

CHALMERS



Voice communication system for air traffic control Development and evaluation of a prototype

Master of Science Thesis in Software Engineering and Technology

ERIK LEVIN
SAVASH ZAYNAL

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
Göteborg, Sweden, October 2011

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

Voice communication system for air traffic control
Development and evaluation of a prototype

ERIK LEVIN
SAVASH ZAYNAL

© ERIK LEVIN, October 2011.

© SAVASH ZAYNAL, October 2011.

Examiner: ALI SALEHSON

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Cover:

A United States Army air traffic controller at work by the radio. Image credit:

Expert Infantry (2011). *DA-ST-86-06255*.

<http://www.flickr.com/photos/expertinfantry/5467599262> (2011-08-02)

Department of Computer Science and Engineering
Göteborg, Sweden October 2011

Preface

This project was performed in collaboration with the Air Traffic Management division of Saab Security Systems AB, Göteborg, Sweden. We had great help from our company supervisor, Bert Axelsson, and all the staff at the division. We would also like to thank our university supervisor and examiner, Ali Salehson for all the support.

Abstract

This report details the development and evaluation of a prototypical, Internet Protocol (IP)-based voice communication system (VCS) for air traffic control. Such a system is concerned with connecting air traffic controllers to the radio which is used to communicate with aircraft pilots. Some benefits with a VCS are the integration of telephone and radio services, recording of voice communications and data, retransmission on several radio channels. Using IP also entails interoperability between components of different manufacturers. The prototype was to be the smallest possible that could allow two users to connect simultaneously to a radio. Because of its safety critical nature, the requirements for a VCS are strict, and documents from the European Organisation for Civil Aviation Equipment (Eurocae) were used as a basis for the specification. From the documents, a subset of the requirements concerned with the functionality for basic radio calls, communication initiation between the VCS and radio, performance of the system, and maximum allowed delays were chosen.

The radio gateway that was selected for the project did not fulfil several of the chosen requirements, since it was using an experimental firmware and was not developed fully in accordance with Eurocae's specifications. To develop a full-fledged system, a radio gateway in line with the required specifications must be used.

One part of the project was to test requirements concerning the maximum latency in the system. Different methods were used to measure the latencies. It was discovered that they sometimes show different results, which leads to the conclusion that care must be taken when deciding what measurement methods to use and how to use them.

Contents

1	Introduction	1
1.1	Background	1
1.1.1	Overview	1
1.1.2	Communication technology—circuits or packets	2
1.1.3	Requirements	3
1.1.4	Functionality	3
1.1.5	Architecture	4
1.2	Purpose	5
1.3	Scope	5
1.3.1	Delimitations	6
1.3.2	Assignment	6
2	Method	8
2.1	Prestudy	8
2.2	Tools	8
2.3	Requirements	9
2.4	Development approach	9
2.5	Evaluation	10
3	Technical considerations	11
3.1	Protocols	11
3.1.1	Signalling and audio transmission protocols	11
3.1.1.1	Session Initiation Protocol (SIP) overview	13
3.1.1.2	Real-time Transport Protocol (RTP) overview	15
3.1.2	Transport layer protocols	16
3.2	Programming language and platform	17
3.3	Software libraries, frameworks and tools	17
3.4	Radio capabilities	17
3.4.1	Radio channels	17
3.4.2	Radio signalling	18
3.4.2.1	SIP-specific event notification	18

CONTENTS

3.4.2.2	RTP header extension	19
3.4.2.3	Audio stream suppression	19
3.4.2.4	Real-time Transport Control Protocol (RTCP)	19
3.4.2.5	Choice of signalling method	20
3.4.3	Cross-coupling	20
3.5	Radio gateway	21
3.6	Network architecture	21
3.7	The software	22
3.7.1	Graphical user interface	22
3.7.2	Classes and interactions	23
3.7.2.1	Enabling or disabling a channel	26
3.7.2.2	Switching between modes	26
3.7.2.3	Cross-coupling or decoupling	27
3.7.2.4	Push-to-talk	27
3.7.2.5	Handling squelch	27
3.7.2.6	Keep-alive messages	28
3.7.2.7	Library changes	28
4	Delay measurements	29
4.1	Delay requirements	29
4.1.1	Frequency key activation response time	30
4.1.2	Transmitter activation delay	30
4.1.3	Aircraft call indication delay	31
4.1.4	Ground transmission voice delay	31
4.1.5	Ground reception voice delay	31
4.1.6	Loopback delays	32
4.1.6.1	Transmitter activation and aircraft call indication loopback delay	32
4.1.6.2	Ground transmission and reception voice loopback delay	32
4.2	Methods	32
4.2.1	Oscilloscope	33
4.2.2	Software timers	33
4.2.3	Time logging	33
4.3	Set-up	34
4.3.1	Equipment	34
4.3.2	Frequency key activation response time	35
4.3.3	Transmitter activation delay	37
4.3.4	Aircraft call indication delay	37
4.3.5	Ground transmission voice delay	37
4.3.6	Ground reception voice delay	38

CONTENTS

4.3.7	Transmitter activation and aircraft call indication loop-back delay	38
4.3.8	Ground transmission and reception voice loopback delay	39
5	Results	40
5.1	Measurement results	40
5.1.1	Frequency key activation response time	40
5.1.2	Transmitter activation delay	40
5.1.3	Aircraft call indication delay	41
5.1.4	Transmitter activation and aircraft call indication loop-back delay	41
5.1.5	Ground transmission voice delay	41
5.1.6	Ground reception voice delay	41
5.1.7	Ground transmission and reception voice loopback delay	41
5.2	Fulfilment of requirements	41
5.2.1	Requirements for milestone 1	42
5.2.1.1	Summary for milestone 1	43
5.2.2	Requirements for milestone 2	44
5.2.2.1	Delay requirements	44
5.2.2.2	Summary for milestone 2	46
5.2.3	Requirements for milestone 3	46
5.2.3.1	Summary for milestone 3	49
5.2.4	Requirements for milestone 4	49
5.2.4.1	Summary for milestone 4	50
6	Discussion and conclusions	51
6.1	The delay measurements	51
6.1.1	Summary of the measurement methods	51
6.1.2	Measurement differences between methods	52
6.1.3	Differences between transmission and reception delay	54
6.2	Gateway evaluation	55
6.2.1	Gateway summary	57
6.3	Software development	57
6.3.1	Protocol library	57
6.3.2	Programming language	57
6.4	Testing environment	58
6.5	Additional features	58
6.5.1	Best signal selection and climax mode	58
6.5.1.1	Best signal selection	58
6.5.1.2	Multi-carrier mode	59
6.5.2	Simultaneous transmissions detection	59

CONTENTS

Glossary	60
Bibliography	62
A Selected requirements	65
A.1 Requirements for milestone 1	65
A.2 Requirements for milestone 2	67
A.3 Requirements for milestone 3	69
A.4 Requirements for milestone 4	72
B Delay measurement figures	74
B.1 Figures: Delay specification	74
B.1.1 Transmitter activation delay	74
B.1.2 Aircraft call indication delay	75
B.1.3 Ground transmission voice delay	75
B.1.4 Transmitter activation and aircraft call indication loop- back delay	76
B.1.5 Ground reception voice delay	76
B.1.6 Ground transmission and reception voice loopback delay .	77
B.2 Figures: Oscilloscope set-ups and readings	77
B.2.1 Frequency key activation response time	77
B.2.2 Transmitter activation delay	79
B.2.3 Aircraft call indication delay	80
B.2.4 Ground transmission voice delay	81
B.2.5 Ground reception voice delay	82
B.2.6 Transmitter activation and aircraft call indication loop- back delay	84
B.2.7 Ground transmission and reception voice loopback delay .	85
C Delay result tables	87
C.1 Frequency key activation response time	87
C.1.1 Part a)	87
C.1.2 Part b)	88
C.2 Transmitter activation delay	88
C.3 Aircraft call indication delay	89
C.4 Transmitter activation and aircraft call indication loopback delay .	89
C.5 Ground transmission voice delay	90
C.6 Ground reception voice delay	90
C.7 Ground transmission and reception voice loopback delay	91

Chapter 1

Introduction

The first chapter of this report aims to give the reader a general understanding of the project: an overview about the domain of voice communication systems (VCSs) for air traffic control (ATC), what the goal of this project is, and what it will and will not contain. Subsequent chapters will go into more detail about the methods and findings of the project, and contain the discussions and conclusions.

Note that the digital communications domain is a world of acronyms, and this report will also contain its share of abbreviations. As help for the reader, a glossary can be found on page 60.

1.1 Background

This first section of chapter 1 provides the necessary background to understand the report. It gives an overview of what a VCS involves and what such a system does, in the context of ATC. Afterwards an explanation follows of the technology, architecture, and the functionality of such a system.

1.1.1 Overview

The most important way to exercise ATC is to use voice communication over radio to guide the aircraft pilots (Bardach et al., 2003; Miller and Greenspan, 2008). A VCS is used to connect the ATCs to voice communication technologies such as a radio and the public switched telephone network (PSTN) as can be seen in figure 1.1. The infrastructure of such systems is usually a proprietary circuit-switched network where the work stations of the controllers are connected via an electronic call exchange to the radio and the telephone network as seen in figure 1.2. This project, however, will be based on a packet-switched infrastructure using Internet Protocol (IP) because of its advantages regarding flexibility and in-

1.1. BACKGROUND

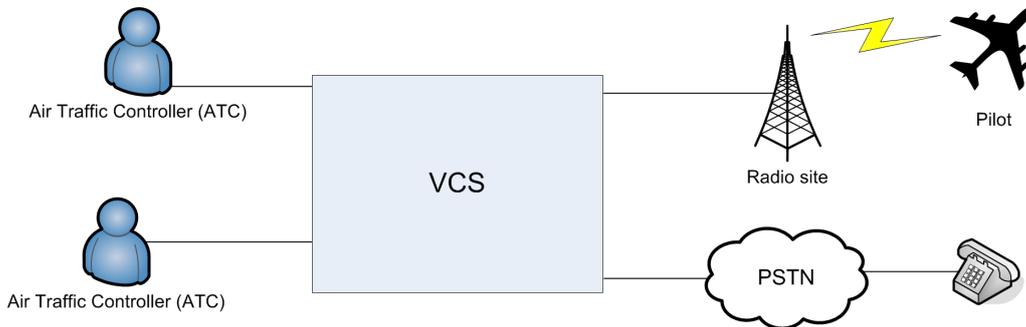


Figure 1.1: General black-box view of a VCS.

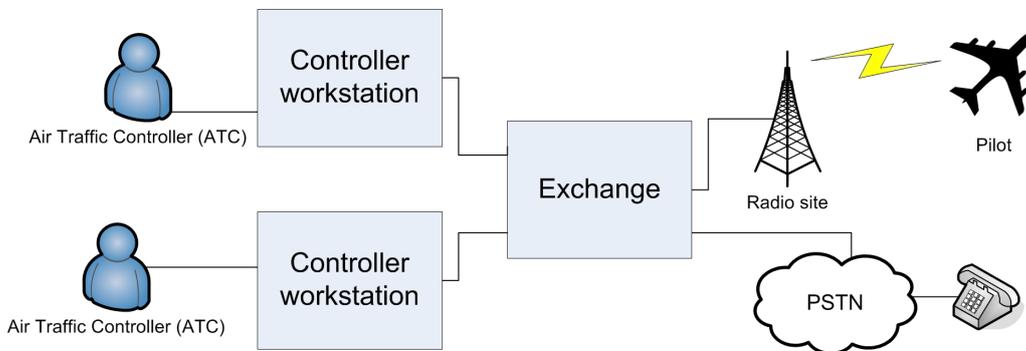


Figure 1.2: Overview of a circuit-switched VCS.

teroperability. With a VCS, the controllers can communicate through both radio and telephone using a single work station, called a controller working position (CWP). Using their work stations, the controllers can choose which radio frequencies to receive from and send to, and also make calls both internally within the airport and externally over the PSTN.

Since the VCS is the means with which to guide the aircraft, it is a system of critical importance. Due to the safety critical environment of an airfield, the quality requirements for a VCS are strict. The system must provide a reliable service of sufficient quality, there must be no long delays in communication, and sound quality should be adequate.

1.1.2 Communication technology—circuits or packets

In recent years, the communications industry has been moving from circuit-switched to packet-switched technology based on the IP. This applies also to voice communications within ATC, where the interest in infrastructures for VCSs has been shifting from circuit-switching to IP technology (Haindl et al., 2009).

1.1. BACKGROUND

There are several advantages of using the open IP standard instead of proprietary circuit-switched solutions. One major advantage is increased interoperability. Using IP allows interconnecting commercial off-the-shelf components from various manufacturers. Special-purpose, expensive software and hardware can be replaced by off-the-shelf software and standard Internet equipment such as routers and switches. Using IP also enables integration with external data services such as weather information, and connections over a wide area network (WAN), so that controllers can perform their task from a remote location (Darilion et al., 2004). Failure recovery is another important area where packet-switching has an advantage over circuit-switching. Failure of a node or link in a circuit-switched network would cause the communication to be interrupted, unless precautions had been made beforehand. On a packet-switched network, an alternative path for the packets will automatically be computed as a consequence of the normal operations of the network (Pointurier, 2002, section 1.1).

However, packet-switched networks also have drawbacks compared with circuit-switched infrastructures. For instance, the former are more unpredictable concerning delays (Kurose and Ross, 2008, section 1.3.1).

1.1.3 Requirements

Both functional and quality requirements for IP-based VCSs for ATC are detailed in the documents ED-136, ED-137 and ED-138 (Eurocae WG-67, 2009*a,b,c*), by the European Organisation for Civil Aviation Equipment (Eurocae). Eurocae is a non-profit organization composed of stakeholders within aviation, concerned with standardization for the industry. Eurocae has been given the task by the European Aviation Safety Agency (EASA), an agency of the European Union, to develop technical standards (Amato, 2008).

The Eurocae documents concerning VCSs specify matters such as the maximum allowed audio and signalling delays, security, data traffic precedence, voice quality, network requirements, protocol usage and of course functionality.

Among the key requirements, which will influence many decisions while developing a VCS, are those deciding what network protocols to use. Regarding this issue, Eurocae mandates the use of the Session Initiation Protocol (SIP) (Rosenberg et al., 2002) for session set-up and management, and the Real-time Transport Protocol (RTP) (Schulzrinne et al., 2003) for audio transfer and in-session radio signalling.

1.1.4 Functionality

The functional requirements of a VCS include characteristics of analogue radio communications such as push-to-talk and channel selection. The air traffic con-

1.1. BACKGROUND

troller should be able to enable any channels for reception, transmission or both. Then, it should be possible to “push-to-talk”, that is, to simply push a button and immediately be able to communicate over the radio. In contrast to conventional telephone communications, it is not necessary for any recipient to hear a ringing, and choose to answer the call. Rather, the radio should answer immediately.

It should be possible to listen to several channels at once for incoming calls, and when multiple connections are ongoing at the same time, the sound from them should be mixed together and played through the same speakers. Furthermore, there should be support for conference calls and prioritization of calls, so that when a call of higher priority is received (for instance an emergency message), calls of lower priority should be suppressed or terminated.

In ATC, the airspace is split into sectors, each with its own radio frequency for communication. Usually, each controller monitors one sector at a time, but as air traffic load varies, it might be desirable to assign sectors differently (Darilion et al., 2004). When the air traffic volume is less, one can combine several radio frequencies into one, a technology called cross-coupling. When sectors are cross-coupled, they are effectively merged, allowing them to be monitored by one single controller (Kurth et al., 2005).

In addition to radio capabilities, the CWP has all the features of a normal telephone system when communicating through the internal telephone or the PSTN, such as call queues, putting calls on hold, and conference calls (Eurocae WG-67, 2009c, chapter 3).

1.1.5 Architecture

An IP-based VCS contains several CWPs: computers with the capabilities needed for communicating with aircraft, and a human-machine interface consisting of devices such as a microphone, speakers, and a touch screen. Here, the controller manages the communications with aircraft and other locations. To integrate radio and telephone into the network, gateways are used to translate the signals and sound from the radio sites and telephone network into voice over Internet Protocol (VoIP) data. In addition the VCS usually has devices for recording the communications, maintenance and configuration terminals, and perhaps gateways to other legacy communication systems. An outline of a possible VCS architecture is shown in Figure 1.3 on the next page.

There might be a central VoIP server, which acts as a proxy to coordinate traffic and forward transmissions to and from the correct CWPs. Alternatively, one could choose a distributed approach that does not use a central server (Hafner and Mahmoud, 2005).

In one network model, the components are interconnected by an isolated local area network (LAN), that is, it is not connected to the Internet and data over the

1.2. PURPOSE

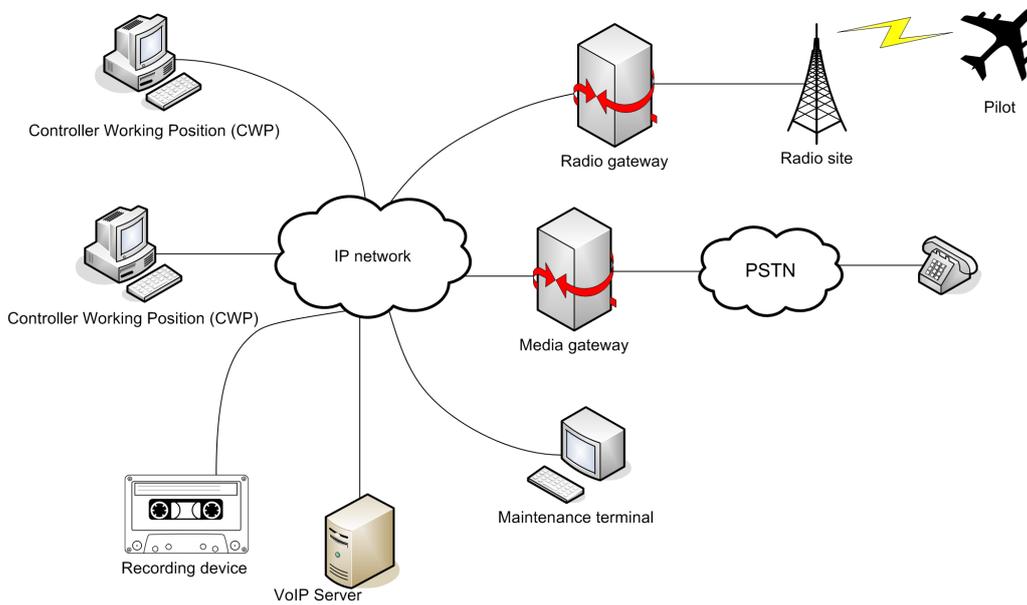


Figure 1.3: Diagram of an IP-based VCS architecture.

system does not compete with traffic unrelated to the service. Another option is to allow connections over WAN, such as the Internet, to interconnect VCSs, or connect remote radio sites and data services to the system (Darilion et al., 2004; Eier and Kampichler, 2010).

1.2 Purpose

As mentioned in section 1.1.1, the IP-based VCS is an essential system for modern aviation, as the primary means to guide the aircraft pilots. Knowledge about the practices and methods regarding this kind of critical system is valuable to the aviation industry. This project aims to find and discuss the different technical issues that must be considered when constructing a VCS, through literature surveys, and through developing and evaluating a prototype.

For instance, one important issue is to find a method for measuring the latency in a VCS to ensure that the system fulfils the quality requirements. Another is what improvements could be made to the chosen radio gateway to make it more suitable for this application. Further considerations regarding technical choices of protocols, platforms, components, and implementation will be presented and discussed.

1.3 Scope

A VCS is a large system with many hardware and software components, and knowledge about such systems is a broad area. Because of time limitations, it is not possible for this project to develop a full-fledged VCS from scratch according to all the standards of the business. This section will therefore define the boundaries of the project.

1.3.1 Delimitations

The goal is to develop a minimum VCS that allows connections to the radio. Telephony integration will not be part of the project, since knowledge about IP telephony is already well researched and widespread. Recording devices will not be considered either.

The demands on the graphical user interface (GUI) of a CWP are important. The GUI requires a high degree of usability and performance so as to not hamper the controllers in their work. Constructing a complete, flawless GUI is also outside the scope of this project. Only a limited interface that fulfils the needs for developing and testing the functionality will be created.

Eurocae mandates that the audio quality of the system is measured using, among other things, the mean opinion score method (Eurocae WG-67, 2009c). In this project, however, no formal user tests of this kind will be performed. The quality considerations will be focused on the delays.

The Eurocae documents also detail how system engineering and management shall be performed, how redundancy shall be employed for greater failure safety and several other areas not concerned with the functionality or delays. While also important for a complete VCS, these areas will not be considered.

Sometimes it is desirable to interconnect a VCS with data services over a WAN, for instance over the Internet (Darilion et al., 2004). No such wide area connections will be considered in this project, and the prototype will operate only inside an isolated LAN.

The functionality of the radio gateway in the VCS also limits what is possible to implement. In this project, we will use an in-house gateway from Saab named *IPG 3000*. The SIP support for the *IPG 3000* is under development in an experimental firmware, and by applying the gateway in a prototypical VCS, it will be evaluated how well it works for this application, and improvements will be suggested. However, attempting to correct any issues that might arise by changing the firmware of the gateway is out of scope for this project, and must be left to future work.

1.3.2 Assignment

The objective of the project is to develop a functioning VCS prototype, and a method for measuring delays in the system. The size of the prototype is to be the minimum required to enable the crucial functionality for radio communication: push-to-talk, selection of radio channels, and coupling of channels. It is clear that a system containing one single CWP cannot support the required functionality, and hence cross-coupling and simultaneous push-to-talk cannot be performed in a meaningful manner. Also, the number of radio channels must be greater than one, to allow selection between different channels.

Thus we can conclude that the minimum VCS that allows this functionality is one with two CWPs and two radio channels. See Figure 1.4 for an outline of such a system. A central voice server may be used to coordinate communications, or it could be omitted in favour of a peer-to-peer architecture.

Not every one of the hundreds of requirements in the Eurocae documents can be carefully studied and considered during this project, so a selection of relevant requirements must be made. Focus will be on the requirements concerned with delays, functionality, and those relating to how SIP and RTP are used for signalling and session management.

