UNIVERSITY OF NAIROBI

SCHOOL OF COMPUTING AND INFORMATICS

A design of an IP Voice Communication System for Air Traffic Control by use of Open Source software

BY

Nixon Baraza

P53/66087/2013

SUPERVISOR

Dr. L. Muchemi

7th September 2017

A project report submitted in partial fulfillment for the requirements of Master of Science Degree in Distributed Computing Technology.

DECLARATION

This research project is my original work and has not been presented for a degree or masters in any other University.

Signature ………………………………… Date ………...................……………..

NAME: Nixon Wandera Baraza

REG NO: P53/66087/2013

This project has been submitted with my approval as the university supervisor department of ……………………..

Signature ………………………………… Date ……………………...………………

NAME: Dr. L. Muchemi

School of Computing and informatics

ACKNOWLEDGEMENT

I thank all for supporting me to the completion of this project in one way or another. First, I thank God for protection and ability to do the project work.

I'm very much grateful to Dr. L. Muchemi for accepting my request to be my supervisor and for the guidance he has given me throughout the project work.

My sincere thanks also goes to my employer the Kenya Civil Aviation Authority for allowing me to use materials at our Kenya Airports

I am also so thankful to my classmates whose challenges and productive critics provided new ideas to work.

Last but not least I'm profoundly grateful to my family who encouraged me thought the research work.

ABSTRACT

The aim of this paper is to suggest a design of an IP based Voice Communication System for Air Traffic Control by use of Open source software. With the advent of new VOIP standard for Air Traffic Control, it has resulted in a research opportunity. This research opportunity is driven by the fact that unlike the old technology we have legacy open source IP telephony software which can be modified through research to provide necessary VOIP functionalities for Air Traffic Control voice. Also, this research opportunity is being driven by the fact that Telecommunication service providers are replacing analog/Time division multiplexing communication links with IP based communication links. Most Voice communication system manufacturers have ceased investment and production of the old voice communication system, and therefore the cost of spares are bound to increase. A subset of the requirements of ED 137 B standard was used in the design of a prototype.

Design Research as a strategy was employed in this project because it is concerned with design or improvement of artifacts. The selection of this strategy was informed by the fact it is linked to Design Science Research DSR paradigm which underpins the philosophy of this project. The general methodology for design research by Vaishnavi and Kuechler (2015) was employed to guide the activities of the project. In the development phase of the general method for design research, we applied software reengineering methodology to develop four algorithms using legacy telephone software as the base software.

The design was subjected to experimental evaluation to check to on functionality as per ED 137 B VOIP standard and usability. The form of experimental evaluation used was a simulation, and with this, we used SIP tester simulation software. The results of the evaluation showed that some functionality of the standard were met. The above proved that using open source software; we can be able to design an IP based Voice Communication Control System for Air Traffic Control.

Table of Contents

List of abbreviations

VOIP - Voice over Internet Protocol.

TDM - Time Division Multiplexing.

PCM - Pulsed Code Modulation.

FAA - Federal Aviation Administration.

EUROCONTROL - European Organization for Air safety of Air Navigation.

ANSP - Air Navigation Service Providers.

ICAO - International Civil Aviation organization.

ITU - International Telecommunication Union.

GNU - General public license.

SIP - Session In-session.

RTP - Real-time Protocol.

ANSPS - European Air Navigation service Providers.

GANP - Global Air Navigation Plan.

IETF - Internet Engineering Task Force.

SDP - Session Description Protocol.

UAC - User agent client.

UAS - User agent server.

URI - Information in the request.

OSS - Open Source Software.

OSI - Open Source Initiative.

DSR - Design Science Research.

RTCP - Real Time transport protocol.

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

1.1 Background

Communication is the firmness of any air traffic services and the Air Navigation Services Providers (ANSP) are responsible for providing reliable communication services to aircrafts to ensure air safety (Himeda *et al*, 2016). Voice communication is one type of the communication performed by air traffic controllers in providing safety to aircraft. To accomplish this function, a system called voice communication system specific to air traffic control is used. This system sits between the air traffic controller and audio devices (radios and telephones) which could in the remote place or within the premise where air traffic control is taking. For a long time, the technology for Air Traffic Control Voice Communication Systems was based on analog and more recently digital in the form of Time Division Multiplexing/Pulsed Code Modulation (TDM/PCM) technologies. The road map for Voice over Internet Protocol (VOIP) was presented in the GNAP which was adopted by the 38th Session of ICAO Assembly (APANPIRG/26, 2015). With the advent IP based voice communication Air Traffic Control are now moving to IP based Voice Communication system for Air Traffic Control. Figures 1.1 and 1.2 below explain the Voice Communication System black box and the components of the traditional voice communication system respectively.

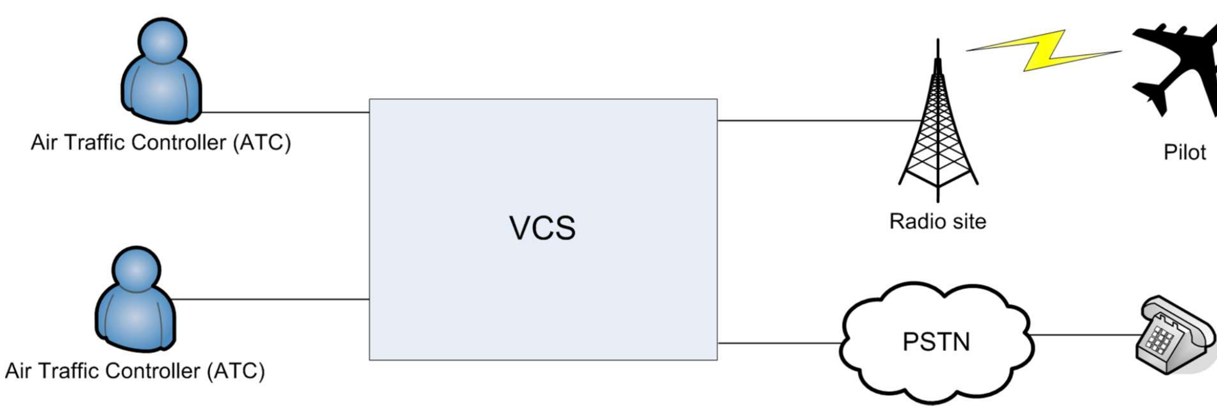

Figure 1.1 Basic block diagram view of the Voice Communication System.

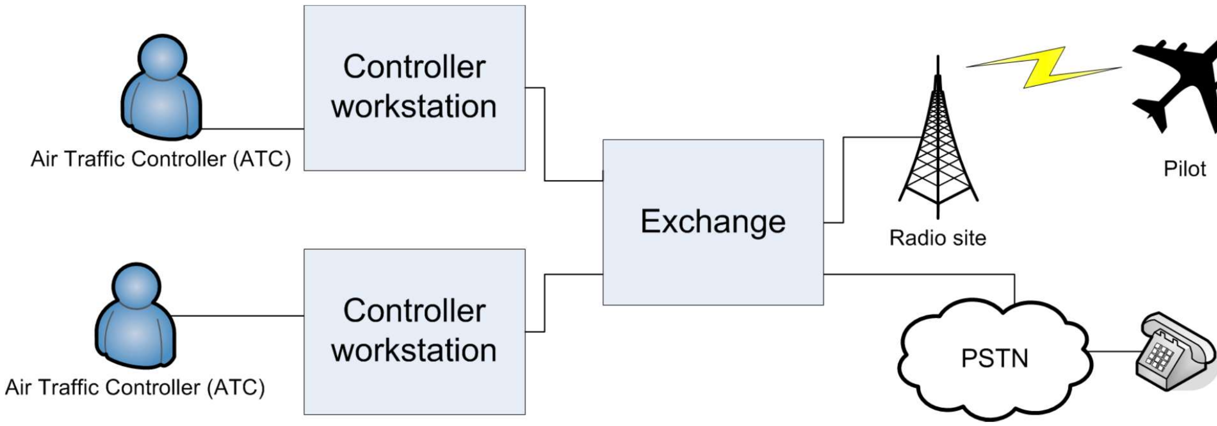

Figure 1. 2 Components of the tradition voice communication system.

The above diagrams show a traditional Voice Communication System that present a very complex structure for the exchange component. With IP technology the exchange part can be made as simple as a Local Area Network LAN switch. This kind of setup removes the most expensive element of the system which is the voice switch and also reduces single point of failure. Apart from these, two key factors are driving this change in this aviation community.

- Telecommunication service providers are now staging out their leased line TDM/analogue services. These leased lines are used to create links for remote Radios or telephone lines to other ATC unit in another airport. This builds a case for the extension of the end equipment (Voice communication system and Radios).

- The European Organization for Air safety of Air Navigation EUROCONTROL now prescribes interoperability in order to increase handling of increased air traffic and to rip benefits that will accumulate from the use of IP Voice communication system. This sets the agenda for the discontinuation of TDM voice switches for air traffic control. Consequently, no Equipment manufacturer would want to continue investing in TDM switches despite the fact that many of the Air navigation service providers ANSP are still using TDM switches. Spares for this Equipment are bound to become expensive

IP communication for Air Traffic Control has the potential of increasing reliability in the sense that TDM systems usually rely on duplicating expensive centralized equipment. IP communication systems migrate core switching intelligence away from the central Equipment to the peripheral equipment thus a breakdown in a single unit does not affect operation in the rest of the system.

Besides that, it costs less in the sense that using an IP infrastructure builds synergies in an organization procurement, operation and maintenance, which then leads to notable savings. IP communications is scalable, so ATC authorities no longer need to invest in large systems right from the beginning as compared with traditional voice communication systems.

The European Organization for Civil Aviation Equipment EUROCAE WG-67, with cooperation from EUROCONTROL, European industry, and ANSPs, developed the first VoIP in Air Traffic Management standard. This standard was released in February 2009 as a set of documents known ED136-138 defining the operational concept, the interoperability and the network- associated requirements. A new version of the standard ED137B was issued in February 2012 after a review which included integration of requirements from the United States Federal Aviation Administration FAA.

According to 2013-2028 ICAO Global air navigation capacity and efficiency plan, migration to VOIP will begin in block 0 with expected completion in 2020.

1.2 Statement problem

For voice communication system of air traffic control to operate, it depends on two key elements. First, there must be a voice switch to link the air traffic controller to the Radios and telephones. Second, for the remote radios and telephones, there must be a telecommunication line either leased from a service provider or privately owned by Air Traffic Control ATC Organization. Because of competition, most Telecommunication service providers have already migrated to the packet switched lines correctly using the IP technology. For a legacy system to be connected to such line, an interface/gateway is required. The requirement for sharing resources using VCS have emerged. Standardization for VOIP in air traffic control has been completed, and Equipment manufacturers have started producing IP based voice communication in line with the recommendation from civil aviation regulators like; ICAO the International Civil Aviation Organization. In summary, the following issues have emerged.

- Standardization organization e.g. International Telecommunication Union ITU, EUROCAE have stopped working on standards for legacy networks.
- Telecommunication service providers who provide telecommunication links to ATC voice communication system have discontinued or will discontinue analog, 64K, and E1 links.

- Legacy VCS systems are not ready for new interoperability requirements for sharing Radios
- Maintenance and upgrade for legacy VCS will become harder in future due to obsolete Technology.

There exists a lot of Open Source software libraries that are used to make VOIP products specifically for telephone systems. These software libraries can be extended by including modules for IP Radio communication and ATC to enable Radio communication. Most of the Equipment manufacturers avoid the use of open source software because they would like to hold on to their code. Companies like Red Hat produces software that powers mission-critical systems (Brad, 2011) yet their software is based on open source software. Therefore, the aim of this project is to take the opportunity of Open Source software line of software to design an IP based Voice Communication System for Air Traffic Control by incorporating an extra module for Radio communication.

1.3 Objectives:

1. Design algorithm by use of open source software to include ED 137B standard for use in Radio communication for Air Traffic Control.
2. Implement algorithm to make a prototype.
3. Simulate input data and use it on the prototype.
4. Use standard evaluation methods to perform an assessment.

1.4 Justification:

Telephone IP libraries are implemented as full duplex while radio communication is full duplex. Research can be done on how the library can be converted to a half-duplex and also on how the headers can be modified to provide signaling required in Radio communication using the open source software.

1.5 Scope:

Air traffic Voice Communication Control system for Air Traffic Control is a complex system and therefore in regard to this, it was not possible because of time to produce a complete system from scratch and based on EUROCAE documents (ED 137 B). Therefore, in this section we define the boundary of the project. The goal is to design an IP based Voice Communication System for Air Traffic by modification of SIP/RTP library where we shall introduce RTP extension header. The project will, therefore, make use of the existing mature tried and tested SIP/RTP libraries. The minimum functionality is to provide two-way communications between the Air Traffic Controller and the pilot.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

In this section, we discuss theoretical background which relates IP voice communication for air traffic control, the standards required, the protocols employed namely the Session In-session SIP protocol and the Real-time Protocol RTP. Others which relate include; Open source SIP/RTP library, the modification needed to Radio communication on SIP/RTP library, Design Science Research DSR, Design evaluation methods and lastly the conceptual framework.

2.2 IP voice communication for Air traffic control

The aim of VoIP Voice over Internet Protocol is to transmit audio data via IP packets. VoIP network combines the technology of the voice networks with that of data network. This union can be likened with the marriage of two different characters, with the hope that through the union some synergy will accrue. Basically there are two types of network voice. First, is the connection-oriented network which is used to construct the Public Telephone Switched Network. In this network, the network sends a tone, and then the user dials the phone number and the destination pickups the receiver, a point to point channel is established via the different switches in the network that are used to establish the required path. When either party hung up, the communication is terminated, and the channel resource is released and made available to be allocated for another communication session. In contrast to the voice networks, data networks e.g. IP are classified as connectionless networks in that data packets have source and destination addresses included in each packet. This packet is sent on to the network that can route the packets via different paths. Since the transmission of packets happens through an IP network, the communications is normally less robust than a circuit switched public telephone network. Using IP then there is no network mechanism that ensures that data packets are not lost and that they are delivered in sequential order. This is termed as a good effort without Quality of Service (QOS) guaranteed. There are several metrics which are available for measuring VoIP call quality. Any VOIP network including those used for Air Traffic Control must ensure that these parameters are met.

2.2.1 VOIP Standardization for Air Traffic Control:

The European Organization for Civil Aviation Equipment EUROCAE Working Group 67 defined standards for IP Voice ATM Systems which go along in guaranteeing a safe and secure voice communication for ATC. The standards are listed below:

1. ED136 VoIP ATM System Operational and Technical Requirements.
2. ED137B Interoperability Standards for VoIP ATM Components – Part 1: Radio Interface.
3. ED137B Interoperability Standards for VoIP ATM Components – Part 2: Telephone Interface.
4. ED137B Interoperability Standards for VoIP ATM Components – Part 3: Recording Interface.
5. ED137B Interoperability Standards for VoIP ATM Components – Part 4: Supervision.
6. ED138 Network Requirements and Performances for VoIP ATM Systems – Part 1: Specification.
7. ED138 "Network Requirements and Performances for VoIP ATM Systems – Part 2: Design Guideline.

The first set of documents were issued in February 2009, and a new version for the interoperability standard ED 137B was released in February 2012 after a review which eventually led the inclusion of requirements from the USA Federal Aviation Administration FAA. The following membership of EUROCAE WG67 was involved in coming up with this standards:

European Air Navigation Service Providers ANSPS.

a. Federal Aviation Administration.
b. VCS and Radio manufacturers.
c. Regulator bodies namely International Civil Aviation Organizations ICAO and the European Organization for safety of air navigation EUROCONTROL.

2.2.2 Aviation System Block Upgrade ASBU.

According to Global Air Navigation Plan GANP VOIP is one of the enablers in link media as shown in figure 2.1. As per plan it was envisaged that migration to VOIP would start form 2013 and continue to block 3 which ends in 2028. Therefore, it means that VOIP in ATC is still in its formative stage.

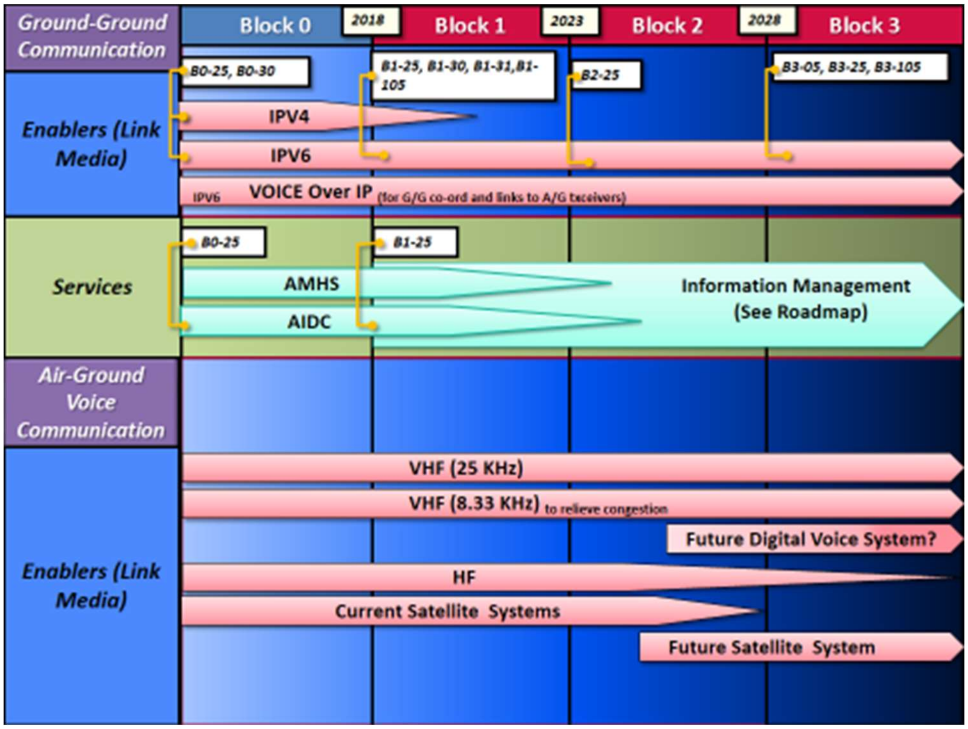

Figure 2. 1 ASBU ground to ground communication road map (Solar, 2014).

2.3 VOIP Protocols.

The standard protocols used in VOIP application are discussed here. The deficiencies associated with this protocols are highlighted together with what needs to be done to address the requirements for IP Radio communition for Air Traffic Control.

2.3.1 Session-Insession Protocol (SIP).

This is an application-layer control protocol which was designed by the Internet Engineering Task Force (IETF). The protocol was designed with ease of implementation and is it's also scalable and flexible.

This specification is available in the form of RFCs, where the most important one is RFC3261 (IETF) which contains the key protocol specification. The function of the protocol is to create, modify, and terminate sessions with one or more participants. Sessions are a set of senders and receivers that communicate, and the state kept in those senders and receivers during the communication (Rosenberg *et al*., 2002). All SIP communications sessions share at least the following three functions :

1. Provision of the signaling between participating aprties to set up a session.
2. SDP Session descrtion protocol to exchange communication parameters.
3. Using suitable protocol SIP applies connvey information in the session for different situations. For example starting a session and putting a call on hold, different protocols would be employed.

2.3.1.1 Sip main components:

The following are the main elements of SIP;

a) UAC User agent client

This is the the client side components that communicates to the User agent server (UAS). UAC client can initiate up to six SIP requests to User Agent Server. The six requestscan be one of the following: INVITE, ACK, OPTIONS, BYE, CANCEL and REGISTER. In making request, UAC SIP component determines information which is essential for the request, and this includes the protocol, the port and the IP address of the Server where the request is will be sent.

b) UAS User agent server

This is the application runs the as Server and is the one that receives the SIP requests from a UAC, responds by sending a response to the to the UAC. From preceding, we note that SIP operates in client/server mode. When say A requests for connection to start communication with B the recipient B becomes the Server, and A becomes the client. The roles keep changing depending on who makes the response and makes the reply. SIP packet consists of SIP header shown in figure 2.2 below and a message which could be a request or a response.

Sip messages: It a question or a response, which can be of different types. The header contains the request or response message, which is usually in clear text and thus can be read by human beinggs. Below is a sample SIP message header.

INVITE sip:nixon@Fenix.org SIP/2.0

Via: SIP/2.0 /UDP pc22.whereelse.org; bbranch=z8LH4bM776asdhds

Max Forwards: 80

To: nixon <sip:nixon@Fenix.org>

From: Mary <sip:jane@Jenix.org>;tag=1523301774

Call-ID: a54b2c76e44710@pc22.Jenix.org

CSeq: 333159 INVITE

Contact: <sip:jane@pc22.Jenix.org>

Content-Type: application/sdp

Content-Length: 130

From the SIP message above the first statement shows an INVITE message type which is areq uest that is used start commuicationt. . The first line also contains the SIP address and the SIP version that is used. The second line contains Via field which indicates the address to the destinationt.The request message makes its way and each node appends its address to the Via field, so as ensure that responses is routed back correctly. SIP transactions are tracked by an identier in the Via field

To-and-From , show identy and locations of the sender and recipient respecively. Call ID field is a sole identifier that ensures unique identication of SIP sessions. CSeq number is the number of requests that have been sent within a a particular session.

