A Comparative Study of In-Band and Out-of-Band VoIP Protocols in Layer 3 and Layer 2.5 Environments

Georgios Pallis

Submitted in partial fulfilment of the requirements of
Napier University for the degree of
Master of Science in Advanced Networking

Sponsored by inAccess Networks SA and QoSient LLC

School of Computing

EDINBURGH NAPIER UNIVERSITY

August 2010

Authorship Declaration

I, Georgios Pallis, confirm that this dissertation and the work presented in it are my own achievement.

Where I have consulted the published work of others this is always clearly attributed.

Where I have quoted from the work of others the source is always given. With the exception of such quotations this dissertation is entirely my own work.

I have acknowledged all main sources of help.

If my research follows on from previous work or is part of a larger collaborative research project I have made clear exactly what was done by others and what I have contributed myself.

I have read and understand the penalties associated with Academic Misconduct.

I also confirm that I have obtained **informed consent** from all people I have involved in the work in this dissertation following the School's ethical guidelines.

Signed:

Date: 23 / 08 / 2010

Matric No: 07001710

Data Protection Declaration

Data Protection Declaration Under the 1998 Data Protection Act, The University cannot disclose your grade to an unauthorised person. However, other students benefit from studying dissertations that have their grades attached.

Please sign your name below *one* of the options below to state your preference.

The University may make this dissertation, with indicative grade, available to others.

The University may make this dissertation available to others, but the grade may not be disclosed.

The University may not make this dissertation available to others.

To my parents for their support. To Ioanna and Katerina.
To all the engineers and scientists who share their knowledge with the
rest of us in order to make this world a better place to live.

Abstract

For more than a century the classic circuit-switched telephony in the form of PSTN (Public Service Telephone Network) has dominated the world of phone communications (Varshney et al., 2002). The alternative solution of VoIP (Voice over Internet Protocol) or Internet telephony has increased dramatically its share over the years though. Originally started among computer enthusiasts, nowadays it has become a huge research area in both the academic community as well as the industry (Karapantazis and Pavlidou, 2009). Therefore, many VoIP technologies have emerged in order to offer telephony services. However, the performance of these VoIP technologies is a key issue for the sound quality that the end-users receive. When making reference to sound quality PSTN still stands as the benchmark.

Against this background, the aim of this project is to evaluate different VoIP signalling protocols in terms of their key performance metrics and the impact of security and packet transport mechanisms on them. In order to reach this aim in-band and out-of-band VoIP signalling protocols are reviewed along with the existing security techniques which protect phone calls and network protocols that relay voice over packet-switched systems. In addition, the various methods and tools that are used in order to carry out performance measurements are examined together with the open source Asterisk VoIP platform. The findings of the literature review are then used in order to design and implement a novel experimental framework which is employed for the evaluation of the in-band and out-of-band VoIP signalling protocols in respect to their key performance metrics.

The framework is composed of open standard methods and tools so as to provide flexibility and applicability to real-life networks. The major issue of this framework though is the lack of fine-grained clock synchronisation which is required in order to achieve ultra precise measurements. However, valid results are still extracted. These results show that in-band signalling protocols are highly optimised for VoIP telephony and outperform out-of-band signalling protocols in certain key areas. Furthermore, the use of VoIP specific security mechanisms introduces just a minor overhead whereas the use of Layer 2.5 protocols against the Layer 3 routing protocols does not improve the performance of the VoIP signalling protocols.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to extend thanks to Prof. Bill Buchanan not only for his supervision of this project, but mostly for his overall support all these 3 years that I am part of the Edinburgh Napier University family. I would also like to thank Mr. Robert Ludwiniak for his guidance and for being part of the marking process. Furthermore, I would like to thank the following people: First of all Dr. Michael Manousos of inAccess Networks SA and Mr. Carter Bullard of QoSient LLC for their free technical support regarding the Asterisk IP PBX and the Argus flow monitoring software suite respectively. Second, my good friend and flatmate Dr. Hayes Mawindi Mabweazara from the School of Arts and Creative Industries of our university. His academic experience was invaluable. At this point I would like to congratulate him for his recent doctorship. Third, Mr. Phil Zimmermann for providing me the ZRTP patch for Asterisk. Numerous anonymous open source and VoIP enthusiasts had also their part in this project. Their help in the VoIP related IRC channels and forums is highly appreciated. Finally, I would like to mention two very special persons: RMS and Linus. This project would never happen without their contribution in the IT world. Last but not least I would like to thank my favourite hard rock and heavy metal bands. Their music kept me alive during the formation of this MSc thesis.

CHAPTER 1
Introduction

1.1 Project overview

As it is stated by Abbasi et al. (2005) “In recent years, there have been strong efforts to develop Voice over IP (VoIP) protocols that can operate over the current Internet infrastructure and perform gracefully with large scale global deployment.” This shows the need for fast VoIP protocols which can offer toll quality telephony over packet-switched networks.

Against this backdrop, this project aims to evaluate different VoIP signalling protocols in terms of their key performance metrics and the impact of security and packet transport mechanisms on them by using a novel evaluation framework. In order to provide valid results the suggested framework is based on open standards and tools.

1.2 Background

Over the years the Internet has thrived as one of the main communication means by introducing a variety of new applications and services. One of the most important is VoIP which emerges as a significant alternative to conventional circuit switched telephone networks (Karapantazis and Pavlidou, 2009). There are plenty of reasons for this, but cost is the most significant one. With IP telephony it is possible to avoid huge charges when making long distance domestic or international calls. This is very important especially for transitional or undeveloped countries where there is a lack of competition among major service providers (Robert et al., 2008). VoIP can also offer common features like voicemail, call waiting or call forwarding for free. In addition to these, new services can be introduced offering extended possibilities to the end users and more revenues to the phone operators. Furthermore, as a result of its computer mediated nature, VoIP enables users to make phone calls by using a number of devices (Ranjbar, 2007). Either special VoIP phones can be used or software that is installed in mobile equipment (laptops, personal digital assistants, tablet PCs, and so on) adding to flexibility, mobility and productivity, especially when combined with applications like instant messaging or e-mail. Scalability is also a strong asset of VoIP networks (Chong and Matthews, 2004). Updating their capacity is easy and more cost-effective, while circuit switched networks are composed of expensive components that are difficult to maintain and upgrade.

In this light, there is no question why VoIP usage has increased through the years. Already by the end of 2002 in China VoIP telephone traffic had surpassed traditional telephone traffic in both domestic and international calls (Wang and Hu, 2004). Furthermore, Internet giants like Google have started to provide VoIP services (Park, 2010). However, the traditional PSTN (Public Switched Telephone Network) still dominates voice communications. Despite its costly services and complex maintenance, for more than a century PSTN continues to serve as the main network for relaying voice. Regarding sound quality and availability it stands as the benchmark for telephony. Indeed, these are the key issues for VoIP (Bross and Meinel, 2008). Due to their nature, VoIP systems are extremely sensitive to delay, jitter (variation of delay) and packet loss. This is a big problem in the modern converged networks that carry both voice and data. Furthermore, VoIP systems can be affected by electrical power failures, thus reducing their uptimes, while PSTN succeeds 99,999% availability (Varshney et al., 2002). Finally, since VoIP technology depends on packet networks, security attacks, like eavesdropping or DoS (Denial-of-Service), have to be considered as well (Materna, 2006). However, the available security mechanisms can affect the performance of VoIP networks and degrade voice quality. A summary of the advantages and constraints of IP telephony compared to the traditional PSTN is given in the following table.

Indicator	**PSTN**	**VoIP**
Switching	Circuit switched	Packet switched
Bandwidth	Fixed	Variable and easily adjustable
Scalability & features	Reprogramming or changes in the network design are required	Easily added without major changes
Flexibility	Very low	High, easy to combine with other services
Cost	Expensive equipment and maintenance	Cheap to deploy and maintain
Service availability	Extremely high (99,999% uptime)	Variable, depending on the packet switched network availability
Quality of Service	Extremely high	Variable, depending on delay, jitter and packet loss.
Latency	Extremely low	Variable, depending on the network congestion
Security	Generally high level offered	Eavesdropping and DoS is easier to carry out

Table 1.1: *Comparison of PSTN and VoIP telephony.*

Generally, VoIP is a revolutionary technology that introduces flexible and affordable voice communications. However, it is a new technology with a number of limitations which have to be closely examined before being deployed.