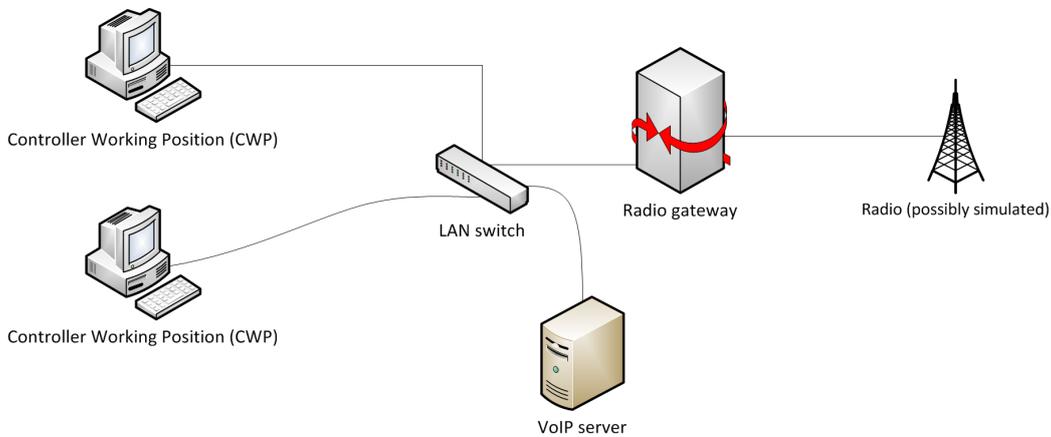


Figure 1.4: Architectural diagram of the prototype VCS

Chapter 2

Method

This chapter describes the different phases of the project: the prestudy, the selection of tools for development, how the requirements were chosen and desired features for the hardware components, and also the prototype milestones.

2.1 Prestudy

In the initial phase of the project, a prestudy was conducted, where information was gathered about radio gateways and voice communications over IP in general, and about SIP and RTP in particular. The Eurocae specifications, information from vendors about gateways, research papers, and generalized architectural scenarios from industry were studied to provide an understanding of the problem at hand.

2.2 Tools

It was decided that the project would not be concerned with implementing SIP and RTP for audio, since there are many existing implementations already available for use. The project regards only an experimental prototype, and therefore it was deemed appropriate to first and foremost search for open source libraries, as they are free of cost, and also extendible should it be necessary to customize their functionality.

It was a desire from the company that the prototype should be developed in one of the programming languages used within the company, that is, C++ or C# *.Net*. To ensure easy usage in the prototype, it was a criterion for the SIP and RTP library that it too should be written in one of these languages.

To develop the software itself, several development tools were provided by the company, such as the integrated development environment (IDE) *Microsoft Visual*

2.3. REQUIREMENTS

Studio 2008 and the issue tracking system *Trac*. These well-established tools were recommended by the company, and were deemed adequate for the project at hand. They were thus used for development.

2.3 Requirements

The documents from Eurocae mentioned in section 1.1.3 were decided upon as the basis for requirements for the prototype, because of their status as acknowledged technical standards. The documents were studied and a subset of the requirements were chosen for consideration. The selected requirements were those that concerned the goals of the project: functionality for basic radio communications, protocol usage, and the maximum allowed delay for different actions.

Some examples of what the requirements specify are that audio shall be encoded using pulse code modulation (PCM), with a standard packetization interval of 20 ms, and that the protocols used in the system must be SIP and RTP, implemented as detailed in Rosenberg et al. (2002) and Schulzrinne et al. (2003), respectively. The RTP header shall be extended to allow radio signalling information to be transmitted.

When neither party of a session is transmitting, the audio stream is paused. To ensure that the connection has not been broken, RTP packets without audio data are periodically sent as keep-alive messages.

It is furthermore specified that initiation of a SIP session must never originate from the radio side, that is, from the gateway. Instead, a CWP shall initiate a session with the gateway when it wants to communicate over a radio frequency connected to that gateway. The system must be able to support seven simultaneous connections to the same radio channel. The requirements also detail how cross-coupling should function.

All of the selected requirements are described in greater detail in appendix A.

2.4 Development approach

For the software of the prototype, four milestones were decided to be developed in consecutive order, with the selected requirements split up between them. This section specifies how each of the milestones was defined.

Milestone 1 For the first milestone two clients should be able to communicate over a network using SIP and RTP as specified in the selected requirements (derived from the Eurocae documents). The clients do not need to have all the

2.5. EVALUATION

functionality of a fully fledged CWP; the focus of this milestone is on the communication on a technical level, and its purpose is to lay the foundation for the prototype.

Milestone 2 The second milestone requires that a CWP can communicate over radio, through the radio gateway. This milestone is where the push-to-talk mechanism should be implemented.

Milestone 3 In milestone three, the CWP functionality selected for implementation should be developed. This includes the possibility for cross-coupling (see section 3.4.3) and prioritization of calls.

Milestone 4 For the last milestone, at least two CWPs will be able to communicate over the radio simultaneously, and all selected functionality has been enabled.

2.5 Evaluation

Finally, the prototype was checked against all the requirements. It was evaluated which requirements the prototype did meet and which it did not.

Also, different methods to test delays were applied. Ten measurements for each requirement and for each applicable measurement method were done, and the results were collected. From the results, conformance to the delay requirements was evaluated, and the result were discussed, particularly in the case of any requirements not being met.

Chapter 3

Technical considerations

While constructing a VCS, many technical decisions must be made, which will affect how well the system will comply to both functional and quality requirements. When developing such a critical system, much care must be taken when deciding about issues such as the choice of architecture, platforms, protocols, and hardware components, as well as how to implement the functionality, and measure the quality of the service in a reliable manner. This chapter details these considerations and specifies what solution was chosen for each of them.

After coming to decisions about these considerations, the software was developed. This chapter contains an explanation of the software, its GUI, its functionality and the interactions between its classes.

3.1 Protocols

One fundamental decision for any communication system is what protocols to use in the different layers of the communication stack. It has already been established that the network layer protocol that this project concerns is IP, but choosing which protocols to use for the other layers are still to be decided.

3.1.1 Signalling and audio transmission protocols

Within voice communication applications, several different protocols can be used. There is a need for signalling protocols such as SIP and H.323 (ITU-T, 2010) for call signalling and session management. For the real-time transfer of media data, other protocols such as RTP are used. There are also closed, proprietary protocols such as the Skype protocol (Baset and Schulzrinne, 2006). Using open standards, however, increases interoperability between endpoints where the equipment comes from different vendors (Haindl et al., 2009).

3.1. PROTOCOLS

Today, the major competition among open protocols when implementing VoIP applications is between H.323 or SIP (Hillenbrand et al., 2005). H.323 is a recommendation put forth by the International Telecommunication Union (ITU), and it is a standard for real-time audio and video communication, including specification for a whole protocol stack and corresponding architecture (Liu and Mouchtaris, 2000). SIP is a protocol specified by the Internet Engineering Task Force (IETF) concerned only with signalling and session management. As such, it is more lightweight and less complex than H.323, and because it restricts itself to only consider a subset of the functionality required for VoIP, it is easier to combine with different protocols as required. Furthermore, SIP uses a clear text syntax that is similar to the well known web protocol Hypertext Transfer Protocol (HTTP), while H.323 uses binary encoding. According to Basiccevic et al. (2008), SIP is more well-fitting to the protocols and practices of the Internet, while H.323 retains some unnecessary baggage from the circuit-switched telephony environment.

SIP has been regarded as the more suitable protocol for VCSs within ATC. One reason is its rapid development and interoperability when compared to H.323 (Darilion et al., 2004). Another is because its extendibility allows for implementing most of the needed ATC features (Haindl et al., 2009). SIP is also the protocol required by the Eurocae documents, which mention the following advantages of SIP over H.323: better loop detection, scalability, and interoperability (Eurocae WG-67, 2009b, Appendix C). Because of their inclusion in the Eurocae documents, SIP was decided for this project.

Since SIP is only concerned with session management, another protocol is needed for streaming the audio data. For that task, the dominant open protocol is RTP, defined by the IETF. RTP is concerned only with transporting the media, and an additional protocol, Real-time Transport Control Protocol (RTCP) is used for control and monitoring (Schulzrinne et al., 2003). Since RTP is the most common protocol for voice applications, and also mandated by Eurocae, it was chosen for this project.

It is also conceivable for an IP-based VCS to not use any session management protocol at all and send RTP packets directly to the intended receiver, without setting up any session beforehand. For such a solution, however, several things must hold. Firstly, the location of each node in the network must be known by all other nodes at all times, since RTP itself does not have any support for address resolution. This means that the network has to remain static, and will not be easily extended and modified. Another issue is that parameters such as what codecs or packet sizes that are supported by different nodes must be hard coded into the system, or a session control protocol would be necessary to negotiate such session parameters.

While these limitations could possibly be overcome for a system concerned only with direct, straightforward communications, such as a VCS connected only

3.1. PROTOCOLS

to a radio, omitting session management protocols would not work very well if telephony functionality is to be implemented. Protocols such as SIP are literally made to support telephony functionality: ringing, being busy and putting calls on hold. A system without session protocols, using only audio packets over User Datagram Protocol (UDP) for its data transfer was used by Hafner and Mahmoud (2005) for simulation purposes.

3.1.1.1 SIP overview

Understanding the details in the remainder of this report requires basic knowledge about the protocol suite that was chosen for the project: SIP and RTP. For complete details about SIP, see Rosenberg et al. (2002).

SIP location resolution SIP has support for location resolution using so called registrars. A user can send a message to a registrar server which will store information about that user's current whereabouts. In this way, the user can be reached by anyone querying the registrar for the user's location (Rosenberg et al., 2002, section 10).

SIP messages SIP is, as mentioned above, quite similar to HTTP. Each SIP message is either a request or a response, of which there are several types. Which request or response a message represents is written in its header, formatted in clear text and thus easily read by humans. An example of a SIP message header is seen in figure 3.1.

```
INVITE sip:bob@somewhere.com SIP/2.0
Via: SIP/2.0/UDP pc33.elsewhere.com;branch=z9hG4bK776asdhs
Max-Forwards: 70
To: Bob <sip:bob@somewhere.com>
From: Alice <sip:alice@elsewhere.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.elsewhere.com
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.elsewhere.com>
Content-Type: application/sdp
Content-Length: 142
```

Figure 3.1: Example of the header of a SIP request.

The first line shows that this message is a request of the type INVITE. Such a request is used to ask that communication is initiated with the recipient. Then

3.1. PROTOCOLS

follows the SIP address, or Uniform Resource Identifier (URI), of the recipient. A SIP URI contains a user name, and the domain or IP address to the host where this user can be found. The first line also shows what version of SIP is being used.

The `Via` field on the second line shows the address to the host where a response should be sent. As the request makes its way through any proxy servers, each server adds its own address to a `Via` field, to ensure that responses take the same route back. `Via` also contains an identifier for the SIP transaction.

Next are the `To` and `From` fields, which simply contain the display names and addresses of the sender and recipient. The `Call-ID` field is a globally unique identifier to ensure that any SIP session can be uniquely defined. `CSeq` is a sequence number that serves to number the requests that have been sent within a session.

`Max-Forwards` limits the number of hops that the message can do before it will be dropped, in which case an error response is generated. The `Contact` field contains the direct address to the originator. This address is where requests should be sent, while the address in the `Via` field is where responses are to be sent.

Finally there is `Content-Type` and `Content-Length`, which specifies what format the message body is in and how long it is, respectively.

SIP call flows Figure 3.2 shows an example of how the passing of SIP messages could occur during a SIP session.

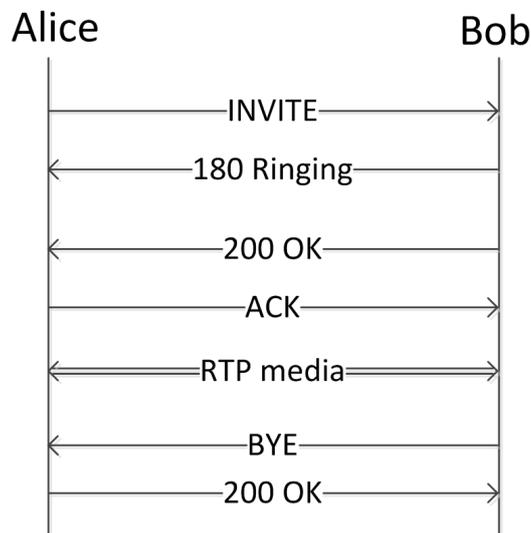


Figure 3.2: An example of a simple SIP call flow.

In the example, Alice sends an `INVITE` request to Bob, upon which Bob's SIP phone responds that it is ringing, waiting for Bob to pick up. When that happens,

3.1. PROTOCOLS

an OK response is sent, which is acknowledged by Alice sending an ACK request¹. After this exchange, the session has been set up, and the actual communication data is sent between both users as a stream of RTP packets. After some time, Bob chooses to close the session, and does this by sending a BYE request, which is met by an OK response from Alice.

A SIP call flow can be much more complicated than this, involving multiple proxy servers, and many more kinds of requests and responses. This was only one of the simplest possible examples. For more call flow examples, the reader can refer to Johnston et al. (2003).

Session parameter negotiation using SIP What media types to use and the parameters for these streams are decided in the beginning of a session. This is done by one user sending an offer: essentially a list of what media streams and parameters that user wants to communicate with, and the port numbers where the media data for each stream should be sent. The other user responds by removing from the list those that it is not able or willing to use, and sends it back, including the port numbers that user wants to use. The result is the intersection of the acceptable parameters for both users, and these are defined as the allowed settings for the session. Additionally, each user knows what ports to connect to.

These offers and answers are sent in a format called Session Description Protocol (SDP) in the bodies of the SIP messages. To continue the example above where Alice contacts Bob, it might be that Alice includes an SDP message containing an offer for parameters for one audio and one video stream in her INVITE request. Bob, however, does not want to use a video stream, but can accept the audio stream that Alice suggests. Bob therefore removes the video stream from the offer, and sends it back in his OK response.

It is also possible to request that the session parameters are changed in an ongoing session, for instance if a user wants to change the port number for one of the media streams. Then, a new INVITE request containing the new SDP offer is sent within the dialogue, and responded to as normal. For more details about how parameter negotiation is done, see Rosenberg and Schulzrinne (2002).

3.1.1.2 RTP overview

RTP is the protocol concerned with transmitting the media data from end to end. Its packets contain sequence numbers so that received packets can be put in the correct order before playback. They also include time-stamps denoting the instant when the contained media data was sampled to make sure that media is played

¹Note that ACK is a request and not a response, although it is never used to initiate communications, but only to acknowledge a received response.

3.1. PROTOCOLS

back smoothly. RTP does not, however, provide handshakes, acknowledgements or retransmissions of lost packets.

RTP packets do not have header fields in clear text, in contrast to SIP. Rather, they are encoded in binary. The RTP header is shown in figure 3.3.

The default header has three fixed 32-bit words with fields of varying sizes, and then between zero and fifteen additional words. The number of additional words depends on the number of contributing source identifiers that are present, a number which is specified in the CC field. After the default header there may be an application-defined header extension. If a header extension is present, the flag X is set to 1, otherwise it is 0. See sections 3.4.2.2 and 3.4.2.5 about how the header extension works, and how it can be used.

Explanations about the source identifiers and the other fields in the header are out of scope for this brief exposition. Further details about RTP are available in its specification: Schulzrinne et al. (2003).

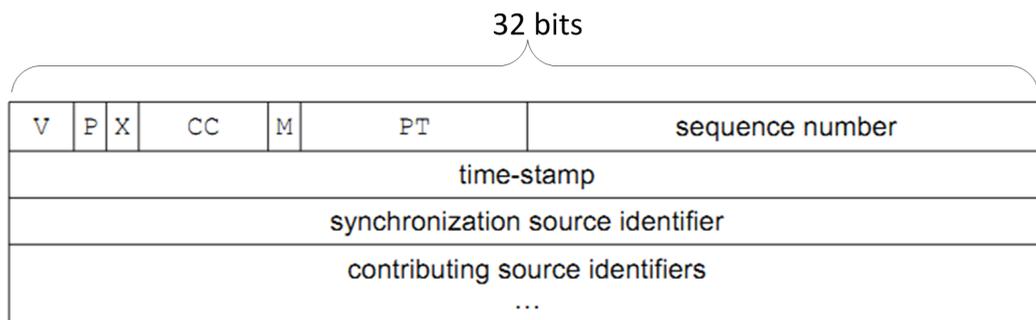


Figure 3.3: The fields of the RTP header.

3.1.2 Transport layer protocols

In addition to SIP and RTP, some transport layer protocol must be used. Here, the most common options are either Transmission Control Protocol (TCP) (Postel, 1981) or User Datagram Protocol (UDP) (Postel, 1980). TCP features an initial handshake, acknowledgement of each packet, and retransmission of lost packets which is meant to ensure a robust connection. On the other hand, UDP reduces the amount of traffic being sent over the network as no retransmission or acknowledgement takes place, and it also has a smaller header.

When TCP is used as the underlying protocol for the SIP communication, the handshake introduces a delay in the system, and its time out parameters are difficult to adjust from the application (Darilion et al., 2003). UDP is a better choice for SIP since the latter has its own management of time outs and retransmissions.

Also for audio transmission over RTP, UDP is usually considered to be the most useful transport layer protocol. This is because TCP is more concerned with reliable transmission than with real time delivery, whereas for audio streaming, some packet loss is more acceptable than untimely delivery (Perkins, 2003, pp. 46–47). Therefore, UDP was the chosen transport layer protocol, both for SIP and RTP traffic.

3.2 Programming language and platform

Different programming languages have different strengths and weaknesses. But for this project, the primary factors were that there should be an appropriate protocol stack written in the language, and that it was one of the languages used within the company. The company primarily used C++ and C#, with Microsoft's framework .NET, so these were the first choices. After finding a protocol stack in C# that seemed useful, C# .NET was decided as the language and underlying platform to use.

3.3 Software libraries, frameworks and tools

Available frameworks for SIP and RTP communication was examined. Desirable characteristics were that it should be open source to allow for inspection and modification, it should be complete and stable enough for immediate use, have enough documentation, and be written for .NET.

The chosen library was the open source C# framework *Lumisoft.Net*, available from <http://www.lumisoft.ee>. It included a stack for SIP, RTP, and audio handling. However, during the course of the project it was necessary to modify the library slightly, as some features required for the VCS were missing, such as support for the RTP header extension (see section 3.4.2.2 on page 19).

3.4 Radio capabilities

This section explains some of the necessary capabilities for an IP-based VCS, and discusses different methods how they can be implemented.

3.4.1 Radio channels

In a radio communication system, the user needs to be able to choose which radio channels to receive from or transmit to. In a legacy system, this could be done

3.4. RADIO CAPABILITIES

through analogue means, but this section explains how different radio channels can be handled in a software system based on SIP and RTP.

Since SIP has been chosen as the signalling protocol, it is clear that to select a channel for listening, a SIP message should be sent to the radio gateway that is connected to the desired frequency. But which message it should be and how the radio channels are addressed need further specification. Kurth et al. (2005) discusses an architecture where every single radio channel has its own radio server. In such a system, each channel would be accessed by a different IP address. Alternatively, if several frequencies are accessible through the same network node, they can be distinguished using the SIP user name. Eurocae specifies an addressing convention for SIP URIs, where information about the radio frequency is put into the user name field, while the domain is named after the physical radio site (Eurocae WG-67, 2009a, section 3.3).

Using the correct SIP URI for the desired radio channel, the CWP must then signal to the radio gateway that it wants access to it. The CWP can for each channel either receive only, or be allowed to also send to the channel.

Within SIP, an INVITE request is used to set up a session, to then allow transfer of media. Therefore, an INVITE is used to signal that a CWP wants access to listen to a radio channel. To specify whether it is desired to also be able to send or only to receive, an attribute to the media session description in the body of the SIP message can be used. Depending on how that attribute is set, the radio gateway or server will allow access for receiving only, or for both sending and receiving.

3.4.2 Radio signalling

After the CWP has been granted access to a radio channel, the occurrence of push-to-talk (PTT) and squelch must be signalled somehow. To do this, many different solutions are possible.

3.4.2.1 SIP-specific event notification

One approach, demonstrated by Darilion et al. (2003, 2004), is to use an extension of SIP that adds support for requests related to signalling events between users. A user that wishes to receive PTT or squelch notifications for a certain radio frequency subscribes to the frequency using a SUBSCRIBE request. When such an event occurs, the corresponding gateway or server sends a NOTIFY request to the subscribers. The NOTIFY requests are then acknowledged by each subscriber. This method is reliable, since it uses acknowledgement, and if a node would be interested in only the PTT and squelch events and not the audio, there is no need to set up an RTP session. Also, it uses only open Internet standards, which allows for interoperability. A disadvantage is that it requires a large number of SIP messages

3.4. RADIO CAPABILITIES

being sent across the system, and if the SIP stack is not fast enough, the creation of that many transactions will increase the delay to unacceptable levels.

3.4.2.2 RTP header extension

RTP packets can be augmented with an extended header (Schulzrinne et al., 2003, section 5.3.1). This extension has the form shown in figure 3.4. It requires a 16 bits long field that can be defined by the application that uses the header extension, for instance for denoting which type of header extension it is in case several are used. Then there is a 16 bit field for the length of the extension, denoted in number of 32-bit words, and then the actual header extension is appended.

Using a custom header extension in the RTP packets of the VCS, information about squelch and PTT status can be passed. This way, signalling status will be transmitted together with every voice packet, which ensures that the signals are always synchronized to the audio stream. Allowing this information to piggyback on the RTP packets avoids creating a new SIP session every time the signalling status changes. A disadvantage of this method is that it requires that an RTP session is set up, even if the audio is not of interest (Darilion et al., 2004).

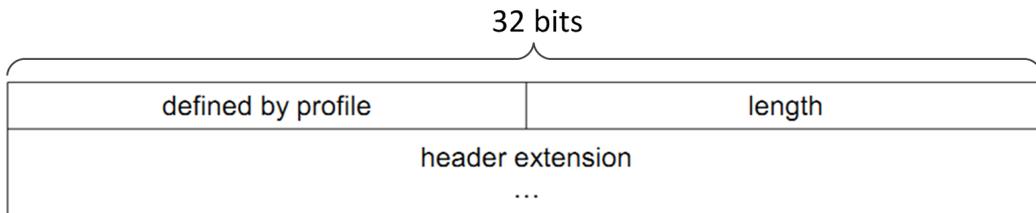


Figure 3.4: The RTP header extension.