Max-Forwards indicates the number of maximum hops that for a message that be tolerated before the message is discrded which would then cause an error messasge to be generated. Contact field contains the location to the originator. Content-Type , specifies the format the of message body and Content-Lenth specifies how long the message is. Figure 2.2 below is shows SIP messages are exchanged by two communicating parties

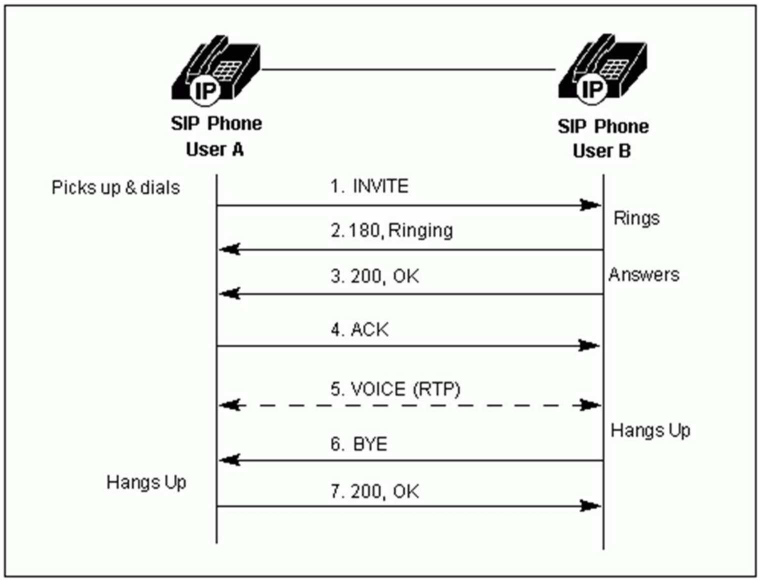

Figure 2. 2 SIP flow call management (Johnston, 2015).

In the figure above, user A starts communication by sending an INVITE request to User B, who whose phone responds by ringing, Finaly when user B pick the handset, a response encoded as OK is sent, which is then acknowledged by user A who sends an ACK encoded reply. After this exchange of SIP messages, the session is now setup, and the original information data/voice is transmitted in full duplex as a stream of RTP packets. To end the session a BYE request is sent, whose is reply is a 200 OK from the response (Darilion et al, *2004*).

Session parameter negotiation using SDP (Session Description Protocol): This is used to convey information about media types and other communication parameters in multimedia sessions to help participants join or gather info of a particular session. This forms an agreement of the agreeable Parameters for both users, that will be complied by both parties during communication.

2.3.2 RTP: Real-time Transport Protocol

The second protocol after SIP is the RTP which is the real-time transport protocol used to carry audio from one end to the other. According to RFC 3550 the functions of RTP are as follows:

- Payload type identification
- Source identification
- Sequence numbering
- Times tamping

RTP packet format

Figure 2.3 below shows the RTP header packet format.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |V=2|P|X|  CC   |M|     PT      |       sequence number         |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                           timestamp                           |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |           synchronization source (SSRC) identifier            |
 +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
 |            contributing source (CSRC) identifiers             |
 |                             ....                              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2. 3 RTP packet format (Schulzrinne *et al*, 2003).

The first 32 bits of the header consists of several control bits.

- The version number (V) SIP which is 2.
- (P) is the padding bit that indicates if there is padding octets inserted at the end of this packet. Padding may be required by some applications with fixed length packet sizes.
- The extension (X) bit indicates if there is an extension after the fixed header.
- The count field (CC) indicates the number of contributing source identifiers (CSRC) following the fixed header
- The marker bit (M) may be used as general marker, for example. Indicating the beginning of a speech burst in communication.
- The payload type (PT) field indicates the payload format.

- The sequence number field contains an incrementing counter which is started by a source from a random number.
- The timestamp corresponds to the generation of time instant of the first octet in the payload.
- The synchronization source identifier (SSRC) is a randomly generated figure that uniquely identifies the source within a session.
- Following the fixed header there are one or more contributing source identifiers which are supplied by the mixer and the payload.

2.4 Radio functionalities.

In this section we describe the necessary functionalities for an IP based VCCS, and later discuss different implementable methods as suggested by (Schulzrinne *et al,* 2003)

2.4.1 Radio Communication Channels.

For Communication Radio, the user requires to be availed with radio channels to choose from. In legacy system, analogue means through the use of analogue interface such as ear and mouth E &M would be used, but in IP communication this requirement is handled by a new set of protocols i.e. SIP and RTP.

2.4.2 Radio signaling.

When in operation once the VCS has established communication session with the Radio, then the VCS and Radio need to signal each other so as to enable transmission and reception of audio. An occurrence of push to-talk (PTT) will signal transmission and while the occurrence of squelch will signal reception. In order to understand how PTT and squelch can be established, following approaches are discussed:

2.4.2.1 SIP event notification

This method, makes use of SIP with an extension. Support for requests related to signaling (PTT/Squelch) events between users (Darilion *et al* 2004). SUBSCRIBE and NOTIFY requests are used to implement PTT and Squelch respectively. A user that wants to use a certain radio frequency SUBSCRIBE to the frequency using and the corresponding Radio sends NOTIFY request to back to the SUBSCRIBER. Because of acknowledgement which is embedded in SIP message this method is reliable. The second advantage is that there is no need to setup RTP session, a host would be interested in PTT and squelch events only and not the audio, the last advantage is that, this method uses only open Internet standards, a disadvantage with this method it increases the number of number of SIP messages.

2.4.2.2 RTP header extension

This method requires an extra header to be added to RTP packets to transmit more header parameters (Schulzrinne *et al*, 2003). This extension is as shown as figure 2.4 shown below:

- Type: a 16 bits' field which is defines by the application that uses the extension,
- Length: a 16-bit field that defines the extension head length,
- Header extension: the appended header extension.

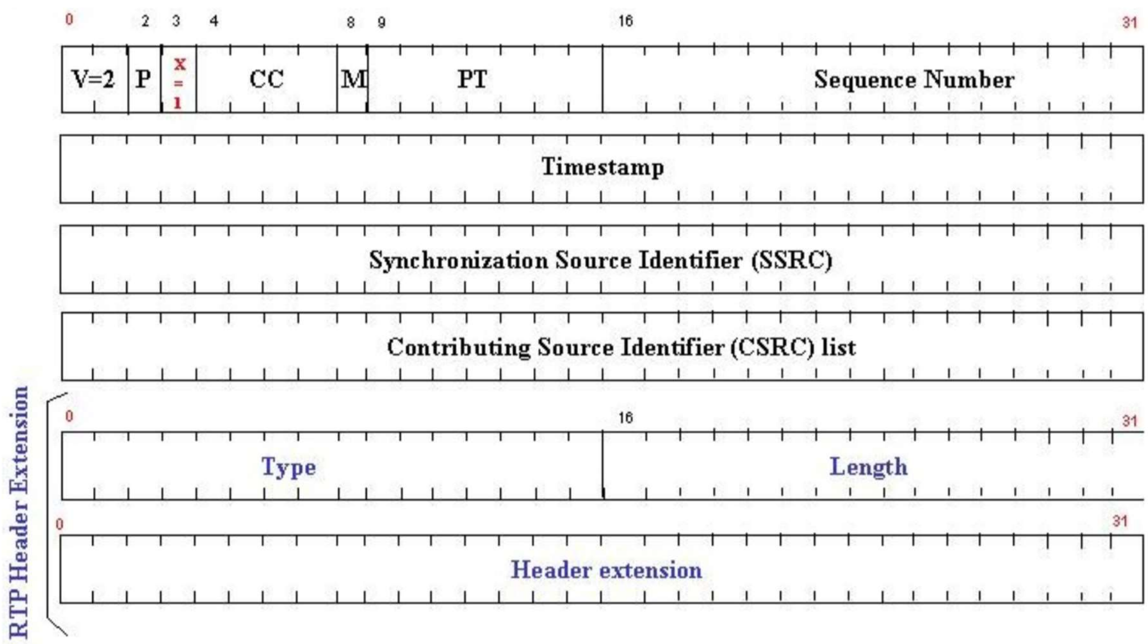

Figure 2. 4 RTP header plus RTP header extension (Schulzrinne *et al,* 2003).

Using a customized header extension, Squelch and PTT can be encoded, which will then convey signaling information. In this method signaling status shall be transmitted together voice packets, and this will ensure presence of signaling information which always be synchronized to the audio stream. When PTT/Squelch is sent together with the RTP packets will avoid creating a new session every time there is a change signaling status. The disadvantage is that a RTP session has to be set up even if audio is not being transmitted or received (Darilion *et al*, 2004).

2.4.2.3 Suppression of audio stream.

For this audio packets are transmitted/received when either Squelch or PTT is active, and muted when neither PTT nor Squelch is not sent. Therefore, this implies that every presence of packets signifies PTT. As compared to the RTP header extension is that this looks to be simpler and the header can be made to be always small (Schulzrinne *et al*, 2003). The disadvantage with this method is it only allows one type of information that is either there are packets of RTP are present or not. With ATC communication being complex signaling some information such as best signal selection and acknowledgement that the links are working are required.

2.4.2.4 RTCP: Real Time transport protocol.

Signaling information can be encoded in Real Time Transport Protocol. No RTP session is needed but there will be a need to modify to suit this situation. However, this method lack acknowledgment and this makes it less appealing than the SIP specific event notification method mentioned earlier, (Schulzrinne *et al*, 2003).

2.4.2.5 The choice of Radio signaling for this project.

The RTP header extension as prescribed by Eurocae but audio with stream suppression was adopted for this project. That is, audio packets will only be transmitted if PTT or squelch has been activated (ED137B). This method is based ED 137 B standard which has been adopted by ICAO

2.5 SIP/RTP open source library.

In this section we discuss implementation of SIP/RTP protocol through open source software. For this project PJSIP an open source library for SIP and RTP was selected for use after satisfying an evaluation criterion. The particular modules that needs to be modified is highlighted

2.5.1 OSS Open source software.

OSS is software that has its source code is open, published and developed by individuals using the Internet to support, collegiate behavior and usually by voluntary effort. In consistence with the principles defined with open source initiative (OSI) the license is usually available at no charge.

General Public License (GNU) This is a license by the Free Software Foundation where GPLv2 and GPLv3 are the two major versions available. In June 2007 GPLv3 was introduced later as a major update to GPLv2. GPL has the following major key points e:

- Programs are used without any restrictions and commercial use is allowed.
- Programs can be distributed for free or for money but the source code must be part of the package.
- Modification of internal architecture to suit different requirements is allowed.
- Modified versions of the source code are distributable under the terms open source software.

Lesser GNU General Public License:

With LGPL a program doesn't need to be GPL licensed. The library is structured in the same way as a standard GPL license but the difference is that it can be made to be both free and proprietary software.

Why Use of Open Source Software

We now endeavor to make the case for use of OSS in our project and why Software with Open Source license is important.

- Reduced information system costs since there is no charge for the code.
- Technical independence from software publishers since one is free to change without consent.

- Integration and interoperability repositioning as a major concept of your information system development strategy.
- The quality of the code is improved if more people can have access to it. As many people interact with the code, it implies that many issues will arise about the code which may result in the revelation of more bugs.
- Publication academic papers can enhance learning if the source code of software written for these papers cane made available to enhance learning or research

2.5.2 SIP/RTP Open source library

This project required the use of SIP/RTP library. SIP/RTP software libraries implement SIP/RTP protocols discussed above. For radio communication an additional header is required to carry radio signaling information. We therefore made an extensive search for an appropriate SIP/RTP library that can be modified for this project. The following evaluation criteria was used to select the:

License:

The basic requirement for design and development for this project was to base all infrastructures on Open Source license. The main idea for this was to reduce cost and for the system to enjoy the benefits that accrue with open software.

Documentation:

Documentation is one of the key thing when starting a project. Project team should waste time because of missing documentation during project implementation. This is important especially when using an OSS SIP library. Unclear or missing documentation is a problem often seen in open source projects (Vaishnavi *et al*, 2013).

Complexity of Integrating:

One of the most important task is to integrate the library within the tool chain so as to deploy it on the embedded board. There are libraries available with embedded systems already in their design which means such libraries are generally easier to port to the destination board. There is situation that require external dependencies to other libraries which must also be ported.

Resource Requirements:

Embedded platforms usually have limited resources. This includes physical memory, disk/flash storage and CPU power. It is important to check resource requirements before settling on a particular library.

Sample Applications:

Sample applications can make use of the library easier, leading to custom applications being derived from same.

Community and user base:

A huge community together with a user base is usually an indicator for the quality of the projects. Open source projects are not subjected to aggressive marketing behind, what counts are satisfied users. Users will not be willing to choose a library that is badly written and barely usable.

COMPARISON OF VARIOUS SIP/RTP LIBRARY

We made a deep and detailed investigation, compared different possibilities of SIP/RTP libraries as shown in the following three tables. PJSIP SIP/RTP an open source software library which was chosen for this project is the reference SIP/RTP library.

Table 2. 1 Comparison of PJSIP vs. .NET C# SIP libraries (Schulzrinne *et al*, 2003).

	PJSIP	Unity SIP .NET SDK	Independentsoft SIP.Net	OZEKI C# SDK
Link	Library: http://www.pjsip.org Wrapper for .NET: (https://sites.google.com/site/sipekvoip/)	http://www.konnetic.com/products/products_sip_sdk_std.aspx	http://www.independentsoft.com/sip/index.html	http://www.voip-sip-sdk.com
Supported platforms	• Windows • Mac • Linux • Windows Mobile (using wrapper) • iOS • Symbian • *Android and Blackberry are scheduled for next release*	• Windows • Linux (using Mono)	• Windows • Linux (using Mono) • Windows Mobile	• Windows • Android • iOS • Windows Mobile
License	GPL 2.0	Commercial	Commercial	Commercial
Price	Free	• 1 developer license - 299 Euro • Unlimited developer license - 699 Euro • Blueprint edition (source code) - 3999 Euro	**Quantity** **Unit price** 1 - 4 200.00 USD 5 - 9 180.00 USD 10 + 160.00 USD	**$ 700-349.000**
Licensing comments	Free usage is allowed if the final software is licensed under GPL 2.0 or later. *(Commercial products are allowed by the license, but the source code is required to be available in public)*.Custom open source license can be requested, with some exceptions.	The entire software base of the library can be sold freely without any limitation. There is a 30-day trial.	No limitations for commercial redistribution of library binaries. Unit means – one developer license. Trial exists and can be extended by request but only for research purposes.	Licensing is based on the quantity of simultaneously processed calls from 1 for 700$ to 349.000$ for 10 000 calls. Different bundles are available. More details at: http://www.voip-sip-sdk.com/p_28-price-list-for-ozeki-voip-sip-sdk-voip.html
Advanced features	• **Mobile platforms support** • Video support • Good documentation and community support		• **Mobile platforms support**	• **Mobile platforms support** • **Facebook calls support** • **Good documentation and customers support** • **Paid training for developers**

Table 2. 2 Comparison of PJSIP vs. Java SIP/RTP libraries (Schulzrinne *et al*, 2003).

	PJSIP	Ceridwen 3M SIP Circulation Library	JAIN-SIP	MjSIP
Link	Library: http://www.pjsip.org Wrapper for Java: (http://sourceforge.net/projects/pjsip-jni/)	http://www.ceridwen.com/selfissue/3m-sip/	http://java.net/projects/jsip/pages/Home	http://www.mjsip.org/
Supported platforms	• Windows • Mac • Linux • Windows Mobile (using wrapper) • iOS • Symbian • *Android and Blackberry are scheduled for next release*	• All Java supported platforms	• All Java supported platforms	• All Java supported platforms
License	GPL 2.0	GPL 3.0	Unknown	Unknown
Price	Free	Free	Unknown	Unknown
Licensing comments	Free usage is allowed if the final software is licensed under GPL 2.0 or later. *(Commercial products are allowed by the license, but the source code is required to be available in public)*. Custom open source license can be requested, with some exceptions.	Ceridwen Self Issue Client is available as open source under GPLv3. Contact development@ceridwen.com for other licensing options.	Library's main site lacks information. Commercial usage is not mentioned at all. Seems to be custom licensed open source.	Library's main site lacks information. Commercial usage is not mentioned at all. Seems to be custom licensed open source.
Advanced features	• **Mobile platforms support** • Video support • Good documentation and community support	3M SIP version 2 protocol (also known as SIP2). It includes both client and server implementations and supports both telnet and socket-based communications.	This project publishes a full implementation of RFC 3261 Specification as well as support for several SIP RFCs.	• All standard SIP layers and components • Various SIP extensions (already defined within IETF) • Some useful Call Control APIs (e.g. Call-Control, UserAgent, etc.) • Reference implementation of some SIP systems (Proxy Server, Session Border Controller, and UA)

Table 2. 3 Comparison of PJSIP vs. Python libraries (Schulzrinne *et al*, 2003).

	PJSIP	Shtoom	SIP SIMPLE Client SDK
Link	Library: http://www.pjsip.org Wrapper for Python: (http://www.pjsip.org/pjsua.htm)	Home site: http://divmod.org/projects/shtoom (temporary down) Library sources: http://sourceforge.net/projects/shtoom/	http://sipsimpleclient.com/
Supported platforms	• Windows • Mac • Linux • Windows Mobile (using wrapper) • iOS • Symbian • *Android and Blackberry are scheduled for next release*	• Windows • Mac • Linux	• Windows • Mac • Linux
License	GPL 2.0	LGPLv2	GPLv3
Price	Free	Free	Free (under GPL)
Licensing comments	Free usage is allowed if the final software is licensed under GPL 2.0 or later. *(Commercial products are allowed by the license, but the source code is required to be available in public)*. Custom open source license could be requested, with some exceptions.	LGPL Libraries can be used commercially by binary linking.	For alternative non-GPL license and support contract you may contact AG Projects at sales-request@ag-projects.com
Advanced features	• **Mobile platforms support** • **Video support** • **Good documentation and community support**		• **Good documentation and support**

Each programing language usually has a platform-specific library, but as depicted above it is PJSIP library which is presented in all three matrices. PJSIP can be used on different desktop and mobile platforms due to its native C++ implantation and a wide range of wrappers available. Due to it is good functionality, community support and documentation, PJSIP appears to be the best choice. Consequently because of these reasons we settled on PJSIP for our project.

2.6 Design Science Research.

The objective of this project was to design an IP based Voice Communication System for Air Traffic Control. The design was supposed to come up with algorithms which would then be implemented by coding. We analyzed table 4 below on research paradigms and their underlying philosophical assumption as suggested by (Vaishnavi and Kuechler, 2015).

Table 2. 4 Philosophical assumptions as suggested by Vaishnavi and Kuechler, 2015.

	PHILOSOPHICAL ASSUMPTIONS			
RESEARCH PARADIGMS	**ONTOLOGY**	**EPISTEMOLOGY**	**METHODOLOGY**	**AXIOLOGY**
Positivist	-Single, stable reality - Law-like	-Objective - Detached observer	-Experimental - Quantitative - Hypothesis testing	-Truth (objective) -Prediction
Interpretive	-Multiple realities - Socially constructed	-Empathetic - Observer subjectivity	-Interactional - Interpretation - Qualitative	-Contextual understanding
Critical/ Constructionist	-Socially constructed reality - Discourse - Power	-Suspicious - Political - Observer constructing Version	-Deconstruction - Textual analysis - Discourse analysis	-Inquiry is value-bound - Contextual understanding - Researcher's values affect the study
Design	-Multiple, contextually situated realities	-Knowing through making - Context-based construction	-Developmental - Impact analysis of artefact on composite system	-Control - Creation Understanding

From the table, paradigm of design science research (DSR) was adopted for this study. DSR is usually used in the design/creation of artefacts or innovations whose aim is to solve problems or change the status of the world (Vaishnavi and Kuechler (2015). The philosophical of the methodology of DSR which highlights developmental and impact analysis of artefact on composite system is in line with the objectives of this project.

2.6.1 Strategies for Design Science Research

Strategies usually used in DSR are design research and action research approaches. Table 1 above highlights the philosophical stance for this kind of research. When new systems like IT or IS are introduced in organizations, action research is conducted to find the socio-technical effect on users or organizational management and the operations. Design research in contrast as a strategy is concerned with improvement/design of constructs in context, knowing or learning through invention and making or enhancement.

For this project, design research approach was more suited since the focus of the research was on the construction of an artifact. In Design research, we can distinguish three research cycles as shown in figure 2.5 below.

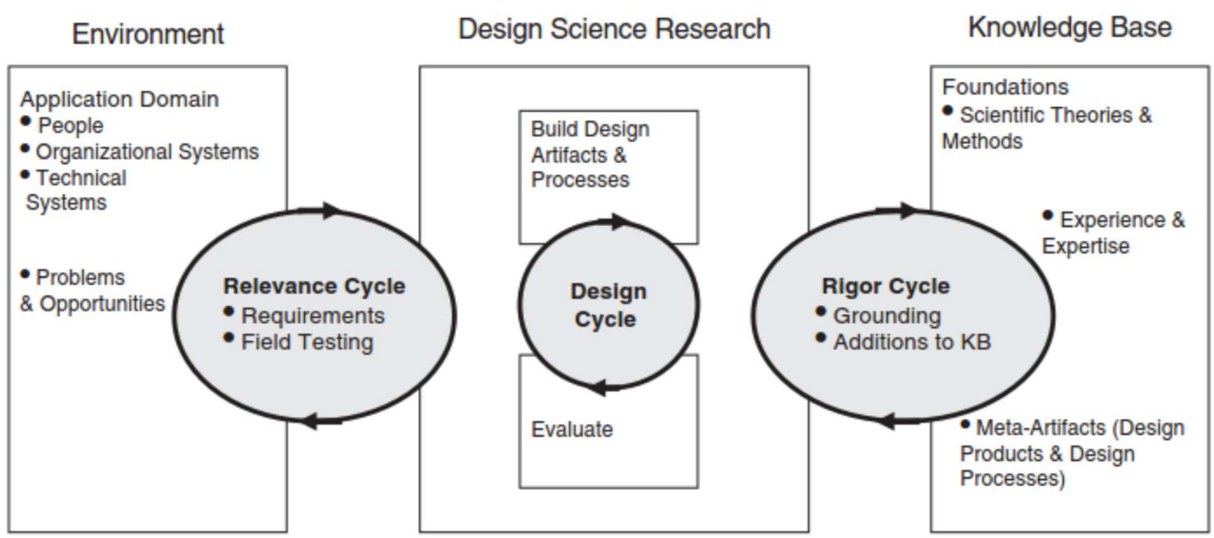

Figure 2. 5 Design research cycles and research relevance and rigor (Hevner *et al*, 2010 p. 16)

Relevance cycle:

• The research problem or the need for the research and the research environment is explained in the relevance cycle, the application domain initiates Design research by providing the following

a. The requirements which include the opportunity, problems, potentiality, etc.
b. Design artifact evaluation - Acceptance Criteria.

