gadget might require a channel selector switch or fasten to permit clients to switch between various radio channels.

Button for power: To turn the device on or off, it may require a power button.

Ergonomics: To ensure efficient and comfortable use, the device's user interface should be designed with ergonomic principles in mind.

Simple design: The gadget's UI ought to be intended to be natural and simple to use, with clear marks and guidelines.

Compatibility: The gadget's UI ought to be viable with a great many clients, incorporating those with various degrees of specialized mastery, actual capacities, and language foundations.

4.8.2 Hardware Interfaces

- Power supply
- Audio input and output
- Radio module
- Microcontroller
- Audio codec

- Connectors

4.9 Communication Interfaces

1. Radio interface: The device must have a radio interface that allows it to transmit and receive radio signals over the network. This interface should support the relevant radio frequency, modulation, and encoding standards used by the network.

2. Audio interface: The device must have an audio interface that allows it to receive and transmit audio signals. This interface should support the relevant audio encoding and decoding standards used by the network.

3. Ethernet interface: The device may have an Ethernet interface that allows it to connect to a wired network for data transfer or remote control.

4. Wireless interface: The device may have a wireless interface, such as Wi-Fi or Bluetooth, that allows it to connect to other devices or components for data transfer or remote control.

5. Monitoring interface: The device may have a monitoring interface that allows users to monitor its status and performance, such as LED indicators, a display, or a web-based dashboard.

4.10 System Features

This is a comprehensive software system designed to simulate radio communication networks and enable voice communication experiments in a controlled environment. The system provides an extensive range of features that enable users to simulate various real-world communication scenarios and enhance their communication skills. The system features include:

4.10.1 Peer-to-Peer (P2P) Communication

The system uses a P2P communication approach, which enables users to communicate directly with each other without the need for a central server. This approach ensures low latency, reduces network congestion, and enhances communication quality. The P2P approach also makes the system highly scalable, enabling users to add or remove nodes from the network as required. The system supports various P2P communication modes, including ad-hoc, infrastructure, and hybrid modes, to cater to different communication scenario

4.10.2 Simulation of Radio Networks

The system can simulate various radio network topologies, including star, mesh, and ring topologies, to simulate real-world communication scenarios. The system can also simulate different transmission powers, frequencies, and modulation schemes to provide a realistic simulation of radio networks. The system also supports multi-radio interfaces, enabling users to simulate different radio networks simultaneously. The system's simulation capabilities make it an ideal platform for research and experimentation in the field of wireless communication.

4.10.3 Voice Communication

The system supports voice communication using a variety of protocols, including simplex, half-duplex, and full-duplex modes. The system can also simulate various noise and interference conditions to provide a realistic simulation of real-world communication scenarios. The system supports various audio codecs, including G.711, G.729, and GSM, to ensure high-quality voice communication. The system also supports voice encryption, ensuring that communication remains secure and confidential.

4.10.4 Real-time Monitoring and Analysis

The system provides real-time monitoring and analysis of the communication traffic, enabling users to monitor and analyse various metrics, including signal quality, latency, and packet loss. The system also provides real-time feedback and analysis to enhance users' communication skills. The system supports various monitoring and analysis tools, including packet sniffers, network analysers, and spectrum analysers. The system also supports visualization tools, enabling users to visualize various communication metrics in real-time.

4.10.5 User Management

The system provides user management capabilities, allowing administrators to create and manage user accounts and access levels. The system also provides role-based access control, ensuring that users only have access to the features and functions they require. The system also supports various authentication and authorization mechanisms, including LDAP, RADIUS, and TACACS+.

4.10.6 Scalability

The system is highly scalable, enabling users to add or remove nodes from the network as required. The system can also support many users, enabling multiple users to communicate simultaneously. The system's scalability makes it an ideal platform for large-scale communication experiments and simulations.

4.10.7 Cross-Platform Compatibility

The system is designed to be cross-platform compatible, enabling users to run the system on various operating systems, including Windows, Linux, and macOS. The system supports various programming languages, including C, C++, Python, and Java, enabling users to develop custom applications and scripts.