3.4.2.3 Audio stream suppression

In this simple solution, RTP packets are only sent when squelch or PTT is active, and suppressed otherwise. Therefore, the very presence of packets signifies PTT. The advantages of letting the occurrence of RTP packets itself signal PTT are basically the same as for using the RTP header extension, with the additional advantage that the method is simpler and, of course, the header can be kept at a normal size. The disadvantage is that the method only allows one bit of information: either there are RTP packets or there are not. More complex signalling information such as the priority of the signal, and acknowledgement that the connection is working, both examples of data that Eurocae uses, cannot also be sent in this manner.

3.4. RADIO CAPABILITIES

3.4.2.4 RTCP

Customized RTCP packets can be used to transmit signalling information. This way, no RTP session is needed. However, this method could be less robust than the SIP-specific event notification method, since there is no acknowledgement for RTCP (Darilion et al., 2004).

3.4.2.5 Choice of signalling method

The method for signalling mandated by Eurocae is the RTP header extension, but audio stream suppression is also required. That is, audio packets should only be sent if PTT or squelch is activated (Eurocae WG-67, 2009a, section 5.6.3). The requirements specify that the header extension shall have a length of one 32-bit word, where information about PTT and squelch status are put. The header extension should also contain data used for simultaneous transmissions detection and best signal selection, features that were not implemented for this project, but are explained in 6.5.

To be in line with the Eurocae requirements, this combination of the RTP header extension with audio stream suppression was decided to be employed for squelch and PTT signalling in this project.

3.4.3 Cross-coupling

Cross-coupling, briefly mentioned in section 1.1.4, is the functionality to combine several physical radio frequencies into a single logical frequency. There are several different ways to accomplish this. The simplest one is local coupling, in which the CWP that created the coupled group handles retransmission of the signal between the coupled channels. That is, when audio is received on one of the channels, the CWP retransmits the same signal on all the other coupled channels to make them act like a single logical channel. This simple method has its limitations, however. For instance, if a truly decentralized approach is chosen, then nothing prevents different CWPs from putting channels into configurations that could lead to unwanted cross-coupling chains or loops. Say one controller couples frequencies F1 and F2 together. Another controller couples F2 with F3. Now, a signal being received at F1 will be retransmitted to F2 by the first controller, but then the second controller will retransmit F2 on F3. Then F1 and F3 will appear coupled, even though they do not appear in the same cross-coupled group at any CWP.

Another drawback of local coupling is that it adds additional load to the CWPs, and could incur a large latency (Kurth et al., 2005). Nevertheless, local coupling is essentially the method specified by Eurocae, but in addition, there are require-

3.5. RADIO GATEWAY

ments that no frequency should occur in two different cross-coupled groups in any CWPs, which could warrant the use of an additional cross-coupling server.

An approach that can be used when each radio channel has its own radio server is a master-slave model, where each of the SIP servers can either be the master of a group, or a slave in it, when channels are cross-coupled (Kurth et al., 2005). The server selected as the master will coordinate the traffic in the cross-coupled group, and decide if a CWP wishing PTT access can gain it or not. The master forwards squelch status originating from either its own radio, or from the radio of one of its slaves, and slave servers forward audio from their radios to the master. Putting the responsibility of handling cross-coupling on radio servers lessens the load on CWPs and can avoid cross-coupling chains.

For this project, local cross-coupling was decided to be implemented, since it is required by Eurocae. Comparing other approaches to cross-coupling is left to future work.

3.5 Radio gateway

To translate the output from an analogue radio into packets using the chosen protocols, a radio gateway with support for these protocols is required. The gateway needs to be able to convert analogue audio into digital sound packeted in RTP, and should have full support for SIP as specified in Rosenberg et al. (2002), to allow interoperability with any SIP application. Further desired requirements of a gateway is for it to conform to Eurocae's specifications as much as possible, including support for the RTP header extension, and being able to support at least two radio channels.

As mentioned in section 1.3.1, the in-house *IPG 3000* was chosen, a gateway currently under development. It was specified as having support for both SIP and the header extension for RTP defined by Eurocae. The gateway has interfaces for two radio channels via RJ-45 jacks (the common Ethernet cable).

3.6 Network architecture

Decisions about network architecture are of great importance in any network based system. Whether to employ a distributed peer-to-peer, or a centralized client-server approach has a large impact on the system.

A centralized server architecture where the server forwards all the audio transmissions between clients, has the advantage that it is possible to exert full control over the media sessions, but one drawback is that if the server ceases to function, the whole system goes down, given that there is no backup. A fully distributed

peer-to-peer architecture has no single point of failure, and another advantage is scalability, as workload is divided among the clients (Hafner and Mahmoud, 2005).

More common for SIP is the intermediate method, where proxy servers and registrars are used to locate clients by user name and routing SIP messages, while RTP media streams are sent directly between clients. This way, audio latency will not be as large as when a server would have to process every audio packet. However, if the location of the destination is known, there is no need to send the SIP request to an intermediate server either; it can be sent directly to the recipient (Eurocae WG-67, 2009a, part 1, section 3.4.3).

For this prototype, it was deemed likely that all components would have fixed, known locations in the network, which means that it would not be necessary to employ proxy servers at all. Therefore, it was decided that the starting point should be a server-less, distributed architecture. To begin with, the functionality would only be implemented in the CWP module, and only if it later proved necessary, a server software should be developed. This was to lessen the number of components needed for the prototype and ease both development and testing.

3.7 The software

This section demonstrates how the prototypical user interface looks, explaining the interesting classes and how they interact when the program receives input of different kinds. Details about classes from external frameworks are out of scope for this report and will not be discussed. Also, the source code of the software cannot be disclosed in this report, since it is owned by the company where the project was carried out.

3.7.1 Graphical user interface

Two examples of how the GUI looks are shown in figure 3.5. The program window displays two radio channels as a strip of two buttons each. The buttons function both as input and visual indicators of the channel status. The topmost button is for enabling and disabling the channel. When enabling a channel, it will initiate the receive-only mode, indicated by “Rx” on a light yellow background on the second button. Pushing the bottommost button switches the channel between receive-only mode and transmit-receive mode, which is indicated by “TxRx” on a green background. When a squelch is received from the radio in either mode, the topmost button displays “Receiving” on a yellow background.

Below the channel strips is a text field with the IP address of the host to be contacted. Below this text field is a set of controls. The “Push to talk” button

3.7. THE SOFTWARE

enables PTT, allowing the user to send audio towards the radio. “Cross-couple” and “De-couple” enable and disable cross-coupling between the two channels, respectively. In an actual system, the user would have to select which channels to cross-couple, but with only two channels there is only one possible configuration. Therefore, only a button which couples the two existing channels is needed in the GUI. However, the underlying software allows for arbitrary configurations. When a channel is cross-coupled with another, it is indicated by the bottommost button displaying “XC” on a blue background. The button labelled “Terminate all calls” does exactly what it says on the tin: all ongoing SIP calls are terminated in a controlled manner, and the channels are reset to disabled status. Lastly, there is “Create new CWP window”: a button used only for testing purposes with the intent of quickly allowing the launch of a second instance of the program. This is to quickly test functionality locally without having to connect to a remote computer, by having one instance set up a SIP call to the other, .

In figure 3.5 a), channel 1 is in receive-only mode, and is receiving a signal. Channel 2 is in transmit-receive mode, and is not receiving any signal. Figure 3.5 b) shows that channels 1 and 2 are cross-coupled. A signal is being received at channel 1. The program retransmits this signal onto channel 2, but there is no visual indication of this.

Note that the GUI, as was explained in section 1.3.1, was not developed to be fully realistic. It was developed only for the purpose of testing the prototype, and some of the GUI elements described above would be different or absent in a real VCS.

3.7.2 Classes and interactions

The CWP program is a Windows form application, outlined as a class diagram in figure 3.6. The base class of the program is the `MainFrame` class, containing graphical components and their event handlers. This class contains a `CWP` object, which represents the logical functionality of a CWP. The `CWP` class contains lists of the existing radio channels and all the active calls, and keeps track of what channels are cross-coupled.

Each active call is represented by an instance of the `Call` class, which is responsible for handling the audio stream, as well as squelch and PTT signalling for the call. It also has knowledge about the SIP session it corresponds to, so that it can instruct the SIP session to terminate when needed, and an instance of `EurocaeHeaderExtension`, a class that represents the RTP header extension defined by Eurocae, as explained in section 3.4.2.2.

The `Channel` class represents a radio channel. It has knowledge of its corresponding `Call`, what number the channel is, and what state it is in. It must be exactly in one of the states `Off`, `Rx` (receive only), `TxRx` (send and receive) or

3.7. THE SOFTWARE

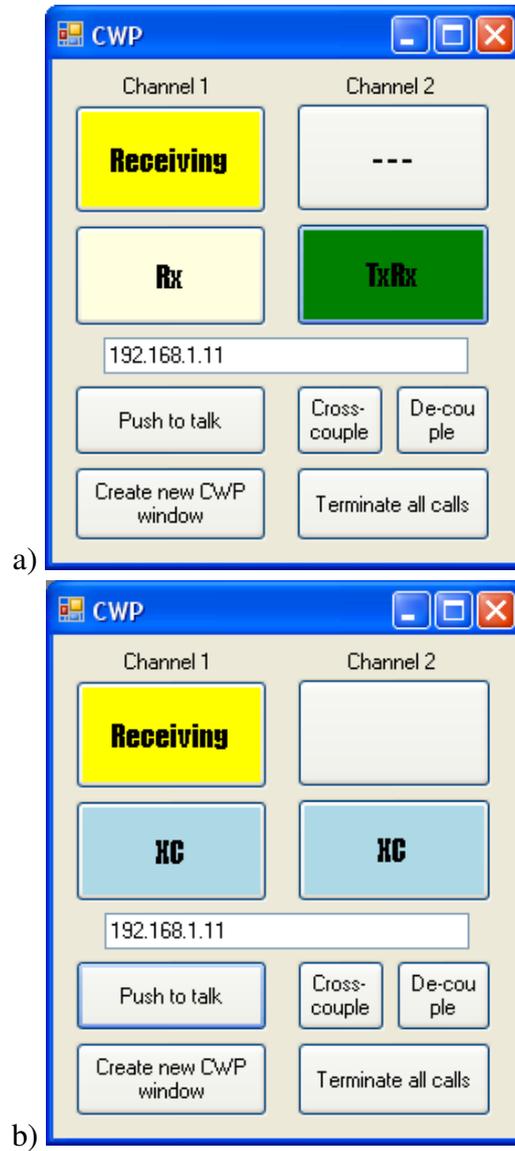


Figure 3.5: The GUI as it appears in two different situations. a) Channel 1 is in receive-only mode, and channel 2 is in transmit-receive mode. b) The channels are cross-coupled.

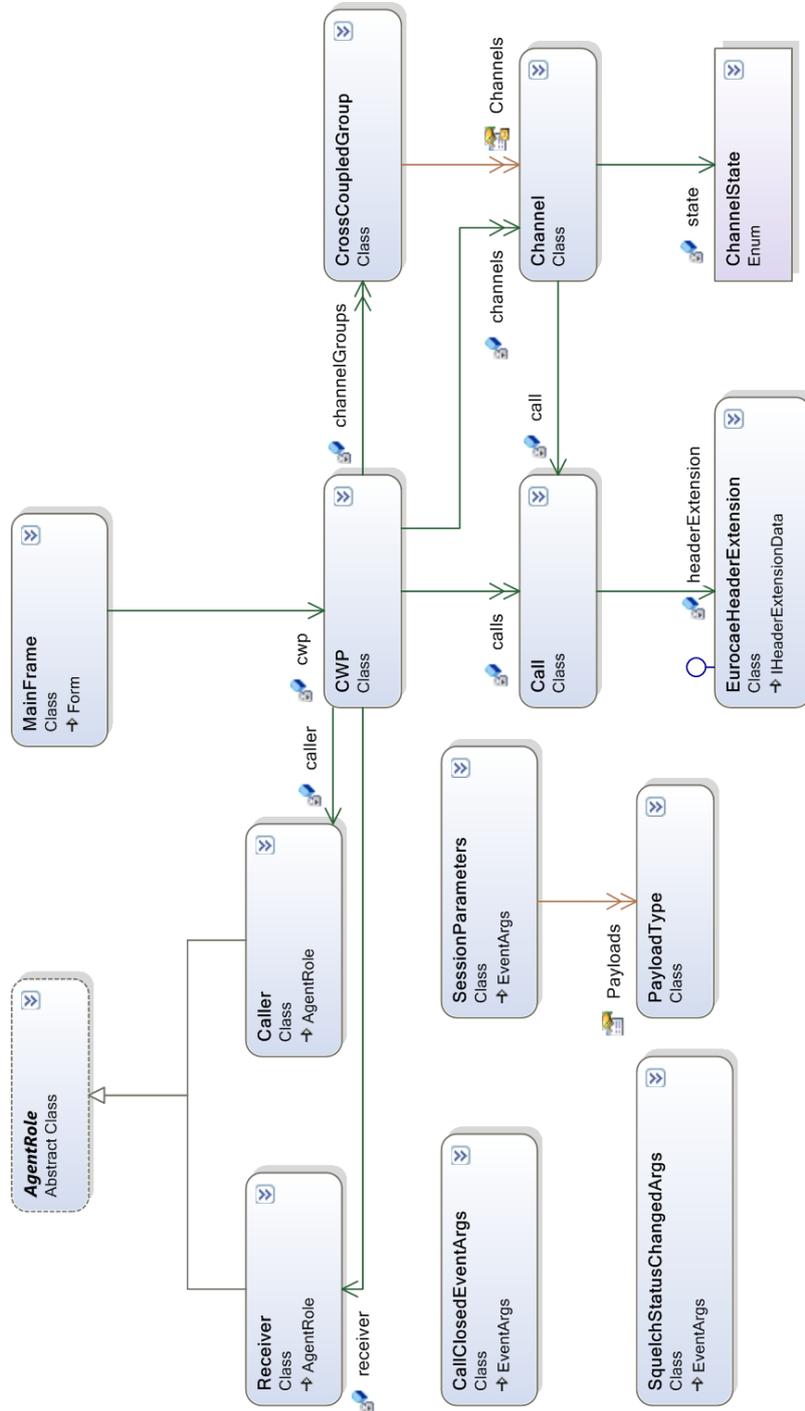


Figure 3.6: Class diagram of the CWP software.

3.7. THE SOFTWARE

CrossCoupled, which are present in the ChannelState enum. Upon any change of channel state, the Channel fires an event which the MainFrame listens to, so that the GUI is updated with the proper visual indicators.

The CWP class also has access to the two classes Caller and Receiver, which handle outgoing and incoming SIP requests, respectively. Caller and Receiver are logically similar, and have similar member variables. Therefore, they are both descendant of the abstract class AgentRole. They have access to the SIP stack, and contain a list of all accepted audio codecs, which infers what parameters can be accepted during negotiation of the SIP session.

There are also four different classes that act as arguments to events firing. SessionParameters is used with events that occur when a SIP handshake is completed and a Call has been set up. Its members include a unique identifier that allows the CWP to find the Call that has been initiated, and all the needed parameters for the session, such as a list of the RTP payload types that are allowed in the session. Next is CallClosedEventArgs, used when a call is terminated. It contains a unique identifier that allows the CWP to find the Call that has been closed. Then there is SquelchStatusChangedEventArgs, which is used when the CWP notices that the squelch signal has been turned on or off from the radio gateway. It has just a boolean representing the current squelch status, and the name of the channel it concerns.

3.7.2.1 Enabling or disabling a channel

When the MainFrame receives a click on the button to enable or disable a radio channel, it forwards the button click to the CWP class. If the Channel object with the correct number is in its Off state, CWP creates a Call object, and then uses Caller to attempt to initiate a SIP session with the supplied IP address. The SIP session will have the receive-only attribute set. If it is successful, the Call is notified, and starts its audio and RTP handling. Also, the Channel is set to Rx state, and it is associated with the Call.

If the Channel was On when its enable-disable button was clicked, the CWP finds the Call associated with the Channel and instructs it to terminate. The Channel's state is set to Off.

3.7.2.2 Switching between modes

As a mouse click on the bottommost button of a channel strip is registered by MainFrame, it is forwarded to the CWP class. If the channel is in the Rx or TxRx state, the CWP finds the Call associated with the Channel and terminates it. Then it uses the same procedure as above to create a new Call, with the stream mode of the SIP session set to receive-only or send-and-receive accordingly. Normally

3.7. THE SOFTWARE

in a SIP application, the stream mode of a session can be updated by simply sending a new INVITE request, inside the session. However, the radio gateway does not support within-dialogue INVITE, and therefore the whole session must be terminated and a new must be initiated to change the stream mode. If the channel is Off or CrossCoupled when the button is clicked, nothing happens.

3.7.2.3 Cross-coupling or decoupling

A click on the “Cross-couple” button calls a method in CWP that takes a list containing the numbers of the channels to be cross-coupled. No radio channel can be in more than one cross-coupled group, so if any of the Channels is already CrossCoupled, the method does nothing. Otherwise, Channels that are Off or Rx are initiated into sessions with send-and-receive mode on, and all involved Channels have their state set to CrossCoupled. They are also added to a new CrossCoupledGroup.

3.7.2.4 Push-to-talk

Clicking the “Push-to-talk” button makes the CWP find all Channels that are in a TxRx or CrossCoupled state, and call a method in their associated Calls. The Call object switches its boolean field ptt, and changes its EurocaeHeaderExtension object to reflect the current PTT status. Then, depending on whether PTT was turned on or off, the Call starts or stops the audio handling object that receives audio from an audio input device and sends it as RTP. Finally, it locks the change of the RTP header into place, and starts transmitting RTP packets signalling PTT with the new header, and containing audio taken from the input device.

3.7.2.5 Handling squelch

Incoming squelch signals from the network are handled by using events. The occurrence of a squelch is first registered by a Call, which can elicit from the RTP packets whether squelch is on or off. According to the Eurocae specifications, squelch should be indicated by a flag in the extended RTP header. The functionality to read squelch status from this flag is implemented in the Call class. Unfortunately, the radio gateway does not use the RTP header extension. Instead, the gateway starts transmitting an audio stream via RTP to the caller immediately upon reception of a SIP INVITE request, without waiting for a squelch to occur. Then, when squelch occurs, the gateway starts sending a new, duplicate audio stream to an address that can be configured in its maintenance interface. This behaviour unfortunately makes it impossible to properly implement a general solution for reading a squelch signal from this gateway, as the audio stream

3.7. THE SOFTWARE

that is transmitted upon squelch is sent to an address not related to the user who initiated the SIP call.

A very limited solution, that is, for only one CWP, was implemented in the following way: in the maintenance interface of the gateway, the address to this single CWP is entered. Then, when connecting to the gateway, an RTP stream immediately starts transmitting from it. Upon squelch, the second stream is sent to the same address. The appearance of a second RTP stream can be detected by the program, and this is exactly what the `Call` object does to detect a squelch from the gateway.

When the `Call`—by which ever method—has registered a squelch, it starts the object that receives RTP packets from a stream, decodes them, and outputs audio on an audio output device. `Call` also fires an event which is received by the CWP, and from there propagates to the `MainFrame` which updates the GUI.

3.7.2.6 Keep-alive messages

The Eurocae specifies a way to ensure that the connection to the other host is not broken. This is done by the periodical emission of RTP packets without audio data during silent periods. The packets use the header extension to signal squelch or PTT, and indicate whether the node is receiving packets from the other node or not. When a node is no longer receiving keep-alive packets (nor any other RTP packets) for a certain time, it times out the session, terminates it and attempts to restart it. This functionality is implemented in the `Call` class, so that it transmits these keep-alive packets at regular intervals.

3.7.2.7 Library changes

A few things had to be changed in the *Lumisoft.Net* library during development. For instance, on SIP session set-up, the library automatically tried to contact an address over the Internet, for IP address resolution. There was no option for disabling this behaviour. Since the prototype was to operate in a LAN with no Internet access, this behaviour had to be removed by removing the code related to it.

Furthermore, the library did not have support for the RTP header extension. Therefore, this functionality had to be implemented. A class named `RTP_HeaderExtension` was added to the library, containing a `length` field, a `type` field, an array representing the extension data, and a method that encodes the extension into a byte array. A reference to this class was added to the already existent class representing an RTP packet. When it comes to the Eurocae-specific implementation of the header extension, a class representing it was created, including the fields specified

3.7. THE SOFTWARE

by Eurocae, and also containing a method that writes the extension into an array of bytes. This class was named `EurocaeHeaderExtension`.

In this way, an RTP stream can be set to use a particular `RTP_HeaderExtension` object, which can in turn be loaded with the byte array representation of an instance of the `EurocaeHeaderExtension`. When the stream sends any packet, the `RTP_HeaderExtension` is encoded in an array of bytes and added to the packet header.

Chapter 4

Delay measurements

As previously mentioned, there are many requirements for voice quality set forth in the Eurocae documents. Several of these are related to the maximum amount of delay that may occur for signalling and voice transmission. To verify that a VCS satisfies the requirements, one or more ways to measure these delays are necessary. This section discusses different methods for this.

In a VCS, delays can in general be caused by several reasons. These include network latencies such as queuing delay and propagation delay, the performance of the CWP software, the radio and other control centre equipment, but also VoIP latencies such as packetization, de-jitter buffer, and serialization (Bardach et al., 2003; Hafner and Mahmoud, 2005). In the prototype, however, network latencies are negligible, and no real radio was used.

In its most general sense, a measurement for latency requires the observation of at least two events at some sample points in the system, and the possibility to register and compare the time at which the events occurred, so that a time difference can be deduced. The sample points can be different points in the system, as in the case of a one-way test, in which one sample point is at the source and one at the destination. They could also be at the same place in the system, in case of a round-trip measurement.

Only one diagram depicting the specification for a delay requirement is included in this chapter. Specifications and measurement set-ups for all the requirements can be found in appendix B.

4.1 Delay requirements

Here, the different delay requirements that were selected from the Eurocae documents are explained, including what their sample points are.