- Testing of the Research results in the field to answer questions like
    a. Is the environment improved due the Design artifact?
    b. How do we measure the improvement?
- Repeat the relevance cycle as need be

The artifact may have deficiencies in behavior and qualities which will lead to the restatement of the requirements and when the artifact has been modified field testing is done again.

The rigor cycle:

It uses current knowledge bases which include methods, theories, products, processes, artifacts, expertise and experiments that provide a foundation for thorough design research.

The design cycle:

This cycle encompasses the build/design and evaluation which are the main activities and actions. Requirements are validated through a thorough evaluation which may result in problem statement being revisited for improvement are revisited for improvement. DS methodology as and shown in figure 2.6 was employed for this project.

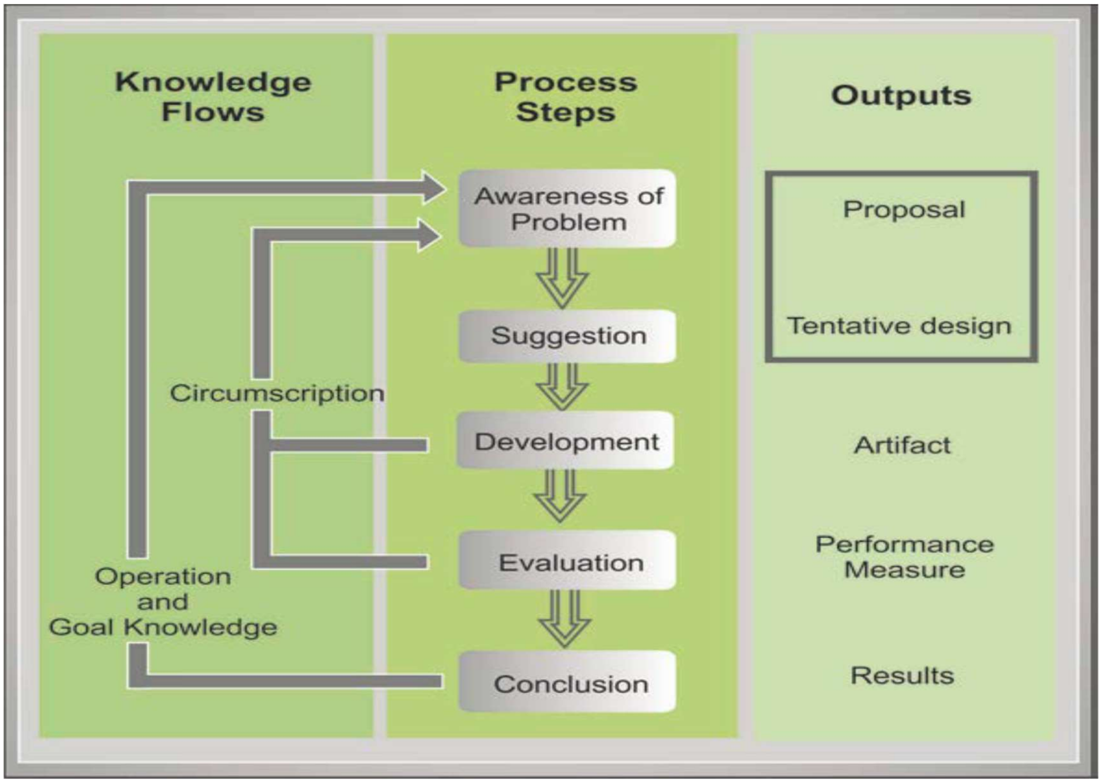

Figure 2. 6 General methodology for design research (Vaishnavi and Kuechler 2015)

Vaishnavi and Kuechler (2015) describes the design research process as shown below:

1. Awareness of problem. This is the first step in which a problem is sighted by the researcher. Identification of need or an idea or Problem where design and creation of a model, framework, artefact, construct, theory or method can result to solutions is the characteristic of this phase. A research proposal is the result of this step.
2. Suggestion. Using relevant existing theories and knowledge solutions are suggested in this second step. Finally, a possible design or solution is selected among many possible solutions.
3. Development. This phase leads to development of an artifact.
4. Evaluation. Qualitative or Quantitative evolution technique is used to perform evaluation so as to measure performance of the artefact.
5. Conclusion. In the conclusion phase, results of the research are used to enhance the body of knowledge.

In the three stages of development, implementation and evaluation, new findings elicit new problem/ideas, which lead to an iteration of some of the processes. It may happen that "circumscription process" which is iterations of the process done several times, may be conducted before the design is fully complete.

2.6.2 Program evolution - Development methods:

No software program within a changing environment will ever have all the requirements for the future nor will be immune to change. Rather, it will become more and more complex to cope with the changes of the outside world. This is reflected in two of the basic f software evolution laws:

- The Law of Continuing Change — a program that is in existence must change, or with time it will becomes less useful.
- The Law of Increasing Complexity — as a program is enhanced, with time it becomes more complex. This will continue until work is done to maintain or reduce the complexity.

Components reuse and software reengineering are some of the examples of how one can cope up with continued change and increasing complexity.

Software components reuse

In the present world it is recommended to design the system with individual components that can be replaced or changed as the system evolves. These components have individual interfaces, they encapsulate internal details and they have separate documentation. One main advantage of constructing software with components is reuse. Software reuse is when existing software is used to construct new software. Examples of software reuse are design patterns and software frameworks. A design pattern is a reusable software design. They are not code in themselves, they rather express a commonly tried and tested approach in software. A framework uses these design patterns to form the architecture for an application.

Software reengineering

Re-engineering is the analysis and modification of a software system. The goal with software reengineering is to decipher and understand the current software and then improve the system. Re-engineering can consist of reverse engineering, refactoring and forward engineering.

- **Reverse engineering** is a process where one analyzes an existing software and creates a representation of this system on different abstraction levels. Reverse engineering can be done in the following steps (Dugerdil, 2006)
    1. Scope: Firstly, one must decide which part of the system should be refactored/restructured and why. All parts may not be important or critical to restructure.
    2. Domain: Identify business tasks that the system should support. These tasks can then be divided into use-cases. These should be documented and should be mapped to their data structures (entity, table column etc.).
    3. Architecture recovery: Cluster software elements together that support the business tasks. This is done by tracing which software elements connect to certain tasks. This can be done by seeing which data structure is connected to what software element. In a system where there is no separation between views and models this is hard.
- **Refactoring or restructuring** is the transformation of a system at the same abstraction level.
  It keeps the external behavior of the system.
- **Forward engineering** is the traditional way of designing systems. It usually starts from high abstraction and ends in an actual physical implementation of the system.

Software reengineering approach:

Software re-engineering can be done in different ways (Rosenberg *et al*, 2002).

a. **Incremental:** In this approach, system sections are re-engineered and added incrementally as new versions of the system are needed to satisfy new goals. The project is broken into re-engineering sections based on the existing system's sections.

b. **The "Big Bang":** This approach, also known as the "Lump Sum" approach, replaces the entire system at one time. This approach is often used by projects that need to solve an immediate problem, such as migration to a different system architecture.

c. Evolutionary: In the "Evolutionary" approach, as in the Incremental approach, sections of the original system are replaced with newly re-engineered system sections. In this approach however, the sections are chosen based on their functionality, not on the structure of the existing system.

2.6.3 Design evaluation methods

The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well executed evaluation methods (Hevner *et al,* 2004). As such evaluation is an important stage in Design Research. ICT artifacts can be evaluated in terms of functionality, performance, completeness, reliability, consistency, usability accuracy, fit with the organization, and other relevant quality attributes. The evaluation of designed artifacts usually employs methodologies available in the knowledge base. These methods are summarized in Table 2.5 as shown below. The selection of evaluation methods must be matched appropriately with the designed artifact and the selected evaluation criteria.

Table 2. 5 Design methods of evaluation (Hevner *et al,* 2004).

1. Observational	Case Study: Study artifact in depth in business environment
	Field Study: Monitor use of artifact in multiple projects
2. Analytical	Static Analysis: Examine structure of artifact for static qualities (e.g., complexity)
	Architecture Analysis: Study fit of artifact into technical IS architecture
	Optimization: Demonstrate inherent optimal properties of artifact or provide optimality bounds on artifact behavior
	Dynamic Analysis: Study artifact in use for dynamic qualities (e.g., performance)
3. Experimental	Controlled Experiment: Study artifact in controlled environment for qualities (e.g., usability)
	Simulation – Execute artifact with artificial data
4. Testing	Functional (Black Box) Testing: Execute artifact interfaces to discover failures and identify defects
	Structural (White Box) Testing: Perform coverage testing of some metric (e.g., execution paths) in the artifact implementation
5. Descriptive	Informed Argument: Use information from the knowledge base (e.g., relevant research) to build a convincing argument for the artifact's utility
	Scenarios: Construct detailed scenarios around the artifact to demonstrate its utility

The above methods of evaluation were critically analyzed. The primary goal of evaluation through observation is to determine in a comprehensive manner how the artifact behaves in a practical environment place (Hevner *et al*, 2004). This method of evaluation, researcher becomes an observer without participating directly. In the analytical method the artifact is evaluated to determine the interaction with the environment (Hevner *et al*, 2004). For this method of evaluation, the purpose is to check on performance and how it can be improved. Next is the experimental evaluation which can done using controlled experimental like in a laboratory setup or through simulation (Hevner *et al*, 2004). The fourth evaluation method is which can be done through functional testing and structural testing (Hevner *et al*, 2004). The final method of evaluation is Descriptive evaluation method. In this method one can form informed argument using information from the knowledge base or use contract detailed scenarios about the artifact to depict the utility. The fourth evaluation method is which can be done through functional testing and structural testing (Hevner *et al*, 2004). The requirements for this project were based on ED 137 B standard. To be able to verify whether this requirement was met, real environment where the Air Traffic controller communicates to the aircrafts was not possible. Therefore, we settled on Experimental evaluation method employing simulation. Simulation was employed because the artifact was executed using artificial data

2.8 Conceptual Framework.

From the foregoing it emerges that in order to design an IP based voice communication that can meet Air Traffic control requirements using Open Source SIP/RTP library, then the most important element that needs to be addressed is the Radio signaling. Different approaches have been discussed but in order to comply with standards for Air traffic control communication ED 137B standard which specifies that an RTP packets should have an extended header. This extra header should carry PTT which is required for audio transmission and Squelch which indicates to the VCS that audio is coming. In addition, the extra header is required to check link status. Since once SIP has setup a telephone session audio can flow in both directions at any time, there is a need suppress audio transmission when the operator is not talking. Figure 2.7 below is the operation concept showing the conceptual framework.

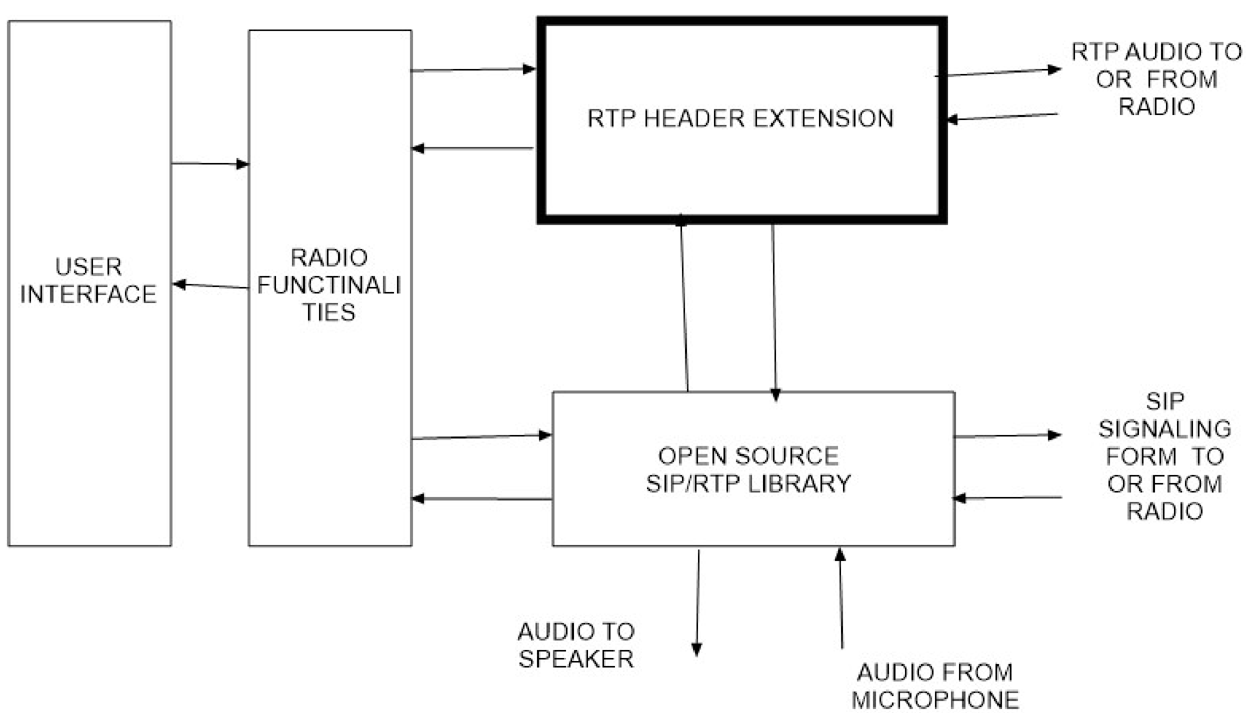

Figure 2. 7 Conceptual framework (Johnston, 2015)

Description of the main components

a) User interface and radio functionality module:
   - Set communication via SIP/RTP library
   - enable channel selection by the operator
   - Enable audio transmission by the operator.
   - Cease audio transmission by the operator
   - Provide visual indication for audio transmission and reception.
   - Give visual indication of channel status to the user.

b) Real-time Transport Protocol Header extender:
   - Provide PTT and Squelch signaling information for transmitter selection and receiving audio.
   - Enable link monitoring.

c) Session-Initiation - Protocol/Real-Time Transport Protocol library
   - Open source SIP/RTP library in this case PJSIP will be employed. This library will provide all the SIP and RTP processing functionalities.

CHAPTER 3: METHODOLOGY

3.1 Introduction:

This project is based on design science research DSR. It is used in the design of artifacts or to find solution to problems which could change the status of the universe (Vaishnavi and Kuechler, 2015). Two strategies are used with DSR; Action research and design research as discussed earlier the literature review. The strategy that was used for this project is Design research approach. This is because the main aim design research is developing and creating new artifacts. The following sections give details of this method and how it was used to archive the research objectives.

3.2 Research Design

For this project, the research design method which has been suggested by Vaishnavi and Kuechler (2015) and shown below in Figure 3.1 was employed design IP based Voice Communication System for Air Traffic Control.

The following description gives the five steps in the design research process and how it was used to achieve the objective of the research project (Vaishnavi and Kuechler, 2015).

1. **Awareness of the problem:** We took a thorough analysis of the voice communication situation in air traffic control. Using the new standard for voice communication in ATC coupled with aviation system block upgrade ASBU plan for the aviation industry we came to the realization of the following issuers:
   - New IP voice standard ED 137 B promoted by the industry regulator (ICAO) for interoperability purpose.
   - Investment in the analogue/TDM voice switch technology has reduced. This means spares will become expensive.
   - Telecommunication service providers are in the process of discontinuing or have discontinued communication links based on the old technology.

   The output of this phase is was the proposal which was presented and accepted.

2. **Suggestion**: Preliminary investigation into the opportunities presented by IP based telephone communication software was done. The gap that can be filled by the new ICAO VOIP standard was identified which led to a realization of a tentative design as shown in the literature review. Since the basis of this design was based on existing

work in IP communication, two methods namely software component reuse and software reengineering were explored. Software components reuse involves building software from scratch but at the same time making use of design patterns or software framework made earlier. This method is quite involving and considering that there are software libraries for telephone communication which can be modified using software reengineering we settled on software reengineering as the methodology for the development phase. As was indicated in the literature review the open source software to be reengineered was settled on PJSIP library.

3. **Development:** This phase entails the development of the artifact. The goal of this project was to design an IP based Voice Communication for Air Traffic Control using open source software in order to add the necessary functionality to provide for the new VOIP standard by ICAO there was need to employ a software development methodology that would allow the understanding of some critical parts of PJSIP library and application. As indicated in the suggestion phase, software reengineering methodology was chosen as the framework design the algorithms. The goal of reengineering the PJSIP library and the application was to refactor portion of the two software so that selected requirements of ED 137 B for voice communication in ATC are met. Since the overriding factor in the design of an IP based voice communication system for ATC is functionalities based on ED 137B evolutionary approach software reengineering was chosen.

Since this development entailed moving from the existing system to a target system software reengineering model proposed by Eric j. Bryne shown in figure 3.1 was used.

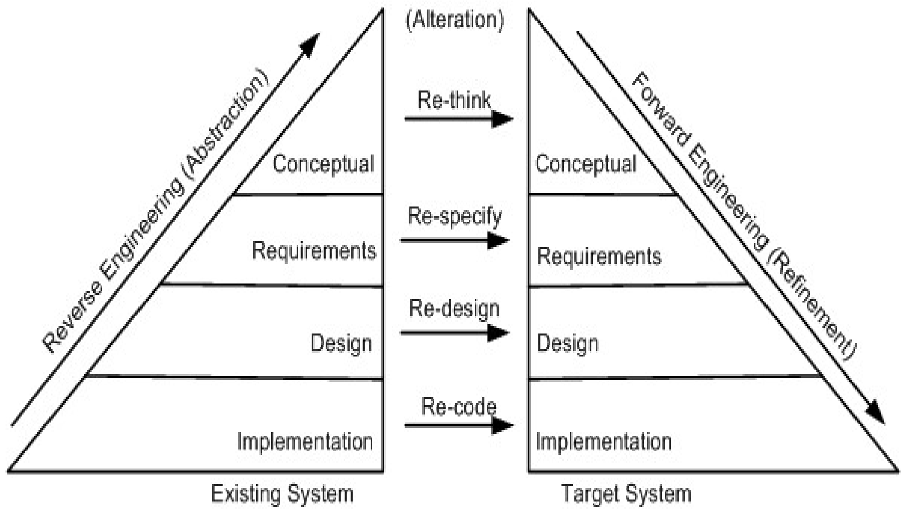

Figure 3. 1 General model of software reengineering IEEE 1992 (Dugerdil, 2006).

From the software reengineering model, the following steps were employed;

a. Reverse engineering (Abstraction):

In order to understand a software system, sufficient information must be extracted from the system for the re-engineering goal to succeed. Reverse engineering does just that; it creates and abstract a description of the system (Dugerdil, 2006).

The following reengineering steps as discussed in the literature were employed to achieve the reverse engineering goals;

1. Scope: The following were identified as the scope.
   a) PJSIP application PJSIP_sampl_simple_ua: This is a telephone application that runs above PJSIP library. It was chosen because of it is simplicity in the implementation of IP telephone communication.
   b) PJSIP library version 2.4 and it is associated PJMEDIA library: The scope within two libraries zeroed in on the components that deal with SIP and media.

2. Domain: A sip flow diagram was drawn to indicate the IP based communication tasks that occurs using the PJSIP library.

3. Architecture recovery: The architecture to be used for refactoring was recovered using the PJSIP library and the application.

    a. **Re-specify (Alteration):** We analyzed the architecture recovered in the reverse engineering section together with the selected ICAO VOIP standard. The result of this was a new SIP flow diagram showing the sequence of communication between the proposed voice communication system and the IP Radio (simulated for test)

    b. **Forward engineering (Refinement).** The results got in b (Re-specify) were used to come with a prototype using prototyping software development methodology. The activities of the prototype are by the diagram below.