1.3 Aim and objectives

The aim of this project is to evaluate different VoIP signalling protocols in terms of their key performance metrics and the impact of security and packet transport mechanisms on them by using a novel evaluation framework. In order to meet this aim the following objectives are defined:

1. Critically review the main VoIP protocols and their security mechanisms, the different network protocols which are used in order to transport voice and the main methods for performing traffic measurements.
2. Design a flexible and applicable to real-life networks novel evaluation framework based on the findings of the literature review in order to evaluate VoIP protocols and the security and packet transport mechanisms that can affect their performance by utilising open standard methods and tools.
3. Implement the framework by setting up the selected methods and tools and evaluate the VoIP protocols based on the results produced by carrying out a number of experiments as set in the design by using this framework.

1.4 Thesis Structure

This thesis is organised in seven chapters as follows:

Chapter 1 **Introduction.** This chapter provides the overview of the project and the background to the subject of VoIP. It also presents the aim of the project and the thesis structure.

Chapter 2 **Theory.** This chapter presents the factors that affect the performance of IP telephony as well as the main VoIP protocols..

Chapter 3 **Literature review.** This chapter provides a critical analysis of VoIP protocols and their security mechanisms along with a review of the packet transport methods that are used to relay voice and the work in the field of traffic measurements. A well-known open source VoIP platform is also examined.

Chapter 4 **Design.** This chapter describes the design of an experimental framework for the evaluation of VoIP protocols based on the findings of the literature review. The choices made for this design are justified here.

Chapter 5 **Implementation.** This chapter presents the details of the implementation of the evaluation framework. Snippets of the configuration of the various framework components are given in order to provide a better understanding of the whole system.

Chapter 6 **Evaluation.** This chapter presents and analyses the results of the experiments that were carried out using the designed framework.

Chapter 7 **Conclusions.** This chapter provides a summary of the entire project. Furthermore, recommendations for further research in the field of VoIP are discussed.

CHAPTER 2

Theory

2.1 Introduction

This chapter provides some basic background information in respect to VoIP. In particular, the key VoIP performance metrics and QoS (Quality of Service) requirements for VoIP telephony are introduced according to the studies of Goode (2002) and Karapantazis and Pavlidou (2009). Determining these metrics is significant for the evaluation of VoIP technologies. Furthermore, the main VoIP signalling and transport protocols are presented in short. Finally, this chapter introduces the notion of in-band and out-of-band VoIP signalling protocols.

2.2 QoS requirements of voice

As it was discussed in Chapter 1 when making reference to sound quality PSTN stands as the benchmark. In order to achieve the same level of quality, VoIP systems have to meet extremely strict QoS requirements. The next sections analyse the key indicators that affect the performance of VoIP systems.

2.2.1 End-to-end delay

End-to-end delay, also known as one-way delay or mouth-to-ear delay, is the total time interval required to deliver the sound to the receiver from the moment the sender starts speaking. Figure 2.1 (Salah, 2006) illustrates the end-to-end VoIP components and the delay they introduce at each stage of a typical VoIP phone call:

(i) *Encoding delay:* The first step is to convert the analog voice signal into a digital signal. This is handled by the *encoder* which samples the original voice in a constant rate and assigns a value to each sample, creating a fixed bitstream. This rate depends on the algorithm (codec) used, but typically with PCM (Pulse Code Modulation) 8-bit samples are generated every 0,125ms (resulting in a data rate of 64kbps). The whole process introduces the encoding delay.

(ii) *Packetisation delay:* The *packetizer* is the next component in the VoIP call path. It encapsulates a fixed number of voice samples into packets, adding the appropriate protocol headers. The time for that is the packetisation delay.

(iii) *Network delay:* Finally, the digitised voice is ready to be send to the receiver through the packet switched network. The total time interval needed to transmit and propagate the voice packets from one end to the other leads into the network delay.

(iv) *Playback delay:* At the receiver's end, a critical component is the *playback buffer* which tries to smooth the different delays when receiving successive packets. The time interval required for that represents the playback delay.

(v) *Decoding delay:* Following the playback buffer, the initial voice samples must be de-encapsulated from the packets by passing through the *depacketizer*. Eventually, the *decoder* turns the digital speech samples back into analog sound. This whole process of reconstructing the original voice signal introduces the decoding delay.

Generally, a delay of up to 200 ms is considered to be acceptable, but a maximum of 100 ms has to be achieved, if higher quality is desired.

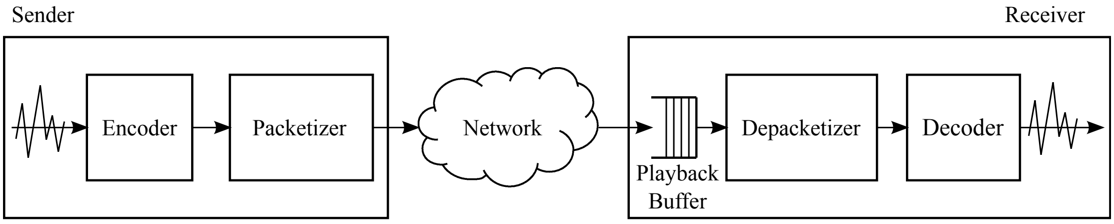


Figure 2.1: *VoIP end-to-end components (Salah, 2006).*

2.2.2 Jitter

One factor that can dramatically degrade the quality of a VoIP call is jitter. Jitter is defined as the variation of delay and it is the result of congestion that might occur in the network over time. While the sender transmits the voice packets at a steady rate, it cannot be ensured that they are received in an equally constant rate, exactly because of possible network congestion. The playback buffer tries to smooth these delay variations, as it was mentioned earlier. Jitter should not be more than 30 ms, but values up to 75 ms are tolerable.

2.2.3 Packet loss

Packet loss is another key metric for VoIP quality that expresses the number of packets which fail to be delivered at the receiver's end. Like jitter, it can happen because of network congestion along the data path. Other contributing factors can be low reception in wireless networks, high end-to-end delay or a full playback buffer. It is recommended that packet loss should be as low as 1% or less.

2.2.4 Bandwidth utilisation

A significant metric is also the required bandwidth for a single call. It is affected by the encoder and the algorithm (codec) used for converting the analog voice signal into digital. The overhead caused by the protocol headers added by the packetizer or the possible security mechanisms is also an important factor. Typically, the more bandwidth is required by the encoder, higher the quality is. On the other hand, since bandwidth is not infinite, its utilisation must be kept as low as possible, offering at the same time the best achievable voice quality.

2.3 Signalling and transport protocols

Apart from the end-to-end VoIP components, that were examined earlier, there are also various other elements needed in order to deploy VoIP. Essentially, a *gatekeeper* or *call-manager* must be installed. Its function is to find and reserve the appropriate resources for call routing, establishment and maintenance. Another node in the call path can be a *gateway* that handles the interconnection with the PSTN, other external networks or old analog phone devices. The above are impossible without the use of another set of components which is constituted by various communication protocols.

In general, communication protocols define the mechanisms in a call session that the endpoints must agree upon to achieve connection. Within the VoIP context, there are two types of communication protocols: Signalling and transport. The signalling protocols have to perform a number of tasks related to the call session, while the transport protocols carry the voice packets between the participants. However, there are signalling protocols that apart from the call session relevant functions, can relay the voice payload without depending on the transport protocols. Therefore, *out-of-band* and *in-band* signalling protocols are defined, respectively. In any case, according to Ganguly and Bhatnagar (2008) they have to perform the following tasks:

1. **Callee location:** The current location of the called party must be discovered by contacting the appropriate entity.
2. **Availability determination:** It must be determined if the callee is available and if not, whether the call must be redirected to voicemail, forwarded to another user or simply dropped.
3. **Session parameter negotiation:** In order to set up a call, several parameters have to be negotiated like codec type, encryption, etc.
4. **Session modification:** During a call, the participating parties can change the initial parameters agreed, when better bandwidth utilisation or improved quality must be realised.
5. **Session termination:** At the end of the call, every involved node has to be informed, so that it will free all the reserved resources.

Following, the main in-band and out-of-band signalling and transport protocols are presented.