4.1. DELAY REQUIREMENTS

4.1.1 Frequency key activation response time

This delay requirement has two parts.

- a) When a channel is selected for receiving at the CWP, received audio should be heard by the controller within 300 ms. See figure 4.1.
- b) When a channel is selected for transmitting, it should be possible to transmit to the channel within 300 ms. See figure 4.2 on the next page.

The first event of interest for both parts of this requirement is the activation of a channel at the CWP. The second event is for a) when sound from the radio channel is heard at the CWP's audio output, and for b) when audio can be transmitted to the frequency that was selected. That the radio is ready for transmitting can be noticed in the CWP by the reception of an OK response from the gateway.

Therefore, the first sample point is at the CWP user interface, when the button for a channel is pushed. For a), the second sample point is at the audio output of the CWP, and for b), it is the point in the application where a SIP response is detected.

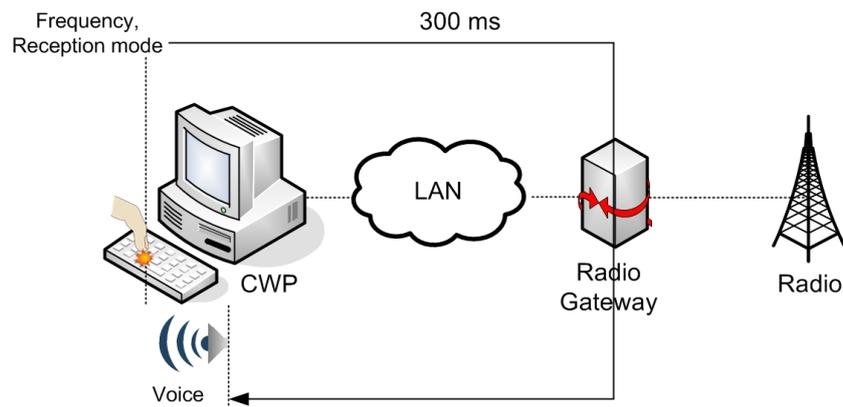


Figure 4.1: Specification for frequency key activation response time, a) receive mode

4.1.2 Transmitter activation delay

This requirement is concerned with the delay from the moment PTT is activated on the CWP until it reaches the radio transmitter. The maximum delay allowed to the radio is 80 ms and another 20 ms for the radio to output it. The first event to be registered here is the occurrence of PTT, with a sample point at the interface of the CWP. The other sample point is when the radio is available to transmit. See figure B.1.

4.1. DELAY REQUIREMENTS

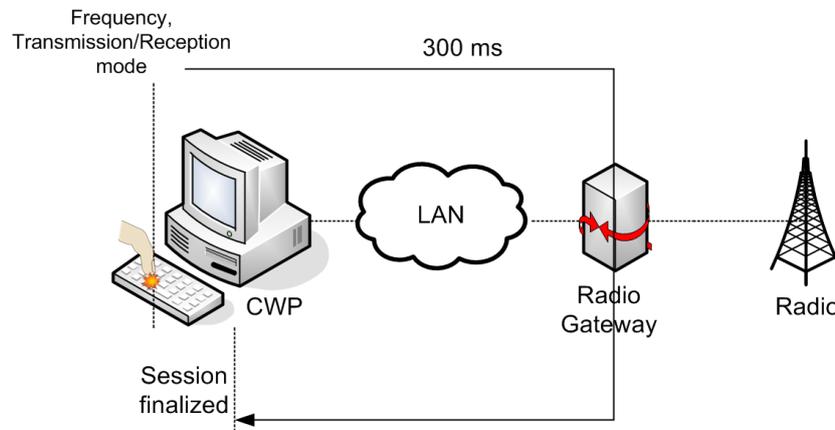


Figure 4.2: Specification for frequency key activation response time, b) transmit-receive mode

4.1.3 Aircraft call indication delay

This requirement specifies the delay from that a transmission from an aircraft call is received at the radio, until it is being indicated at the CWP. The maximum delay allowed is 100 ms, where 50 ms is from the time the radio gets the call until it outputs it, and another 50 ms is allowed for the call to be indicated on the CWP. Therefore, the first sample point is at the radio, where the event to observe is the reception of a radio signal from an aircraft. The other sample point is the CWP interface, where the indication should be observed, as illustrated in figure B.2.

4.1.4 Ground transmission voice delay

This requirement is concerned with the voice delay of a transmission from the CWP. The maximum delay allowed is 130 ms, where 120 ms is from the time a controller speaks into a microphone at the CWP until it reaches the radio transmitter. Another 10 ms is for the time from when sound is input to the radio transmitter until it is output. The first sample point will be the audio input device of the CWP where the event of interest is the appearance of a sound. The second sample point is where the audio is being output from the transmitting radio. See figure B.3.

4.1.5 Ground reception voice delay

The ground reception voice delay is essentially the same as the previously described requirement, but in the other direction: from radio to CWP. Again, it is specified to a total delay of 130 ms, of which 10 ms is reserved for delay in the

radio and 120 ms for the sound to reach from the radio output to the audio output of the CWP, see figure B.5. The sample points here are the radio with the appearance of a reception as the event, and the audio output of the CWP.

4.1.6 Loopback delays

This requirements does not exist in the Eurocae documents, but is a unification of two delay requirements. Since it was not possible to use the software timer method on any of the delay requirements except for frequency key activation response time, a loopback set-up was used to solve this issue since it makes it possible to have both of the sample points on the same device: the CWP. However, it is only possible to measure the sum of the delays for the joined requirements. The loopback delays were also measured using the oscilloscope method for comparison.

When constructing a loopback delay out of two different requirements, it can be known that at least one of them has failed, if the loopback time is greater than the sum of the allowed delay for the two requirements. However, it must be noted that if it takes less than this sum, that does not mean that both requirements have been fulfilled. It could also be that one of them takes longer than required, while the other is below its limit.

4.1.6.1 Transmitter activation and aircraft call indication loopback delay

This delay requirement is a joint of transmitter activation delay and aircraft call indication delay. The maximum delay for the loopback in question is $80 + 50 = 130$ ms. This means if it takes more than 130 ms, either the requirement for transmitter activation or aircraft call indication delay has failed. See figure B.4.

4.1.6.2 Ground transmission and reception voice loopback delay

This delay measurement is a joint of ground transmission voice delay, and ground reception voice delay. The maximum delay for the loopback is the sum of the delays for both requirements that the loopback consists of, $120 + 120 = 240$ ms. See figure B.6.

4.2 Methods

This section explains three different methods to measure delays: measuring electrical signals with an oscilloscope, using software timers, or using time logging.

4.2.1 Oscilloscope

An oscilloscope is connected to each of the sample points of the latency test. If the event of interest is not already the emission of an electrical signal, the system is set up so that a signal is emitted when the event occurs. The signals will be registered by the oscilloscope, and it is then possible to establish the time difference between them.

This method has the advantage that it measures the whole system from an external vantage point, and unless the system was changed to emit electrical signals for the test, it requires no knowledge about how the system is implemented internally. It is therefore a suitable method if the latency requirement includes components that are black-box to the testers, and whose internal workings cannot be probed directly. A disadvantage of this method is that emitting an electrical signal that is not part of regular system operations can lead to an additional delay that must be taken into consideration.

4.2.2 Software timers

Timers can be created inside the software that start and stop when certain events of interest occur. The advantage of this method is that it is easy to implement, as it only requires the addition of a few lines of code. A drawback is that this method requires both events to occur on the same software.

Measuring on packets One approach is to create timers inside the software that start and stop with the sending and reception of certain RTP packets. This can be implemented by checking the information within the RTP packets.

Loopback signals This can be implemented by connecting an output from the gateway straight to an input. One advantage with loopback measuring is that it enables both events to occur on the same node. An example of when this method is of interest is when testing voice reception or voice transmission which are one-way measurements between two different nodes. Since the events occur on different nodes, they do not have a shared timer to start and stop, but by looping back the audio, a timer could be started and stopped at the CWP, as explained in section 4.3.8.

4.2.3 Time logging

Another software method that can be used is by logging the time that the events of interest occur. The delay can then be measured by calculating the difference between the times recorded in the log. An advantage with this method compared

to the previously mentioned software timer method, is that logging enables the measurement of sample points when they are at different nodes. But for this to be realized in a correct manner the devices need to have their system clocks synchronized. This can be done with the clock synchronization protocols Network Time Protocol (NTP) or Precision Time Protocol (PTP). They are employed by using a time server that the clients synchronize their system clocks against. But for this to work, access to the source code is required for necessary configuration in both clients. Therefore this method was not possible to implement since the source code of the gateway was not available.

4.3 Set-up

This section is concerned with the technical details of the equipment that was used during the delay measurements, and the set-up for each of the measurements. Sample outputs for each of the measurements are provided in appendix B.

4.3.1 Equipment

The delays caused by the radio cannot be affected by the development of the software or network, and the radio could be a given factor if the VCS is to be deployed at an airfield that already has radio sites. The radio can therefore be considered to be outside of the scope of a VCS, which is why the Eurocae delay requirements often give an alternative version, excluding consideration of the radio. Thus, a radio and the delays it causes were excluded from the delay measurements, and no radio was connected to the system during testing. Instead, any necessary signal, audio, or action from the radio was simulated. This was done by exposing the wires in the RJ-45 cable that normally connects the gateway to a radio. The exposed wires were also used as sample points for measurements concerning output to or input from the radio. To check if PTT was signalled from a CWP, pin 7 and 8 of the cable were measured. The gateway has a relay that it signals PTT with, by connecting it which leads to a closed circuit in the radio. In figure 4.3, the dashed line represents the radio, which is simulated by using a power source, a resistor, and a relay. A squelch can be simulated by connecting pins 1 and 2 together to create a closed circuit in the gateway. When disconnected the potential difference is circa 4 V and when connected it is 0 V.

To simulate audio from the radio, a tone generator emitting a sine wave was connected to pins 3 and 6 of the RJ-45 cable, which are the wires used for audio input to the gateway. The tone generator was also used as the audio from a controller by connecting the audio output of the generator to the microphone input of the CWP.

4.3. SET-UP

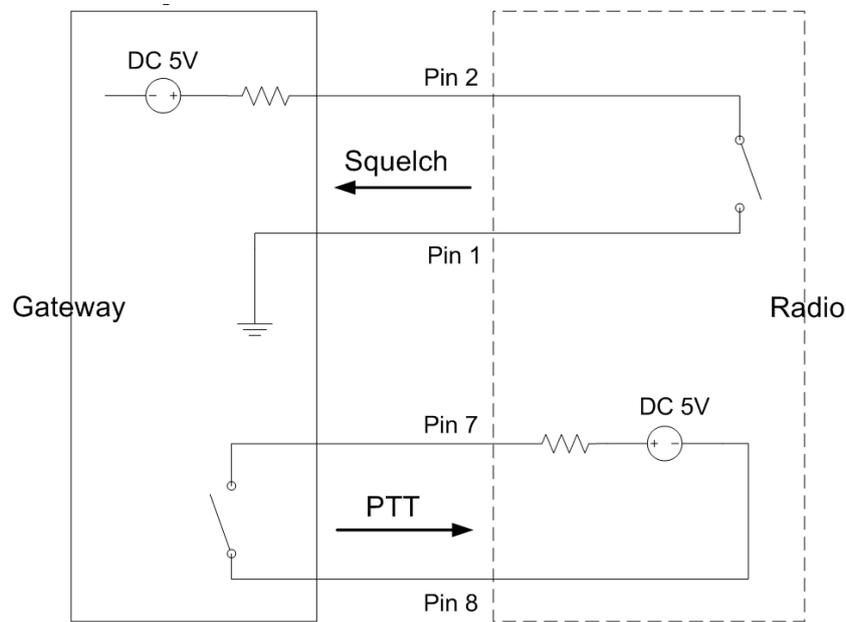


Figure 4.3: Schematics over how push-to-talk and squelch signals are sent to and from the gateway.

An RS-232 cable was connected to the serial port on the CWP to create an electric signal measurable by an oscilloscope with the occurrence of events such as button clicks and packet reception. Upon the detection of such events in the software, a certain pin was set high (+10 V) or low (-10 V). This was used as input to the oscilloscope when needed.

An oscilloscope with two channels was used to register when events were triggered between one or two sample points depending on the delay under measurement.

4.3.2 Frequency key activation response time

This delay requirement consists of two parts. Part a) is a round trip measurement with both sample points at the CWP, one event being the activation of a frequency in receive mode, the other being the reception of audio from an aircraft transmission with a maximum delay of 300 ms. See figure 4.1 on page 30.

Part b) concerns the delay from the time when a frequency is selected in transmit and receive mode until the moment it becomes usable for transmitting, which is from the time the transmit and receive button is clicked until an OK response is received. The maximum delay is 300 ms. See figure 4.2 on page 31.

Oscilloscope measurements

Part a) The CWP was connected to both channels of the oscilloscope. One channel registered when a frequency was selected to be in receive mode, while the other registered audio output from the CWP. A tone generator was connected to the audio input of the radio gateway. The squelch was active to enable a constant audio transmission from the radio side. See figure B.7.

An example of the oscilloscope measurements can be seen in figure B.8. Channel 1 registers the selection of a frequency in receive mode while channel 2 registers the received audio of the CWP.

At the beginning of the measurement the frequency was not selected. This is indicated by the output of channel 1 being -11 V. This can be verified by channel 2 displaying 0 V which indicates no audio output from the CWP even though the radio side is transmitting audio. When the frequency is selected in receive mode the signal on channel 1 shifts to +11 V. Shortly after that audio is detected on channel 2 which is indicated by a sine wave. The response time can be obtained by measuring the difference between the moment when channel 1 changed to +11 V and the first sign of audio in channel 2.

Part b) The serial port of the CWP was connected to one oscilloscope channel to register when a frequency was selected for transmit mode and when it was ready for use. See figure B.9 on page 78.

An example from the oscilloscope measurements can be seen in figure B.10 on page 79. The selection of a frequency to be in transmit mode is indicated when channel 1 shifts from -11 V to +11 V. When a session is finalized for that frequency it is ready to be used in transmit mode. This is indicated by a shift in channel 1 back to -11 V. The response time can be obtained by measuring the difference between the moment channel 1 shifted to +11 V and the moment it changes back to.

Software timer

Part a) When a frequency was selected for receive mode, the reception of its button click event caused a timer to start. The timer was stopped when the software received the first RTP packet.

Part b) As in part a), the timer starts with a button click event, but for the selection of a frequency in transmit and receive mode. Here, the timer was stopped on reception of an OK SIP response.

4.3.3 Transmitter activation delay

This delay requirement is concerned with the time it takes from the moment when PTT is pushed on the CWP until the moment when it reaches the radio side. The maximum delay requirement is 80 ms. See figure B.1.

Oscilloscope The CWP was connected through the serial port to one channel on the oscilloscope, to register when PTT was pushed, while the other channel was connected to an RJ-45 cable on the radio side to register when the PTT was received, which can be seen in figure B.11.

A sample from the oscilloscope measurements can be seen in figure B.12. Channel 1 outputs the status of the CWP serial port, which indicates PTT, while channel 2 outputs the PTT status on the radio side. The delay can be obtained by measuring the difference between the moment when channel 1 changes from -11 V to +11 V and the moment when channel 2 changes from +5 V to 0 V.

4.3.4 Aircraft call indication delay

This delay requirement is concerned with the time it takes from the moment squelch is activated on the radio side until the moment when it is indicated on the CWP. The maximum delay requirement is 50 ms. See figure B.2.

Oscilloscope measurements The CWP serial port was connected to one of the channels in the oscilloscope to register when a squelch was indicated on the CWP. The other channel of the oscilloscope was connected to the squelch input of the gateway.

A sample from the oscilloscope measurements can be seen in B.14, where channel 1 displays the squelch activated on the radio side and channel 2 displays the indication of squelch in the CWP. The delay can be obtained by measuring the difference between the moment when channel 1 changes from 4 V to 0 V and the moment when channel 2 changes from -11 V to +11 V.

4.3.5 Ground transmission voice delay

This delay requirement is concerned with the time it takes from the moment a controller speaks until the moment when the sound reaches the radio side. The maximum delay requirement is 120 ms. See figure B.3.

Oscilloscope measurements A tone generator was connected to the microphone input of the CWP. This connection was intercepted by one of the oscilloscope

4.3. SET-UP

channels to register when a controller was talking into the microphone. The audio output from the gateway was connected to the other channel of the oscilloscope to register the audio transmitted to the radio side. See figure B.15.

An example from the oscilloscope measurements can be seen in figure B.16, where channel 2 displays the transmitted audio from the CWP while channel 1 displays the audio output from the gateway. Here, the delay can be calculated by measuring the difference between the first peak of channel 2 to the first peak of channel 1.

4.3.6 Ground reception voice delay

This delay requirement is concerned with the time it takes from the moment audio is received on the radio side until the moment when it is output at the CWP. The maximum delay requirement is 120 ms. See figure B.5.

Oscilloscope measurements A tone generator was connected from the radio side to the audio input on the gateway. This connection was intercepted by one of the oscilloscope's channel to register audio reception. The other channel of the oscilloscope was connected to the audio output of the CWP. See figure B.17.

An example from the oscilloscope measurements can be seen in figure B.18, where channel 1 displays received audio from the radio side while channel 2 displays the audio when output from the CWP. Here, the delay can be calculated by measuring the difference between the first peak of channel 1 and the first peak of channel 2.

4.3.7 Transmitter activation and aircraft call indication loop-back delay

This delay measurement is concerned with the time that it takes from the moment the PTT is activated until the moment a squelch is received, on the CWP. See figure B.4. For this loopback a circuit was created, depicted in figure B.19(b). The PTT signal is used as a control signal in the circuit, so that a squelch is activated on reception of a PTT.

Oscilloscope measurements The CWP was connected through the serial port to one channel on the oscilloscope to register when PTT was pushed by setting a pin high. The loopback circuit ensures that squelch is sent back when PTT reaches the radio side of the gateway. When the squelch was indicated on the CWP, it was registered through the serial port, by setting the same pin low. The delay can be obtained by measuring the difference between the moment when channel 1

4.3. SET-UP

changes from -11 V to +11 V and the moment when it changes back. See figure B.20.

Software timer When the PTT button click event is received by the software, it starts a timer. Finally when the squelch is indicated on the CWP, the timer stops.

4.3.8 Ground transmission and reception voice loopback delay

This delay measurement is concerned with the time that it takes from the moment when audio is input until the moment same audio is output on the CWP. See figure B.6. To enable transmission and reception simultaneously, both PTT and squelch had to be active before the measurement could start.

Oscilloscope measurements A tone generator was connected to the microphone input of the CWP. The connection was intercepted by one of the channels of the oscilloscope to register when a controller was transmitting. The loopback was realized by connecting the audio output with audio input on the gateway. The other channel of the oscilloscope was connected to the audio output of the CWP to register audio reception. See figure B.21.

An example from the oscilloscope measurements can be seen in figure B.22, where channel 1 displays the audio input to the CWP while channel 2 displays the audio output. Here, the delay can be calculated by measuring the difference between the first peak of channel 1 and the first peak of channel 2.

Software timer A timer was started when audio was input to the CWP. The timer was stopped on reception of the first RTP packet.

Chapter 5

Results

This chapter presents the results that were achieved throughout the project. First the results of the delay measurements are presented. Then, all the chosen requirements from appendix A are gone through, and it is evaluated whether the prototype conforms to each of them or not.

5.1 Measurement results

In this section, the results for the delay measurements are presented. For requirement descriptions, see section 4.1. The results are further analysed and discussed in chapter 6.

5.1.1 Frequency key activation response time

The delay requirements for frequency key activation response time part a) and b) both had a maximum delay requirement of 300 ms. There was only one occasion where the delay measurement did not meet its requirement. However, their average was well below the required maximum delay. For part a) see table C.1 and for part b) see table C.2.

5.1.2 Transmitter activation delay

The maximum delay requirement for transmitter activation delay was 80 ms. All measurements were well below the required maximum delay, with an average of 46 ms. See table C.3.

5.2. FULFILMENT OF REQUIREMENTS

5.1.3 Aircraft call indication delay

The maximum delay requirement for aircraft call indication delay was 50 ms. All measurements were well below the required maximum delay, with an average of 30 ms. See table C.4.

5.1.4 Transmitter activation and aircraft call indication loopback delay

The sum of the delay requirements for transmitter activation and aircraft call indication loopback delay was 130 ms. All measurements were well below this sum, with an average of 80 ms for oscilloscope, and 91 ms for software timer, which implies that at least one of the two requirements has been fulfilled. See table C.5.

5.1.5 Ground transmission voice delay

The maximum delay requirement for ground transmission voice delay, was 120 ms. None of the measurements were below the required maximum delay, and the average was 144 ms. See table C.6.

5.1.6 Ground reception voice delay

The maximum delay requirement for ground reception voice delay was 120 ms. All measurements were below the required maximum delay, with an average of 104 ms. See table C.7.

5.1.7 Ground transmission and reception voice loopback delay

The sum of the two delay requirements was $120 + 120$ ms. Only two of the oscilloscope measurements were below this delay, and the average was 245 ms. All measurements that were carried out with a software timer passed well below the required maximum delay with an average of 115 ms. See table C.8.

5.2 Fulfilment of requirements

In this section, all the chosen requirements will be gone through. For each, it will be explained how the requirement was met, or why it was not met. The requirements will not be explained here, instead the reader is encouraged to refer to appendix A.

5.2.1 Requirements for milestone 1

AUDIO 4: Voice packetization interval requirement Priority: 3

This requirement was partially met. The CWP sends a SIP INVITE specifying that it desires to use 20 ms packets, and the gateway also supports this parameter. However, the gateway supports only sending packets with 20 ms of audio in each, and it only accepts packets with a multiple of 20 ms per packet, according to its documentation (Saab Danmark A/S, 2009). Thus, the gateway would not support packets with 10 or 30 ms of audio, and neither was this possibility implemented in the CWP.

AUDIO 6: Voice coding requirement Priority: 3

This requirement was met. Both the CWP and the gateway support PCM A-law and μ -law. Support for the optional voice compression codecs was not implemented.

COMMUNICATION MODEL 3: Applicable protocol Priority: 3

This requirement was met. SIP and RTP are the used protocols.