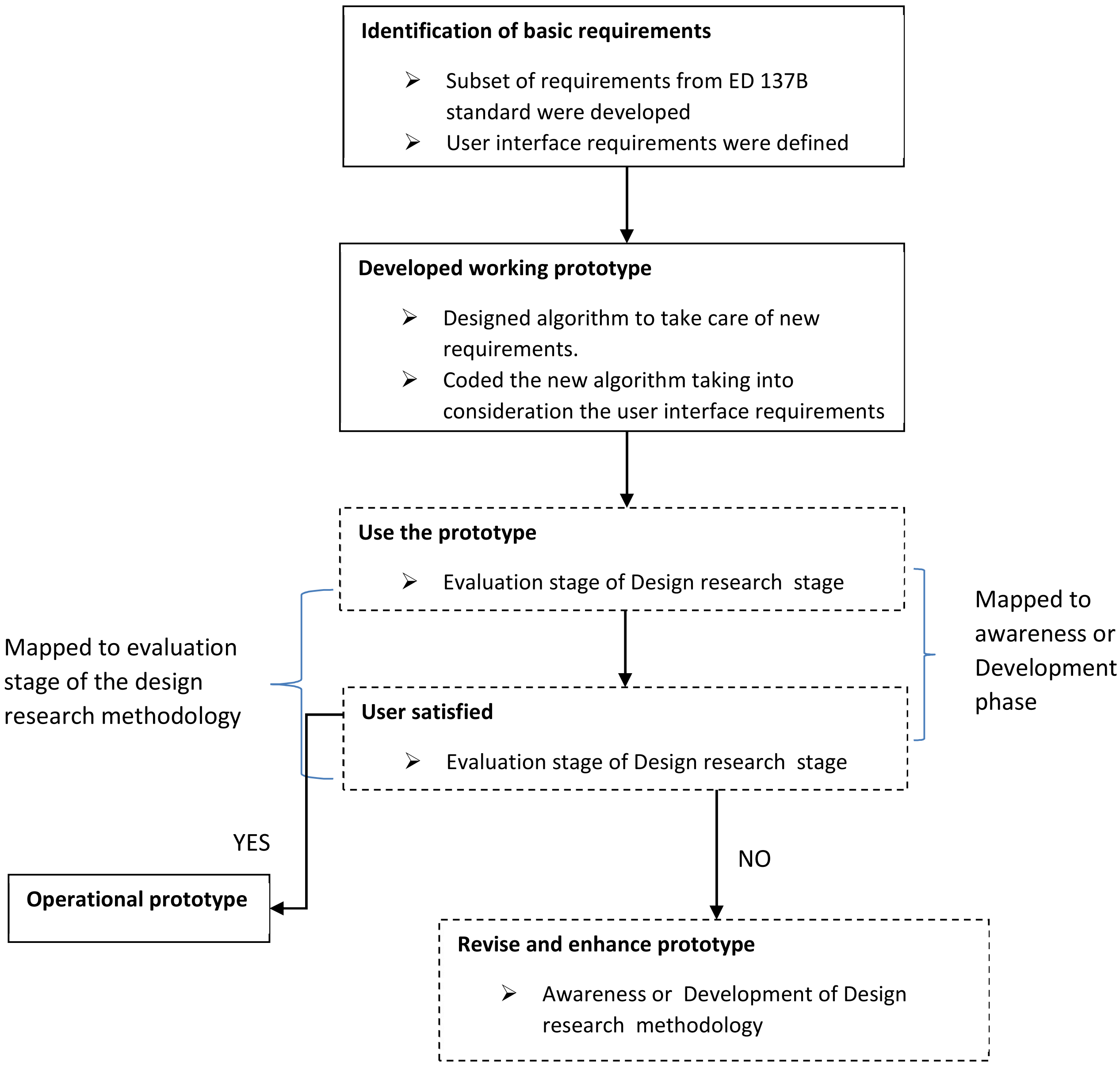

Figure 3. 2 Outline of the Development stage (Dugerdil, 2006).

3.3 Evaluation

The selected ICAO VOIP requirements were used to evaluate the prototype to find out whether the objectives of the project were met. The requirements were mainly centered on functionality and VOIP quality parameters such delay and audibility of audio signal.

Using figure 3.3, the selected evaluation methods were matched appropriately with the designed artifact.

1. Observational	Case study: Study artifact in depth in business environment
	Field study: Monitor use artifact in multiple projects
2. Analytical	Static analysis: Examine structure of artifact for static qualitied (e.g., complexity)
	Architecture analysis: Study fit of artifact into technical IS architecture
	Optimization: Demonstrate inherent optimal properties of artifact or provide optimality bounds on artifact behvaio
	Dynamic analysis: Study artifact in use for dynamic qualities (e.g., performance)
3. Experimental	Controlled experiment: Study artifact in controlled environment for qualities (e.g., usability)
	Simulation- Execute artifact with artificial data
4. Testing	Functional (black box) testing: Execute artifact interfaces to discover failures and identufy defects
	Structural (with box) Testing: Perform coverage tseting of some metric (e.g., execution paths) in the artifact implementation
5. Descriptive	Informed argument: Use information frim the knowledge base (e.g., relevant research) to build a convincing argument for the artifact's utility
	Scenarios: Construct detailed scenarios around the artifact to demonstrate its utility

Figure 3. 3 Design methods of evaluation (Henver *et al*, 2004).

Method: After analyzing the above methods shown in figure 3.1 we selected simulation method which is a subset of Experimental evaluation as the evaluation method. The reason behind this was due to:

a. No live Equipment was available to act as the Radio. Kenya where the author resides is yet to implement IP based Radio for Air Traffic Control.
b. We managed to get simulation software for the Radio end which could simulate most critical functions of the Radio.
c. In Kenya to be able to communicate with the aircraft one requires an Air Traffic control license.

Star Trinity Sip tester (Sip Tester n.d) was chosen as the simulation software for Radio part of the communication. There were other software for simulation like MAP SIP emulator from GL communication include (GL communication n.d) but the license was too expensive. Script statements written using CallXML (Sip Tester n.d) were used evaluate the artifact as indicated in the requirements.

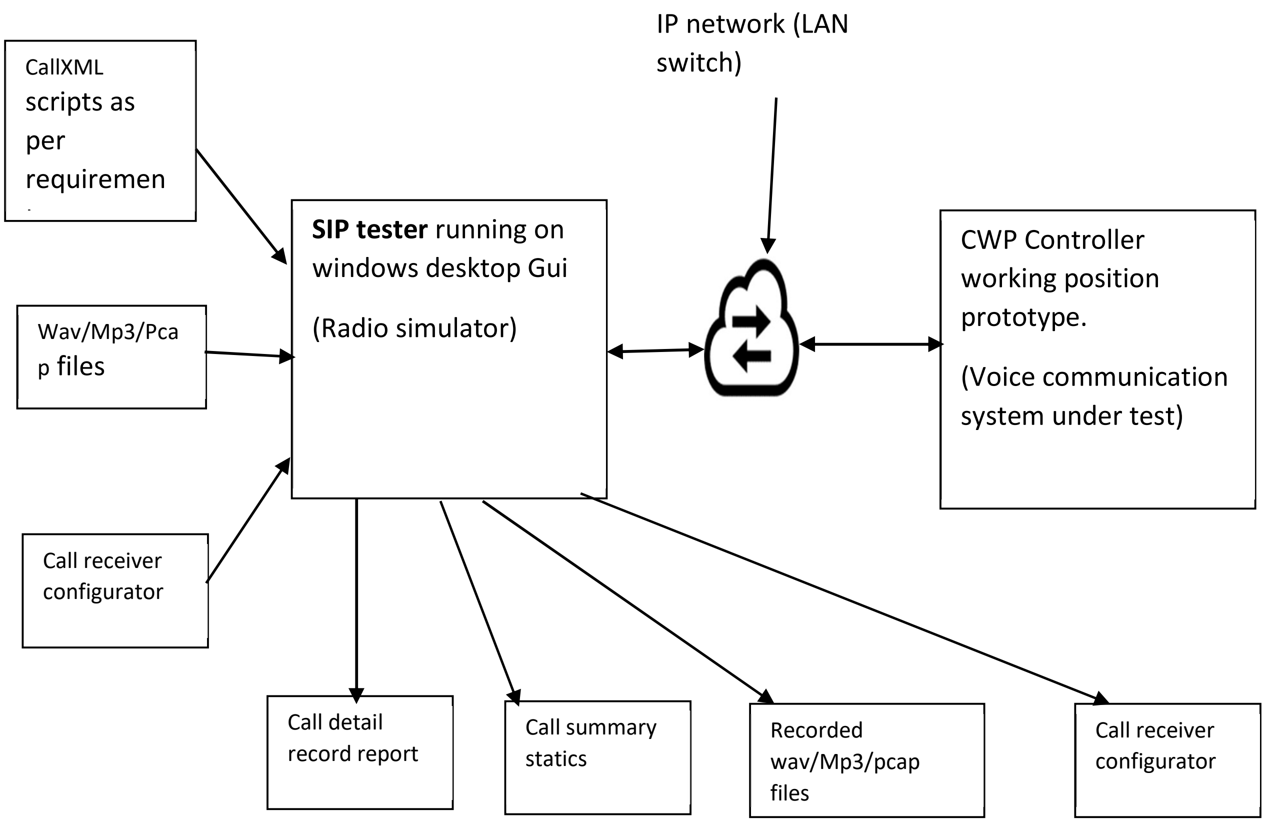

Figure 3. 4 Architecture employed for evaluation (Henver *et al*, 2004).

3.4 Conclusion

The evaluation results were discussed as those which past and those which didn't pass. Knowledge on how legacy IP based telephone can be extended to cater for VOIP Radio communication in Air traffic control was expounded.

CHAPTER 4: DEVELOPMENT

4.1 Introduction

This section details the development work that was done to design the IP based Voice Communication System for Air Traffic Control. The development work was based on software reengineering methodology. Consequently, reverse engineering which is the first task of software engineering is described first. We then describe Refactoring work which is the second task in software reengineering. Lastly but the least we describe the work that was done during the forward engineering stage.

4.2 Software reengineering.

The activities of software reengineering to achieve the objectives of the project are described here in terms of reverse engineering, re-specification and forward engineering:

4.2.1 Reverse engineering

4.2.1.1 Scope

The following was identified as the scope of the project:

a) Application: PJSIP sample_simple_ua:
b) Telephone library: PJSIP library version 2.4 (PJSIP and PJMEDIA)

Figure 4.1 below shows the scope for the project as indicated above

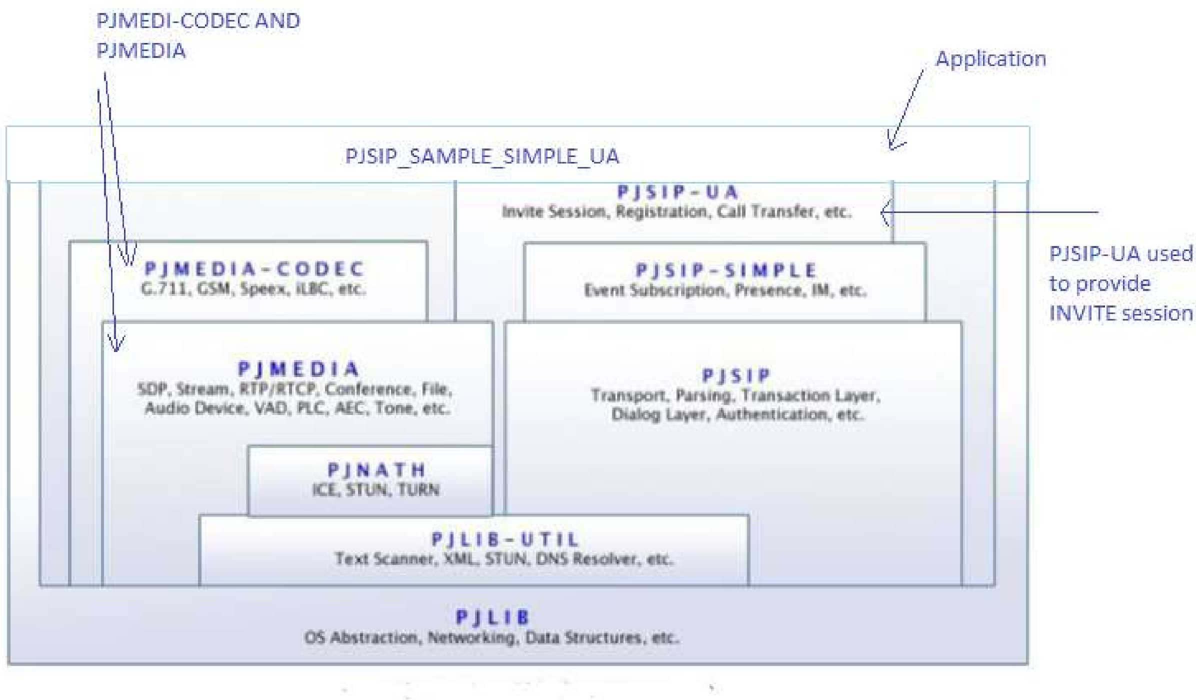

Figure 4. 1 PJSIP library and Application showing scope for Reverse engineering (Handley & Schooler, 2002).

SIMPLEUA.C: This is a basic simple SIP user agent complete media. The user agent does the proper SDP negotiation and there after starts RTP media session once SDP negotiation has finished.

PJSIP-UA: This is a high level abstraction of INVITE sessions. The library also provides SIP transfer functionality and client registration,

PJMEDIA and PJMEDIA-CODEC: are multimedia and placeholder for codecs respectively. These two components when operational can be described by the media flow diagram shown in figure 4.2 below.

4.2.1.2 Architecture recovery

The following section details the essential architectural components that we found necessary to understand in the process of reverse engineering. We first highlight the PJMEDIA and some of its internal components structure which was part of the. Finally, we show the logic of the PJSIP application (simpleua.c) in terms of a flow chart.

PJMDIA: The following pjmedia flow diagram

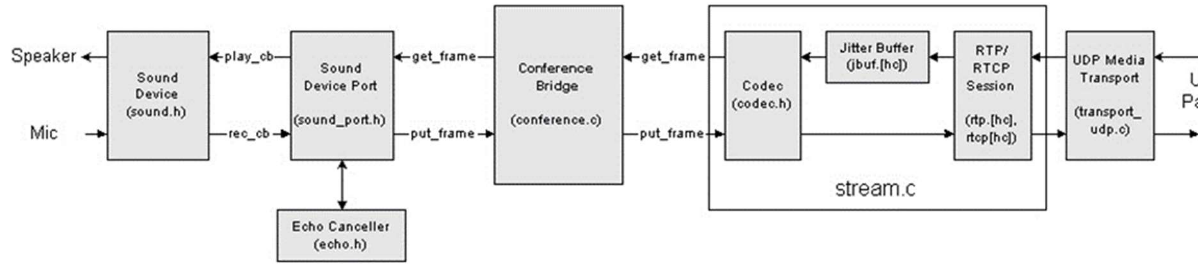

Figure 4. 2 PJMEDIA media flow (Handley & Schooler, 2002).

The main building block for the above diagram are:

- Sound Device Port which is a thin wrapper for the Sound Device Abstraction to translate sound device's rec_cb()/play_cb() callbacks into call to.
- Conference Bridge: This allows telephone conference but is not necessary for Radio communication.
- Media Stream that is created for each call.
- Media Transport that is attached to the Media Stream to receive/transmit RTP/RTCP packets

4.2.1.3 DOMAIN

The following sip flow diagram figure 4.2 shows the major components involved in telephone communication at higher level of abstraction using PJSIP library. This explains the domain of the legacy system that was employed in the design of an IP based system for Air Traffic Control.

Figure 4.3 below show SIP call flow diagram for the legacy system.

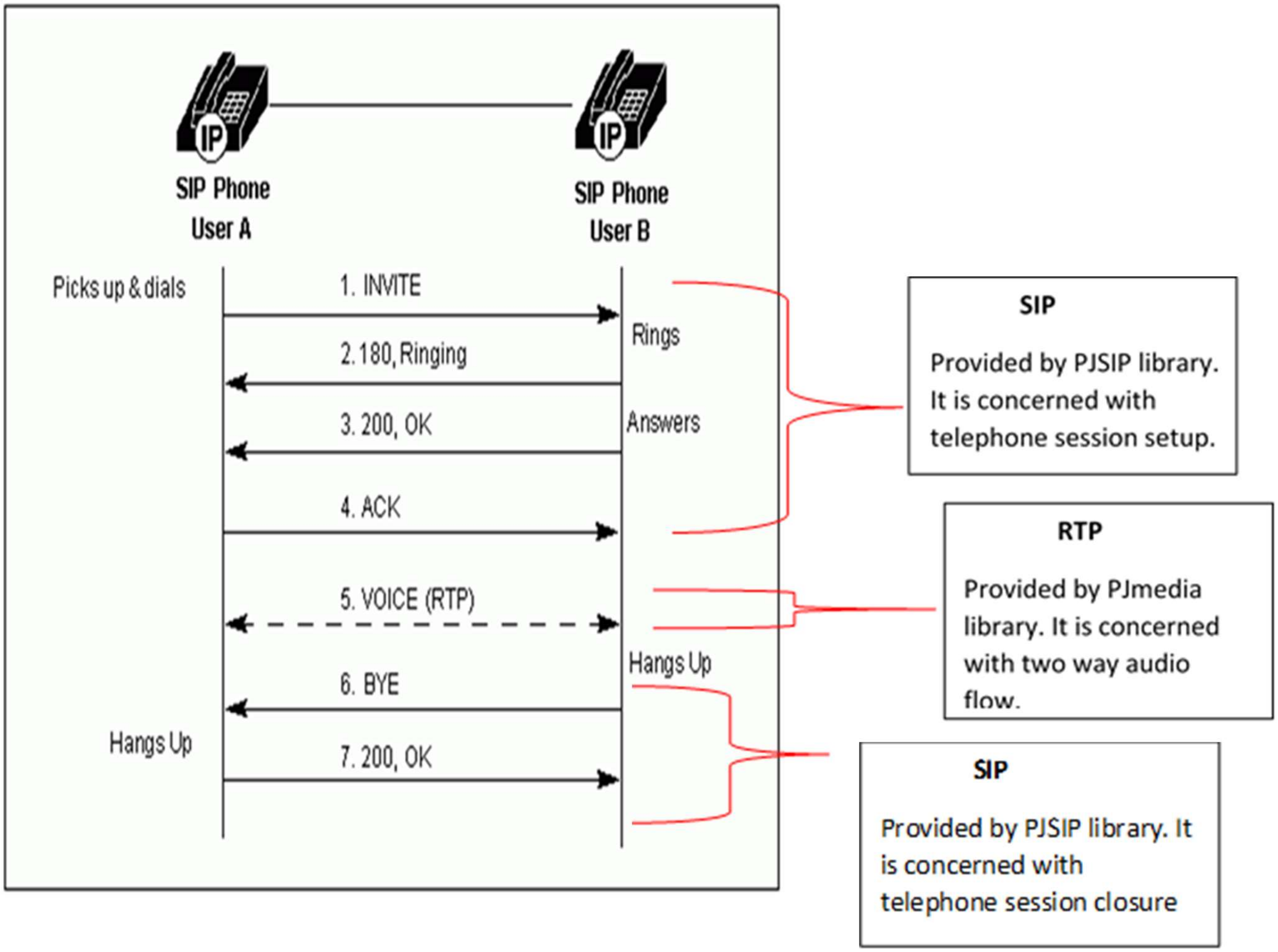

Figure 4. 3 Sip flow

Description of the SIP call flow diagram:

From the above figure, user A sends an INVITE message (flow number 1) request to user B, upon which B's SIP Phone responds by ringing (flow number 2), waiting for Bob to pick up. When Bob picks the handset an OK (flow number 3) encoded response is transmitted back, which is then ACK acknowledged by (flow number 4) (Priyadarshi *et al*, 2015). This exchange, results in session is set up, and the actual communication intelligence (audio or video) is sent in full duplex mode using RTP packets stream (flow number 5). To end the session, one of the uses e.g. B replaces the handset which the results in a BYE (flow number 6) request, which is then replied by sending an OK (flow number 7) response from A.

CONFRENCE BRIDGE

The conference bridge provides a programing interface API to control audio routing between the various audio objects. The idea is that it is it connects audio source to audio destination, which will make the audio flows from the source to destination, an example of the audio object from media flow in the figure 4.2 above is the sound device port and media stream port found in stream.c.   The diagram in figure 4.4below shows a conference

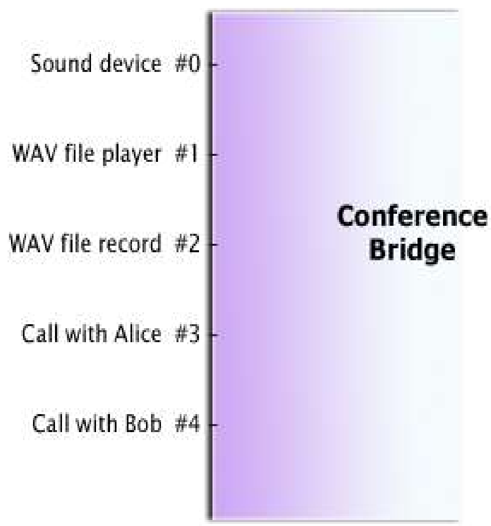

Figure 4. 4 Conference Bridge

As shown above in the diagram above the conference bridge has the following objects:

- sound device, which is always attached at slot number 0 as a convention.
- at slot 1 a WAV file to be played back
- at slot 2 a WAV file for recording
- at slot 3 an active call to Allice
- at slot 4 an active call to Bob

For slot 3 and 4 the media object is formed by stream.c object. As shown in the diagram above whenever a media object is plugged-in to the bridge, it will not be connected to anything, so media will not flow from/to any of the objects.

Telephone call

Usually a telephone call, establishes a bidirectional audio with the remote person and in this case Alice, the media object for Alice can be connected to any free slot apart from slot 0 which is always reserved for microphone and speaker.

The conference bridge diagram below shows the interconnection for Alice call.

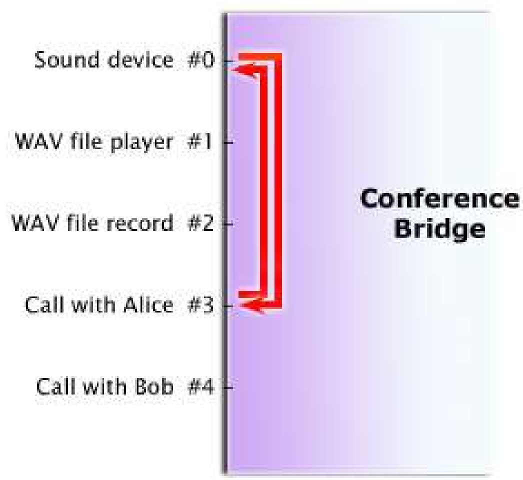

Figure 4. 5 Conference bridge with two way communication

As shown in the above diagram a call that is bidirectional between sound device (speaker/microphone) and Alice has been established as indicated with the red lines.