2.3.1 H.323

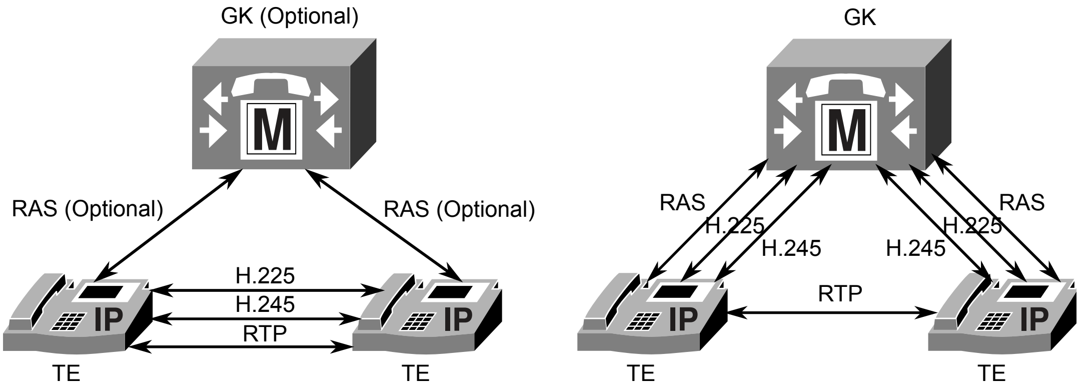
H.323 historically is the first protocol standardised for real-time multimedia communications. It was ratified by ITU (International Telecommunication Union) and it is an out-of-band signalling protocol. An H.323 network is composed by several *zones*. Each zone consists of the following units (Karapantazis and Pavlidou, 2009):

1. **Terminal:** The client's endpoint in H.323 is referred as a terminal or TE.
2. **Gatekeeper:** A gatekeeper or GK is a key component in the H.323 architecture. Apart from the expected tasks of call routing, establishment and maintenance, it is responsible for many other functions. Typically, it has to perform call admission, bandwidth control and zone management.
3. **Gateway:** A gateway or GW in an H.323 network offers the standard functionality of interconnection with the PSTN.
4. **Multipoint Control Unit:** A multipoint control unit or MCU provides conference facilities for three or more terminals or gateways.

H.323 is actually a protocol suite that defines several other protocols. Each one of them has a specific role in the whole communication setup process (Soares et al., 2008):

1. **H.225/RAS (Registration Admission Status):** This protocol is used for registration of the terminals in the gatekeeper, call admission, bandwidth allocation and exchange of status messages. It is also used for communication between gatekeepers across multiple zones for address resolution purposes.
2. **H.225:** This is the signalling component of ITU's VoIP protocol suite, therefore it carries out all the operations for setting up, maintain and tear down a call session. Actually, it is a subset of Q.931 standard for ISDN (Integrated Services Digital Network).
3. **H.245:** This is the control protocol. Control data is necessary in order to exchange information regarding the capabilities of each terminal, negotiate master-slave relationships and handle the operation of logical channels.
4. **H.450:** Supplementary services, such as call transfer, call forwarding or call hold, are offered by H.450. Furthermore, it provides decentralised or distributed control when this is required.
5. **T.120:** This is a protocol suite that implements services for multimedia conferencing.

Based on the above, two call models are supported by H.323: Direct calls and routed calls through the gatekeeper (Liu and Mouchtaris, 2000). This is shown in Figure 3.1.



(a) Direct call model

(b) Gatekeeper-routed call model

Figure 2.2: *H.323 call models.*

2.3.2 SIP

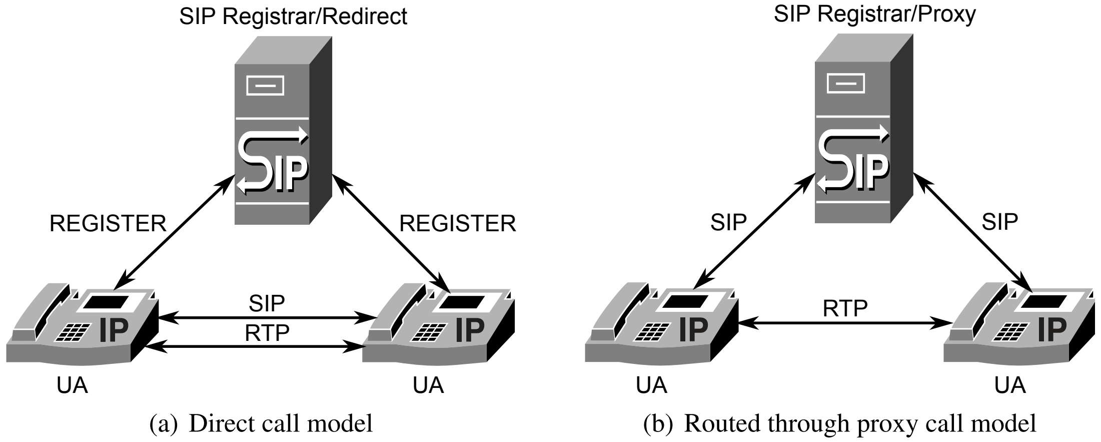
SIP (Session Initiation Protocol) was introduced by IETF (Internet Engineering Task Force) as an alternative to ITU's H.323 and it is an out-of-band signalling protocol too. SIP follows the classic client-server model. The following components are defined in a SIP network (Goode, 2002):

1. **User Agent:** A user agent or UA is the endpoint that makes and receives calls. It can function either as a user agent client (UAC) or as user agent server (UAS). An endpoint can have both of these attributes, but only one per session is active, depending on whether it initiated or responded to a SIP request, respectively.

2. **Network Server:** Four different types of network servers are specified in SIP. These are:

    - *Registrar server:* This is where a UA authenticates and registers its location and contact list.
    - *Proxy server:* Requests by UAs are received by proxies and forwarded appropriately. In essence, they are intermediate entities which provide routing functions, but they can also enforce policies regarding admission control or security. However, a proxy server is not always necessary, since direct calls are possible.
    - *Redirect server:* In order for a caller to reach directly the callee, it may need to consult a redirect server (for example when the callee has changed its location). Redirect servers do not accept or forward requests.
    - *Location server:* User details and updates of its current location are stored in this type of SIP servers. They also forward this information to proxies and redirect servers.

The above functionalities can be integrated in just one entity. This is shown in Figure 3.2 which illustrates the direct and routed call models supported by SIP.



(a) Direct call model (b) Routed through proxy call model

Figure 2.3: *SIP call models.*

2.3.3 IAX

IAX (Inter-Asterisk eXchange) was initially developed for the Asterisk IP PBX (Private Branch Exchange) platform, but adopted also by other soft switches. The initial version was soon deprecated by IAX version 2 or IAX2 (still commonly referred as IAX). IAX is an in-band signalling protocol. In an IAX network all the involved entities, such as the endpoints and servers, can relay signalling messages or the voice payload. Therefore, they are referred as peers. This is demonstrated in Figure 3.3.

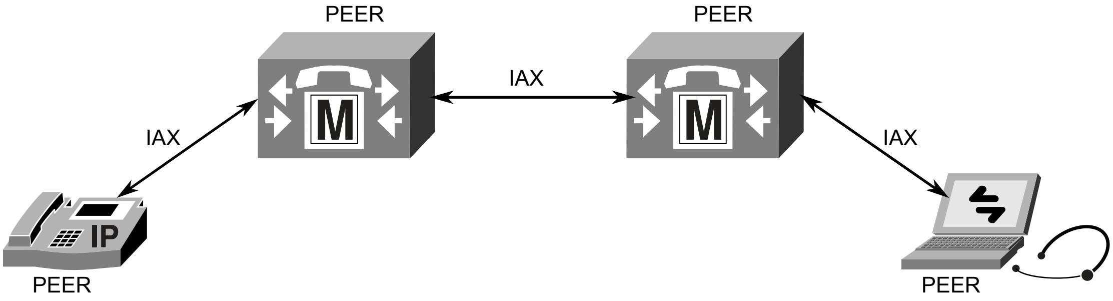


Figure 2.4: *IAX call model.*

2.3.4 RTP

RTP (Real-time Transport Protocol) is a transport protocol designed to carry data with real-time characteristics, such as voice. As a transport layer it can use either TCP (Transmission Control Protocol) or UDP (User Datagram Protocol). However, the extra reliability and retransmission of lost packets offered by TCP, is meaningless in voice communications. Hence, for VoIP calls RTP data packets are relayed over UDP for fast delivery. Furthermore, RTP complements UDP's unreliability by providing time-stamping, sequence-numbering and reordering of the packets. RTP over UDP is used by out-of-band signalling protocols, like H.323 and SIP, for transmitting voice.