RTP 2: RTP header specifications Priority: 3

This requirement was partially met. The CWP transmits packets with the correct header, but the gateway does not send packets with the header extension.

RTP 11: RTP and RTCP UDP port number Priority: 3

This requirement was met. The prototype system always uses an even port number for RTP and the next odd number for RTCP.

RTP 19: Keep-alive messages Priority: 3

This requirement was partially met. The CWP includes the parameters for keep-alive messages in its SDP, and if those parameters are accepted by the other endpoint, the CWP will send such packets accordingly, and register those that come from the other node. However, once again the gateway does not support the requirement. It will ignore the SDP parameters for keep-alive messages, and does not have the capability to send such messages.

SIP 1: SIP version Priority: 3

SIP version 2 is used in the system.

5.2. FULFILMENT OF REQUIREMENTS

SIP 4: SIP message body (SDP) Priority: 3

This requirement was partially met. When the CWP sends SIP messages including an SDP body, the required fields are present. But the gateway does not support all of the fields, such as the ones related to the keep-alive messages.

SIP 2: SIP supported requests Priority: 2

This requirement was partially met. The essential SIP request types for setting up and taking down a session are supported by both CWP and gateway, but the SUBSCRIBE / NOTIFY event package was not implemented in the CWP. As far as it is known, the gateway does not support SUBSCRIBE and NOTIFY and the event package either.

SIP 3: SIP supported responses Priority: 2

This requirement was partially met. There are dozens of different responses, and implementing realistic support for each of them would not have been feasible for this prototype. Only support for the most essential responses was implemented in the CWP. It was not tested whether the gateway supports all of the required responses.

SIP 5: SIP address format Priority: 2

This requirement was not met. As explained earlier in this chapter, only a CWP was developed, and in addition the system consists of a gateway and connections between them. No registration service or proxy server was used, and addressing is done with IP addresses.

To call the two radio channels on the gateway, SIP URIs with the same IP address but different user names are used. The two channels were simply given the user names “radio1” and “radio2”, respectively. For example, to connect to channel 1 an INVITE is sent to the address `sip:radio1@<gateway IP>`.

According to the requirements, the user name should contain the frequency and what type of air traffic service unit it is, but since the radio is only simulated, there is no physical frequency associated with the channels. Therefore, only simple placeholder names were used.

The CWP uses the user name “cwp”, but since it is only the CWP that initiates sessions and never the radio side, what user name it has bears little significance as long as internal telephone functionality is not implemented.

5.2.1.1 Summary for milestone 1

The requirements with priority 3 were all at least partially met. Those that were not fully met were all due to limitations in the gateway, such as its lack of support

5.2. FULFILMENT OF REQUIREMENTS

for the RTP header extension. They were all implemented as far as it was possible with no possibility to make changes to the behaviour of the gateway. For the priority 2 requirements, some were not fully implemented because it was decided that the effort it would take would outweigh the return of interesting results of, for example, implementing realistic support for every one of the dozens of responses.

5.2.2 Requirements for milestone 2

SIP 8: SIP call set up procedure Priority: 3

This requirement was partially met. The RTP header extension, which is mentioned in this requirement, is not used by the gateway. The other parts of this requirement were met.

COMMUNICATION MODEL 5: Communication initiation between VCS and radios Priority: 3

This requirement was partially met. Again, the missing part is that the gateway does not use the RTP header extension to send squelch signals. Otherwise, the requirement was met.

RTP 14: RTP header extension description Priority: 3

This requirement was met. Whenever the RTP header extension is present, it follows this requirement.

RTP 15: RTPTx information field Priority: 3

This requirement was met. The RTP header extension that the CWP can send follows these requirements.

RTP 16: RTPRx information field Priority: 3

This requirement was not met. The radio side, that is the gateway, never sends RTP packets with the extended header. Therefore, the RTPRx variant never appears in the system.

5.2.2.1 Delay requirements

The complete results for all of the different measurements are found in section 5.1. The results of these measurements will be analysed here and it will be discussed what they imply for the fulfilment of the requirements.

5.2. FULFILMENT OF REQUIREMENTS

RADIO FUNCTIONAL 10: 300 ms max frequency key activation response time Priority: 2

This requirement was met. Frequency key activation response time was measured using both an oscilloscope and software timers. All of the ten software timer measurements were well below the limit, but of the oscilloscope measurements, one was 53 ms above. This was the only measurement above the limit, so its appearance could be considered anomalous. If so, the requirement has been met.

RADIO PERFORMANCE 3: 100 ms max transmitter activation delay Priority: 2

This requirement was met. The oscilloscope measurements were all 25 ms or more below the limit.

RADIO PERFORMANCE 4: 100 ms aircraft call indication delay Priority: 2

This requirement was met. All measurements were below the limit by at least 13 ms. Transmitter activation and aircraft call indication were also measured together as a loopback measurement. These measurements were also within the sum of the limits for the two requirements.

RADIO PERFORMANCE 6: 130 ms max ground transmission voice delay Priority: 2

This requirement was not met. The latency was 15–23 ms above the limit, according to the oscilloscope measurements.

RADIO PERFORMANCE 8: 130 ms max ground reception voice delay Priority: 2

This requirement was met. The measurements were consistent, and all below the limit by at least 12 ms.

SAFETY 5: No quality of service (QoS) degradation of IP-voice Priority: 2

This requirement was met. The prototype VCS operates on a closed LAN that it does not share with any other service that could degrade the QoS of the VoIP service.

SAFETY 1: Call type discrimination during call establishment phase Priority: 1

This requirement was not met. Support for the SIP header field for priority was not implemented in the CWP, and not tested for the gateway.

5.2. FULFILMENT OF REQUIREMENTS

SYS ENG 16: Detection of end-to-end connection loss Priority: 1

This requirement was not met. The system does not continually check connections between nodes. While this is an important reliability requirement, its implementation is fairly advanced and it was therefore left out.

SYS ENG 35: Speech activity detector with pseudo squelch signal generation

Priority: 1

This requirement was not met. A way to discover voice signals being transmitted without a squelch was not implemented.

5.2.2.2 Summary for milestone 2

Again, the cause of priority 3 requirements not being fully met was largely the gateway lacking support for the extended header. The CWP software was developed as far as possible to meet the requirements. Among the delay requirements, the prototype passed all of them except one. This fact is discussed further in chapter 6.

5.2.3 Requirements for milestone 3

RADIO FUNCTIONAL 1: Radio access modes of operation Priority: 3

This requirement was met. The four different radio channel states were implemented in the CWP logic. A channel can be put in each of the states by interaction with the GUI of the CWP.

RADIO FUNCTIONAL 14: Simultaneous multiple frequency selections for transmission and reception Priority: 3

This requirement was met. By design, it is possible for the CWP to activate any number of channels for transmission and reception. However, the gateway only allows two radio channels, so no situation with more than two simultaneously selected channel will occur in the prototype.

RADIO FUNCTIONAL 15: Cross-coupling facility to be implemented Priority: 3

This requirement was partially met. Only duplex mode was implemented. A signal received on a cross-coupled channel will always be retransmitted on the other channels in the group. Simplex mode was deemed as the less interesting variant, as retransmission is the important part of cross-coupling. Therefore, simplex mode was left out of the prototype.

5.2. FULFILMENT OF REQUIREMENTS

RADIO FUNCTIONAL 16: Quantity of frequencies in a cross-coupled frequency group Priority: 3

This requirement was met. The CWP does not limit the number of radio channels in a cross-coupled group. However, only two radio channels exist in the prototype system, so for this reason there could never be a group with 28 channels.

RADIO FUNCTIONAL 17: Two cross-coupled frequency mode Priority: 3

This requirement was partially met. The cross-coupling functionality is largely implemented, except that simultaneous reception on both channels does not cause one of them to block.

RADIO FUNCTIONAL 20: Identical frequencies not allowed in two cross-coupling sessions Priority: 3

This requirement was met. The CWP cannot put a channel in CrossCoupled state in another cross-coupled group. If the channel was cross-coupled at another CWP, it would also be prevented from being cross-coupled again, since the channel would be busy on the gateway. The gateway does not support multiple SIP sessions on the same channel.

RTP 6: RTP PTT transmission performance Priority: 3

This requirement was met. The functionality concerning PTT has been implemented in the CWP.

RTP 7: RTP aircraft call transmission performance Priority: 3

This requirement was not met. The gateway does not conform to this requirement as it does not use the RTP header extension.

SIP 6: Basic call functionalities Priority: 3

This requirement was partially met. Usage of the priority header was not implemented in the CWP. It was not tested whether the radio gateway supported the priority header field or not. The remainder of this requirement has been implemented.

RADIO FUNCTIONAL 22: PTT precedence over cross-coupling transmission Priority: 2

This requirement was not met. The suppression of retransmissions in case of a PTT from the controller was not implemented. Only a basic cross-coupling functionality was implemented, so this requirement was omitted.

5.2. FULFILMENT OF REQUIREMENTS

RADIO FUNCTIONAL 24: First received aircraft call used as a cross-coupled group's incoming frequency Priority: 2

This requirement was not met. No designation for incoming frequency is done in the CWP. If multiple receptions occur, the CWP will attempt to retransmit all of them to the other channels in the group.

RADIO FUNCTIONAL 33: PTT and aircraft call locked-on condition prevention Priority: 2

This requirement was not met. No facility to discover the occurrence of an erroneous, indefinite PTT or squelch signal has been implemented.

RADIO FUNCTIONAL 35 Duplex and simplex cross-coupling functionality support Priority: 2

This requirement was partially met. As stated before, only duplex cross-coupling was implemented.

RTP 12: RTP radio PTT activation/deactivation Priority: 2

This requirement was met. The mandatory parts were implemented, even though the recommended redundant transmission of several packets upon the release of PTT was not implemented. When PTT is released, the CWP simply starts transmitting keep-alive packets at their normal rate.

RTP 13: RTP radio aircraft call activation/deactivation Priority: 2

This requirement was not met. The gateway does not conform to this requirement. It does not utilize the RTP header extension. It also transmits voice packets even in the absence of squelch, and it does not repeat any packet to signal the end of a squelch.

SYS ENG 31: Quantity of frequencies in a cross-coupled group Priority: 2

This requirement was partially met. By design, the CWP puts no limit on the number of frequencies in a cross-coupled group. The system management facility was not implemented.

SYS ENG 32: No limit on number of cross-coupled groups Priority: 2

This requirement was met. The software poses no architectural limitation on the number of cross-coupled groups. However, since the prototype only contains two radio channels, no more than one cross-coupled group is possible.

5.2. FULFILMENT OF REQUIREMENTS

RADIO FUNCTIONAL 18: Three cross-coupled frequency mode Priority: 1

This requirement was partially met. Cross-coupling over a group of arbitrary size has been implemented in the CWP. However, the blocking of simultaneous incoming transmissions and the cross-coupled PTT inhibition period were not implemented.

5.2.3.1 Summary for milestone 3

The requirements of highest priority for milestone 3 were not all met down to the last detail. Features considered less important such as implementing simplex cross-coupling in addition to duplex and implementing support for the SIP priority header field were omitted to allow more time to implement the more essential parts of this functionality. In another case, the requirement is restricted by the gateway not supporting the extended RTP header.

5.2.4 Requirements for milestone 4

COMMUNICATION MODEL 4: Simultaneous communications between VCS and radios Priority: 3

This requirement was not met. The gateway supports only one SIP session per channel.

SYS ENG 12: No architectural limit on active CWPs Priority: 3

This requirement was not met. Since the previous requirement is not met, multiple CWPs can only be connected to the gateway if they do not try to connect to the same channel. In the prototype, this limits the number of active CWPs to two, since there are only two radio channels.

RADIO FUNCTIONAL 2: Transmit configuration description Priority: 2

This requirement was not met. Priority levels for different controllers were not implemented.

RADIO FUNCTIONAL 12: Facility isolation / disconnection warning Priority: 1

This requirement was not met. Automatic discovery of disconnections has not been implemented.

5.2. FULFILMENT OF REQUIREMENTS

RADIO FUNCTIONAL 40: PTT identity notification Priority: 1

This requirement was not met. Identification of users when they activate PTT was not implemented in the prototype.

RADIO PERFORMANCE 5: 250 ms max cross-coupled PTT inhibition period, XC2 Priority: 1

This requirement was not met. The inhibition period was not implemented.

SIP 7: SIP audible tones control Priority: 1

This requirement was met. No call project tones are used when connecting to the radio.

SYS ENG 13: Simultaneous use of any system functionality combination

Priority: 1

It is unknown whether this requirement was met or not. The requirement does not outline any way to test that it has been met. In any case, systematic testing of system features in combination was not performed.

SYS ENG 39: Alarm for non-selected frequencies Priority: 1

This requirement was not met. There is no alarm if a radio channel is not selected.

5.2.4.1 Summary for milestone 4

Most of the requirements for milestone 4 were not met. One of the greatest obstacles is the failure of the first requirement in the section. The gateway only supports one SIP session per radio channel. Therefore it would be very hard to implement true support for multiple CWPs without changing the gateway or introducing some workaround.

Chapter 6

Discussion and conclusions

In this final chapter, the results from chapter 4 and chapter 5 are discussed, and conclusions about them are drawn. This includes analysis of the delay measurements, performance of the software, the requirements that were unfulfilled, an evaluation of the gateway, and reflections about the software development. Suggestions for further work are presented.

6.1 The delay measurements

This section discusses the results regarding delay measurements, including the methods for measuring delays that were used and why the different methods show different results.

6.1.1 Summary of the measurement methods

In this report, the idea of delay measurements was generalized to two events being registered at two sample points in the system. Three different methods to measure delays were presented, and two of them were tried on the delay requirements that were chosen for the project.

The oscilloscope method is very general, and has the advantage that the system under test can be viewed as a black box, as long as the events are electrical signals that are output from or input into the system. If the events are electrical signals sent between components of the system, it would need to be a little more transparent, so that it is possible to tap into these signals. Sometimes, however, it is not easy to correspond an event exactly to any electrical signal that could be tapped into, such as was the case with the measurements from chapter 4 where the first event consisted of clicking a button. Even though an electronic signal travels from the mouse to the computer at the click of a button, this signal would be cum-

6.1. THE DELAY MEASUREMENTS

bersome to tap into using an oscilloscope. If it cannot be done, then an electrical signal that the oscilloscope can register must be generated beyond the normal operations of the system, which in this project was done by having the software alter a pin in the serial port of the computer. Such a change requires that the system is transparent and changeable. It could also potentially affect the measurement.

This is contrasted to the method of using software objects that measure time. It requires a fully changeable system, so that these timers can be created, started and stopped whenever it is required. It is also difficult to implement when two events are at different components of the system. It is not clear how it would be possible to transfer a software stopwatch object between two different nodes of a network and still have it measuring time reliably. The method has the advantage that it is fairly easy to add the test directly into the code, and no external equipment is needed. Care must be taken to where in the code to start and stop the timer object, also, how and when to create and destroy the object, so that the measurement is affected as little as possible.

Another method is to use time logging. This works well for measurements with sample points in different nodes on the network. The requirement is, however, that those components are kept synchronized to a time server. If the events of interest are the reception or transmission of packets that already have the ability to contain time-stamps, then the logged time-stamps are simply used for those packets. Otherwise, this method as well requires some changing in the software of the system, to decide when the time-stamp should be acquired to provide the most reliable result.

In summary, regardless of what method one uses, it is often necessary to approximate the placing of the sample points, possibly adding some test-specific behaviour to the system, when it is impossible or difficult to measure at the exact sample point of the requirement.

6.1.2 Measurement differences between methods

When comparing the results in section 5.1 between requirements that were tested both with oscilloscope and software timers, the measurement values are sometimes significantly different. For the frequency key activation response time part a), for instance, the oscilloscope measurements showed on average circa 40 ms more than those made with software timers. What are the differences between the methods that could cause this difference? The actual requirement specifies a delay between that the user takes an action, and sound being presented to that user. The oscilloscope takes one of its sample points from the actual event that the requirement concerns: the output of audio from the CWP. The other, however, is an artificially created electric signal not part of normal system operations: the raising of a pin on the serial port of the CWP. The sample point could have been

6.1. THE DELAY MEASUREMENTS

approximated closer to the user action by tapping into the wire between the input device and the computer, but the simpler serial port method was chosen instead.

The software timer, on the other hand, has sample points inside the code of the software: one as soon as the CWP class registers a click on the button, and one upon reception of an RTP packet containing audio, which decoded has a volume above a certain threshold, see figure 6.1.

Thus, both measurement methods could affect the result. The operating system takes an unknown amount of time to register the mouse click and forward it to the application. The serial port takes an unknown additional time to activate after the actual mouse click. The analysis of the volume of each RTP packet could place an additional load onto the system. Any of these explanations could apply to why the measurements conducted with the two methods differ.

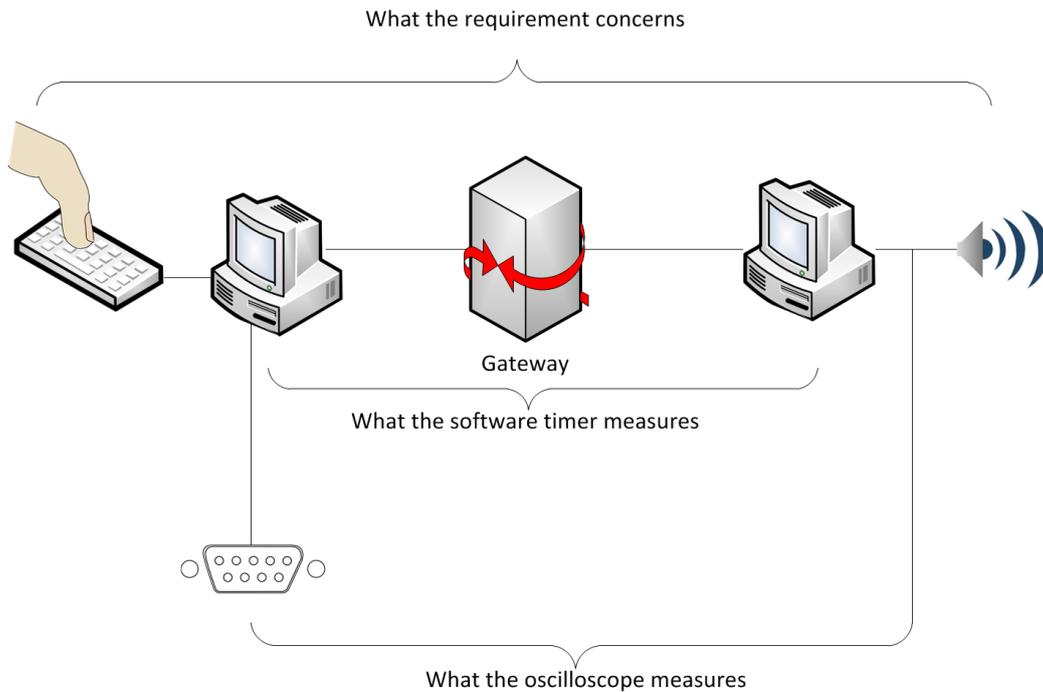


Figure 6.1: An example of discrepancies between measurement methods and the actual requirement, in the case of frequency key activation response time. Note that the the same CWP is depicted twice in this schematic; there are not two CWPs in the measurement.

Between different measurements, the oscilloscope method sometimes shows a lower delay than the timer method, and sometimes the other way around. For frequency key activation response time part b), the two results are closer, the timer results being only 7 ms greater. Aircraft call / transmitter activation loopback

6.1. THE DELAY MEASUREMENTS

delay has the timer results 13 ms greater. But in the measurement for ground transmission and reception voice loopback delay, they differ with as much as 130 ms on average, oscilloscope measurements being the greater. This means that no general conclusion that one of the methods usually shows a longer delay than the other can be drawn. It seems to be entirely dependent on how the tests are executed. This leads to the conclusion that it is important to consider what the difference is between the chosen sample point and that which according to the requirement is the actual sample point. Which factors does the test leave out, and which does it add? In the example above, the software timer leaves out two things. Firstly, the travel of a signal from the mouse to the computer, the registration of this mouse click in the operating system, and the forwarding to the application. The other thing is the output of audio to the speaker after the timer has been stopped, and its playback in that device. The software timer method also adds a factor to the system: the existence of the timer objects, and analysis of the volume of received RTP packets. Similarly, the oscilloscope method above also omits the signal from the mouse to the CWP, and the playback of audio in the output device. It also adds something: the activation of the serial port.

In fact, if the test adds something that takes extra time, it is less harmful than if it leaves anything out which could add to the delay. The reason being that a test that only adds factors in will overestimate the delay. If it does and still lies below the limit, it can be concluded that the system would be acceptable even without the added factors. For a test that omits something on the other hand, it can never be said with certainty that the requirement is actually met without certain knowledge of how long time the omitted part takes.

6.1.3 Differences between transmission and reception delay

In the prototype, the most serious failure of a delay requirement was the ground transmission voice delay, where all of the tests that were performed consistently resulted in a delay 15–33 ms too large. These measurements were made with oscilloscope. In the preamble of chapter 4 it was mentioned that network latencies such as propagation and queuing delay inside the network, and performance of the components and protocol stacks can all add to delays. However, not all those factors are equally significant in all cases. The prototype in this project was run inside a closed LAN connected with 100 Mb/s Ethernet cabling and a switch with a capacity of 100 Mb/s. The traffic during testing was much too low to throttle the network, and therefore network latencies had negligible bearing on the measurement results. The cause of delays in the prototype must instead be caused mainly by the performance of the hardware and software of the CWP and gateway, including sound cards, application algorithms, and protocol implementations.

It is interesting to note that transmission of voice from CWP to the radio side

took about 40 ms longer than reception of voice from radio side to CWP. It could be that the sound card of the CWP is less efficient in converting analogue audio from its audio input to digital sound than the other way around, or that the *Lumisoft.Net* framework takes longer to encode audio and packetize it into RTP than to unpack incoming RTP packets and decode their contents. The case could also be reversed in the gateway: it is slower to process incoming RTP audio and transmit it to the radio side. It could also be any combination of these cases.

As further work, the components should be tested in isolation to see if it is the CWP or the gateway which causes the bulk of the difference in latency between sending and receiving.