MEDIA STREAM AND MEDIA TRANSPORT

The diagram in figure 4.6 below shows the interconnection between the media transport and the media stream.

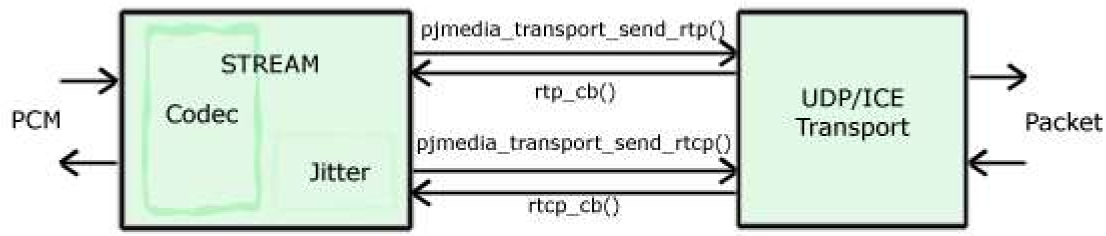

Figure 4. 6 Interconnection between media stream and media transport

From the diagram above we note that we have an input in the form of PCM which is then encoded to RTP packet and the passed on to the media transport which then encodes it to an IP packet. The same is repeated when a packet from the network arrives at the media transport but the process is reversed.

pjmedia_transport_send_rtp (): This Sends RTP packet with a specific media transport. This simple wrapper then calls send rtp () member of the transport. There after the RTP packet will then be delivered to the destination address. As shown in the literature review the PJMEDI RTP packet header is a standard RTP packet header as used in any IP telephone system

rtp_cb: This is the call back to the stream object as audio flows in the opposite direction

The figure 4.7 below highlights the sequence of the events take place when this library is used for IP telephone communication

4.2.2 Re-specification (Alteration)

Reengineering = Reverse engineering + Δ + Forward engineering (Darilion *et al*, 2004).

- This function "Δ" depicts the modifications made to the original system. There are two key dimensions of the alterations: change in functionalities and change in implementation techniques.

- Functionality change comes from a change in the business rules, this is the desired change for our project. The change required s to design an algorithm to implement the new ICAO VOIP standard for Air Traffic Control
- The other change is concerned with implementation. For example, a program can be converted from procedural to object oriented programing. This alteration will not be used.

Consequently, from the ED 137B documentation the following SIP flow diagram figure 4.7 summarizes the call flow sequence for the prototype:

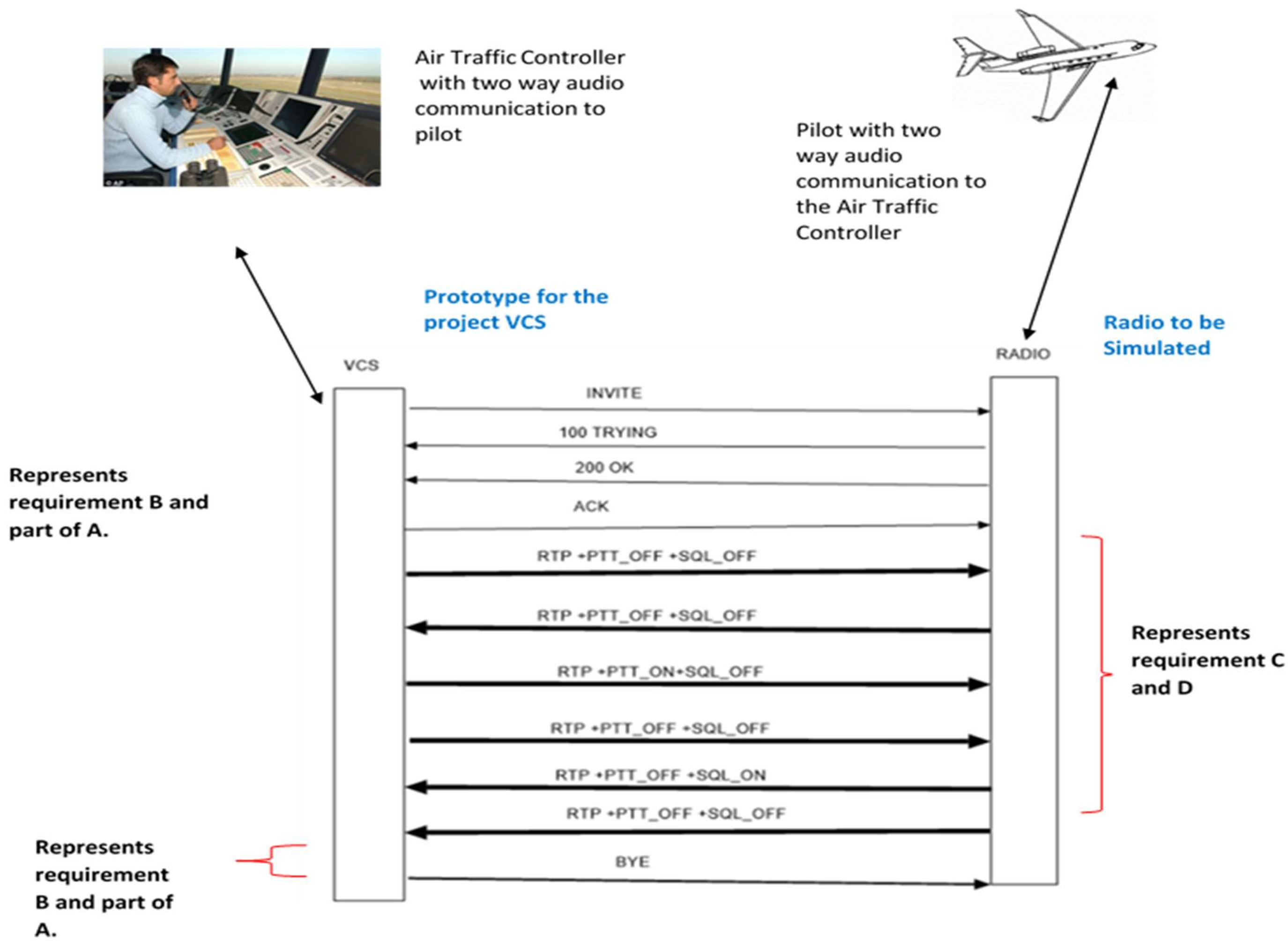

Figure 4. 7 Call flow sequence for the prototype

Description of the requirements (ED137B Radio)

1. **INVITE:** The User Agent at the VCS endpoint sends an INVITE request message containing the following: From: and To: SIP URI addresses, a Call_ID used to identify the call, Cseq command sequence header, the Contact address of the User Agent A1, a Max-forwards value of 70 (used to avoid SIP Radio session requests entering a loop), a Subject header defined as "radio" and a Priority header defined as "normal". The content type defines an SDP message body is being carried with the INVITE method.
2. **100 Trying:** The User Agent at Radio endpoint MAY respond with a 100Trying provisional response containing the same from: and to: SIP URI addresses, the same Call_ID used to identify the SIP Radio session, the same Cseq command sequence header and no SDP message body included.
3. **200 OK:** The User agent at Radio endpoint on receiving the INVITE request message SHALL verify its SIP contents and SDP message body contents. If the GRS endpoint has sufficient resources and the correct media capabilities etc. in order to accept the SIP session establishment request, it will answer with a 200OK final response message, containing an SDP message body that confirms the negotiated designated media capabilities and attributes between the two endpoints.
4. **ACK:** The User Agent A1 at the VCS endpoint then sends an ACK request message containing the same: From: and to: SIP URI addresses, the same Call_ID used to identify the SIP Radio session, a new Cseq command sequence header for the ACK, the Contact address of the User Agent A1, a Max-forwards value of 70 (used to avoid call loops). The content type defines that no SDP message body is being carried with the ACK method**.**
5. **RTP+PTT_OFF+SQL_OFF**: Link monitoring from VCS to the Radio. PTT is set to OFF in extra header when there is no audio for transmission. In response the Radio will transmit RTP packet with extra header by setting squelch OFF after 200 milliseconds. In case there is transmission from either VCS or Radio link monitoring will be suspended and incase of Transmission from VCS go to 6 and for reception from Radio go to 7.
6. **RTP+PTT_ON+SQL_OFF:** In case of transmission from VCS, the VCS will set PTT ON in the extra header and send RTP packet together with RTP extra header to the

Radio. In case there is no further audio transmission from the VCS will set PTT to OFF and resume link monitoring (Go back to 5 above) within 200 milliseconds.

7. **RTP+PTT_OFF+SQL_ON:** In case of reception from the Radio, the Radio will set SQL ON in the extra header and send RTP packet together with RTP extra header to the VCS. In case there is no further audio transmission from the Radio will set SQL to OFF and resume link monitoring (Go back to 5 above) within 200 milliseconds.
8. **BYE:** By deselecting the channel the SIP session shall be closed.

4.2.3 Forward engineering.

This section describes the forward engineering activities that were undertaken to design the IP based Voice Communication System for Air Traffic Control. We first describe how the requirements were gathered. Thereafter we designed four algorithms which we the coded and integrated with PJSIP library. Since the method employed for forward engineering is prototyping which is quite iterative, it overlaps with evaluation whose results feed the forward engineering activities for fine tuning.

4.2.3.1 Requirements

ED 137 B VOIP (Solar, 2014) standard was used as the main source of requirements for this project the requirements that were chosen were those that that deliver the objectives of the project which also coincide with providing basic radio communication functionality using IP for Air Traffic control. The table 4.1 below highlight the requirements that were chosen.

Table 4. 1 ED 137 B Selected requirements for the project (Solar, 2014).

(A) RADIO COMMUNICATION MODEL	
3	**[COMMUNICATION MODEL] Applicable Protocols**
	The SIP, RTP and R2S protocol **SHALL** be the minimum requirements necessary for the implementation so as to provide VoIP communication between the e VCS and GRS endpoints.
5	**[COMMUNICATION MODEL] Communication initiation between VCS and combined GRS**

	In the case that a GRS Transceiver or a GRS Transmitter/Receiver located at the same site and accessible by one SIP URI, the communication between the VCS end point and a GRS endpoint SHALL be performed in two distinct phases. • Phase 1: The SIP session **SHALL** always be initiated from the VCS endpoint towards the GRS endpoint (transceiver, transmitter or receiver). Phase 2: Once the SIP session is established, both VCS and GRS endpoints **SHALL** use the "Keep Alive" mechanism of the R2S protocol to control the link between the VCS and the GRS. R2S-Keepalive packets will always be exchanged between end points in the case that no audio is present.
(B) PROFILE STANDARD FOR THE USE OF SIP IN AN AGVN	
1	**[SIP] SIP Version**
	An Air Traffic Services VoIP Communications System SHALL support SIP version 2 as specified in RFC 3261 [12].
2	**[SIP] SIP Supported requests – see ED137/1B document**
3	**[SIP] SIP Supported response– see ED137/1B document**
4	**[SIP] SIP Message body (SDP**
	Those SIP message bodies containing a description of the session, time and media SHALL be encoded in the Session Description Protocol (SDP) (RFC 2327 [7]).
7	**[SIP] Normal SIP session establishment** SIP session establishment request sent by a VCS endpoint to a Ground Radio Station End point in normal operational conditions **SHALL** use a SIP Priority Header field set to "normal" and a SIP subject header field set to "radio".
	(C) AUDIO
2	**[AUDIO] Voice quality**
	The voice quality of a radio communication is defined using a voice quality estimation methodology nominated "Mean Opinion Score" (MOS) rating.
3	**[AUDIO] Voice latency time performance.**
	The system delay shall respect ITU-T Recommendation G.114

4	**[AUDIO] Voice Packetization interval requirements.**
	The VCS and GRS endpoints SHALL communicate using voice packet sizes of 10, 20 or 30ms.
5	**[AUDIO] Voice coding requirement.**
	The VCS and GRS SHALL support the following codecs according to ITU-T G.711PCM A-law or μ-law G.711 PCM. In order to improve robustness, the ITU-T G.711PLC codec [34] MAY be used;
(D) RTP: REAL-TIME TRANSPORT PROTOCOL	
1	**[RTP] RTP Radio Signaling Audio and protocol requirement.**
	Within an IP-network, the audio transmission and specific radio signaling SHALL be performed by the Real-time Transport Protocol (RTP).
5	**[RTP] RTP PTT transmission performance.**
	PTT signal is used to activate transmission at the GRS transceiver/transmitter. It is activated when the controller at the VCS endpoint selects the PTT key at the Controller Working Position.
6	**[RTP] Squelch transmission performance.**
	Squelch) signal is active when the GRS transceiver/receiver detects an incoming radio call.
7	**[RTP] RTP Header Extension description**
	The RTP header extension is used to transmit additional information necessary for Radio communication. (I.e. PTT activation info, Squelch indication, signal quality index, etc.). The extension SHALL be implemented according to RFC 3550 [21]
	• **REAL TIME SESSION SUPERVISION**
16	**RTP] Keep alive messages**
	The Real Time Session Supervision SHALL be employed between VCS endpoints and GRS endpoints.

To summarize from table 4.1 above it suffices that alteration will involve the following components.

SIP:

The functionality of SIP during session setup, active session, and session termination will have to be modified to take care of ED 137/B requirements.

RTP.

The functionality for two-way radio communication will have to be modified so that we have:

a. Half duplex communication instead of full duplex communication
b. During silence period there will be exchange of keep alive packets.
c. There will be a need to provide signaling information to signal each end incase audio communication needs to occur.

The proposed by ED137 B to take care of header extension is shown in figure 4.8 below

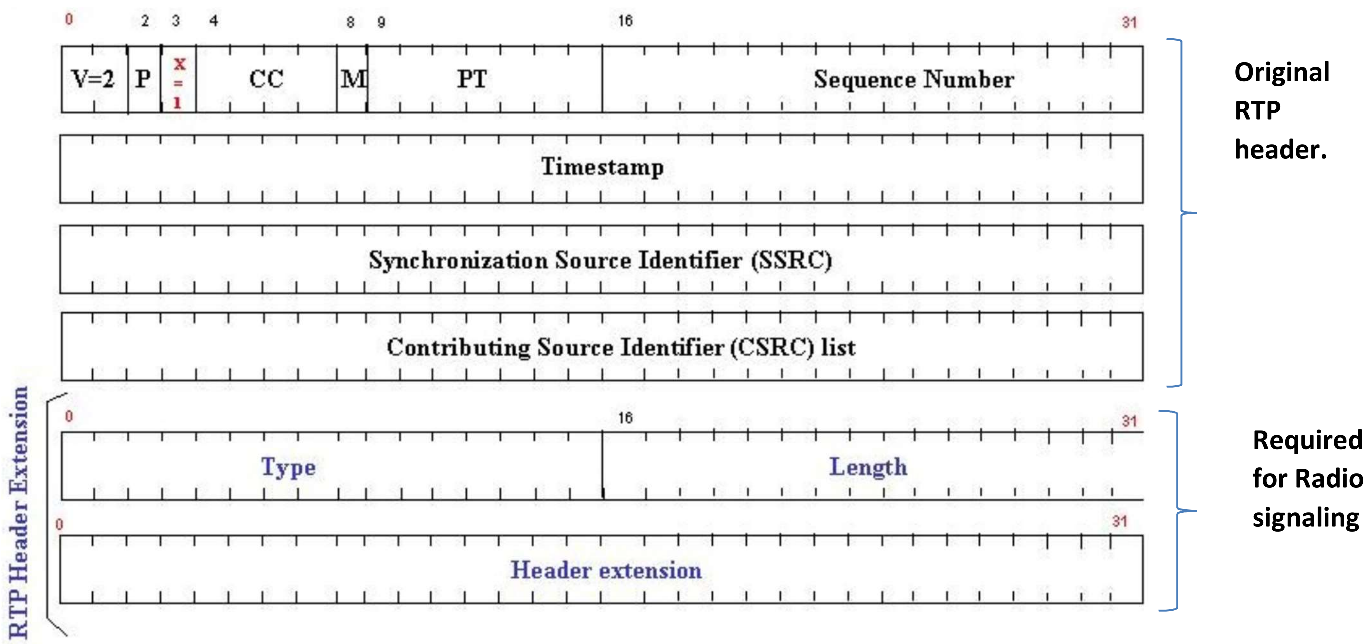

Figure 4. 8 RTP header plus RTP header extension (Darilion *et al*, 2004).

RTP HEADER EXTENSION BY USE OF ED 137B STANDARD

According to ED 137B document audio packets are supposed to carry signaling information for the following purposes.

1. From user to Transmitter to signal that transmission is requested.
2. From Receiver to User to indicate that audio is coming.
3. To monitor session when no audio transmission or reception.

The standard specifies the use the following packet structure to carry the signaling information.

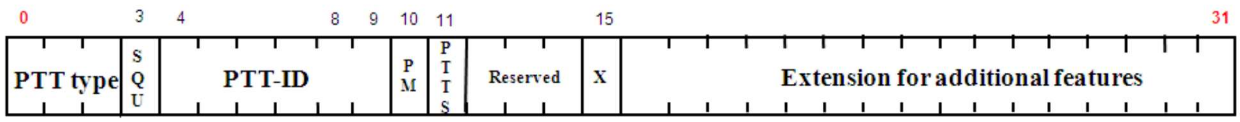

Figure 4. 9 RTP header extender (Darilion *et al*, 2004).

PTT-type (3 bits: b0 to b2)): This field defines the PTT-type sent by the VCS endpoint towards the Transceiver/Transmitter endpoint to activate transmission over the air.

SQU (1bit: b3): This field is used during the reception from the Radio. It is set to one otherwise 0

PTT-id (6 bits: b4-b9): This field is used by the VCS endpoint to send its PTTid previously assigned to it by the Radio endpoint during the SIP session establishment.

PM (1 bit: b10) – PTT Mute: This field is used for signaling PTT_ON to non-transmitting (not selected) transmitters in a coverage group.

PTTS (1bit: b11) – PTT Summation: This field is used by the VCS endpoint to indicate a PTT summation of multiple RTP audio streams in the VCS, e.g. if the corresponding radio interface in the VCS is configured for PTT summation and two users of this VCS are pressing PTT simultaneously.

Reserved (3 bits b12-b14): These three bits are reserved for future extensions.

X (1 bit: b15): This field indicates a marker bit, that SHALL be set to 1 if extended information for additional features is used.

Extension for Additional Features: The information in the other function block is coded in the Type-Length-Value (TLV) format. For backward compatibility, it SHALL be mandatory to set the proper values within the fixed part even if there is redundant information within the additional feature content. When the Extension field is not used, all bits SHALL be set to 0. With the above information regarding ED137B standard, we captured the requirements for Radio Communication for Air Traffic Control as shown in figure 4.10 below

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
V=2		P	x=		CC			M				PT						S	E	Q	U	E	N	C	E	N	U	M	B	E	R
			1																												
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
							T	I	M	E	S	T	A	M	P																
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
							S	S	R	C																					
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
							C	S	R	C																					
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
					T	Y	P	E											L	E	N	T	H								
0	0	0	0	0	0	0	1	0	1	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
PTT TYPE			SQ	PTT-1D						M	PTTS	RESERVED			X			EXTENSION FOR ADDITIONAL FEATURES													
0	0	0	0	X	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

THIS VALUE SHOULD BE RETRIEVED FROM SDP AFTER 200 OK RESPONSE HAS BEEN RECIEVED
IT WILL REMAIN THE SAME FOR BOTH TRANSMIT AND RECEIVE

NOTES:

1. BITS IN RED NEED TO BE CODED
2. PTT TYPE 000 AND SQ 0 IS FOR RTP PACKET THAT HAS NO AUDO (MIC MUTED)
3. PTT TYPE 001 AND SQ 0 IS FOR AUDIO TRANSIMISSION PTT BUTTON SELECTED
4. INCOMING PACKET WILL BE CHECKED FOR X=1 IF NOT REJECT THE PACKET. THEN IF SQ =1 AND PT-ID CORRECT THEN ALLOW AUDIO TO TO COME IN OTHERWISE IF SQ = 0 AND PTT-ID = CORRECT, RECEIVE PACKET (WITH DISABLED SPEAKER).

Figure 4. 10 Radio Communication for Air Traffic Control .

The conference bridge was also modified to take care of the half-duplex communicating. Figure 4.11 shows how communication will occur from the Air Traffic Controller (sound device) to Pilot (Alice) which is also called Transmission (TX).