2.4 Conclusions

This chapter provided some basic background information in respect to VoIP. In particular, the key VoIP performance metrics and QoS (Quality of Service) requirements for VoIP telephony were determined. These are relevant to end-to-end delay, jitter, packet loss and bandwidth utilisation. Defining these metrics is significant in order to achieve the aim of the evaluation of the VoIP signalling protocols. Apart from the above, the main in-band and out-of-band VoIP signalling protocols were presented in short. The next chapter will further discuss their architecture.

CHAPTER 3

Literature Review

3.1 Introduction

The previous chapter provided some basic information regarding VoIP technology and introduced the notion of in-band and out-of-band VoIP signalling protocols. It was discussed that in order to offer good quality packet telephony end-to-end delay, jitter and packet loss must be as low as possible. Bandwidth utilisation is another issue that must not be disregarded. Apart from the VoIP protocols themselves, other factors that have a great impact on these metrics are security and the network infrastructure that relays the voice packets. It can be seen that the VoIP engineer has a variety of issues to examine when designing a VoIP solution (Goode, 2002). Selecting a signalling protocol that will cover the needs for quality real-time voice communications is just one of the things to perpend. Securing the calls is also important, but this must happen without degrading the sound quality. The available network infrastructure should also be considered. But taking all of these decisions for VoIP deployment is one thing. Monitoring the deployed system and measuring its performance is also critical. At any point its good operation must be ensured and any problems that may occur must be detected in a fast and effective way. Therefore, monitoring systems have a vital role in the VoIP ecosystem.

In this context, the current chapter critically examines the features and architectures of in-band and out-of-band VoIP signalling protocols. Security solutions and their impact on VoIP are also discussed. Moreover, various voice transport techniques are assessed. Additionally, the latest methods and developments around network performance measurements are evaluated. Finally, a well-known IP PBX platform is reviewed for its viability in offering VoIP services. The findings of this literature review were used in order to design a novel experimental framework for achieving the aim of evaluating different VoIP protocols and the various factors that can have an impact on their performance as it was set in Chapter 1.

3.2 Comparison of VoIP signalling protocols

Since the dominant VoIP signalling protocols are H.323 and SIP, the majority of the work is focused on the differences between these two (Karapantazis and Pavlidou, 2009). One of the first studies was made by (Schulzrinne and Rosenberg, 1998). The authors compared the aforementioned protocols in relation to complexity, extensibility, scalability and features. They argued that SIP can offer similar services like H.323, but in a less complicated way, while in the meantime is more extensible and more scalable. (Dalgic and Fang, 1999) agrees with the above and also comments about the difficulty in implementing and debugging H.323. This is due not only because of its generally complexity, but also because of the ASN.1 binary format of the messages exchanged by H.323 components during a call. On the other hand the HTTP type of messages used by SIP are easier to implement and debug. Once again the complexity of H.323 is also supported by Basicevic et al. (2008). A number of different call scenarios was examined and it was shown that depending the case up to 28 H.323 messages need to be exchanged for call establishment, while SIP may need only up to 12. Finally, an extensive study on the architecture of the two protocols was carried out by Glasmann et al. (2003) where they pointed out the differences in the philosophy followed for offering VoIP services. H.323 provides solutions for things like QoS, it is strict regarding backwards compatibility with previous versions and specifies supplementary services in more detail. Especially the last one is critical for better interoperability with the traditional circuit-switched PSTN. Generally, H.323 is oriented more towards real-time multimedia communications like telephony, which explains its rigorous specifications, while SIP is a more generic protocol for setting up and tearing down sessions of any kind.

With respect to IAX as a newer VoIP signalling protocol it has not yet compared to H.323 and/or SIP in terms of architecture (Karapantazis and Pavlidou, 2009). Its main advantage stems from its in-band nature. Transferring the signalling messages and voice payload through a single UDP port, allows it to easily traverse through firewall and NAT (Network Address Translation) devices. On the contrary, out-of-band protocols like SIP and H.323 cannot easily tackle firewall and NAT traversal issues unless VoIP aware security devices, proxies or protocols like STUN (Simple Traversal of UDP through NAT) are used, but in the cost of increased expense and/or perplexity. (Goode, 2002; Yeryomin et al., 2008). Apart from being more flexible with security appliances, IAX is also more bandwidth efficient. Numerous streams between the same pair of peers can be multiplexed in a single trunk channel (Karapantazis and Pavlidou, 2009; Boucadair, 2009). For this to be possible, messages are switched in a binary format that the protocol defines as frames. Unfortunately though, IAX has not gained any great popularity between VoIP vendors and service providers. SIP and H.323 have a broader spectrum of offered services, while IAX is strictly designed for carrying voice and video.

In terms of comparing the performance of these protocols few studies have been realised. De et al. (2003) compared SIP and H.323 and found that the former is more capable in establishing a greater percentage of call sessions under heavy traffic. Abbasi et al. (2005) measured the sound quality that SIP can achieve compared IAX. The results showed that IAX can offer better sound quality. This was attributed to the small size of the IAX mini frames which are used to convey voice payload. However, since the experiments were carried in the Ottawa MAN (Metropolitan Area Network) the network conditions were controlled by using the NISTnet emulator. Lastly, Montoro and Casilari (2009) performed some experiments in order to measure the bandwidth utilisation of SIP and IAX. As already suggested by Karapantazis and Pavlidou (2009) and Boucadair (2009), they proved that IAX uses less bandwidth when its trunking (multiplexing) mode is used. Otherwise, bandwidth consumption is almost the same. Processor and memory utilisation was also similar for both of the protocols, but independent from the number of concurrent calls when it comes to the latter. They did not comment though on the end-to-end delay, jitter and packet loss metrics.

3.3 Security options for VoIP

It was mentioned earlier that VoIP is truly considered as another application over packet networks. Therefore, it suffers from security issues that include authentication of the users as well as confidentiality assurance of their voice sessions. Various safeguard mechanisms already provide solutions for these problems with respect to other applications such as email or file transfer. These mechanisms are adopted for protection of the VoIP signalling and transport protocols as well. Additionally, other VoIP specific methods are introduced. An overview of the above is provided by Cao and Malik (2006) and Eren and Detken (2007). TLS (Transport Layer Security), MIKEY (Multimedia Internet KEYing) and IPSec (Internet Protocol Security) are some of the common security protocols that are employed by VoIP technologies for protection of the signalling plane.

In particular, the H.235 specification of H.323 is the one that defines various security profiles and a combination of those profiles by adopting the above solutions (Porter et al., 2006). Each of these profiles can be considered as a module that consists of a set of terms, definitions, requirements and procedures. In regards to SIP, a detailed analysis for the available security options was presented by Callegari et al. (2009). Explicit to SIP is the use of HTTP (Hypertext Transfer Protocol) Digest and S/MIME (Secure/-Multipurpose Internet Mail Extension), but TLS and IPSec were also discussed. IAX simplifies things by handling VoIP security in a different way. It has integrated support for MD5 (Message-Digest algorithm 5) or RSA (Rivest, Shamir and Adleman) authentication and AES (Advanced Encryption Standard) 128-bit encryption (Boucadair, 2009). This is in contrast to H.323 and SIP which they have to employ various security standards on top of them increasing complexity.

Protection of RTP should also be mentioned. SRTP (Secure RTP) is the typical method used and relies on HMAC-SHA-1 (Hash-based Message Authentication Code Secure Hash Algorithm) for authentication and integrity and AES for encryption of the voice payload (Bassil et al., 2005). However, the key exchange is an issue for SRTP which makes it vulnerable to Man-in-the-Middle attacks. An improvement of SRTP is ZRTP[1] (Zfone RTP) developed by Phil Zimmermann (Gupta and Shmatikov, 2007; Petraschek et al., 2008; Bresciani and Butterfield, 2009). ZRTP introduces the concept of SAS (Short Authentication String) which is a cryptographic hash calculated out of two Diffie-Helman values for key confirmation. SAS hashes are displayed to the end users through the user interface of their VoIP devices. In order to authenticate each other they have to confirm these values verbally over the phone. After validation shared secrets are cached for future calls between the same users. The above procedure is called ZRTP enrolment.