The sound card in the CWP was of standard issue for personal computer usage, and the audio was handled and encoded by the open source library *Lumisoft.Net*, which did not have a performance guarantee. The effect of using a professional sound card more adapted to the application and a more optimized software library could therefore also be investigated to attempt to lower the ground transmission voice delay.

6.2 Gateway evaluation

A number of requirements were not met due to limitations in the gateway. The gateway in this project was seen as a black-box component, and it was out of scope for the project to change its behaviour by editing its source code. A way that these discrepancies between gateway behaviour and system requirements could have been overcome to enable the missing functionality in this project would have been to introduce a new node into the system. This node would act as an application gateway, translating Eurocae-compliant traffic from the remainder of the system into traffic that the *IPG 3000* understands. However, this additional node would cause additional latency in the system, since all traffic must pass through it; each packet must be unpacked and inspected, processed and repackaged. Also, some amount of work would have to be put into implementing a whole new node and inserting it in the prototype. Instead, the decision for this prototype was to only attempt simpler workarounds. When furthering development towards a fully functional VCS, it is necessary to improve the software of the gateway, adding the missing functionality. The gateway should be developed as an integral part of the system, and have the chosen specifications as a basis.

Below, the shortcomings of the gateway with regards to the requirements are summarized.

Size of RTP payload The gateway does not allow RTP to be configured with packet payload sizes of 10 or 30 ms, as the requirement *AUDIO 4: Voice packetiz-*

6.2. GATEWAY EVALUATION

ation interval requirement dictates that all nodes should. RTP itself readily allows this to be configured, so the ability must only be added to the gateway.

RTP header extension The lack of support for the RTP header extension renders many requirements more difficult to satisfy. Instead, the gateway signals squelch by transmitting a second media stream, and it registers PTT simply on the existence of an RTP stream. This behaviour should be removed, the Eurocae RTP header extension should be implemented, and used instead to signal and register PTT and squelch.

Keep-alive messages The radio gateway does not send or recognize keep-alive messages, and it ignores the SDP parameters used for this functionality. The part of the gateway that handles received SDP messages must be improved to handle these parameters, and the gateway should transmit keep-alive messages according to the parameters when the session has been started.

Suppressing RTP in the absence of squelch *RTP 13: RTP radio aircraft call activation/deactivation* states that nodes must suppress the RTP stream when there is no PTT or squelch signal. The gateway does not conform to this, but instead starts transmitting RTP immediately when the SIP session is set up, regardless of whether squelch is active or not. On squelch, it then transmits a similar stream to an address manually typed into its web interface.

Instead of doing this, the streaming behaviour of the gateway should be that it sends to the address that called it using SIP, when and only when squelch for that channel is active.

Multiple SIP sessions per radio channel Only one SIP session per channel is allowed by the radio gateway. If another user calls a channel which is already in use, the gateway will send an error response indicating that it is busy and not allow the new user to listen to the channel. This makes *COMMUNICATION MODEL 4: Simultaneous communications between VCS and radios* impossible to fulfil, as it states that it should be possible to maintain seven SIP sessions for each radio channel. The number of possible SIP sessions must thus be extended from one to seven per channel, and audio mixing functionality and similar must be added so that the gateway can correctly handle simultaneous calls on the same channel. SIP already has conference call functionality, which might be used to implement this requirement.

6.2.1 Gateway summary

The functionality that was most critically missing from the gateway during the course of this project was lack of support for the Eurocae header extension, and the ability to allow multiple SIP sessions per radio channel, which caused several desired requirements not to be met. The size of RTP payload is the least critical omission as 20 ms is the usual payload size in most RTP applications. At any rate, further work must be put into developing a gateway that is fully compliant with the requirements for a VCS to allow a full-fledged system to use it without needing to resort to unreasonable workarounds.

6.3 Software development

This section discusses the development of the software, what issues that arose, and what conclusions can be drawn from them.

6.3.1 Protocol library

As mentioned in section 3.7.2.7 on page 28, it was necessary to make some changes to the open source protocol stack to suit the needs for the application. The library unfortunately had several more flaws apart from those mentioned, and left much to be desired in terms of structure and documentation. Many questions also remain about how well it is implemented concerning issues such as reliability, security, performance, and maintainability. Nevertheless, using it did help to shorten the development time compared to if protocol support had been developed from scratch, but for a real system, it would most likely be better in the long run to spend some effort to acquire a more complete and robust library.

6.3.2 Programming language

C# with Microsoft's .Net framework was used as the programming language during development of the CWP software. It was sufficient in fulfilling the needs required from the programming language for this type of system, and most likely, any of the major object-oriented languages would have sufficient functionality.

However, the choice of programming language could also affect the performance of the VCS. Lower level languages like C or C++ are commonly thought to be faster than C#. Comparing different languages was outside the scope for this project, but the performance effect of developing parts of the VCS in different languages could be investigated further in future work.

6.4 Testing environment

Because of the high level of safety and reliability requirements for an ATC system, any tests concerning this domain need to be validated that they are free from failure and malfunction, and produce reliable results. The regulations from authorities concerning the testing environment for aviation systems must be followed. Therefore, the test environment is an important part of such development, and it has to be as close to reality as possible. In this project, only rudimentary testing was performed, but in a real system, a stricter procedure and environment for testing must be developed.

6.5 Additional features

This section discusses the most interesting features and issues of VCSs concerning radio communication that were not implemented. More research could be done to provide new specifications for them, or to improve current ones.

6.5.1 Best signal selection and climax mode

Depending on the size and terrain of an air control sector, the strength and quality of the radio signals between aircraft and the control centre can vary. For this reason a sector can have one or more radio transmitters and receivers depending on the size and terrain of the sector, enabling the pilots and the control centre to communicate with each other regardless of whether there is a line of sight between them or not.

6.5.1.1 Best signal selection

When dealing with air to ground receptions among the receivers in a sector, they will have different signal strength depending on the terrain. Based only on its own signal strength, a receiver cannot decide if it is the one with the best signal. In order to get a good reception, the controller has to either switch between different receivers to find the one with best signal reception or listen to two receivers simultaneously. The controller would have to add the tasks of switching between the receivers and dealing with problems such as echo and fading that can be caused when listening to several receivers simultaneously. This is both time consuming and cumbersome. A solution is to automate the signal selection process so that a receiver can be chosen without the involvement of a controller. This can be done by having the receivers send their signal strength information to the system, so that the VCS can decide which signal to forward to the CWP.

6.5. ADDITIONAL FEATURES

Best signal selection is a mandatory requirement in a VCS and it is defined as “... the process by which a particular radio signal is selected 'as best' for presentation to the user and retransmission if cross-coupling is selected” (Eurocae WG-67, 2009a).

6.5.1.2 Multi-carrier mode

When dealing with ground to air transmissions, the pilot does not have the same possibility as a controller to switch between the receivers since he or she is already listening to the only receiver. However, the signal in the pilot's headset will be the result of the reception from several transmitters within the sector, which can lead to unwanted echo effects. This problem can be solved by using multi-carrier mode where the transmitter frequency is offset by different amounts. This will prevent the degradation in audio quality caused by receiving multiple transmissions on the same frequency, as the aircraft can choose to apply a particular offset, which will result in reception of only one of the transmissions.

6.5.2 Simultaneous transmissions detection

A situation with a large number of aircraft in the same sector increases the probability of simultaneous transmissions from two or more aircraft, on the same frequency. This can lead to a situation where the controller is not aware of other ongoing transmissions, which is a potential safety hazard. Eurocae WG-67 (2009c) presents several cases for how this can occur with recommendations for solving each case. However, these recommendations have not been implemented and deployed in an ATC environment. Since there is no data concerning the recommendations, no criterion for successful detection ratio has been defined yet. Thus, no specification is given by Eurocae WG-67 (2009c): “As a first step, a 'best effort' algorithm is awaited here from the radio and VCS manufacturers.” Future work should further investigate the problem of simultaneous transmissions detection and search for good solutions.

Glossary

ATC Air Traffic Control, pp. 1–4, 12, 58, 59

CWP Controller Working Position. The work station of an air traffic controller, within a VCS. It is also the name of a class in the software prototype, pp. 2, 4, 6, 7, 9, 10, 18, 20–23, 25–39, 42–50, 52–55, 57, 58, 63–67, 69, 70

EASA European Aviation Safety Agency, p. 3

Eurocae The European Organisation for Civil Aviation Equipment. An international organisation that publishes standards related to aviation. In this report, mention of the organisation usually refers to Eurocae WG-67 (2009*a,b,c*), pp. iv, 3, 6–9, 12, 18–21, 23, 27–29, 32, 34, 55–57, 62

GUI Graphical User Interface, pp. 6, 11, 22–24, 28, 46

H.323 A collection of protocols for multimedia communication over IP, defined by the ITU, pp. 11, 12

HTTP Hypertext Transfer Protocol, pp. 12, 13

IDE Integrated Development Environment, p. 8

IETF Internet Engineering Task Force, p. 12

IP Internet Protocol, pp. iv, 1–6, 8, 11–13, 17, 18, 22, 26, 28, 43, 45, 60, 64, 66

ITU International Telecommunication Union, pp. 12, 60

ITU-T ITU Telecommunication standardization sector, p. 62

LAN Local Area Network, pp. 4, 6, 28, 45, 54

NTP Network Time Protocol, p. 34

- PCM** Pulse Code Modulation, pp. 9, 42, 62, 63
- PSTN** Public Switched Telephone Network, p. 1, 2, 4
- PTP** Precision Time Protocol, p. 34
- PTT** Push-To-Talk. The action taken by the controller to switch to transmitting mode. Also taken to mean the signal that travels through the system in case of such an action, pp. 18–23, 27, 28, 30, 34, 37–39, 47–49, 56, 64, 65, 67–70, 81
- QoS** Quality of Service, pp. 45, 66
- RTCP** Real-time Transport Control Protocol, pp. vi, 12, 19, 20, 42, 63
- RTP** Real-time Transport Protocol, pp. v, vi, 3, 7–9, 11–13, 15–21, 23, 26–28, 33, 36, 39, 42–44, 47–49, 53–57, 62–65, 67–69
- SDP** Session Description Protocol, pp. 15, 42, 43, 56, 63, 64
- SIP** Session Initiation Protocol, pp. v, 3, 6–9, 11–23, 26–28, 30, 36, 42–45, 47, 49, 50, 56, 57, 62–64, 66, 68–70
- squelch** A mechanism for suppressing output when no audio is received by the radio, to prevent noise from being output to the user. In the context of this report, squelch is often used to denote the signal sent from the radio when its squelch circuit is activated, signifying that an aircraft call has been received, pp. vi, 18–23, 26–28, 34–39, 44, 46, 48, 56, 64–66, 68, 69, 81
- TCP** Transmission Control Protocol, p. 16
- UDP** User Datagram Protocol, pp. 13, 16, 17, 42, 63
- URI** Uniform Resource Identifier, pp. 13, 18, 43, 64
- VCS** Voice Communication System, pp. iv, 1–7, 11, 12, 17, 19, 23, 29, 34, 44, 45, 49, 55–59, 63–67, 69
- VoIP** Voice over Internet Protocol, pp. 4, 12, 29, 45, 63
- WAN** Wide Area Network, pp. 3, 4, 6

Bibliography

- Amato, G. (2008). European standardization and collaboration to a global concept. In *Integration and Harmonization of NextGen and SESAR into the Global ATM Framework*. ICAO. 8–10 September, 2008. Montreal, Canada. Presentation.
- Bardach, H., Kampichler, W., Prinz, J. and Göschka, K. M. (2003). IP communications in air traffic control (ATC). *e & i: Elektrotechnik und Informationstechnik*, vol. 120, nr. 11, pp. 405–407.
- Baset, S. A. and Schulzrinne, H. (2006). An analysis of the Skype peer-to-peer internet telephony protocol. In *IEEE INFOCOM, International Conference on Computer Communications*. pp. 1–11. 23–29 April, 2006. Barcelona, Spain.
- Basicevic, I., Popovic, M. and Kukolj, D. (2008). Comparison of SIP and H.323 protocols. In *International Conference on Digital Telecommunications*. pp. 162–167. 29 June–5 July, 2008. Bucharest, Romania.
- Darilion, K., Kampichler, W. and Göschka, K. M. (2003). Event-based radio communication signalling using the Session Initiation Protocol. In *IEEE International Conference on Networks*. pp. 489–494. 28 September–1 October, 2003. Sydney, Australia.
- Darilion, K., Kurth, C., Kampichler, W. and Göschka, K. M. (2004). A service environment for air traffic control based on SIP. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS-37)*. 5–8 January, 2004. Big Island, Hawaii, United States.
- Eier, D. and Kampichler, W. (2010). Eurocae WG-67 standards for voice-over-IP in ATM for advanced NEXTGEN conops. In *Integrated Communications Navigation and Surveillance Conference (ICNS)*. pp. C8–1 – C8–9. 11–13 May, 2010. Herndon, Virginia, United States.

BIBLIOGRAPHY

- Eurocae WG-67 (2009a). *Interoperability standards for VoIP ATM components*. ED-137. The European Organisation for Civil Aviation Equipment, Working Group 67.
- Eurocae WG-67 (2009b). *Network requirements and performances for voice over internet protocol (VoIP) air traffic management (ATM) systems*. ED-138. The European Organisation for Civil Aviation Equipment, Working Group 67.
- Eurocae WG-67 (2009c). *Voice over internet protocol (VoIP) air traffic management (ATM) system operational and technical requirements*. ED-136. The European Organisation for Civil Aviation Equipment, Working Group 67.
- Hafner, F. and Mahmoud, M. (2005). A simple voice communication system for human-in-the-loop air traffic control simulations. In *Summer Simulation Multiconference*. 24–28 July, 2005. Cherry Hill, New Jersey, United States.
- Haindl, B., Kampichler, W. and Prinz, J. (2009). VoIP in ATC communications—where are we today. In *IEEE/AIAA Digital Avionics Systems Conference (DASC)*. pp. 1.B.2–1 – 1.B.2–9. 23–29 October, 2009. Orlando, Florida, United States.
- Hillenbrand, M., Götze, J. and Müller, P. (2005). Voice over IP—considerations for a next generation architecture. In *EUROMICRO Conference on Software Engineering and Advanced Applications*. pp. 386–393. 30 August–3 September, 2005. Porto, Portugal.
- ITU-T (2010). Packet-based multimedia communications systems. In *Series H: Audiovisual and Multimedia Systems*. No. H.323. International Telecommunication Union.
- Johnston, A., Donovan, S., Sparks, R., Cunningham, C. and Summers, K. (2003). *Session Initiation Protocol (SIP) Basic Call Flow Examples*. RFC 3665 (Best Current Practice). The Internet Engineering Task Force.
- Kurose, J. F. and Ross, K. W. (2008). *Computer Networking: A Top-Down Approach*. 4th edition. Pearson Education, Inc. Boston, Massachusetts, United States.
- Kurth, C., Kampichler, W. and Göschka, K. M. (2005). Frequency cross-coupling using the Session Initiation Protocol. In *Networking - ICN 2005*. Vol. 3420 of *Lecture Notes in Computer Science*. pp. 680–689. Springer Berlin / Heidelberg.
- Liu, H. and Mouchtaris, P. (2000). Voice over IP signaling: H.323 and beyond. *Communications Magazine, IEEE*, vol. 38, nr. 10, pp. 142–148.

BIBLIOGRAPHY

- Miller, C. A. and Greenspan, R. L. (2008). Air-traffic control. In *AccessScience*. McGraw-Hill Companies.
- Perkins, C. (2003). *RTP: Audio and Video for the Internet*. Pearson Education, Inc. Boston, Massachusetts, United States.
- Pointurier, Y. (2002). *Link failure recovery for MPLS networks with multicasting*. Master's thesis. School of Engineering and Applied Science, University of Virginia. Charlottesville, Virginia, United States.
- Postel, J. (1980). *User Datagram Protocol*. RFC 768 (Standard). The Internet Engineering Task Force.
<http://www.ietf.org/rfc/rfc768.txt>
- Postel, J. (1981). *Transmission Control Protocol*. RFC 793 (Standard). The Internet Engineering Task Force.
<http://www.ietf.org/rfc/rfc793.txt>
- Roach, A. B. (2002). *Session Initiation Protocol (SIP)-Specific Event Notification*. RFC 3265 (Proposed Standard). The Internet Engineering Task Force.
- Rosenberg, J. and Schulzrinne, H. (2002). *An Offer/Answer Model with Session Description Protocol (SDP)*. RFC 3264 (Proposed Standard). The Internet Engineering Task Force.
- Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and Schooler, E. (2002). *SIP: Session Initiation Protocol*. RFC 3261 (Proposed Standard). The Internet Engineering Task Force.
- Saab Danmark A/S (2009). *IPG 3000 Technical Manual*.
- Schulzrinne, H., Casner, S., Frederick, R. and Jacobson, V. (2003). *RTP: A Transport Protocol for Real-Time Applications*. RFC 3550 (Standard). The Internet Engineering Task Force.

Appendix A

Selected requirements

This appendix lists all the requirements from the Eurocae documents (Eurocae WG-67, 2009*a,b,c*) that were chosen for consideration in this project. They are only briefly, informally explained to allow understanding of this report. For complete details, the reader must refer to the original documents. The requirements will be presented in order of what milestone from chapter 2 they relate to, and they have in this project each been assigned a number between 1 and 3 denoting the priority for their implementation, where 3 is the highest priority and 1 is the lowest. The priorities were chosen with regards to the goals and delimitations of the project. They are listed with the name they have in the Eurocae documents, to make it easier to find them in the original source. For more information about the delay requirements, see also chapter 4.

A.1 Requirements for milestone 1

AUDIO 4: Voice packetization interval requirement Priority: 3

All nodes in the system must support RTP packets with a payload of 10, 20 or 30 ms of audio. This packet size should be negotiated during the set-up of each particular SIP session. The default shall be 20 ms.

AUDIO 6: Voice coding requirement Priority: 3

This requirement defines what different audio codecs should be supported by the system. The allowed encodings are PCM A-law or μ -law as specified by the ITU Telecommunication standardization sector (ITU-T) in the specification G.711. Also, voice compression is allowed according to the standards G.728 or G.729 from the ITU-T. The requirement also specifies what RTP payload type numbers shall be associated with each codec.

A.1. REQUIREMENTS FOR MILESTONE 1

COMMUNICATION MODEL 3: Applicable protocol Priority: 3

The protocols to use for VoIP communication between controllers and the radio shall be SIP and RTP.

RTP 2: RTP header specifications Priority: 3

Specifies how the RTP header is used, and what values are allowed in its fields. In particular, specifies that the bit flag signifying use of an extended header shall be set.

RTP 11: RTP and RTCP UDP port number Priority: 3

RTP data must use an even port number, and the RTCP packets are required to use the next higher port number.

RTP 19: Keep-alive messages Priority: 3

Keep-alive messages are used to control the connection between CWP's and radios. Using them, one node can let the other know that the connection is working in both directions. If keep-alive messages cease or indicate that the connection has been broken in one of the directions, the node can time out the session and attempt to reconnect.

This requirement details how the RTP packet used as a keep-alive message should look. Its header fields are set to certain values to distinguish them from normal RTP packets carrying audio. They should carry the header extension, but no payload. The requirement also specifies how parameters should be negotiated during SIP session set-up that decide how often keep-alive packets should be sent, and how a host should time-out a session.

SIP 1: SIP version Priority: 3

SIP version 2 shall be used in the VCS.

SIP 4: SIP message body (SDP) Priority: 3

Specifies what different fields are allowed in the SDP message body that accompany SIP messages, and what values are allowed for those fields. In particular, only the media type "audio" is allowed, and the media protocol should read "RTP/AVP". Four different values are allowed in the "rtpmap" attribute, which designates what RTP payload types are allowed in the session. The default payload type is to be PCM A-law, which should have the payload type number 0.

A.2. REQUIREMENTS FOR MILESTONE 2

SIP 2: SIP supported requests Priority: 2

Many different SIP requests shall be supported by the different SIP entities in the system. This requirement specifies which they are, which must have an SDP body and which must not, and which may not be sent under certain circumstances. Of interest is the specification of an event package to be used with the SIP specific event notification (Roach, 2002). It consists of simple formatting rules for NOTIFY events used to inform nodes about what user is transmitting on what radio frequency.

SIP 3: SIP supported responses Priority: 2

Specifies which SIP responses shall be supported and how they shall be used.

SIP 5: SIP address format Priority: 2

SIP addresses should be formatted in a certain manner within the VCS. A SIP URI for a radio channel, for instance, should contain the frequency it concerns and an identification for the radio site.

A.2 Requirements for milestone 2

SIP 8: SIP call set up procedure Priority: 3

Gives an example of how a SIP call should be set up. The controller sends a SIP request to the radio. If the recipient's IP address is known, the request can be sent to the IP address directly. If not, a request with a SIP URI can be sent to a SIP proxy server, which will resolve the URI and forward or redirect the request. Upon receiving the request, the radio responds with a SIP message containing information about what parameters such as audio codecs the radio can accept.

The radio shall be called using SIP every time the radio channel is selected. Upon reception of an OK response from the radio, RTP streams according to the decided parameters are established, with the header extension being used to signal information such as squelch and PTT. It shall also be possible for both CWP and radio to terminate an ongoing call. The radio should only terminate the call if it receives another call with higher priority, however.

It must never be the radio that initiates a SIP call, and the radio is recommended not to accept calls from unknown hosts.

COMMUNICATION MODEL 5: Communication initiation between VCS and radios Priority: 3

The connection is to be initiated from the CWP to the radio. SIP and SDP shall be used to negotiate what parameters that should be used. Then, RTP is used

A.2. REQUIREMENTS FOR MILESTONE 2

to transport audio data, and the header extension is used for squelch, PTT and other signals. Keep-alive messages are also sent to determine that the connection is functioning correctly.

RTP 14: RTP header extension description Priority: 3

This requirement defines how the extended RTP header is to be used in a VCS to send signalling information. An extended RTP header in line with Schulzrinne et al. (2003) always has a field for type and one for length of extension. Next comes the extension itself, in the number of 32 bit words specified in the length field. For the header extension specified in this requirement, the type field contains the hexadecimal value 67₁₆, and the extension should have a length of 1.