4.11Non-functional Requirements

In addition to the functional requirements, the Half-Duplex Radio Net Emulator for Voice Procedure Lab has a set of non-functional requirements that ensure the system's quality and performance. The non-functional requirements include:

4.11.1 Usability

The system must be user-friendly and easy to use. The system's user interface must be intuitive and straightforward, enabling users to navigate the system with ease. The system must also support various input methods, including keyboard and mouse, touch screen, and voice commands. The system must also provide users with sufficient documentation and training materials to enable them to use the system effectively.

4.11.2 Performance

The system must be fast and responsive, enabling users to communicate in real-time. The system must support low latency and high throughput, ensuring that communication is smooth and uninterrupted. The system must also be scalable, enabling it to handle a large number of users and communication traffic.

4.11.3 Security

The system must be secure, ensuring that communication remains confidential and protected from unauthorized access. The system must support various encryption and authentication mechanisms, including SSL/TLS, AES, and RSA. The system must also provide access control, enabling administrators to restrict access to sensitive data and functions.

4.11.4 Reliability

The system must be reliable, ensuring that communication remains stable and consistent. The system must be available 24/7, ensuring that users can access the system whenever they need it. The system must also be fault-tolerant, ensuring that communication remains uninterrupted in the event of a system failure or error.

4.11.5 Maintainability

The system must be maintainable, enabling administrators to update and modify the system as required. The system must be modular and extensible, enabling administrators to add or remove features and functions easily. The system must also be well-documented, enabling administrators to understand the system's structure and functionality.

4.11.6 Compatibility

The system must be compatible with various hardware and software systems, enabling users to integrate the system with other tools and platforms. The system must also support various communication protocols, ensuring that users can communicate with different devices and networks.

4.11.7 Scalability

The system must be scalable, enabling administrators to add or remove nodes from the network as required. The system must also be able to handle many users and communication traffic, ensuring that communication remains smooth and uninterrupt

Chapter 5

Analysis

A half-duplex radio net emulator for a speech process lab is a difficult project that involves careful consideration of the requirements, design, implementation, restrictions, user experience, and regulatory compliance. It is critical to specify the precise features and functions that the emulator should provide and to build a system architecture that matches those requirements while keeping money, time, and resources in mind. Best practises for hardware and software development should be followed, as well as rigorous testing and quality assurance procedures. The emulator should be easy to use, accessible, and compatible with all applicable norms and standards. Overall, proper preparation and execution may result in a helpful tool for radio communication training and testing.

Chapter 6

Future Work

The Half Duplex Radio Net Emulator for Voice Procedure Lab is an innovative system that provides a comprehensive and reliable solution for communication experiments. However, there is always room for improvement, and the system can be further enhanced to meet the evolving needs of the users. In this section, we will discuss some of the future work that can be done to improve the Half Duplex Radio Net Emulator for Voice Procedure Lab.

6.1 Integration with other communication protocols

The system currently supports a limited set of communication protocols. In the future, the system can be enhanced to support more communication protocols, such as ZigBee, Wi-Fi, and Bluetooth. This would enable the system to communicate with a wider range of devices and networks, making it more versatile and useful.

6.2 Addition of advanced features

The system can be enhanced by adding advanced features such as noise cancellation, voice recognition, and speech-to-text conversion. These features would make the system more effective in communication experiments, enabling users to focus on their research rather than dealing with technical issues.

6.3 Integration with simulation software

The system can be integrated with simulation software such as MATLAB, allowing users to simulate various communication scenarios before conducting experiments in the lab. This would enable users to test their theories and ideas before conducting experiments, saving time and resources.

6.4 Development of a mobile application

A mobile application can be developed for the Half Duplex Radio Net Emulator for Voice Procedure Lab. This would enable users to access the system from their mobile devices, providing them with more flexibility and convenience. The mobile application can also be integrated with other mobile applications such as messaging and voice call applications, enabling users to communicate with each other more easily.

6.5 Expansion of hardware support

The system can be expanded to support more hardware devices such as sensors, actuators, and robots. This would enable users to conduct experiments in various fields such as robotics, automation, and IoT. The system can also be integrated with various hardware platforms such as Raspberry Pi and Arduino, providing users with more options for hardware development.

6.6 Integration with cloud services

The system can be integrated with cloud services such as AWS and Azure, enabling users to store and analyse communication data in the cloud. This would enable users to access their data from anywhere, providing them with more flexibility and convenience. The system can also be integrated with various cloud-based applications such as data analysis and machine learning, enabling users to extract insights from their data.