[1] http://zfoneproject.com/

A number of researchers studied the effects of user authentication and voice encryption on VoIP performance. Alexander et al. (2009) evaluated the impact of SRTP and ZRTP. They concluded that the processing overhead increases, but this does not affect VoIP performance. Moreover, they argued that ZRTP is slightly slower than SRTP due to the former's authentication procedure in the media channel. For extracting their results though, they used Wireshark which has some limited capabilities on measuring jitter and packet inter-arrival times. Callegari et al. (2009) suggested that VoIP specific solutions should be preferred over more generic ones like TLS or IPSec which impose bigger call setup delays. However, their recommendation was not based on practical experiences, but it was a theoretical approach. On the contrary, Ranganathan and Kilmartin (2003) showed that when IPSec with DES (Data Encryption Standard) and MD5 is used there is a 1,4% increase for SIP call setup times and a 1,6% increase in the voice stream delays. These figures grow exponentially during heavy network traffic. Their results were based on the OPNET simulator though.

The increased bandwidth consumption of IPSec due to the overhead caused by the extra protocol headers should also be underlined (Barbieri et al., 2002; Kazemi et al., 2010). Aire et al. (2004) stated that if IPSec cannot be avoided then the proposed encryption algorithm to use is AES. Lastly, the work of Spinsante et al. (2008) has to be mentioned. Measurements carried out in a simulator showed that SRTP introduces a negligible degradation of the voice quality. Based on the above studies it is suggested that security solutions designed particularly for real-time communications can provide the desirable results just in a fraction of the maximum performance that can be achieved when they are not used. It should be mentioned though that regarding the protection of the control plane of the out-of-band VoIP protocols TLS has become the first standard to consider.

3.4 VoIP and other packet transport techniques

As mentioned in the previous chapter, another important factor that poses a great impact on performance of packet switched telephony is the underlying network mechanisms. The following paragraphs analyse how voice is encapsulated and transmitted using various packet transport methods in relation to the brought up issues of IP telephony.

3.4.1 VoIP

It is already discussed that H.323 and SIP protocols depend on RTP and UDP for relaying voice. Hence, in reality VoIP implies VoRTPoUDPoIP (Voice over RTP over UDP over IP). Therefore, a big overhead of 40bytes in total is caused by the headers of these protocols (Karapantazis and Pavlidou, 2009). This is shown in Figure 3.1.

A solution to this problem is cRTP (Compressed RTP) header compression (Ranjbar, 2007) which reduces notably the size of all the IP, UDP and RTP headers, even though it is implied by its name that only RTP header is compressed. By using this technique the overhead is reduced from 40bytes to 4bytes or even 2bytes, depending on whether checksum is used or not. However, if cRTP is to be used, then it must be enabled from end to end across all links. This is not always feasible, due to possible incompatibilities. On the other hand, the maximum size of an unencrypted IAX frame is 12bytes, which is significantly smaller (Boucadair, 2009).

Apart from bandwidth utilisation, another issue of VoIP is reliability. IP is a connectionless protocol offering best-effort services (Francis-Cobley and Coward, 2004). In an IP network packets may not follow always the same route, which can often result to out-of-order and delayed deliveries. Inevitably, this brings into effect big delays, jitter and packet loss. For all the above reasons, other packet transport techniques have been developed, which try to overcome the above issues.

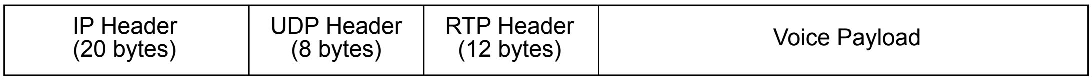


Figure 3.1: *Voice encapsulation in VoIP.*

3.4.2 VoATM

ATM (Asynchronous Transfer Mode) is a connection oriented data link layer protocol, contrary to the unreliable IP. Its operation is based on the use of cells, thus it is referred as a cell-switching technology. Every cell has a 5 bytes header and 48 bytes payload, resulting in a 53 bytes fixed length (Mainwaring, 2000). Moreover, multiplexing is possible and is realised by the use of PVCs (Permanent Virtual Circuits) or on demands SVCs (Switched Virtual Circuits). Additionally, ATM can handle both CBR (Constant Bit Rate) and VBR (Variable Bit Rate) data streams, but also provisions mechanisms for QoS (Kocak et al., 2009). Specifically, AALs (ATM Adaptation Layers) are defined for different types of applications. AAL2 is the one focused towards real-time VBR traffic, such as voice (Kasdirin and Rahman, 2003). The structure of an ATM/AAL2 cell is portrayed in Figure 3.2.

It can be seen that the AAL2 header together with the voice payload form the CPS (Common Part Sublayer), as it is called in the ATM terminology. A padding may also be needed to fill up the fixed length of the cells. This attribute can lead to poor performance for call sessions. Indeed, as Nasr (2003) underlines, cells can be dropped and delays can appear due to cell assembly, during busy hours or when flash crowds occur which result to high volumes of traffic. Ram (2002) and Samhat and Chahed (2005) agree with the above statement. Their studies show that the small footprint of the cells can contribute to really low latencies, but when the networks have to relay a magnitude of calls, then VoIP can scale better.

Kalmanek (2002) noted some other aspects of ATM technology. Despite its disadvantages in huge loads, ATM QoS mechanisms can work sufficiently. Unfortunately, they cannot be applied in a variety of situations. ATM is a highly complicated protocol and as a result this leads to extremely expensive implementations. Hence, VoATM is not always a viable solution.

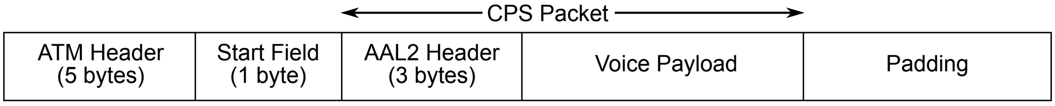


Figure 3.2: *ATM/AAL2 cell structure.*

3.4.3 VoFR

Frame Relay is another data link layer protocol and a successor of X.25 (Chin, 2004). It was introduced as a cheap alternative to the expensive leased lines. Its operation is similar to ATM, as it is based on the concept of PVCs and on demand SVCs multiplexing. The fundamental difference though, is that in this technology the packets or frames have a variable length of up to 4096 bytes. FRF.11 is the frame relay implementation for carrying voice (Groom and Groom, 2006). Each FRF.11 frame consists of a 2 bytes frame relay header, a 2 bytes FRF.11 header, a 2 bytes FCS (Frame Check Sequence) checksum and a 1 byte flag that indicates the end of the frame and the start of the next one. Figure 3.3 shows the format of an FRF.11 frame.

Unfortunately, the related research on VoFR is quite limited. According to Francis-Cobley and Coward (2004) most of the reports are white papers on vendor specific products. Nevertheless, their study showed that frame relay is quite suitable for packet telephony. The small overhead of the frame headers offers great bandwidth utilisation as well as small delays and jitter. Wright (2002) came to agree regarding the significant bandwidth efficiency of frame relay, but he did not comment on the delay issue.

Apart from congestion control though, frame relay does not offer any QoS (McQuerry, 2008). Furthermore, it supports speeds of up to 2 Mbps that the virtual channels need to share in the form of CIR (Commited Information Rate). This means that in case of bursty traffic or link overbooking by the service provider, each virtual channel can get easily overloaded. As a result, the voice quality degrades due to packet loss. Extending the capacity of a frame relay system is not cost-effective and so it is not an appropriate solution for enterprise networks.

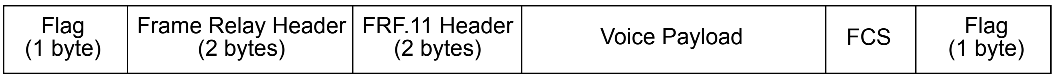


Figure 3.3: *FRF.11 frame.*

3.4.4 VoMPLS

ATM and Frame Relay emerged as the main Layer 2 technologies for WAN connectivity. On the other hand, the simplicity, flexibility and low cost of Layer 3 packet switching were the main reasons for IP adoption. Due to its connectionless nature though, it still depends on Layer 2 protocols, adding to complexity and inefficient use of network resources (Shah and Mohapatra, 2005; Malis, 2006). MPLS (Multi-protocol Label Switching) was standardised in order to offer convergence of the different Layer 2 and Layer 3 services as well as to provide high speed data forwarding.