RTP 15: RTPTx information field Priority: 3

Here, the RTP header extension to be used for packets being sent towards a radio is further specified. Notable is the PTT field, which allows five different values for different types of PTT signals: “PTT off”, “normal PTT”, “coupling PTT”, “priority PTT” and “emergency PTT”. It is specified how the radio should react to different signals of different priority. The bit reserved to indicate squelch is always set to zero in RTP packets travelling towards a radio.

RTP 16: RTPRx information field Priority: 3

In this requirement, the header extension used for packets travelling from a radio is specified. It is largely the same as the RTPTx information field, but the squelch flag is used here. Also, information about the radio signal quality should be present in the header extension.

RADIO FUNCTIONAL 10: 300 ms max frequency key activation response time Priority: 2

When the controller changes the state of a channel, it should take effect within 300 ms. For instance, if a channel is selected in receive only mode and audio is being received on that channel, the audio should be output at the CWP within the time limit. If the channel is selected in send-and-receive mode, it should be possible to transmit within the same time limit.

RADIO PERFORMANCE 3: 100 ms max transmitter activation delay Priority: 2

Transmitter activation delay refers to the latency between a PTT action at the CWP and the radio being enabled for transmission. The time between these events should be at most 100 ms.

A.3. REQUIREMENTS FOR MILESTONE 3

RADIO PERFORMANCE 4: 100 ms aircraft call indication delay Priority: 2

The time between a signal being received at the radio and the indication device (visual or otherwise) for squelch being activated can be at most 100 ms.

RADIO PERFORMANCE 6: 130 ms max ground transmission voice delay Priority: 2

The latency of the audio from the input device of the CWP to the audio being transmitted by the radio is at most 130 ms.

RADIO PERFORMANCE 8: 130 ms max ground reception voice delay Priority: 2

The time between audio being received by the radio and the audio being presented at an output device at the CWP can be at most 130 ms.

SAFETY 5: No QoS degradation of IP-voice Priority: 2

The QoS of the voice shall not be affected by other services within the VCS.

SAFETY 1: Call type discrimination during call establishment phase Priority: 1

The type and priority of a call shall be indicated with the priority header field in the SIP messages when the session is set up. Six different types of calls are defined.

SYS ENG 16: Detection of end-to-end connection loss Priority: 1

The VCS is to automatically check that it is possible to reach all radio sites so that a broken connection can be detected. A broken link must be detected in three seconds or less, and the system management facility shall be notified.

SYS ENG 35: Speech activity detector with pseudo squelch signal generation Priority: 1

In the accidental case that an audio signal is received without a squelch, a device shall register this and make sure that the audio reaches the CWP regardless.

A.3 Requirements for milestone 3

RADIO FUNCTIONAL 1: Radio access modes of operation Priority: 3

Four different modes are possible for a radio channel: “off”, “receive only”, “transmit and receive” or “cross-coupled”. Changing between modes are done by

A.3. REQUIREMENTS FOR MILESTONE 3

the controller with some input device such as a button or screen icon, and a visual and/or audible indication of a mode change must be presented to the controller.

RADIO FUNCTIONAL 14: Simultaneous multiple frequency selections for transmission and reception Priority: 3

It must be possible for a controller to choose up to 28 radio channels to send to and receive from. In case of a received signal on several channels at the same time, the audio from them is to be combined and presented to the controller.

RADIO FUNCTIONAL 15: Cross-coupling facility to be implemented Priority: 3

This requirement describes how cross-coupling should function. A cross-coupled radio channel can be in one of two modes: simplex or duplex mode. Channels in different modes can be mixed in a cross-coupled group. The difference between the modes is that signals received by a channel in simplex mode will not be retransmitted on the other channels in the group. If the channel is duplex, the signal will be retransmitted.

If signals are received on two channels in the same cross-coupled group, only the signal first received will be presented to the controller, while the other will be suppressed.

RADIO FUNCTIONAL 16: Quantity of frequencies in a cross-coupled frequency group Priority: 3

The VCS should allow cross-coupled groups with up to 28 channels.

RADIO FUNCTIONAL 17: Two cross-coupled frequency mode Priority: 3

When two channels are cross-coupled, reception on one channel shall be retransmitted on the other. If the controller signals PTT and transmits audio, the audio shall be sent to both channels.

RADIO FUNCTIONAL 20: Identical frequencies not allowed in two cross-coupling sessions Priority: 3

It is not allowed to have the same radio channel in two different cross-coupled groups.

RTP 6: RTP PTT transmission performance Priority: 3

This requirement is essentially a repeat of previous requirements. It states that a PTT signal is sent when a controller activates some PTT device at a CWP. No voice packets shall be sent when PTT is not activated. The PTT signal shall be transmitted by using a field in the RTP header extension.

A.3. REQUIREMENTS FOR MILESTONE 3

RTP 7: RTP aircraft call transmission performance Priority: 3

Similar to the above requirement, this one summarizes other requirements related to squelch. Squelch is sent when a radio receives a radio signal from an aircraft. The squelch is sent using the RTP header extension. Voice packets shall not be sent when there is no squelch, and the aircraft call indication delay must be no more than 100 ms.

SIP 6: Basic call functionalities Specifies that a radio is to be called using SIP, and that the priority header field shall be used to determine if one call should be suppressed by another with higher priority. The requirement then lists a number of features associated with telephony that must not be allowed.

Priority: 3

RADIO FUNCTIONAL 22: PTT precedence over cross-coupling transmission Priority: 2

When channels are cross-coupled, retransmission of received signals take place. However, a PTT originating from the controller should be prioritized before such retransmissions.

RADIO FUNCTIONAL 24: First received aircraft call used as a cross-coupled group's incoming frequency Priority: 2

Only the audio signal first received shall be presented to the controller and retransmitted. If there are multiple simultaneous receptions at several channels, and it cannot be determined which was first, one of them must be designated as incoming frequency.

RADIO FUNCTIONAL 33: PTT and aircraft call locked-on condition prevention Priority: 2

This requirement specifies that it must not be possible in the system for an erroneous, indefinite PTT or squelch signal to occur, which would block all other communication on the concerned channel.

RADIO FUNCTIONAL 35: Duplex and simplex cross-coupling functionality support Priority: 2

This requirement only repeats how duplex and simplex cross-coupling mode shall function, which has already been specified in other requirements.

RTP 12: RTP radio PTT activation/deactivation Priority: 2

A.4. REQUIREMENTS FOR MILESTONE 4

Here, the signalling regarding PTT activation and deactivation is clarified in greater detail. When PTT is deactivated, it is recommended that three redundant RTP packets, signalling this in their extended header, are transmitted in rapid sequence to ensure that the transmission is deactivated correctly.

RTP 13: RTP radio aircraft call activation/deactivation Priority: 2

Similarly to the above requirement, the deactivation of squelch should be transmitted in redundant packets.

SYS ENG 31: Quantity of frequencies in a cross-coupled group Priority: 2

The system architecture must allow up to 28 channels in a cross-coupled group. However, this number might be configured lower using the system management facility.

SYS ENG 32: No limit on number of cross-coupled groups Priority: 2

The number of cross-coupled groups in a CWP or the entire VCS shall not be limited.

RADIO FUNCTIONAL 18: Three cross-coupled frequency mode Priority: 1

Describes how cross-coupling shall be implemented for cross-coupled groups with three radio channels. Like described in previous requirements about cross-coupling, a received signal on one channel shall be retransmitted on the two others.

A.4 Requirements for milestone 4

COMMUNICATION MODEL 4: Simultaneous communications between VCS and radios Priority: 3

It must be possible for several CWPs to be connected to the same radio channel. Specifically, a radio channel shall be able to support seven different SIP and RTP sessions simultaneously.

SYS ENG 12: No architectural limit on active CWPs Priority: 3

The number of active CWPs in the system shall not be limited by the architecture.

A.4. REQUIREMENTS FOR MILESTONE 4

RADIO FUNCTIONAL 2: Transmit configuration description Priority: 2

Each controller is assigned one out of three priority levels. If several controllers take PTT actions for the same radio channel, the one with highest priority shall gain access to the radio. If several controllers with the same priority level try to send on the same channel, their transmission shall be mixed together and all the audio shall be sent on the channel.

RADIO FUNCTIONAL 12: Facility isolation / disconnection warning Priority: 1

If a radio channel or a CWP is unexpectedly disconnected from the system, an alarm shall alert the controllers.

RADIO FUNCTIONAL 40: PTT identity notification Priority: 1

When a controller takes a PTT action on a radio channel, other controllers having that channel activated shall receive information about who initiated a PTT.

RADIO PERFORMANCE 5: 250 ms max cross-coupled PTT inhibition period, XC2 Priority: 1

After a transmission from a CWP onto a cross-coupled group has been sent, there shall be a short period of time when no retransmission of signals in the group can occur. This period is specified by a parameter named XC2, and it can be at most 250 ms.

SIP 7: SIP audible tones control Priority: 1

Usually, a SIP application gives the user tone indications reflecting if the call is for instance, busy, on hold, or ringing. However, for radio applications, audible tones shall not be used, since the radio should answer immediately on a SIP request and telephone features such as ringing or call on hold are not possible.

SYS ENG 13: Simultaneous use of any system functionality combination Priority: 1

Any functionality in the system can be used at the same time across any number of CWPs and radio channels.

SYS ENG 39: Alarm for non-selected frequencies Priority: 1

In case a certain radio channel is not selected by any CWP, an alarm shall be set off.

Appendix B

Delay measurement figures

This appendix contains images related to the measurement of delays. The first section presents diagrams of the delay specifications. The next section shows pictures of how the oscilloscope measurements were set up, and examples of the readings.

B.1 Figures: Delay specification

B.1.1 Transmitter activation delay

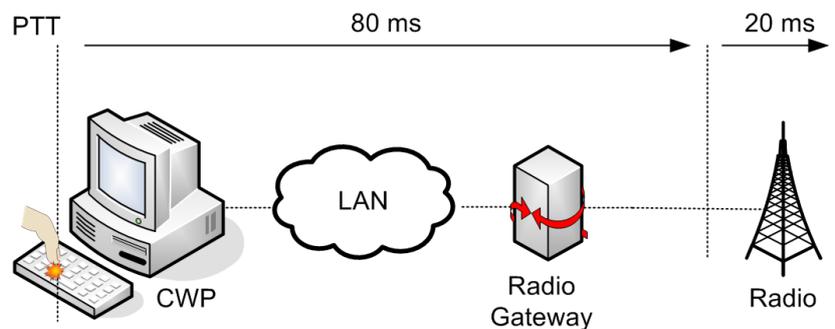


Figure B.1: Specification for transmitter activation delay

B.1.2 Aircraft call indication delay

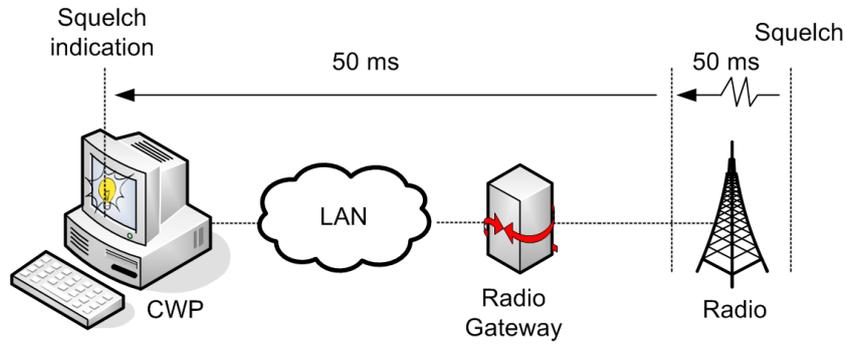


Figure B.2: Specification for aircraft call indication delay

B.1.3 Ground transmission voice delay

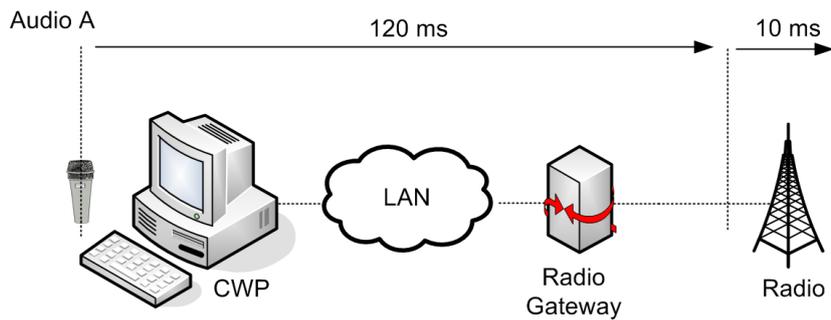


Figure B.3: Specification for ground transmission voice delay

B.1.4 Transmitter activation and aircraft call indication loop-back delay

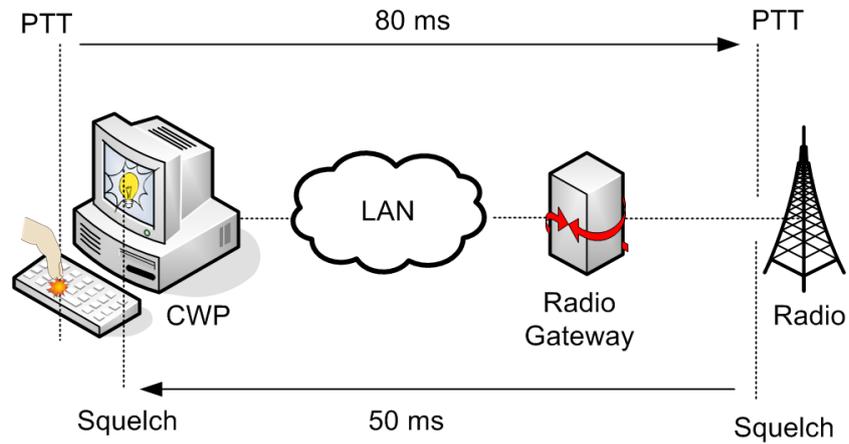


Figure B.4: Specification for transmitter activation and aircraft call indication loopback delay

B.1.5 Ground reception voice delay

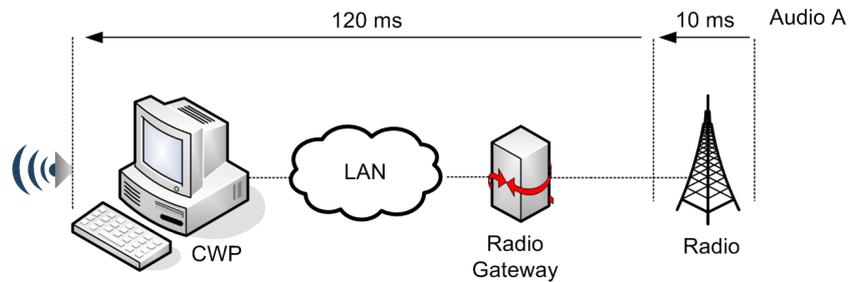


Figure B.5: Specification for ground reception voice delay

B.1.6 Ground transmission and reception voice loopback delay

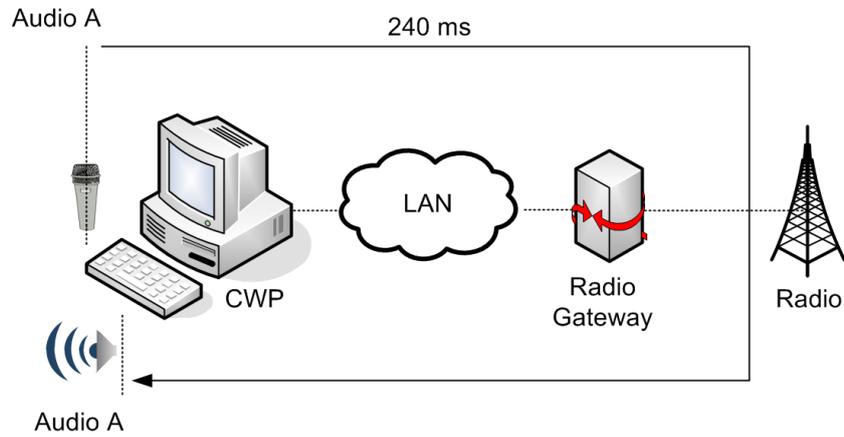


Figure B.6: Specification for ground transmission and reception voice loopback delay.

B.2 Figures: Oscilloscope set-ups and readings

B.2.1 Frequency key activation response time

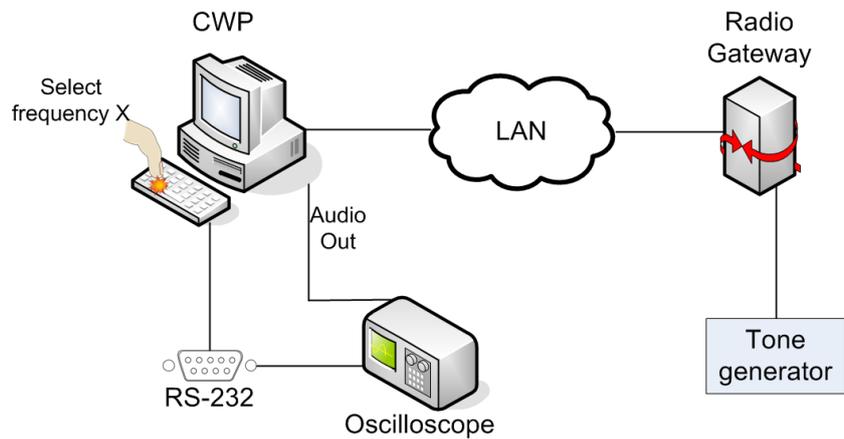


Figure B.7: Measurement set-up for frequency key activation response time, a) receive mode.

B.2. FIGURES: OSCILLOSCOPE SET-UPS AND READINGS

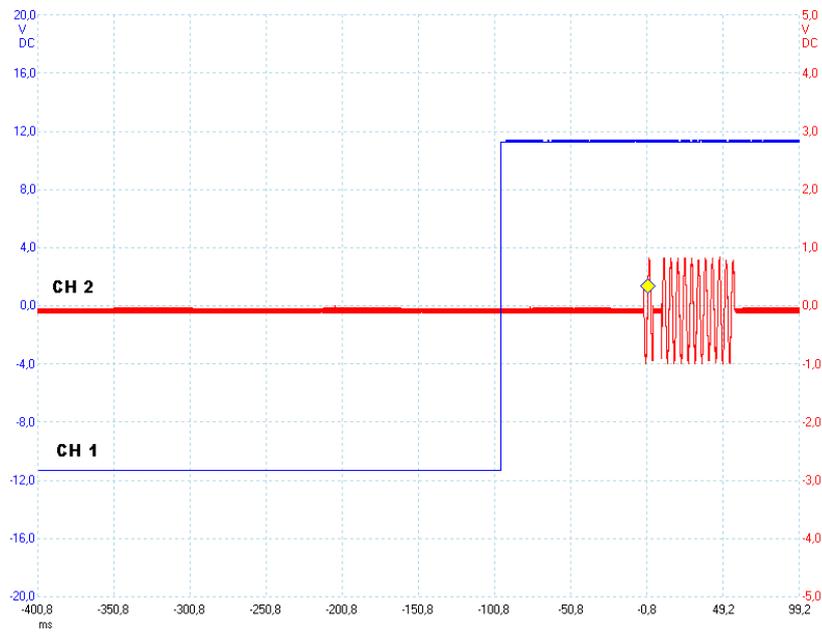


Figure B.8: Oscilloscope measurement of frequency key activation response time a)

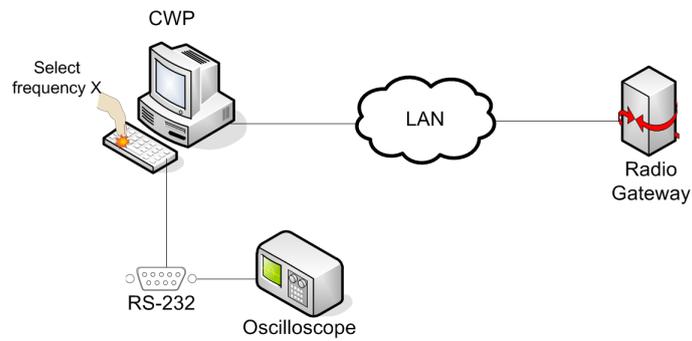


Figure B.9: Measurement set-up for frequency key activation response time, b) transmit mode.

B.2. FIGURES: OSCILLOSCOPE SET-UPS AND READINGS



Figure B.10: Oscilloscope measurement of frequency key activation response time b)

B.2.2 Transmitter activation delay

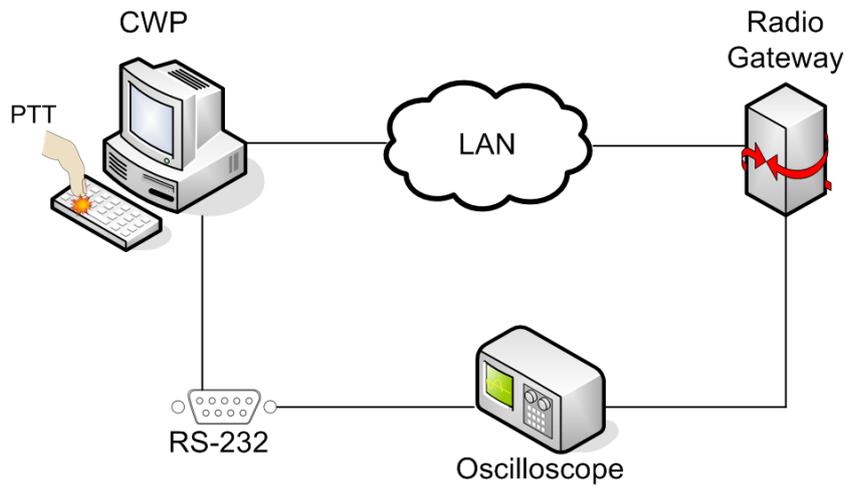


Figure B.11: Measurement set-up for transmitter activation delay.