Chapter 7

Conclusion

The project named 'Half Duplex Radio Net Emulator for Voice Procedure Lab' is a comprehensive and reliable system that provides an effective solution for communication experiments. The system is designed to emulate a half-duplex radio network, allowing users to conduct experiments on various communication scenarios. The system provides various features such as voice transmission, reception, and playback, making it an ideal solution for voice procedure experiments.

The project was initiated by the Signal Training Centre (STC) Kohat, which identified the need for a comprehensive communication experiment system that can be used for training purposes. The Half Duplex Radio Net Emulator for Voice Procedure Lab was developed to meet this need, providing a reliable and effective solution for communication experiments.

The project started with requirements gathering, followed by the design, development, and testing of the system. The development process was guided by the software requirement engineering process, which ensured that the system meets the needs and requirements of the users.

The system is developed using Java programming language, making it platform-independent and easily deployable on various operating systems. The system is also developed using a peer-to-peer (P2P) approach, enabling users to communicate with each other without the need for a centralized server. The system can be easily customized and extended to support various communication scenarios, making it an ideal solution for various research and development projects.

The Half Duplex Radio Net Emulator for Voice Procedure Lab provides various benefits to the users. The system provides a reliable and effective solution for communication experiments, enabling users to conduct experiments in a controlled environment. The system is also user-friendly, with a simple and intuitive user interface, enabling users to easily operate and control the system.

The system can be used by various user classes, such as students, researchers, and professionals, who are interested in communication experiments. The system can also be used by various organizations, such as military and emergency services, for training purposes.

In conclusion, the Half-Duplex Radio Net Emulator for Voice Procedure Lab is a comprehensive and reliable system that provides an effective solution for communication experiments. The system is developed using a P2P approach, making it easily deployable and customizable. The system is user-friendly and can be used by various user classes and organizations. The system provides various benefits to the users, enabling them to conduct experiments in a controlled environment. The Half Duplex Radio Net Emulator for Voice Procedure Lab is a successful project that meets the needs and requirements of the users, providing a reliable and effective solution for communication experiment