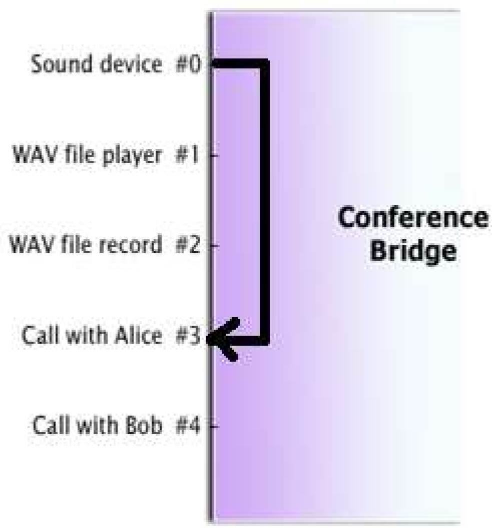

Figure 4. 11 Communication from ATC to Radio Transmission

On reception, the conference bridge will be configured as shown in figure 4.12 below

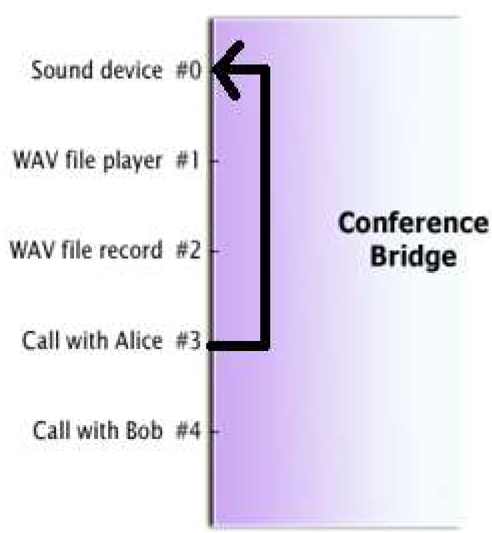

Figure 4. 12 Communication from Radio to ATC

From the fore going the following four algorithm were developed to address the requirements by ED 137 B.

a) **Algorithm for Half duplex communication**

1. If enable channel is pressed go to 2 otherwise check again
2. Connect slot 3 to 1 of the conference bridge
3. Connect slot 3 of the conference bridge to stream object
4. If channel is disabled go to 1 else go to 5
5. If PTT is ON go to 5 go to 6 otherwise go 4
6. Disconnect slot 3 from of conference bridge form sound device
7. Connect sound device to slot 3 of the conference bridge
8. If PTT is still ON check again otherwise go 2

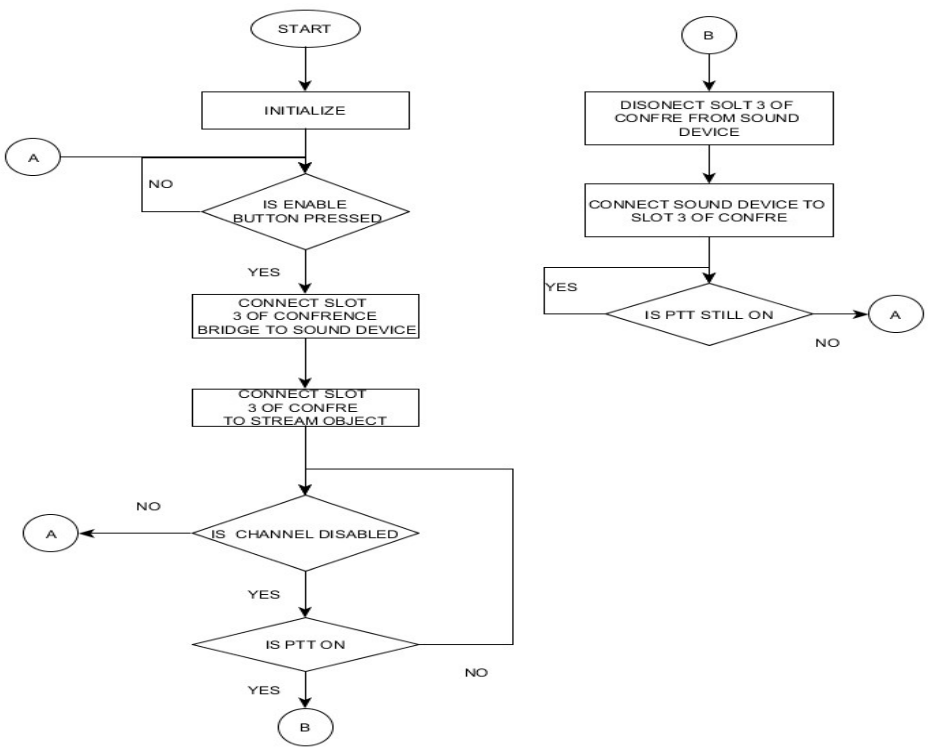

Figure 4. 13 Flowchart for half duplex communication

4.2.4 Algorithm for channel enabled and UP/DOWN link monitoring.

1. A user enables a channel via the user interface.
2. Set CH to EN to indicate transmission mode.
3. If channel is enabled and call is established go to 4 otherwise go to 13
4. Generate Real Time Packet RTP audio packet with no audio
5. Set bit X of the RTP packet header to 1 to indicate RTP packet header extension to follow.
6. Create RTP packet header extension to carry signaling information.
7. On RTP header extension set Push to talk and Squelch to ON.
8. Append RTP header extension to RTP packet.
9. Transmit RTP packet to Radio
10. Disable channel. (set to receive mode)
11. sleep for 200 ms (time required before reply is received)
12. If channel is enabled and call is established go to 4 otherwise go 13
13. If call is established go to 13 otherwise go to go to 19
14. Receive RTP packet from the Radio.
15. If Push to Talk and Squelch is set to ON, then go to 16 otherwise go to 20
16. Radio session up.
17. sleep for 200 ms (minimum time after reception before transmission of the next RTP real time packet)
18. Go to 2
19. Stop
20. Radio session down
21. Go to 19

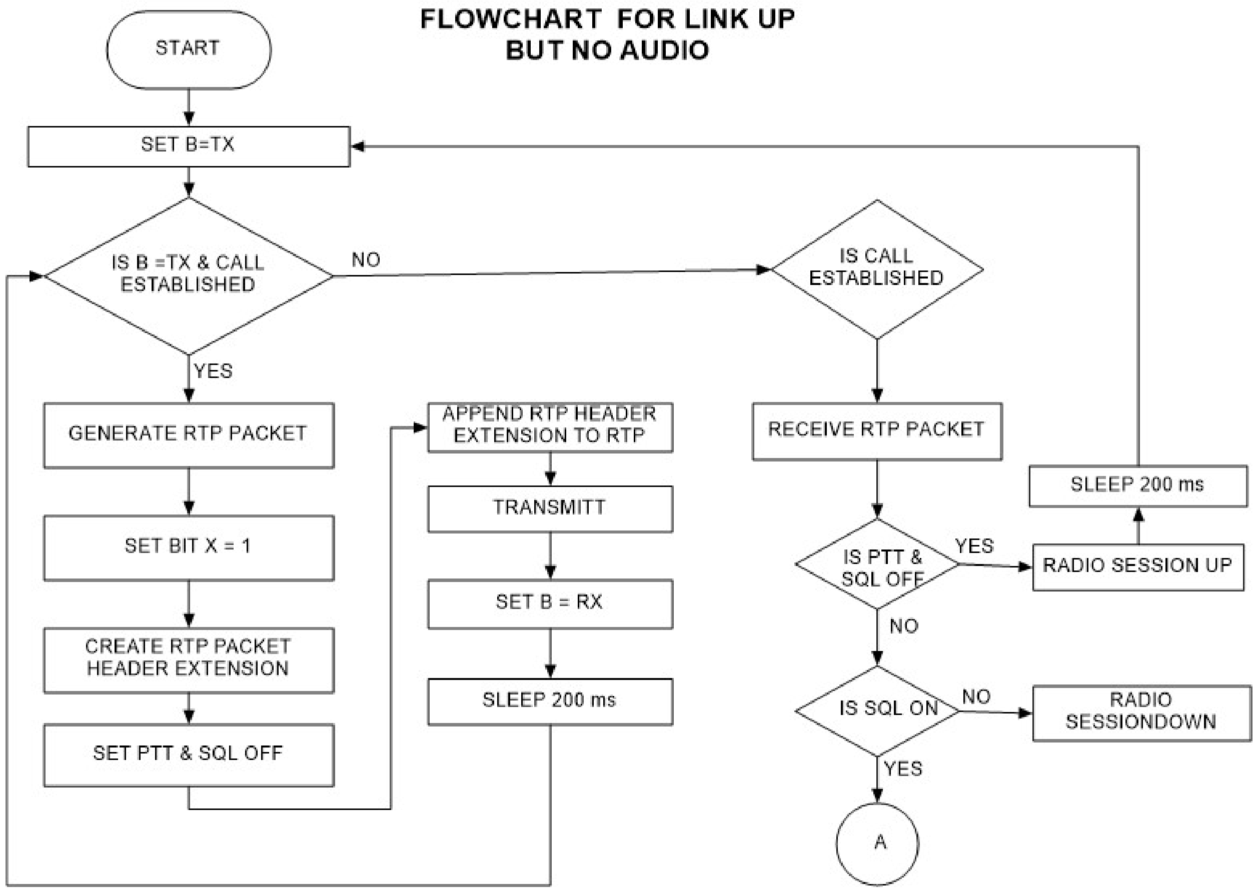

Figure 4. 14 Flowchart for channel enabled and UP/DOWN link monitoring

4.2.5 Algorithm for Audio packet transmission.

1. Through the user interface a user enables transmission of audio packets
2. If transmission is enabled, go to 3 otherwise go to B
3. Receive RTP audio packet from user.
4. Temporarily store the RTP packet
5. Create RTP header extension
6. Set Push to Talk on RTP header extension to ON
7. Append RTP extra header to RTP packet
8. Transmit RTP audio packet to Radio
9. Go to 2

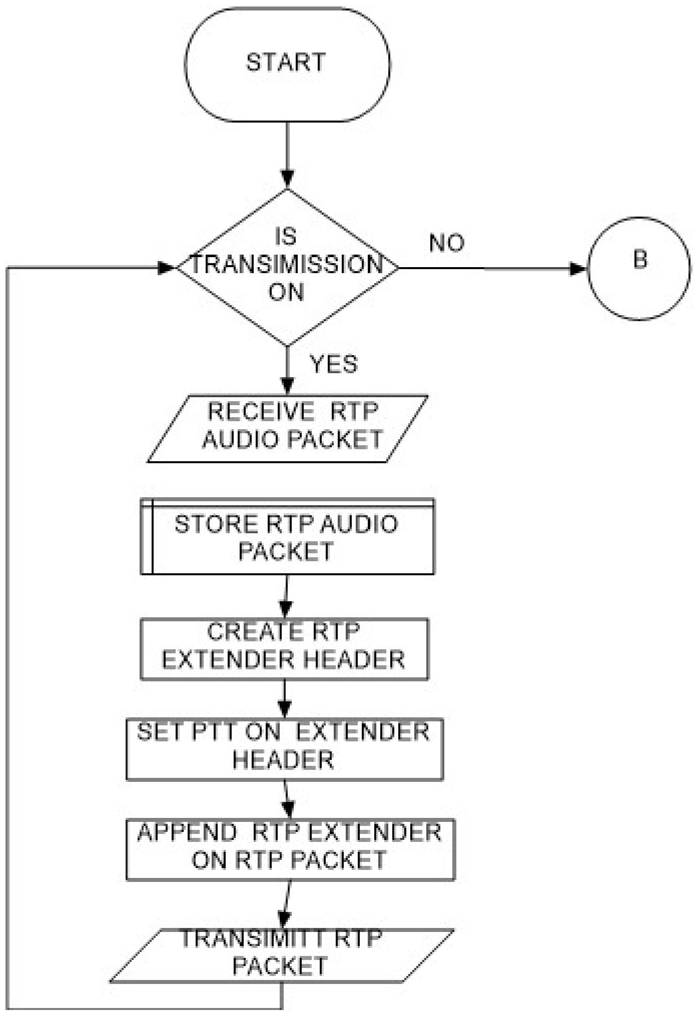

Figure 4. 15 Flowchart for audio transmission

4.2.6 Algorithm for Audio Reception.

1. If reception is ON, go to 2 otherwise go to 6 --
2. Receive and temporarily store RTP audio packet
3. If squelch is ON, go to 4 otherwise go to 6
4. Transmit audio packet
5. Go to 1
6. Sleep 200 ms
7. Go to C

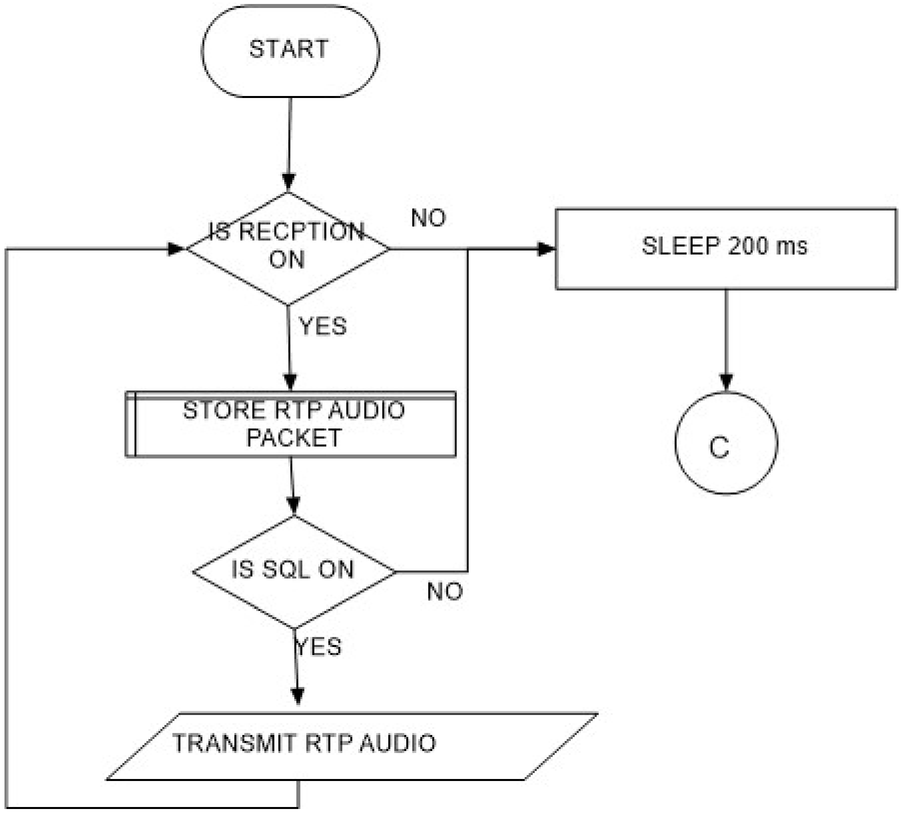

Figure 4. 16 Flow chart for audio reception.

4.2.7 The software

This section describes the software which include the graphical user interface, the modified application (SIMPLEUA.C) and the complete library stack showing the application.

4.2.6.1 Graphical user interface

The program window displays a radio channel as a button each for each channel. The buttons function as input for commands as well as visual indicators of the channel status. Figure 4.17 below shows.

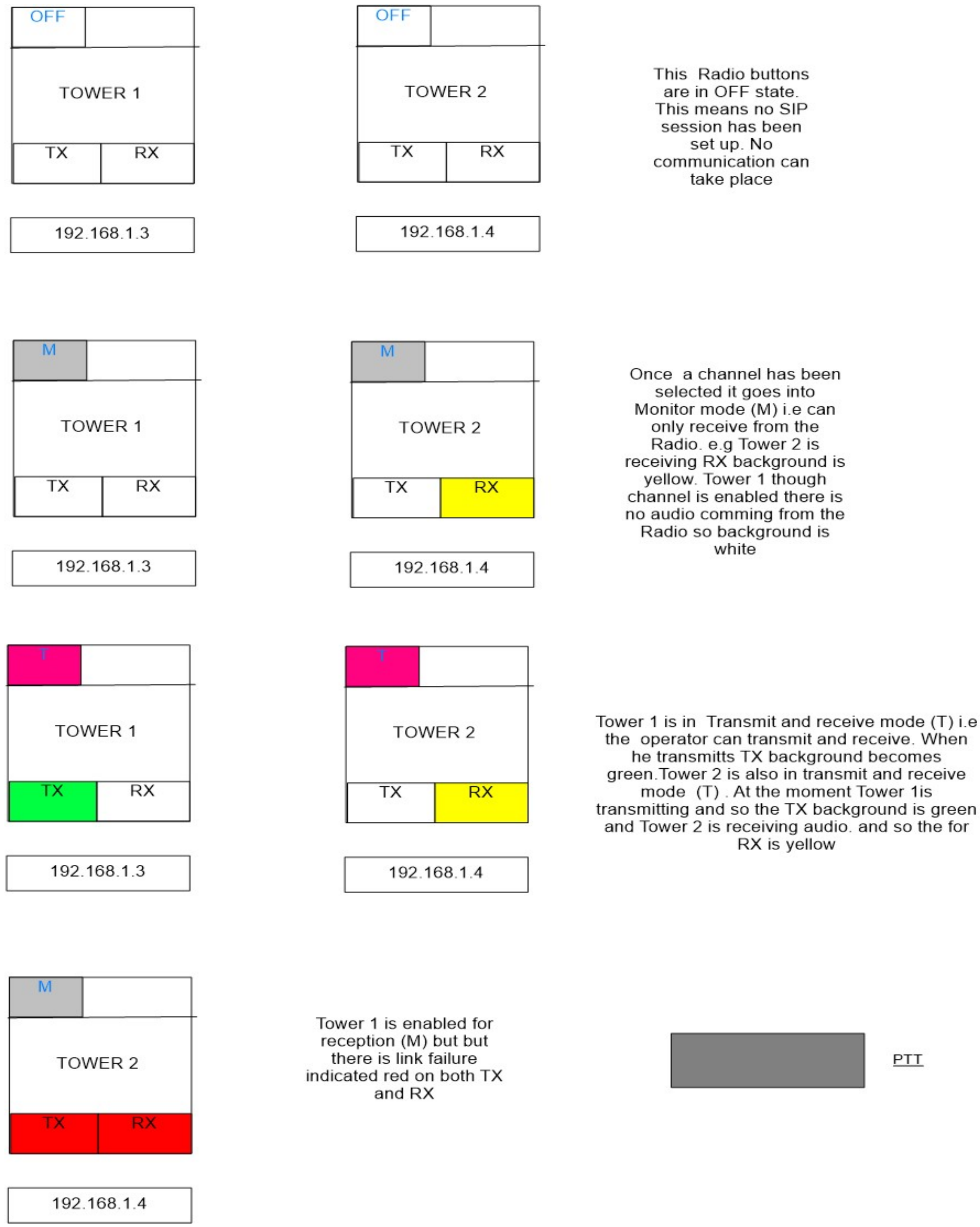

Figure 4. 17 Graphical user interface

Description of the Graphical user interface:

1. **OFF state:** When application starts the radio channels will be OFF state No. communication can take place.
2. **MONITOR state (M):** pressing Radio channel in OFF state will cause the state of the Radio channel to change from OFF sate to MONITOR state which will allow the operator to receive audio from the Radio. In case there is audio the signaling information Squelch will cause background of yellow to appear on RX writing.
3. **TRAFFIC State (T):** Pressing Radio channel when in MONITOR state will cause the channel to change from the channel to change to TRAFFIC state. In this state the operator can transmit audio by first pressing the PTT button that will cause generation of singling information (PTT) to be associated with all channels in Traffic state so that audio can be transmitted.
4. **LINK STATE MONITORING:** When in either MONITOR state or TRAFFIC state and there is no audio transmit or receive the VCS and Radio will exchange packets for monitoring the link. In case the link fails to be ON, TX and RX background will become RED.

4.2.6.2 Application: For the modified application (SIMPLEUA.C) see appendix A

4.2.6.3 PJSIP library plus modified application

Figure 4.18 below shows the complete library stack together with the modified Application SIMPLEUA.c as the final product.

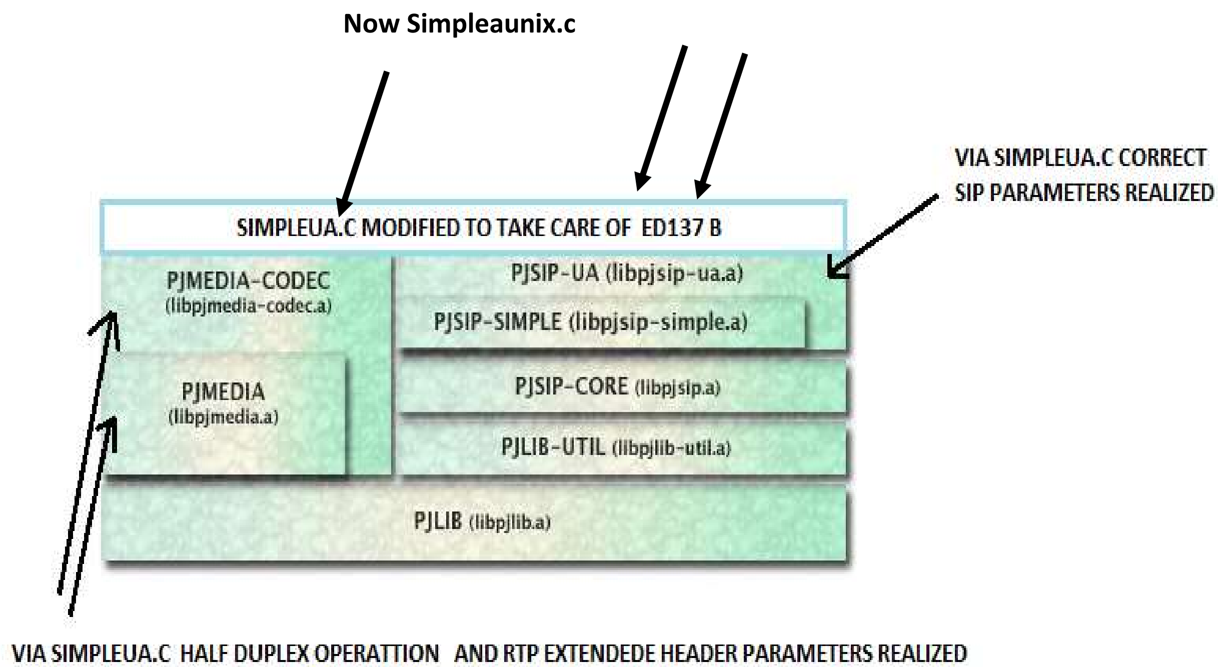

Figure 4. 18 PJSIP with modified library

4.2.8 Hardware: Friendly ARM Tiny210.

The Tiny210 single board computer manufactured by Friendly ARM shown in figure 3.10was chosen among many single board computers that were evaluated. Friendly ARM Tiny210 offers excellent features' that can support a real time application like VOIP. The CPU board is a high-performance Cortex A8 core board uses Samsung S5PV210 as the main processor, running at up to 1GHz.Tiny210 onboard 512 M DDR2 memories can smoothly run advanced operating system, Android, Linux and WinCE6.