Its operation is based on the use of *labels* that are imposed between Layer 2 and Layer 3 headers. Thus, MPLS is characterised as a *Layer 2.5* protocol. According to Vazquez et al. (2004) and Zubairi (2008) in MPLS networks voice can be carried either directly (VoMPLS) or with the VoIP protocol stack (VoIPoMPLS). It is also possible to carry voice using AAL2/ATM (VoMPLSoAAL2oATM) or only AAL2 (VoMPLSoAAL2). Certainly, VoMPLS, which is defined by MFA IA 1.0 standard, is the preferred method, since it has the lowest header overhead. A VoMPLS packet has a header of only 4 bytes and a voice payload that must be a multiple of 4 bytes, so up to 3 bytes padding may be needed. This is illustrated in Figure 3.4.

Unfortunately, end user devices or operating systems are not able to comprehend the MPLS protocol and still rely to IP for transporting data. However, Goode (2002) suggests that MPLS has great features regarding traffic engineering. In an MPLS core network the routing and forwarding functions are based on the use of labels. Routing is performed by LSRs (Label Switching Routers) which create LSPs (Labels Switched Paths) in advance in order to connect the edge routers. Each LSP can forward different sets of packet flows classified as FECs (Forwarding Equivalence Classes). Within an LSP up to 248 calls can be multiplexed (when VoMPLS MFA IA 1.0 standard is used) (Vázquez et al., 2004).

Furthermore, MPLS traffic engineering mechanisms and LSPs can provide bandwidth guarantees, call admission control and support for QoS models, such as IntServ and DiffServ (Lee, 2005; Uzunalioglu et al., 2006). In case of a link failure, fast re-routing can be achieved more effectively compared to the traditional dynamic routing protocols, since backup LSPs may be pre-allocated for rapid restoration (Iselt et al., 2004; Pasqualini et al., 2004a). Additionally, when congestion occurs, an LSP can follow a different route (Andersson and Bryant, 2008). This is not possible with the classic SPF (Shortest Path First) algorithms of routing protocols. Normally, they constantly select the shortest path that was calculated initially, but based on the network conditions, this is not always ideal. All the above traffic engineering features of MPLS allow for low end-to-end delays, jitter and packet loss that are necessary for supreme packet telephony.

According to Rong (2005) though, label distribution and signalling mechanisms for the establishment of LSPs can bring into effect long call setup delays, when the MPLS traffic engineering extensions are used. In addition, Wright (2002) asserted that VoMPLS can pose good to excellent bandwidth utilisation depending on the underlying Layer 2 technology being used. Finally, Barakovic et al. (2006) and Kocak et al. (2009) outlined that MPLS outperforms the traditional IP. In their experiments however they used the traffic engineering extensions and QoS features of MPLS, while IP was run without any optimisations. Moreover, their studies were based on simulation programs and not on real equipment.

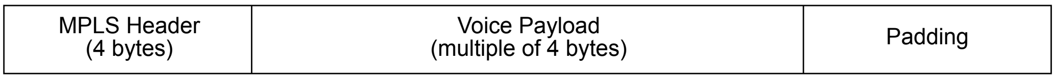


Figure 3.4: *VoMPLS packet.*

3.5 Methods for traffic measurements

In order to evaluate the performance of each VoIP signalling protocol or packet transport mechanism, traffic measurements need to realised. The next paragraphs examine the main methods for network traffic measurements and discuss their advantages and disadvantages.

3.5.1 Active and passive probes

The use of active and passive probes specify the most common approaches for network monitoring and measurements (Agrawal et al., 2006a). Active measurements involve the operation of active probes which inject traffic in the network. They are special software agents installed on designated machines that in their simplest form can send ICMP packets using ping or traceroute in order to measure network properties, like the RTT (Round Trip Time). More advanced active probes can generate traffic (e.g. VoIP calls) when asked and report back their measurements to a central management device. Typically, active probes are able to detect service degradations and give a good picture for the whole network, but they can not point to the node that causes congestion or failures. Thus, they provide a *black box* measurement method.

Contrary to active probes, passive probes just snoop on traffic without injecting any packets (Agrawal et al., 2006b). They can continuously monitor the network performance per application or protocol, while active probes measure the performance for synthetic traffic. Furthermore, passive probes can segment a network and trace problematic links, but only between the points of installation. Hence, it is not possible to provide an overall picture of the network. It must also be noted that passive measurements can be software-based or hardware-based. The first use off-the-shelf NICs (Network Interface Cards) combined with either modified versions of operating systems and device drivers or just monitoring software tools in order to capture traffic (Ubik and Zejdl, 2008). While this method is cost-effective and flexible, it cannot scale well in extremely high speed networks. In such cases specialised monitoring hardware like DAG (Data Acquisition and Generation) cards[2] or other FPGA (Field Programmable Gate Array) based solutions must be adopted, as can be seen in the work of Sommers et al. (2007), Loiseau et al. (2009) and Pezaros et. al (2010), but in the price of increased cost.

Generally, each method has advantages and disadvantages. For example, measuring performance by injecting traffic (active probes) can cause problems to real traffic. On the other hand, clock synchronisation of the monitored systems and timestamping of the captured packets is very critical for fine-grained non-intrusive measurements (passive probes). NTP (Network Time Protocol) and GPS (Global Positioning System) are the most common solutions for clock synchronisation, while daisy-chaining of timestamping clock can be used when DAG cards are installed (John et al., 2010). Additionally, passive probes can have legal or ethical implications regarding privacy (Claffy et al.,

[2] http://www.endace.com/

2009). Another issue for both the methods is the placement of the probes, a big concern especially for huge networks (Chaudet et al., 2005). To overcome all the above, hybrid models have been proposed, like in the work of Zangrilli and Lowekamp (2003) and Ishibashi et al. (2004). According to Agrawal et al. (2007) though, active and passive probes can work equally well.

3.5.2 Packet traces and network flows

As explained earlier, when using passive methods for network measurements the probes installed just "listen" to traffic. Collecting network data can be accomplished by capturing packets with common tools such as the well-known tcpdump[3]. All the captured packets are written in a trace file which can be collected for *offline* analysis (Brownlee, 2001b). *Packet traces* are very useful because they can provide many details about different aspects of a network. Recording and analysing every individual packet is possible to reveal explicit information regarding user, application and protocol patterns. These data can be utilised for fine tuning a network or enhance its *real-time* monitoring (Alcock et al., 2007; Rubio-Loyola et al., 2008). The accuracy of this method can degrade though when having lot of dropped packets (such is the case with commodity PC hardware in extremely high speed networks that was examined previously).

Instead of recording packets, another approach is to record flows. The concept of *network flows* is very important for traffic measurements, but although used for many years a specific definition does not exist. For example in routing a flow is identified as the unidirectional traffic going from one host to another (Brownlee, 2001b). Generally, a network flow is defined loosely as the packets passing through a node at a given time period which have some common properties. Classification of unidirectional flows by a 5-tuple (protocol, source and destination addresses, source and destination ports) is highly appreciated in the research community though. Flow monitoring is a flexible way for performing traffic measurements, since proper probes can be either installed as stand alone devices or embedded in network equipment such as routers. There are however specific requirements that need to be met (Molina et al., 2006). In particular, flow identification and update of flow records must be adequately fast in order to have accurate results. Fast algorithms for packet classification are essential for that and were studied extensively by Gupta and McKeown (2001). Unfortunately, in some cases use of expensive dedicated monitoring hardware is needed. In Figure 3.5 the structure of a flow monitoring system is shown (Molina et al., 2006).

It is worth noting that a combination of the two heterogeneous methods is possible as shown by Zhang (2007). This is useful for cross-validation and complementary reasons, but overall increases complexity.