B.2. FIGURES: OSCILLOSCOPE SET-UPS AND READINGS

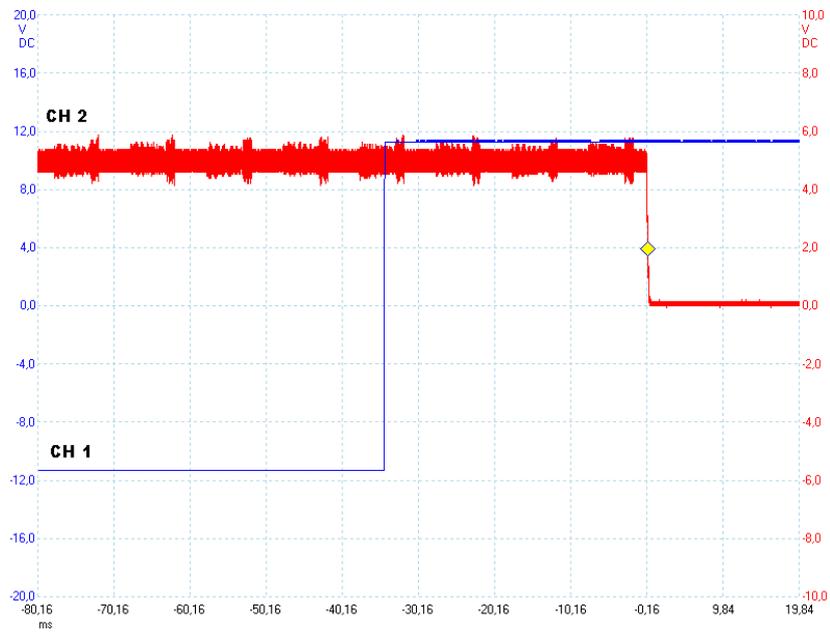


Figure B.12: Oscilloscope measurement of transmitter activation delay

B.2.3 Aircraft call indication delay

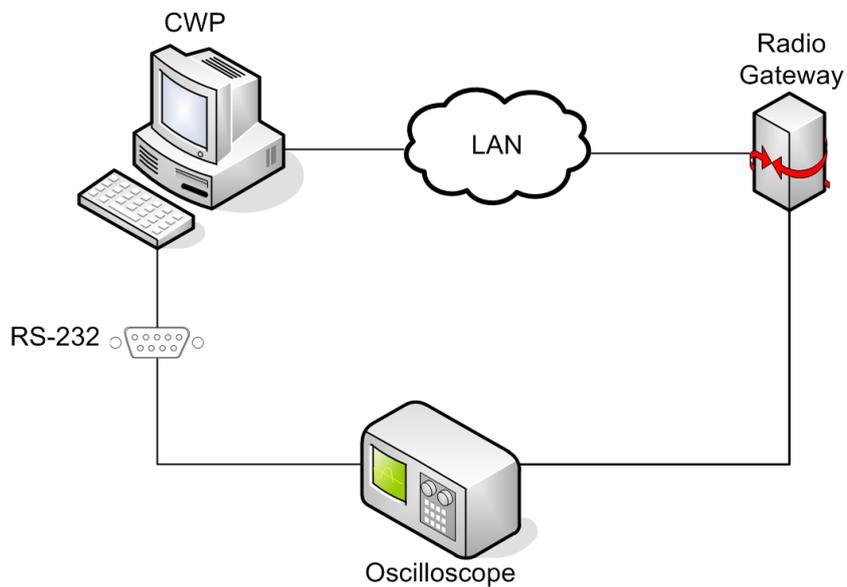


Figure B.13: Measurement set-up for aircraft call indication delay.

B.2. FIGURES: OSCILLOSCOPE SET-UPS AND READINGS

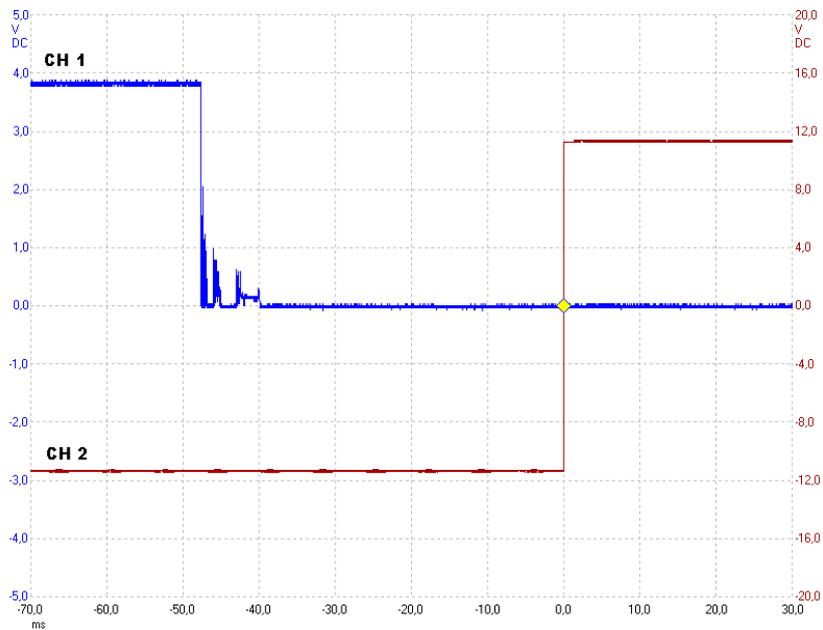


Figure B.14: One measurement of aircraft call indication delay using an oscilloscope. Here, the delay was around 47 ms.

B.2.4 Ground transmission voice delay

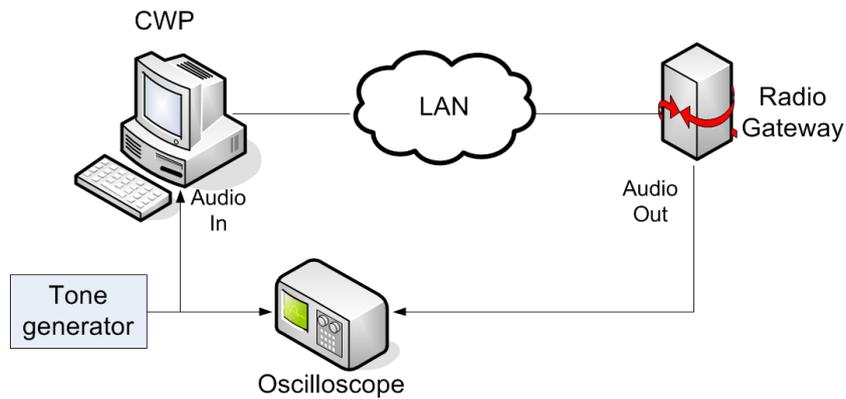


Figure B.15: Measurement set-up for ground transmission voice delay, for component details see chapter 4.3.1

B.2. FIGURES: OSCILLOSCOPE SET-UPS AND READINGS

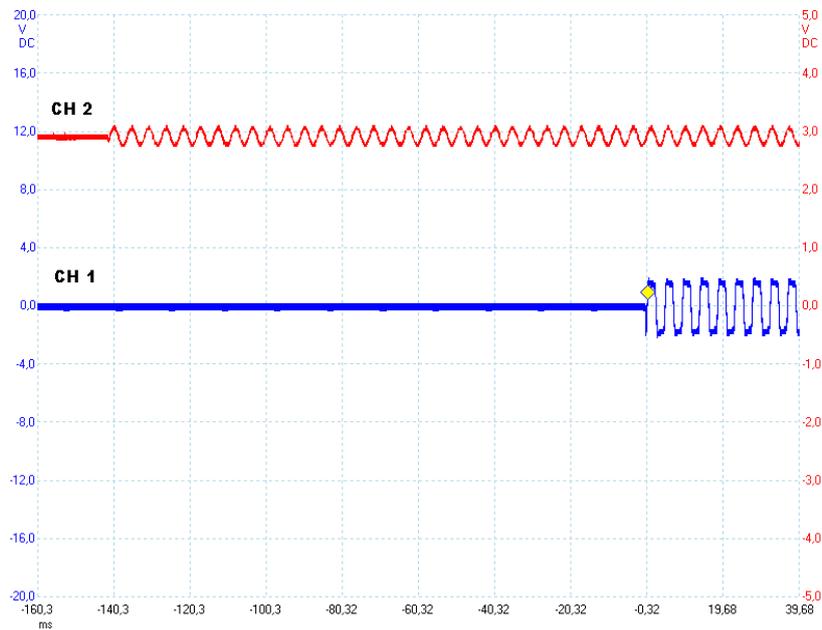


Figure B.16: Oscilloscope measurement of Ground transmission voice delay

B.2.5 Ground reception voice delay

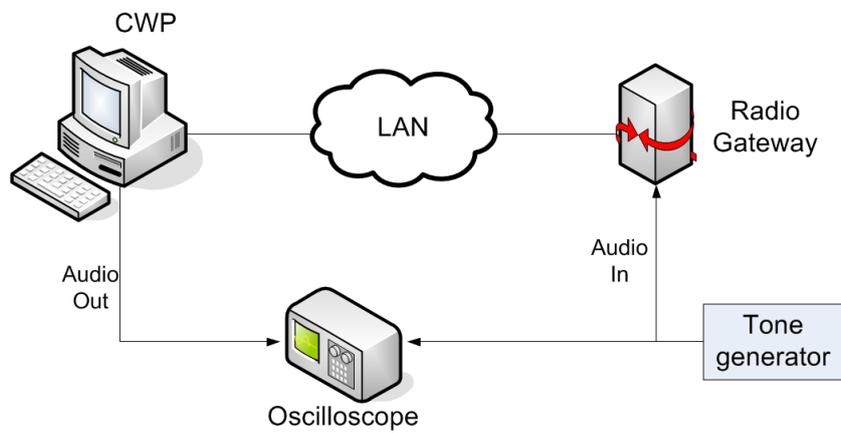


Figure B.17: Measurement set-up for ground reception voice delay, for component details see chapter 4.3.1

B.2. FIGURES: OSCILLOSCOPE SET-UPS AND READINGS

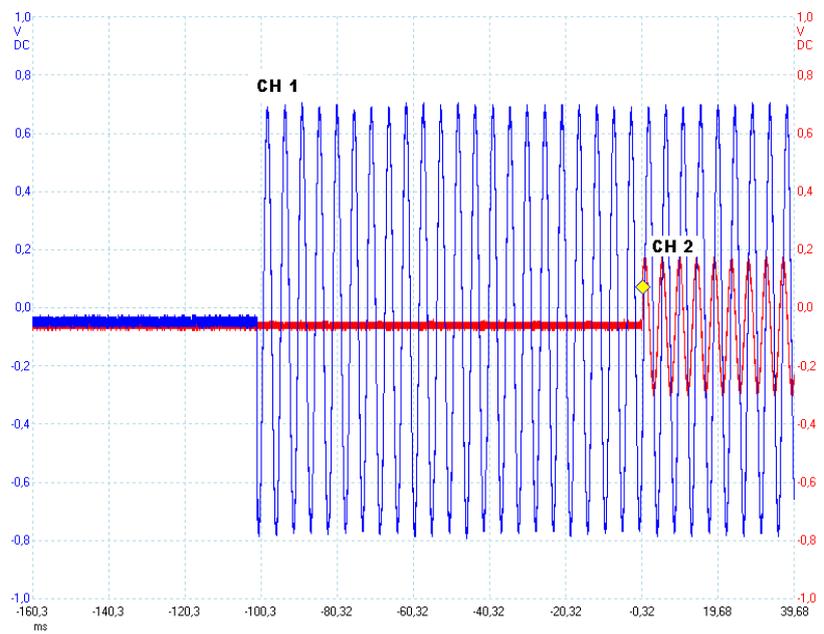


Figure B.18: Oscilloscope measurement of ground reception voice delay

B.2.6 Transmitter activation and aircraft call indication loop-back delay

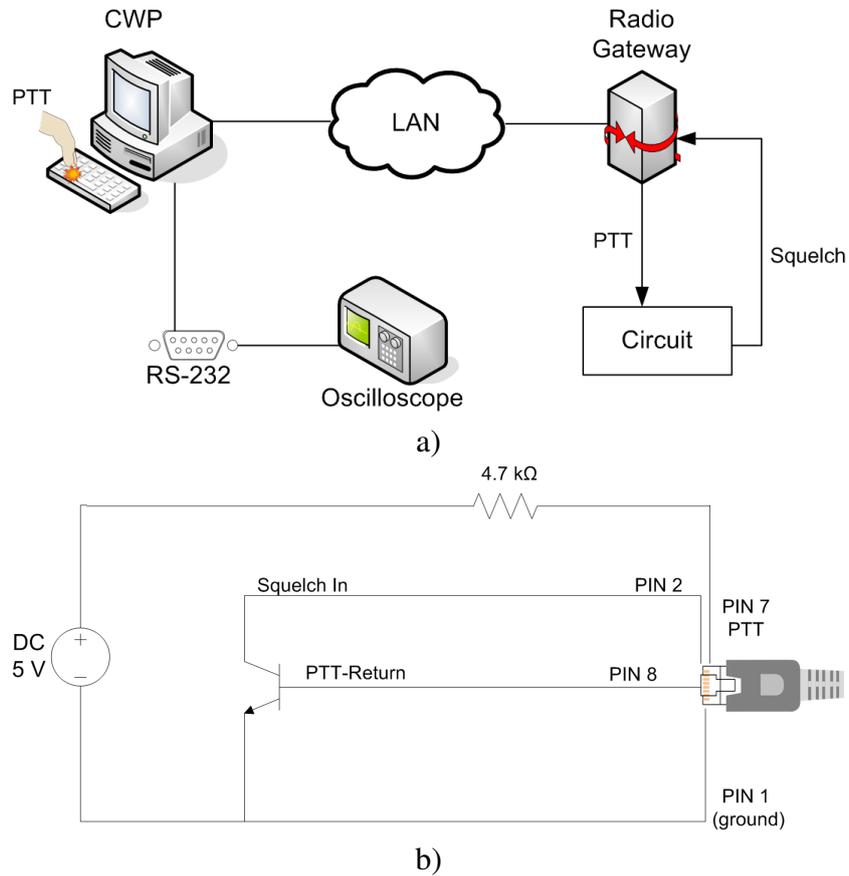


Figure B.19: a) Measurement set-up of transmitter activation and aircraft call indication loop-back delay. b) Circuit schematic for activating squelch with the PTT signal.

B.2. FIGURES: OSCILLOSCOPE SET-UPS AND READINGS

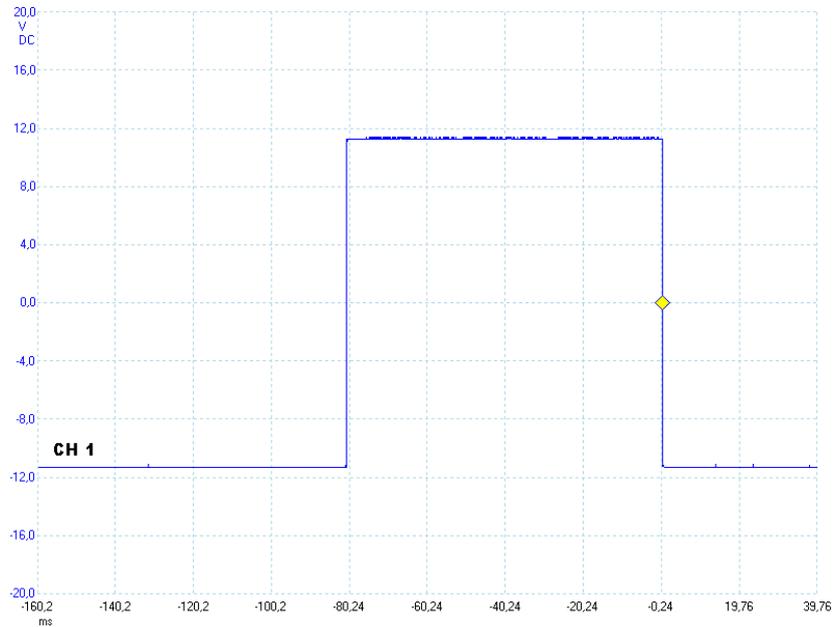


Figure B.20: Oscilloscope measurement of transmitter activation and aircraft call indication loopback delay

B.2.7 Ground transmission and reception voice loopback delay

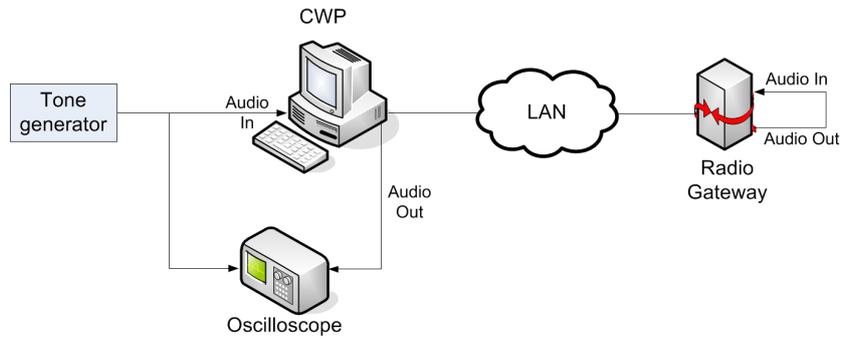


Figure B.21: Measurement set-up for ground transmission and reception voice loopback delay.

B.2. FIGURES: OSCILLOSCOPE SET-UPS AND READINGS

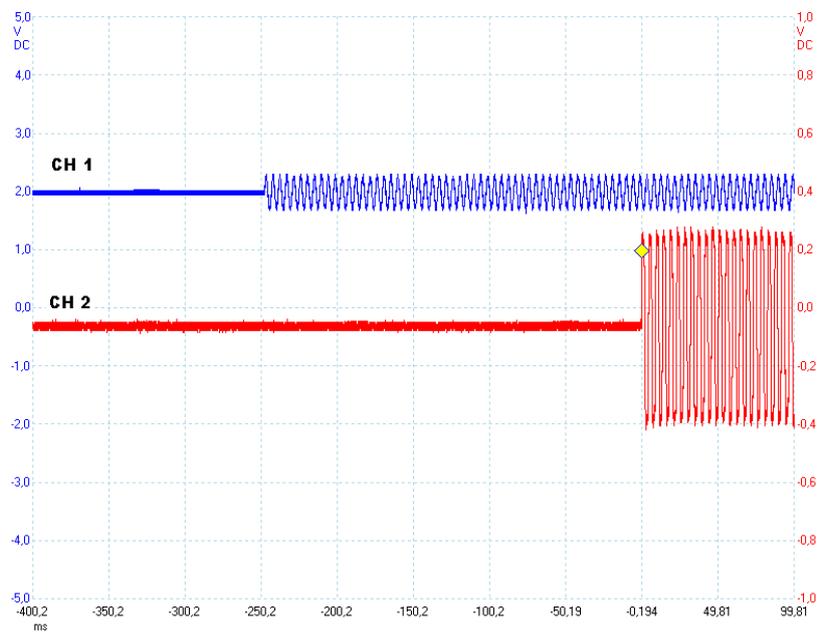


Figure B.22: Oscilloscope measurement of transmission and reception voice loop-back delay.

Appendix C

Delay result tables

C.1 Frequency key activation response time

C.1.1 Part a)

	Oscilloscope (ms)	Software timer (ms)
1.	120	267
2.	284	145
3.	105	144
4.	107	128
5.	353	141
6.	94	125
7.	250	200
8.	97	136
9.	244	124
10.	265	136
Average	192	155

Table C.1: Measurement results for frequency key activation response time, part a) with a maximum delay requirement of 300 ms.

C.2. TRANSMITTER ACTIVATION DELAY

C.1.2 Part b)

	Oscilloscope (ms)	Software timer (ms)
1.	119	122
2.	132	112
3.	114	209
4.	140	111
5.	111	124
6.	113	197
7.	114	113
8.	143	139
9.	111	112
10.	188	113
Average	129	136

Table C.2: Measurement results for frequency key activation response time, part b) with a maximum delay requirement of 300 ms.

C.2 Transmitter activation delay

	Oscilloscope (ms)	Software timer (ms)
1.	40	n/a
2.	44	n/a
3.	39	n/a
4.	55	n/a
5.	55	n/a
6.	43	n/a
7.	43	n/a
8.	46	n/a
9.	51	n/a
10.	35	n/a
Average	46	n/a

Table C.3: Measurement results for transmitter activation delay, with a maximum delay requirement of 80 ms.

C.3 Aircraft call indication delay

	Oscilloscope (ms)	Software timer (ms)
1.	37	n/a
2.	25	n/a
3.	23	n/a
4.	27	n/a
5.	33	n/a
6.	33	n/a
7.	37	n/a
8.	26	n/a
9.	26	n/a
10.	26	n/a
Average	30	n/a

Table C.4: Measurement results for aircraft call indication delay, with a maximum delay requirement of 50 ms.

C.4 Transmitter activation and aircraft call indication loopback delay

	Oscilloscope (ms)	Software timer (ms)
1.	74	75
2.	81	89
3.	25	89
4.	83	73
5.	74	76
6.	73	79
7.	77	83
8.	42	79
9.	80	82
10.	80	91
Average	69	82

Table C.5: Measurement results for transmitter activation and aircraft call indication loopback delay, with the sum of the two requirements being 80 + 50 ms.

C.5 Ground transmission voice delay

	Oscilloscope (ms)	Software timer (ms)
1.	135	n/a
2.	142	n/a
3.	142	n/a
4.	153	n/a
5.	137	n/a
6.	148	n/a
7.	149	n/a
8.	145	n/a
9.	138	n/a
10.	148	n/a
Average	144	n/a

Table C.6: Measurement results for ground transmission voice delay, with a maximum delay requirement of 120 ms.

C.6 Ground reception voice delay

	Oscilloscope (ms)	Software timer (ms)
1.	97	n/a
2.	102	n/a
3.	103	n/a
4.	101	n/a
5.	105	n/a
6.	104	n/a
7.	106	n/a
8.	107	n/a
9.	106	n/a
10.	108	n/a
Average	104	n/a

Table C.7: Measurement results for ground reception voice delay, with a maximum delay requirement of 120 ms.

C.7. GROUND TRANSMISSION AND RECEPTION VOICE LOOPBACK DELAY

C.7 Ground transmission and reception voice loop-back delay

	Oscilloscope (ms)	Software timer (ms)
1.	231	105
2.	248	143
3.	241	100
4.	251	82
5.	249	95
6.	237	132
7.	249	117
8.	252	126
9.	250	125
10.	242	121
Average	245	115

Table C.8: Measurement results for ground transmission and reception voice loop-back delay, with the sum of the two requirements being 120 + 120 ms.