The following is a summary of Tiny210 board features:

- **CPU**: 1 GHz Samsung S5PV210 with PowerVR SGX540 graphics engine
- **DDR2 RAM:** 512MB DDR2 RAM, 32bit data bus, 200MHz
- **FLASH**: SLC NAND Flash: 256MB/1GB
- **Multi-IO:**

2 x 60 Pin 2.0mm space DIP connector

1 x 30 Pin 2.0mm space DIP connector

1 x 51 Pin 1.0mm space SMD connector

On Board:

- HDMI interface
- 4 x User Led (Green)
- 1 x Power Led(Red)

Supply Voltage from 2V to 6V

Mechanical: Dimension: 64x 50x 11mm

Integrated TFT touch screen

It should be noted that the friendly ARM single board computers support TFT touch screens unlike manufactures like Raspberry and Arduino whose boards are not supported by a wide range of TFT screens

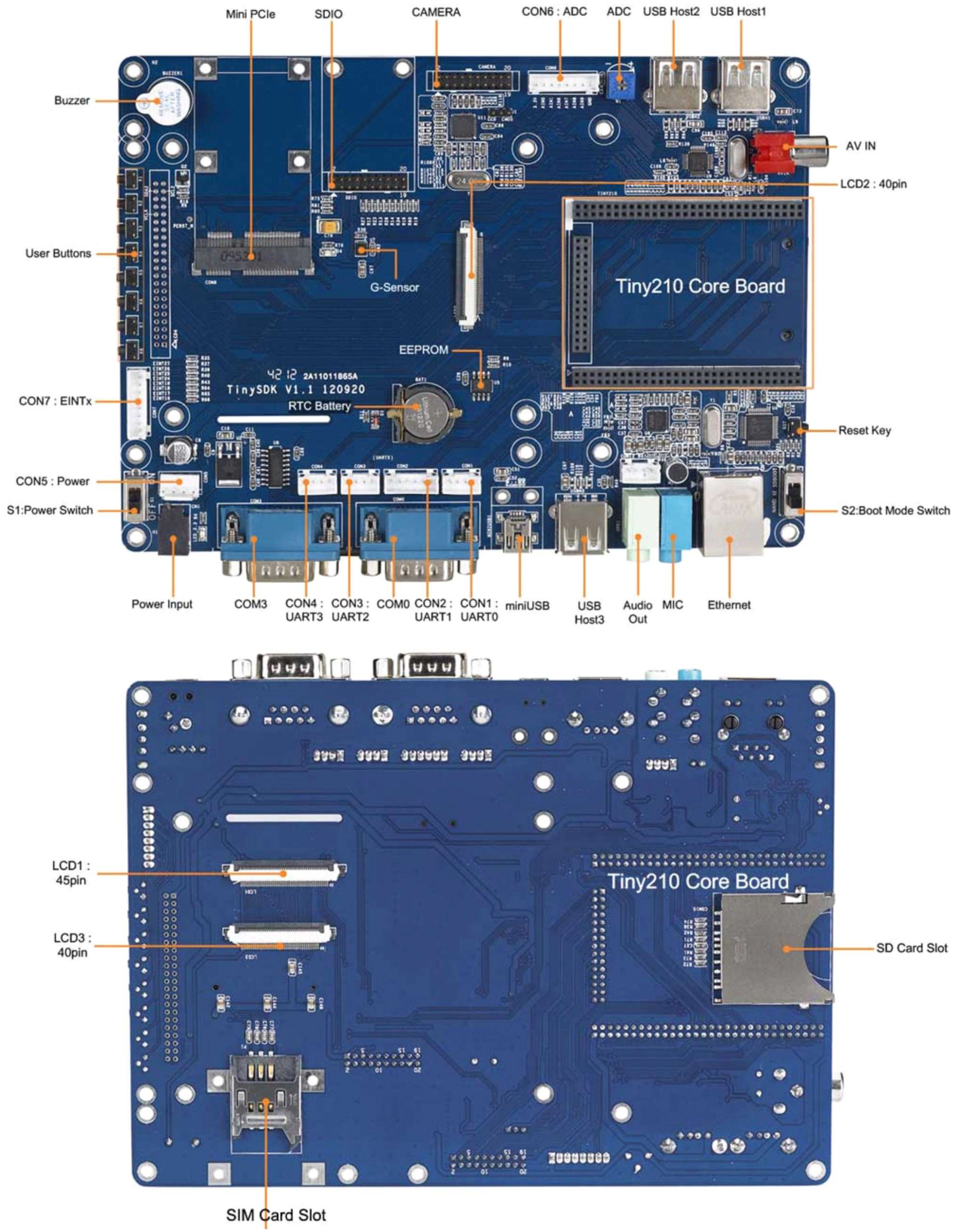

Figure 4. 19 Friendly ARM Tiny210

4.2.7.4 Operating system: LINUX operating system

Linux kernel is monolithic with loadable modules support and provides multitasking capabilities, shared user-space libraries or multitask networking including IPv4 and IPv6. It self does not have real-time capabilities as required by VOIP application — there are generally two ways to add real-time needed features:

- the dual kernel approach and
- Making the kernel natively preemptive.

Dual kernel approach

This method introduces a new kernel (usually a micro- or nanokernel) which runs the whole non-RT Linux system as a thread with the lowest priority. The communication between RT and non-RT tasks is commonly possible using shared memory or FIFOs (First in First Out). This method was not adopted because RT-Linux requires a steep learning curve which would not have gone well in line with short time required for the project.

Modification of the standard Linux kernel to become preemptive

The other approach modifies the standard kernel to enable running both non-RT and RT tasks. A patch called RT-Preempt path was used to modify the standard Linux kernel resulted in making almost all previously uninterruptible system calls pre-emptible. The use of Linux-based operating system was probable from the beginning due to its high modularity it can be run in a variety of forms ranging from small embedded applications to multiprocessor supercomputers. The Linux kernel is one of the kernels optimized to run on the Friendly ARM Tiny210 and by applying a real time patch real time capabilities can be achieved. Tests (Hedera, 2014) have been before to compare real time performance between a standard Linux kernel and a Linux kernel that has been modified with RT-Preempt and the results showed when two systems are fully loaded latency exhibited on the non-modified kernel was 2465µsec and on the modified Linux the latency was 58µsec. The latency exhibited by the modified Linux kernel as compared with the recommendation of ITU-T G.114 recommends a maximum of 150ms one-way latency.

CHAPTER 5 EVALUATION

The following section describes the evaluation tasks that were conducted for evaluation. First we describe the implementation details which highlights the prototype and the objects it was supposed to meet. Next we describe the experimental setup used to perform the evaluation exercises. Finally, we highlight the results of the evaluations.

5.1 Implementation details

The objective of this project was to design an IP based voice communication system for Air Traffic Control using open source software. The IP based Voice Communication system is based on ED 137B Eurocae standard which has been adopted by the International Civil Aviation Authority ICAO.

Four algorithms were designed. As indicated below:

- Algorithm for half duplex communication
- Algorithm for channel enabled and link up/down monitoring
- Algorithm for audio packet transmission
- Algorithm for audio packet reception.

This resulted in final product PJSIP plus application called simple UNIX

5.2 Experimental setup (Simulation setup)

In order to evaluate the prototype, the following setup was employed as shown in the figure 5.1 below.

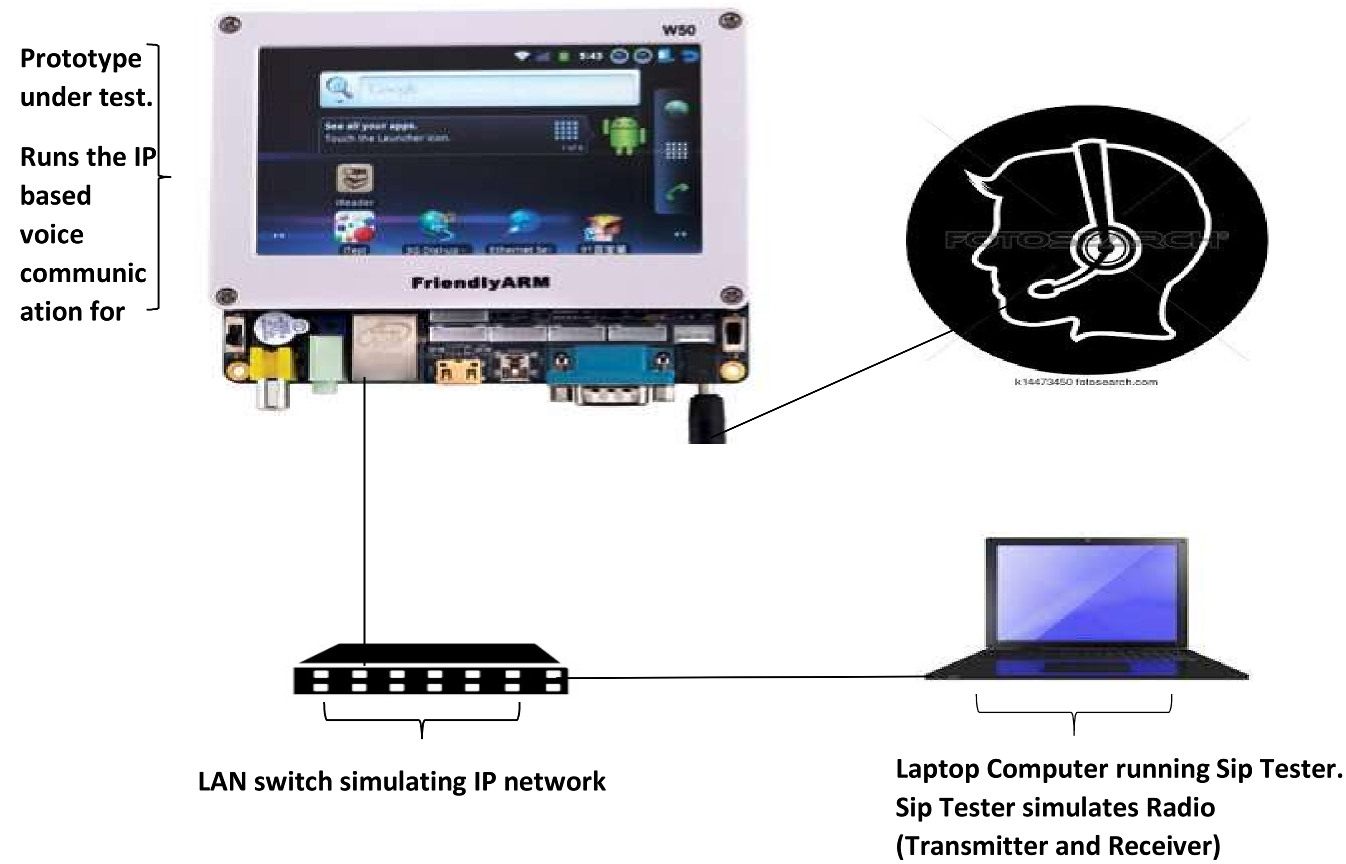

Figure 5. 1 Evaluation setup

The setup consists of IP based voice communication system implemented on the friendly ARM computer. The LAN switch simulates the IP connectivity between end equipment**.** The Laptop computer running Sip Tester simulates the Radios used by Air traffic controllers to communicate with pilots. In order for the simulation to take place the following CallXML scripts performing different function of the Radio were ran on the Laptop computer.

RTP EXTENDED HEADER ATTRIBUTES FOR THE SCRIPTS

Hex Data (optional) - generic RTP extension data in hexadecimal text format. Example: '01C04FFA' **extension ID** (required if 'hex Data' is set) - identifier of RTP extension data format, defined by audio/video profile **wg67_pttType** (optional) - PTT-type field, defined in <u>INTEROPERABILITY STANDARDS FOR VOIP ATM COMPONENTS VOLUME 1: RADIO</u>. If **wg67_pttType** is set to *"rxptt"*, whereby from received RTP it gets PTT-type, i.e. it creates PTT-type RX-TX loopback connection. This loopback connection is used to measure round-trip

RTP audio delay for ED-137 performance testing of air traffic management (ATM) VoIP networks. **wg67_pttId** (optional) - PTT-ID **wg67_squ** (optional, 1 or 0, default is 0) - SQU flag. If **wg67_squ** is set to *"rxptt"*, from the PTT-Type field of received RTP stream a loopback connection is created: If RX-PTT-ID is 0 then TX SQU is 0, otherwise TX-SQU is 1 **wg67_pm** (optional, 1 or 0, default is 0) - PM flag **wg67_ptts** (optional, 1 or 0, default is 0) - PTTS flag **wg67_sct** (optional, 1 or 0, default is 0) - SCT flag.

1. **RTP EXTENDED HEADER EVALUTION FOR SIP AND RTP HEADERS**

```
<callxml>

<accept headers="WG67-Version=radio.01" />
<wait value="2000ms" />
<setrtpextension wg67_pttType="1" wg67_pttId="16" wg67_squ="0" wg67_pm="0"
wg67_ptts="0" wg67_sct="0" />
<playaudio value="music.wav" maxtime="1000s" />
<wait value="400ms" />
</callxml>
```

2. **RTP EXTENDED HEADER EVALUATION FOR R2S (REAL TIME SESSION SUPERVISION PROTOCOL) FOR ED-137**

```
<callxml>
<acceptheaders="WG67-Version=radio.01"sdpAttributes="R2S
KeepAlivePeriod:200|R2S-KeepAliveMultiplier:10|sigtime:1" disableRtp=" true" />
<wait value="2000ms" />
<setrtpextension wg67_pttType="1" wg67_pttId="16" wg67_squ="0" wg67_pm="0"
wg67_ptts="0" wg67_sct="0" />
<playaudio value="music.wav" maxtime="1000s" />
<wait value="400ms" />
</callxml>
```

3. LOOPBACK TEST USING RTP EXTNDED HEADER

The following screen shorts were the outputs of the above CallXML scripts as captured on the simulator

INVITE request from CWP (VCS) to the Radio simulator.

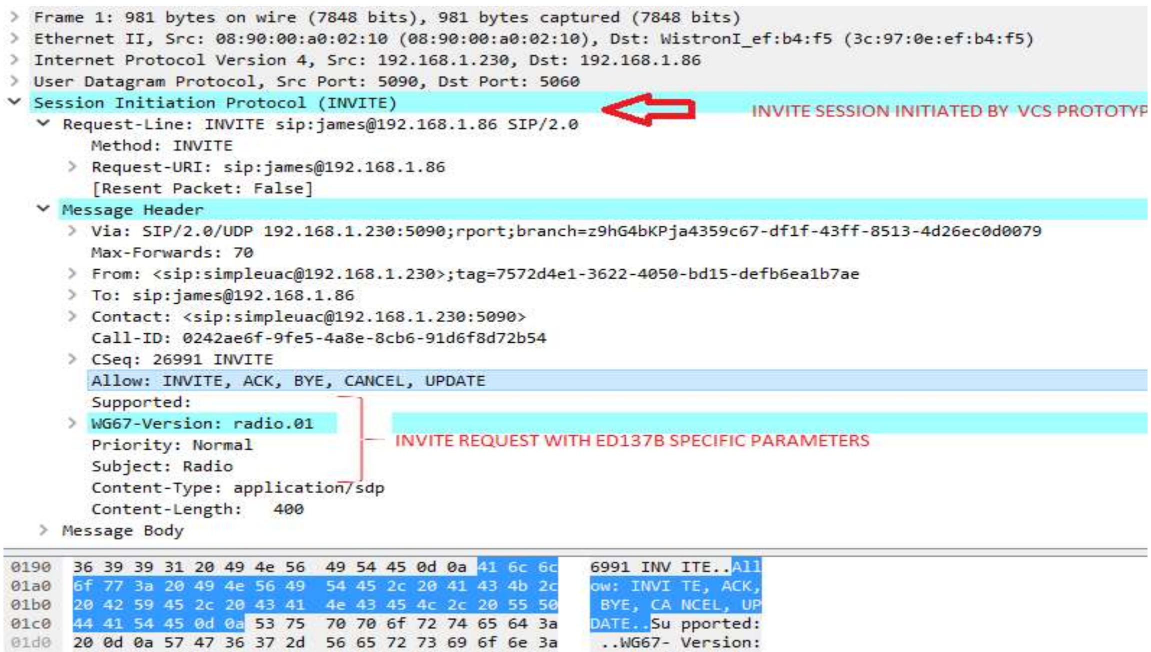

Figure 5. 2 screen short for INVITE request from the prototype

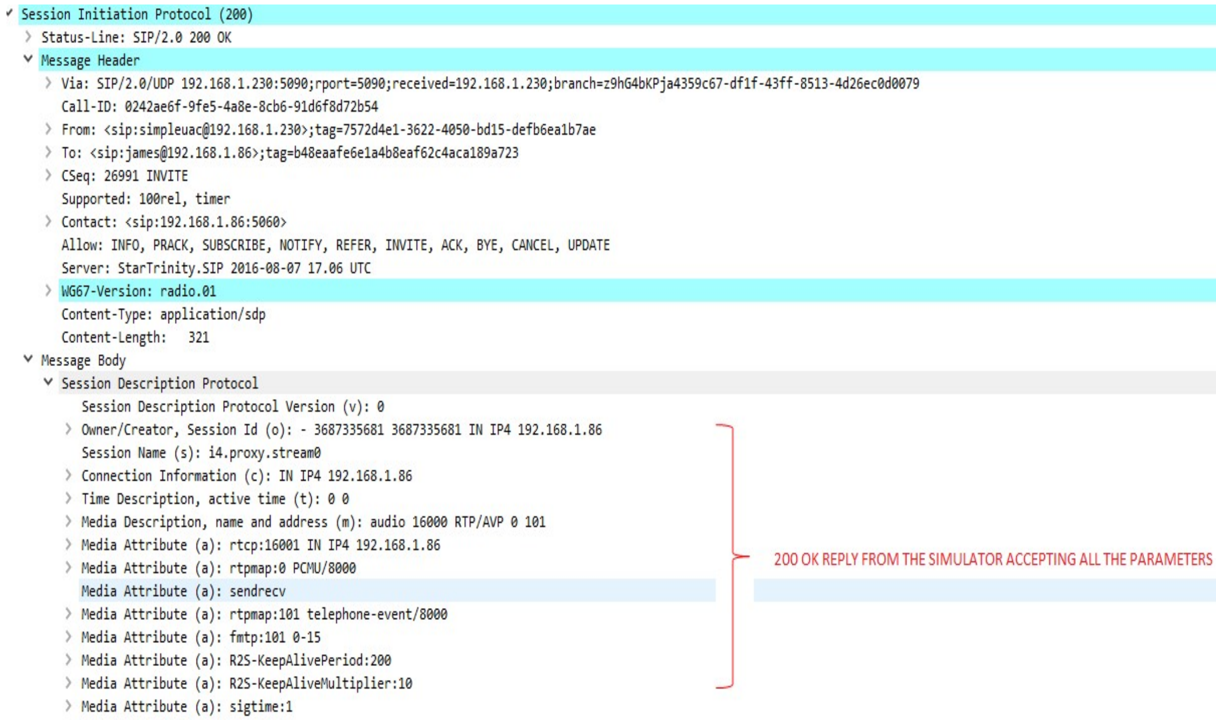

Figure 5. 3 Screen short for 200 OK ACK Response

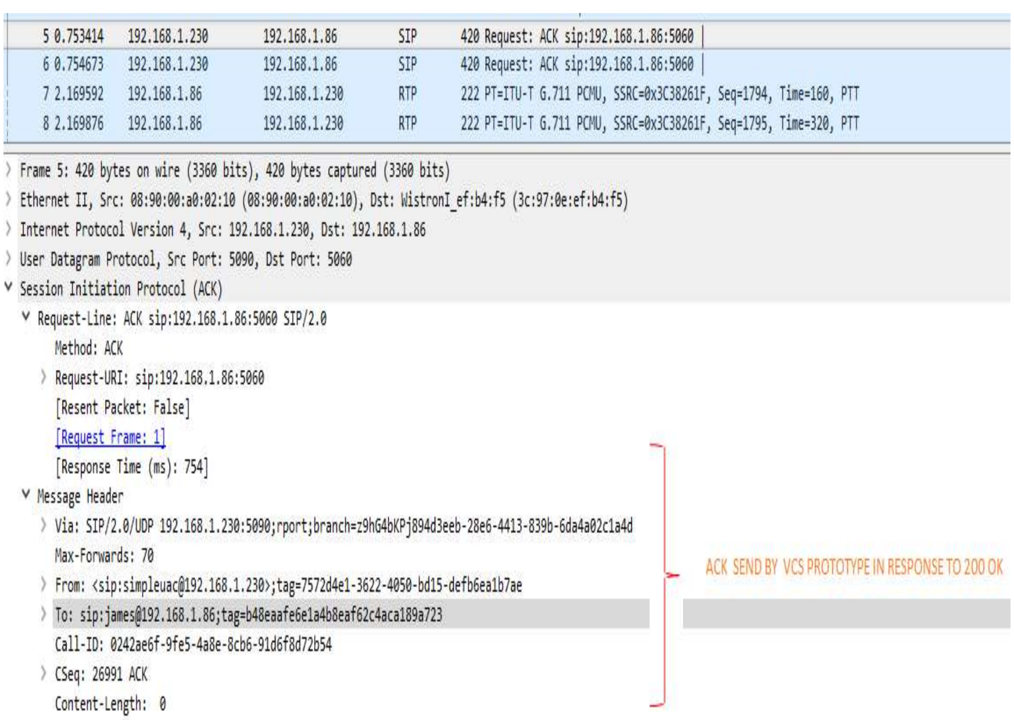

Figure 5. 4 ACK RESPONSE TO 200 OK.