[3] http://www.tcpdump.org/

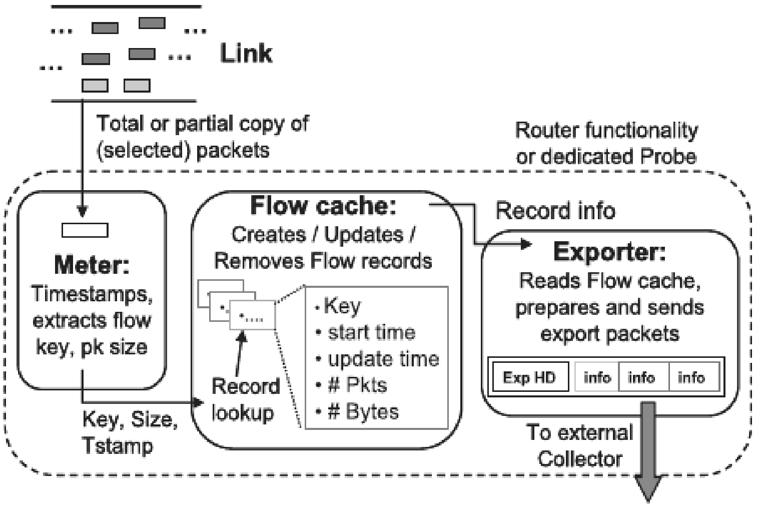

Link
Total or partial copy of (selected) packets
Router functionality or dedicated Probe
Meter:
Timestamps, extracts flow key, pk size
Key, Size, Tstamp
Flow cache:
Creates / Updates / Removes Flow records
Record lookup
• Key
• start time
• update time
• # Pkts
• # Bytes
Record info
Exporter:
Reads Flow cache, prepares and sends export packets
Exp HD
info
info
info
To external Collector

Figure 3.5: *Structure of a flow monitoring system (Molina et al., 2006).*

3.6 Frameworks and tools for traffic measurements

In the previous section, the main methods for realising traffic measurements were discussed. However, since network monitoring is a huge field, based on the above methods a number of frameworks and tools have been developed that try to overcome various issues. Accuracy and scalability are of major concern, but not the only things to consider when designing a monitoring system. Deploying and configuring the system in a flexible and effective way is also of equal significance. The following paragraphs present possible solutions by analysing the merits and drawbacks of their architecture.

3.6.1 NetFlow and IPFIX

NetFlow is a flow monitoring solution designed by Cisco[4]. Initially developed for its routers and multilayer switches, nowadays it can be found in networking equipment made by other vendors as well (Rossi and Valenti, 2010). As described by Zang (2009), a NetFlow system is comprised by the following: An NDE (NetFlow Data Export) and an NFC (NetFlow Flow Collector). The first one is usually found in the networking devices. It records flows and exports them as UDP packets to an NFC. NetFlow in a router can be configured on a per-interface basis, but nevertheless this contributes to an extra processing and memory overhead for the router. This becomes a major concern when massive loads of flows have to be managed especially in high speed networks. In some expensive Cisco platforms this is addressed by using an ASIC (Application Specific Integrated Circuit) chip which offloads the main processor from the task of flow monitoring, but the method used by all NetFlow enabled hardware is that of sampling (McGlone et al., 2008). Many researchers argue that configuring a proper sampling rate is difficult and thus it may introduce inaccuracies when not set properly. Therefore, they propose alternative solutions for improving NetFlow. For example, Estan (2004) suggests the use of adaptive sampling rate which is automatically set in a dynamic way

[4] http://www.cisco.com/web/go/netflow

depending on the traffic mix and scales better. A similar approach was discussed by Choi and Zhang (2006). Choi and Supratik Bhattacharyya (2005) agreed that adaptive sampling would be a great improvement, although his study on NetFlow showed that it performs extremely well.

On that grounds it is no question why NetFlow became over the years the de facto standard for flow monitoring. Various versions of NetFlow exist with v5 being the most widely deployed and v9 being the latest and probably the most important since it is standardised by the IETF (Internet Engineering Task Force) under the name IPFIX (IP Flow Information Export) (Pras et al., 2009). The main innovation behind NetFlow v9 and IPFIX is the use of templates for flexible flow definition which can extend the configurability of the monitoring systems. This is very convenient when different types of traffic traverse a network. Moreover, IPFIX specifies a common file format for flow export and storage and introduces the concept of *mediators* (Trammell et al., 2007; Anderson et al., 2009). NFCs have difficulties to cope in large networks, so mediators act as intermediate entities that receive and process flow data and report back to other mediators or collectors. In that way distribution of processing and storage is achieved, but on the other hand an extra layer of complexity and some additional points of failure are introduced. For this reason the number of mediators used must be seriously considered.

Other issues are related to anonymising and securely transmitting the NetFlow/IPFIX records and remain to get standardised. Especially anonymisation of flow data is very critical to avoid privacy issues, but already some tools offer solutions, like the *anontool* developed by Foukarakis et al. (2007). This software is not the only one though. Due to the popularity of NetFlow/IPFIX a great number of compatible open source and proprietary tools have been developed over the years (van den Nieuwelaar and Hunt, 2004; Pilli et al., 2010). Some of them are complete suites for anonymising, securely transmitting, processing and visualising flow records for a number of metrics including jitter and end-to-end delay. Of particular interest is *Argus*[5] which is described as a superset of IPFIX. It is a real time flow monitoring system which contrary to NetFlow/IPFIX can support bi-directional flows. Argus is appraised not only by the academic community, but by many US government departments as well (McRee, 2007).

Adding to the above it is worth mentioning that NetFlow/IPFIX can be adopted for other applications apart from performance measurements as suggested by Drago et al. (2010). Detecting remote connectivity problems before users take any notice of disruptions caused to offered services is of special significance. A good example of that is YouTube that was not reachable for more than an hour due to a router misconfiguration located at an Internet provider at Pakistan. Monitoring flows at edge routers every 5min can help identifying the cause of such abnormal situations resulting in extremely reduced downtimes. Generally, it can be concluded that NetFlow/IPFIX is a trustworthy and flexible flow monitoring standard even with the debatable limitations of the sampling method.

[5] http://qosient.com/argus/

3.6.2 RTFM

RTFM (Realtime Traffic Flow Measurement) is an IETF standard for flow monitoring. Its architecture, although similar in concept with that of NetFlow/IPFIX, follows a different route. RTFM defines four entities: Meters, meter readers, managers and analysis applications (Brownlee, 1998, 2001b). Meters are the major RTFM components identical to NetFlow NFCs. They gather packets which they aggregate to different bi-directional flows (as opposed to NetFlow's unidirectional flows) based on various rule sets. Meter readers are responsible for collecting the flow records from meters while managers monitor the proper operation of meters and meter readers. They are also in charge of configuring the meters by uploading rules to them. Finally, the analysis applications just process the flow data recorded by the meters. The use of rule sets is a powerful feature of RTFM since it adds to increased adaptability and flow definition. For maximum flexibility a special high-level language was developed, the SRL (Simple Ruleset Language), which allows for easy and fast creation of rules. Having to learn a new language complicates things for users who just want an easy to deploy monitoring system, but certainly is a cunning aspect of RTFM.

Special reference must also be made to the meters. These usually are SNMP (Simple Network Management Protocol) MIBs (Management Information Bases), although RTFM defines RTFM MIBs that can be integrated to routers and switches or be separate units. SNMP is indeed a very easy to implement protocol, which explains its wide adoption, but unfortunately has many limitations. As noted by King and Hunt (2000) it is not scalable in large networks. Collecting huge amount of data is inefficient since a lot of SNMP messages need to be exchanged. Apparently, this contributes to increased management traffic. SNMP is also criticised for its poor performance and inaccurate measurements (Roughan, 2010). Artifacts, errors and missing data are caused partially as a result of bad SNMP implementations, but mainly because of the protocol's nature (SNMP is based on the use of counters which often fail to function properly). Lastly, security issues were not resolved until SNMP v3. An alternative to SNMP MIBs could be RMON2 (Remote Network MONitoring v2) MIBs, but unfortunately RMON2 does not have any notion of bi-directional flows as specified by the RTFM standard.

The classic RTFM implementation is the open source *NeTraMet*[6] distribution. It is comprised by NeTraMet, an SNMP agent designed according to the RTFM MIB specification, and NeMaC, a combined RTFM meter and manager (Brownlee et al., 2003). Few examples of its use include the defunct Kawaihiko university network in New Zealand as well as the UCSD (University of California, San Diego) (Brownlee and Fulton, 2000; Brownlee, 2001a). Another RTFM implementation is the one developed by the KOREN (Korea Advanced Research Network) (Song and Choi, 2006). The researchers concluded that RTFM is a useful flow monitoring framework and that it can provide solid traffic measurements, despite the shortcomings imposed by the SNMP architecture. It can be seen though that it is mainly used for academic purposes. NeTraMet is able to collect and analyse NetFlow records and this is a strong indication of NetFlow's popularity and strong establishment in the flow monitoring market.