Python Code:

```
import tkinter as tk # classic

import tkinter.ttk as ttk #Themed look and feel

from tkinter import *

from tkinter import messagebox as tkMessageBox

from tkinter import scrolledtext

from threading import Thread

from datetime import date, datetime

from http import client

import socket

import threading

import pyaudio
```

```
from pynput import keyboard

from pynput.keyboard import Key, Listener

import time

from datetime import datetime, timedelta

col= [0,10,175, 250]

row= [0,20,70 , 90, 130,   170, 190,   230, 250,   290,310,    350,370,    410,430, 470,510, 550,570]

keeprunning = True

PTT = False   # Press to Talk Key

datachunk = 204

def button1_command():

    print('Button 1 Clicked')

    ipconnect()

    start_keysense()

def button2_command():

    print('Button 2 Clicked')

global window

def main_gui_struct():

    global window

    window = tk.Tk()

    window.title('WALKI-TALKI RADIO')
```

```
window.geometry('365x550+900+50')          # 10,10 means location where it will appear in computer screen
window.resizable(False, False)
window.attributes('-topmost', 'true')
################## FRAME ##################
border_effects = {
    "flat": tk.FLAT,
    "sunken": tk.SUNKEN,
    "raised": tk.RAISED,
    "groove": tk.GROOVE,
    "ridge": tk.RIDGE,
}

frame = tk.Frame(master=window, relief=tk.RAISED, width=365, height=550)
frame.pack()

global prt
global ipadd
global freq
global nis
```

```
    global csign
    prt=tk.StringVar()
    ipadds=tk.StringVar()
    freq=tk.StringVar()
    nis=tk.StringVar()
    csign=tk.StringVar(
    wdth = 15
    lwidth=10
# IMAGE BAKCGROUND
    img= PhotoImage(file=("rr.png"))
    label0=Label(window,image=img, width=350, height=350)#, bg='white', relief=SUNKEN)
    label0.photo = img
    label0.place(x=col[0], y=row[4]+20)#,height=300)
    # Separator object
    separator = ttk.Separator(window, orient='horizontal')
    separator.place(relx=0, rely=0.1, relwidth=1, relheight=0.0)
    #separator.place(x=10, y=row[4]+10)
    # LABELs
    label1=Label(window,text="Port No",width=wdth,font=("arial",12,"bold"))
    label1.place(x=col[1], y=row[0],height=20)
```

```
label2=Label(window,text="IP Adress",width=wdth,font=("arial",12,"bold"))

label2.place(x=col[2], y=row[0],height=20)

label3=Label(window,text="NIS",width=lwidth,font=("arial",12,"bold"))

label3.place(x=col[1]-5, y=row[2],height=20

label4=Label(window,text="Frequency",width=lwidth,font=("arial",12,"bold"))

label4.place(x=col[2]-55, y=row[2],height=20

label5=Label(window,text="Call Sign",width=lwidth,font=("arial",12,"bold"))

label5.place(x=col[2]+70, y=row[2],height=20)

# ENTRIES

entry1=tk.Entry(window,textvariable = prt,width=wdth,font=("arial",12))

entry1.place(x=col[1], y=row[1],height=20)

entry1.insert(tk.END,'999')

global entry2

entry2=tk.Entry(window,textvariable = ipadds, width=wdth,font=("arial",12))

entry2.place(x=col[2], y=row[1],height=20)

entry2.insert(tk.END,'192.168.154.1')

entry3=tk.Entry(window,textvariable = nis, width=8,font=("arial",12))

entry3.place(x=col[1]+10, y=row[3],height=20)

entry3.delete(0, END)

entry3.insert(tk.END,'AB')
```

```
entry4=tk.Entry(window,textvariable = freq, width=8,font=("arial",12))

entry4.place(x=col[2]-40, y=row[3],height=20)

entry4.delete(0, END)

entry4.insert(tk.END,'175.125')

entry5=tk.Entry(window,textvariable = csign, width=8,font=("arial",12))

entry5.place(x=col[2]+85, y=row[3],height=20)

entry5.delete(0, END)

entry5.insert(tk.END,'100')

entry10 = scrolledtext.ScrolledText(master=frame,width=55, fg="black", font=('Arial', 8, 'bold'))

entry10.place(x=col[1], y=row[16],height=30)

entry10.insert(tk.END,'CODE START')

# BUTTON1

button1 = tk.Button(text="CONNECT ON", font=('Arial',8, 'bold'), width=12,height=1, command=button1_command, bg="dark gray",fg="white")

button1.place(x=col[1],y=row[4])

button2 = tk.Button(text="QUIT", font=('Arial',8, 'bold'), width=12,height=1, command=button2_command, bg="dark gray",fg="white")

button2.place(x=col[2],y=row[4]

#label0=Label(window,image=img, width=120, height=220)#, bg='white', relief=SUNKEN)

#sub_btn=tk.Button(window,text = 'Connect', command = ipconnect)
```

```
    #sub_btn.place(x=360,y=293)

PA = pyaudio.PyAudio()

chunk_size = datachunk

audio_format = pyaudio.paInt16

channels = 1

rate = 20000

playing_stream = PA.open(format=audio_format, channels=channels, rate=rate, output=True, frames_per_buffer=chunk_size)

recording_stream = PA.open(format=audio_format, channels=channels, rate=rate, input=True, frames_per_buffer=chunk_size)

def rx_radio():

    global PTT

    while True:

        print('RX',PTT)

        if not PTT:

            try:

                data = MYSOCK.recv(datachunk)

                playing_stream.write(data)

            except:

                pass
```

```
def tx_radio():
    global PTT
    while True:
        print('TX',PTT)
        if PTT:
            data = recording_stream.read(datachunk, exception_on_overflow = False)
            MYSOCK.sendall(data)
MYSOCK = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
MYSOCK.bind(('', 8000))
#MYSOCK.setblocking(False)
def ipconnect():
    global prt
    global ipadds
    global port,ip,e
    global y
    target_port= int(prt.get())
    target_ip  = ipadds.get()
    net_nis  = nis.get()
    net_csign  = csign.get()
    net_freq  = freq.get()
```

```
    MYSOCK.connect((target_ip, target_port))

    # SEND USER DATA to SERVER

    # Send user data to server

    # put code here #

    #MYSOCK.settimeout(0.5)

    print("CONNECTED")

    global rx

    rx = Thread(target=rx_radio,daemon=True)

    rx.start()

    tx = Thread(target=tx_radio,daemon=True)

    tx.start()

def press():

    global PTT

    PTT=True

    #print('PRESS'

def release():

    global PTT

    PTT=False

    print('RELEASE')

def start_keysense():
```

```
    global rx
    import keyboard
    ###################
    keyboard.on_press_key(" ", lambda _:press())
    keyboard.on_release_key(" ", lambda _:release())
#################
main_gui_struct()
window.mainloop()
```