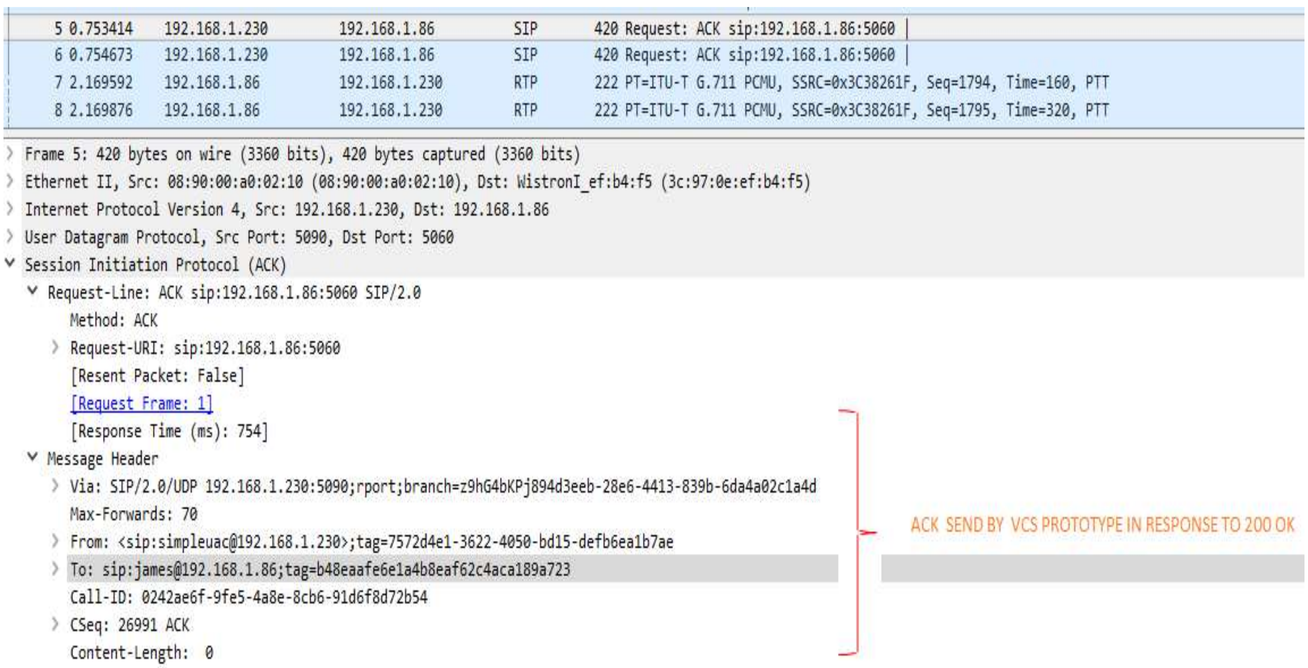

Figure 5. 5 RTP with extension header indication.

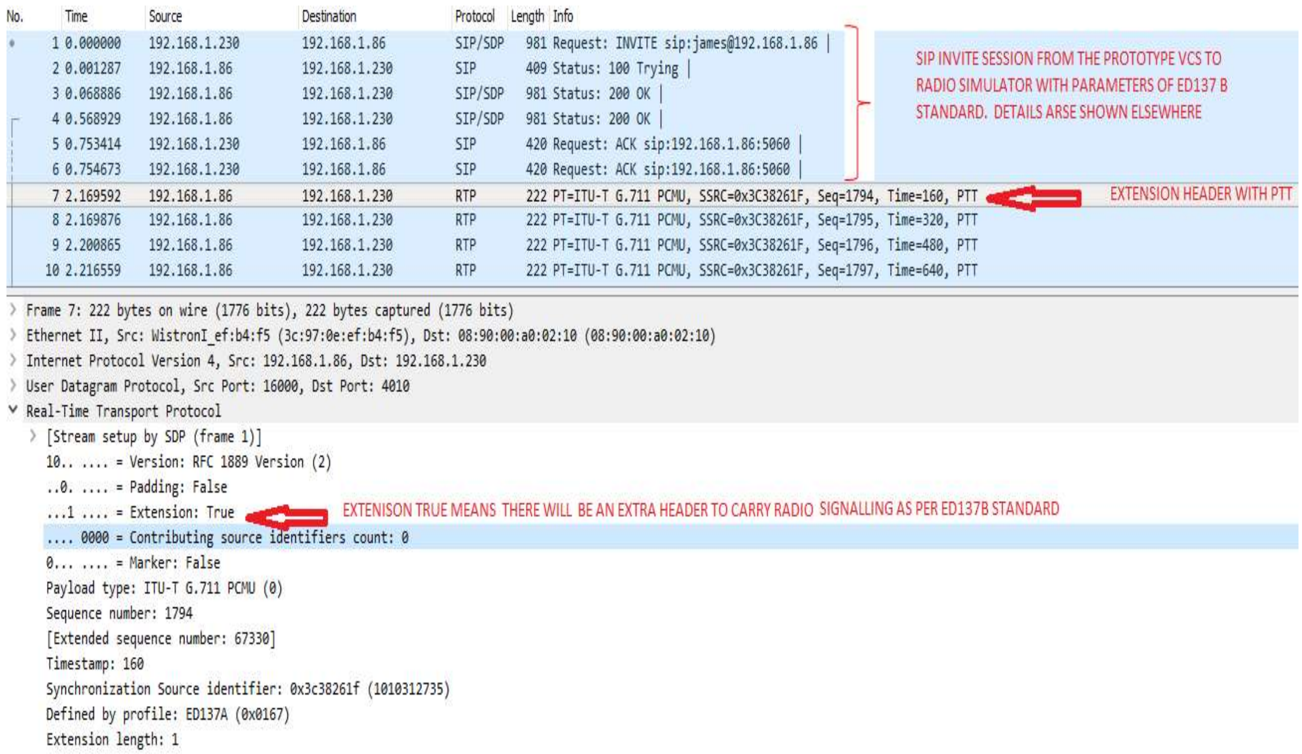

Figure 5. 6 RTP header extension detail

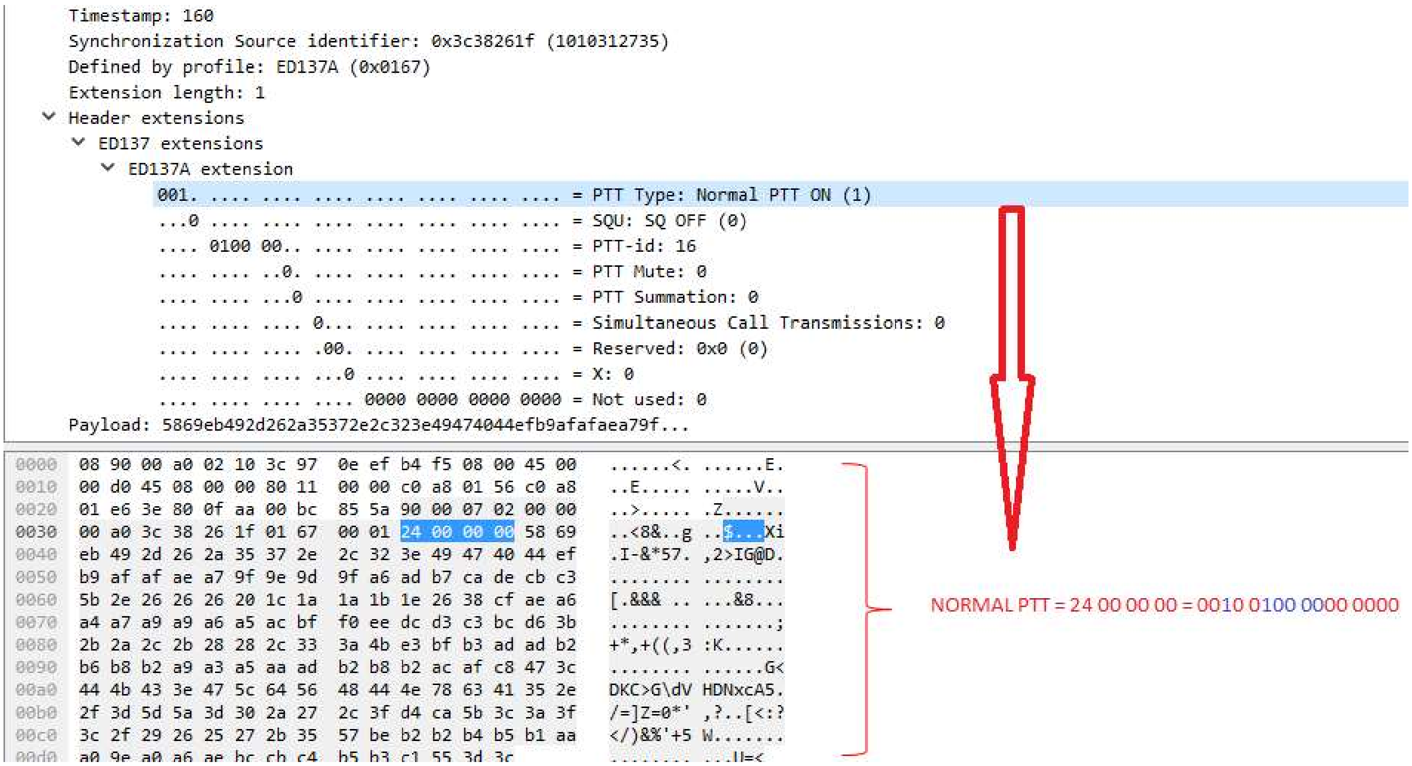

Figure 5. 7 PTT encoding in the extended header

CHAPTER 6 CONCLUSION

This section discusses the results obtained during evaluation exercise. We also discuss how the seven DSR research guidelines as suggested by (Hevner *et al.,* 2010) were met. Finally, we discuss contribution and future works.

6.1 Results

Table 5. 1 Results

<table>
<tr><th colspan="3">(A) RADIO COMMUNICATION MODEL</th></tr>
<tr><td>3</td><td>[COMMUNICATION MODEL] Applicable Protocols</td><td>RESULTS</td></tr>
<tr><td></td><td>The SIP, RTP and R2S protocol SHALL be the minimum requirements necessary for the implementation in order to provide VoIP communication between the User Agents at the VCS and GRS endpoints.</td><td>SIP and RTP headers requirements were met as can be seen in the INVITE and RTP header extension header screen shorts</td></tr>
<tr><td></td><td>[COMMUNICATION MODEL] Communication initiation between VCS and combined GRS</td><td>RESULST</td></tr>
<tr><td></td><td>In the case that a GRS Transceiver or a GRS Transmitter/Receiver located at the same site and accessible by one SIP URI, the communication between the VCS endpoint and a GRS endpoint SHALL be performed in two distinct phases.
Phase 1:
The SIP session shall always be initiated from the VCS endpoint towards the GRS endpoint (transceiver, transmitter or receiver).
Phase 2:</td><td>This requirements was partially met.
Phase 1 of the requirement was met. In all evaluation exercise, the initiation of the session was from the IP based voice communication system to the Radio (simulated).
Phase 2 was not fully met. The prototype could receive the keep alive messages but could not produce response</td></tr>
</table>

	Once the SIP session is established, both VCS and GRS endpoints shall use the "Keep Alive" mechanism of the R2S protocol to control the link between the VCS and the GRS. In the case where audio is present, the R2S-Keepalive packets will be exchanged between endpoints.	
(B) PROFILE STANDARD FOR THE USE OF SIP IN AN AGVN		
1	**[SIP] SIP Version**	**RESULTS**
	An Air Traffic Services VoIP Communications System SHALL support SIP version 2 as specified in RFC 3261 [12].	As per the INVITE screen short this requirement was met
2	**[SIP] SIP Supported requests – see ED137/1B document**	**RESULTS**
3	**[SIP] SIP Supported response– see ED137/1B document**	
4	**[SIP] SIP Message body (SDP**	**RESULTS**
	Those SIP message bodies containing a description of the session, time and media SHALL be encoded in the Session Description Protocol (SDP) (RFC 2327 [7]).	This was met. The SIP meassage contained all the parameters required.
7	**[SIP] Normal SIP session establishment** SIP session establishment request sent by a VCS endpoint to a Ground Radio Station end point in normal operational conditions **SHALL**	This was met. The SIP Priority header was set to normal and the subject header to radio as can be seen from the INVITE request message

	use a SIP Priority Header field set to “normal” and a SIP subject header field set to “radio”.	
	(C) AUDIO	
2	**[AUDIO] Voice quality**	**RESULTS**
	The voice quality of a radio communication is defined using a voice quality estimation methodology nominated “Mean Opinion Score” (MOS) rating.	This requirement was met. The sound from the speakers was audible enough
3	**[AUDIO] Voice latency time performance.**	**RESULTS**
	The system delay shall respect ITU-T Recommendation G.114	This requirement was met
4	**[AUDIO] Voice Packetization interval requirements.**	**RESULTS**
	The VCS and GRS endpoints SHALL communicate using voice packet sizes of 10, 20 or 30ms.	This requirement was partially met. 20ms voice packet size was used
5	**[AUDIO] Voice coding requirement.**	**RESULTS**
	The VCS and GRS SHALL support the following voice codec according to ITU-T G.711PCM A-law or μ-law G.711 PCM. In order to improve robustness, the ITU-T G.711PLC codec [34] MAY be used;	This requirement was partially met. ITU-G.711PCM A-law was used as can be seen from the INVITE request
(D) RTP: REAL-TIME TRANSPORT PROTOCOL		
1	**[RTP] RTP Audio and Radio Signaling protocol requirement.**	**RESULTS**
	Within an IP-network, the audio transmission and specific radio signaling SHALL be performed by the Real-time Transport Protocol (RTP).	This was met as per screen shorts of the RTP header
5	**[RTP] RTP PTT transmission performance.**	**RESULTS**

	PTT signal is used to activate transmission at the GRS transceiver/transmitter. It is activated when the controller at the VCS endpoint selects the PTT key at the Controller Working Position.	The screen short on RTP header extension gives details on how PTT was used to activate transmission.
6	**[RTP] Squelch transmission performance.**	**RESULTS**
	Squelch) signal is active when the GRS transceiver/receiver detects an incoming radio call.	
7	**[RTP] RTP Header Extension description**	**RESULTS**
	The RTP header extension is used to transmit additional information necessary for radio communication. (I.e. PTT activation info, Squelch indication, signal quality index, etc.). The extension SHALL be implemented according to RFC 3550 [21].	This was met. The RTP header extension contains the fields as per the Eurocae standard.
	(E) REAL TIME SESSION SUPERVISION	
16	**RTP] Keep alive messages**	**RESULTS**
	The Real Time Session Supervision SHALL be employed between VCS endpoints and GRS endpoints.	

From the fore going it can be shown that the objective of the project was met as follows

1. Design of an IP based voice communication system for Air Traffic Control.

   This objective was partially met. From the table of results, it can be seen that SIP component was full met but on RTP component some aspects of the signaling were met.

2. Implement code to make prototype.

   This requirement is highly linked to requirement number one and so it was also partially met. For example, R2S signaling was not met.

3. Simulate input data and use it on the prototype.
   This objective was fully met. We were able to simulate data and use it on the prototype to test the functionalities that worked.

4. Use standard evaluation methods to perform an assessment

   This objective was met. We employed experimental evaluation through simulation. Experimental through simulation presuppose that the artefact is tested by use of artificial data. By use of CallXML scripts we were able to simulate Radio functionalities that would communicate with prototype

6.2 Compliance with Design Science Research DSR guidelines

Using DSR research guidelines as suggested by (Hevener *et al*, 2010) and indicted in table 5.2 we discuss the compliance of this project to the guidelines

Guideline	Description
Guideline 1: Design as an Artifact	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
Guideline 2: Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3: Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4: Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

Table 5. 2 Guidelines for Design Science Research DSR guidelines as suggested by (Henver *et al*., 2010).

Guideline 1. Design as an artifact: This project has resulted in the development of four algorithms which have been used to build an IP-based Voice Communication Control System for Air Traffic Control using open source software.

Guideline 2. Problem Relevance: The use of open source software in the design is a response to a research opportunity that has emerged due to the development of a new standard for voice communication in Air Traffic Control ED 137 B by Eurocae which has been adopted by ICAO. The new rule is based on IP communication.

Guideline 3. Design Evaluation: To evaluate the usability and efficacy of the design, the design was subjected to Simulation test which is design evaluation method as suggested by (Hevner *et al*., 2004).

Guideline 4. Research Contribution: The objective of the project was to develop an IP based Voice Communication Control System for Air Traffic Control as per the Eurocae standard (ED 137 B). We developed four algorithms as indicated in the Development chapter. These algorithms were then incorporated into PJSIP library which was then evaluated. The results of the evaluation showed some of the requirements were met. One of the key requirement that was met that communication could be initiated from the prototype with all the extended header parameters to actuate (PTT) the Radio for transmission. We therefore helped unpacked the complex logic ED 137 B standard for VOIP communication in ATC to the open source software. Though this is not a full-fledged voice communication, we can get that we set design foundation for open software.

Guideline 5. Research Rigor: The development and evaluation of the artifact employed standard methods such as software reengineering. In the evaluation stage, we applied standard evaluation process Simulation test as suggested by (Hevner et al. 2004) We also assisted the developer MR. Sergey Alhesin (website) of the SIP tester simulator software to discover a bug in his software which had hindered our progress in the evaluation stage. The above was acknowledged by him.

Guideline 6. Design as a Search process: In meeting this requirement and because a design is an iterative process we employed the general methodology for design research as suggested by (Vaishnavi and Kuechler, 2015). The stages of development and evaluations have loops to the analysis stage which emphasizes rework to fine tune the end product.

Guideline 7. Communication of the Research: The research findings were presented as part fulfillment Masters of Science in Distributed Computing technology. We also presented in a journal suitable for Design science research

6.3 Future works.

This study and past research have sought to gain a greater understanding of the use of open source software to include ED 137B standard for use in Radio communication for Air Traffic Control. This research, in particular, has analyzed how to design the algorithm by coding the algorithm to make a prototype and then stimulate input data and use it in the prototype. Further longitudinal research is needed over an extended period to determine the how this idea has helped in controlling air traffic control. Additional research could also investigate the challenges facing this idea to look for ways of making it perfect.

REFERENCES:

1. APANPIRG/26, (2015) Available at https://www.icao.int/APAC/Meetings/Pages/2015-APANPIRG26.aspx

2. Bardac, H., Kampichler, W., P Göschka, K. M. rinz, J. and (2003). IP communications in ATC. E & i: Elektrotechnik und Informationstechnik, vol. 120,

3. Brad Massey (2011) Breakthrough Capabilities and Value for Mission-Critical Computing Intel and Red hat

4. Clarkk, Wu, Q.,, G. and A., Zorn 2013. RTP Control Protocol (RTCP) Extended Report (XR) Block for Discard Count Metric Reporting (No. RFC 7002).

5. Darilion, K. Göschka, , C., Kampichler, W. and M, K.,, Kurth (2004). A service environment SIP based for air traffic control of 37th Annual Hawaii International Conference on System Sciences (HICSS-37). 5–8

6. DOC9750-AN 2016, https://www.icao.int/publications/Pages/Publication.aspx? docnum = 9750

7. Dugerdil, (2006 February)"Architecture-based software reengineering", Isnet62 Project.

8. Free Software Foundation (2017) GNU General Public License http//:www.fsf.org

9. Handley, M. and Schooler, E. (2002). SIP RFC.

10. Hevner et al, (2010) Design Research in Information Systems

11. Himeda A and Babiker A (2016) Redundancy of Air traffic management networks. University of Khartoum.

12. ICAO http://www.icao.int

13. IETF http://www.ietf.org

14. Johnston, A.B., 2015. SIP: understanding the session initiation protocol. Artech House. https://books.google.co.ke/books?hl=en&lr=&id=GkOPCwAAQBAJ&oi=fnd&pg=PR5&dq=session+initiation+protocol+sip&ots=G2w1whidLr&sig=1WUEXS4-

UyF47YRIVNQbxV5b-NU&redir_esc=y#v=onepage&q=session%20initiation%20protocol%20sip&f=false

15. Lehman, "Programs, life cycles, and laws of software evolution", Proceedings of the IEEE (Volume: 6, Issue: 9, P. 1060 - 1076), 1980

16. Molina, E., Jacob, E., Matias, J., Moreira, N. and Astarloa, A., 2015. Using software defined networking to manage and control IEC 61850-based systems. *Computers & Electrical Engineering*, *43*, pp.142-154.

17. Navigationhttp://www.eurocontrol.int/articles/atm-voice-0.

18. Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R.,

19. Schulzrinne, H., Casner, S., Frederick, R. and Jacobson, V. (2003). RTP: A Transport Protocol for Real-Shelter-applications- application-card_56279-30503.html

20. Singh, S., Jeong, Y.S. and Park, J.H., 2016. A survey on cloud computing security: Issues, threats, and solutions. *Journal of Network and Computer Applications*, *75*, pp. 200-222.

21. Solar Winds Worldwide, 2014 Troubleshooting common issues in VOIP Time Applications. RFC 3550 (Standard). The Internet Engineering Task Force.

22. Vaishnavi and Kuechler, (2015) Design science information system, Available at http://desrist.org/desrist/content/design-science-research-in-information-systems.pdf