Arduino Code

```
#include <Mouse.h>
const int buttonPin = 2;
const int ledPin =  7;
void setup() {
  Mouse.begin();
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
  pinMode(2, INPUT);
}
void loop() {
  //if the button is pressed, send a left mouse click
```

```
  if (digitalRead(2) == LOW) {
    Mouse.click();
    delay(0);
  }
}
```

References

1. [1] Dong Yu and Li Deng. Automatic Speech Recognition. Springer, 2016.

2. [2] Zhong Xin Bai and Xiao-Lei Zhang. Speaker recognition based on deep learning: An overview. Neural Networks, 2021.

3. [3] Vassil Panayiotou, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP), pages 5206–5210. IEEE, 2015.

4. [4] Dong Wang and Xuewei Zhang. Thchs-30: A free chinese speech corpus. arXiv preprint arXiv:1512.01882, 2015.

5. [5] Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng. Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline. In 2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA), pages 1–5. IEEE, 2017.

6. [6] Jiayu Du, Xingyu Na, Xuechen Liu, and Hui Bu. Aishell-2: Transforming mandarin asr research into industrial scale. arXiv preprint arXiv:1808.10583, 2018.

7. [7] Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, et al. Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio. arXiv preprint arXiv:2106.06909, 2021.

8. [8] Daniel Galvez, Greg Diamos, Juan Manuel Ciro Torres, Keith Achorn, Anjali Gopi, David Kanter, Max Lam, Mark Mazumder, and Vijay Janapa Reddi. The people's speech: A large-scale diverse english speech recognition dataset for commercial usage. In Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1), 2021. 1031

9. [9] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. arXiv preprint arXiv:1706.08612, 2017.

10. [10] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. arXiv preprint arXiv:1806.05622, 2018.

11. [11] Yue Fan, JW Kang, LT Li, KC Li, HL Chen, ST Cheng, PY Zhang, ZY Zhou, YQ Cai, and Dong Wang. Cn-celeb: a challenging chinese speaker recognition dataset. In ICASSP 2020- 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 7604–7608. IEEE, 2020.

12. [12] Yi Liu, Pascale Fung, Yongsheng Yang, Christopher Cieri, Shudong Huang, and David Graff. Hkust/mts: A very large scale mandarin telephone speech corpus. In International Symposium on Chinese Spoken Language Processing, pages 724–735. Springer, 2006

13. [13] Tingwei Guo, Cheng Wen, Dongwei Jiang, Ne Luo, Ruixiong Zhang, Shuaijiang Zhao, Wubo Li, Cheng Gong, Wei Zou, Kun Han, et al. Didispeech: A large scale

mandarin speech corpus. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6968–6972. IEEE, 2021.

14. [14] Chinese Academy of Social Sciences and City University of Hong Kong. Language Atlas of China. The Commercial Press, 2012.

15. [15] Jeffrey Weng. What is mandarin? the social project of language standardization in early republican china. The Journal of Asian Studies, 77(3):611–633, 2018.

16. [16] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In IEEE 2011 workshop on automatic speech recognition and understanding, number CONF. IEEE Signal Processing Society, 2011.

17. [17] Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan. CLUE: A Chinese language understanding evaluation benchmark. In Proceedings of the 28th International Conference on Computational Linguistics, pages 4762–4772, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

18. [18] Seyed Omid Sadjadi, Timothee Kheyrkhah, Audrey Tong, Craig S Greenberg, Douglas A Reynolds, Elliot Singer, Lisa P Mason, Jaime Hernandez-Cordero, et al. The 2017 nist language recognition evaluation. In Odyssey, pages 82–89, 2018.

19. [19] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. Journal of machine learning research, 9(11), 2008.

20. [20] Shengkui Zhao, Hao Wang, Trung Hieu Nguyen, and Bin Ma. Towards natural and controllable cross-lingual voice conversion based on neural tts model and phonetic posteriorgram. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5969–5973. IEEE, 2021.