[6] http://www.caida.org/tools/measurement/netramet/

3.6.3 Tstat

Tstat is a traffic monitoring tool developed at the Politecnico di Torino (Mellia et al., 2003, 2005). It is based on libpcap[7] for packet capturing and real-time processing of packet traces. It started as an advancement of tcptrace[8], but the main motivation behind its development was the lack of tools able to automatically produce packet and flow statistical data from traces. Truly, one of its novel features is the ability to measure more than 80 aspects of network traffic such as TCP congestion window size, out-of-sequence segments and duplicated segments. For each one of these statistics instead of dumping single measured data, Tstat builds periodically histograms which estimate the distribution of the performance metrics.

Tstat cannot only analyse the network and transport layers (layers 3 and 4) of the network stack, but also the application layer (layer 7) for easy classification of the network traffic. This is possible due to Tstat's modular design which enables for excellent performance even when used with commodity hardware (Rossi and Mellia, 2006). Every packet is processed by Tstat's trace analyser IP module. Then different modules take care if the packet is a TCP or UDP packet. With this kind of architecture is also easy to include modules for new types of measurements. All these useful features have been utilised and tested extensively in the GARR (Gestione Ampliamento Rete Ricerca) Italian academic and research network as well as for monitoring VoIP and other real-time multimedia traffic in FastWeb, a major Italian Internet provider offering broadband services (Mellia and Meo, 2010; Finamore et al., 2010). Tstat can indeed reveal remarkably detailed information for a monitored network. Unfortunately, as in RTFM's case, it is used mainly for academic purposes. A wider deployment of the tool would be necessary in order to make safer conclusions.

3.6.4 SmokePing

SmokePing is a well-known web based active monitoring tool for measuring traffic rate, latency, latency distribution and packet loss designed by Toby Oetiker (Jeliazkova et al., 2006). It is highly configurable and can use a variety of probes such as fping[9] or Cisco IOS ping. Setting different parameters of probing and pattern matching in order to produce alerts is also feasible and easy to achieve. Through its web interface detailed graphs are produced using *RRDtool*[10] as the database and graphing backend. Multiple sites and devices that often change IP can be monitored at once without any difficulties giving an overall picture of a large network (Jajor et al., 2009). The power of SmokePing lies mainly in its capability of using different probes. Apart from measuring the typical aspects of a network (delay, packet loss, etc) it is also possible to monitor network services such as web, email and IP address resolution. Additionally, its web nature allows for easy integration to other platforms. Generally, SmokePing is an extremely useful monitoring tool. However, like mentioned earlier, the use of active

[7] http://www.tcpdump.org/

[8] http://www.tcptrace.org/

[9] http://fping.sourceforge.net/

[10] http://oss.oetiker.ch/rrdtool/

probes can create problems to real traffic. Furthermore, wrong results can be anticipated as an outcome of a busy device. For example, when probing routers under heavy load their CPU will slow down ICMP responses resulting to incorrect measurements. On that basis it can be concluded that despite SmokePing's easy deployment and operation, the administrators should take special care when setting up its probes.

3.7 The Asterisk IP PBX platform

Asterisk is probably the most popular open source IP PBX. Originally created by Mark Spencer of Digium[11] is well-recognised as a trustworthy VoIP platform. It can manage all of the main VoIP protocols including H.323, SIP and IAX plus interoperate with standard circuit-switched telephone equipment (Qadeer and Imran, 2008). The expected features of a traditional PBX system, like call forwarding, call waiting, music on hold, etc, are implemented by the Asterisk developers (Imran and Qadeer, 2009). All these are possible due to its core architecture which is abstracted from the protocols and hardware interfaces that interconnect it to the PSTN. Specifically, the Asterisk core consists of the following four entities:

1. **PBX switching:** The switching core is the heart of the system responsible for connecting calls originated by users or automated tasks.
2. **Application launcher:** It launches applications for various services such as voicemail, file playback and directory listing.
3. **Codec translator:** It performs audio encoding and decoding of various audio compression formats by using codec modules.
4. **Scheduler and I/O manager:** It is in charge of controlling the low level tasks of the system.

Ahmed and Mansor (2008) evaluated Asterisk for its performance. It was proved that up to 60 concurrent calls can be handled without any problems when a weak processor is used whereas when a powerful dual core processor is purchased then more than 600 concurrent calls are possible. Therefore, there is no question why Asterisk was employed in order to offer VoIP services in large organisations like universities (Yamamoto et al., 2008; Zasepa et al., 2009). An interesting project for creating call management policies in an easy and effective way should also be mentioned. This was implemented by Konstantoulakis and Sloman (2007) further extending Asterisk's capabilities. Its interoperability with SS7 (System Signalling No.7) signalling and other VoIP platforms, such as Cisco Call Manager, was also discussed by Rudinsky (2007) and Chava and How (2007) which they argued that Asterisk can successfully interoperate with other diverse systems. Overall it can be said that over the years Asterisk has been proven as a robust VoIP platform.

[11] http://www.digium.com/

3.8 Chapter overview

This chapter highlighted the different aspects of packet telephony and performance measurements. Firstly, the architectures of the main in-band and out-of-band VoIP protocols were studied. It was seen that H.323 is a quite robust and strictly specified out-of-band protocol compared to SIP. On the other hand SIP is more flexible and more scalable. Furthermore, SIP is far less complex and due to its HTTP nature is easier to integrate in the modern Internet era. On the other hand both of the protocols face difficulties when it comes to firewalls and NAT devices. This is where IAX has a clear advantage. As an in-band signalling protocol it just needs one UDP port to traverse through this type of security equipment. Moreover, it can multiplex many calls in the same stream, thus saving bandwidth. Unfortunately though for its developers, it hasn't gained great popularity in the VoIP market.

When matters come to security then the VoIP engineer has a variety of choices. For H.323 it is the H.235 specification that defines various security profiles and a combination of those for providing security. SIP again can adopt various solutions that come from the web and email world. TLS is the standard though for protecting SIP signalling messages. In regards to the voice payload, SRTP and its extension ZRTP can be used. A more generic solution is IPSec, but it should be avoided, unless this is not possible, due to the big call setup delays that it brings into effect. As for IAX, things are less complicated, since it has integrated support for MD5, RSA and AES 128-bit and therefore does not depend on extra mechanisms on top of it, thus minimising the additional overhead which is caused when security is applied.

Another factor that has a great impact on the performance of packet telephony is the underlying network infrastructure. ATM, Frame Relay and MPLS were examined as alternatives to pure IP networks. It was found that ATM and Frame Relay are not viable solutions for carrying voice. Their main issue lies on the fact that both are incapable of handling properly real-time traffic during busy hours or when flash crowds occur. Moreover, ATM is very expensive to implement. On the contrary, MPLS has great features that are really useful for voice communications, such as bandwidth guarantees, call admission control and extremely fast re-routing mechanisms in case of link failures. Voice directly over MPLS would be ideal, but unfortunately end user devices are not MPLS aware and still rely to IP. Nevertheless, studies show that MPLS outperforms the aforementioned protocols.

Measuring the performance of the above technologies on real-life networks is also critical. Various approaches have been developed for that. The use of active probes provides a black box type of measurement by injecting ICMP packets or even generating real calls. Passive probes just capture traffic and they can be software-based or hardware-based. Capturing traffic can be accomplished either by recording each individual packet or by recording flows. Generally, active and passive methods have advantages and disadvantages, but overall they are considered equally good. The industry standard though is Cisco's NetFlow which became an IETF RFC under the name IPFIX. Equally popular is SmokePing, a web-based tool that can utilise various active probes for easy monitoring of multiple sites and devices.

Finally, the Asterisk VoIP platform was reviewed. Asterisk is an open source IP PBX capable of implementing all of the major VoIP signalling protocols. It was concluded that it can serve as a robust VoIP system which can also successfully interoperate with the traditional circuit switched PSTN.

3.9 Conclusions

As it was seen in this chapter many researchers carried out their studies by using simulation software or utilising non-standard methods in order to measure just few of the key performance metrics for VoIP. Therefore, in order to accomplish the aim of the evaluation of the in-band and out-of-band protocols there is a need for a novel experimental framework. Based on the findings of the literature review the main conclusion is that in order to design and implement a solid evaluation framework a researcher has to define a number of parameters. First the VoIP protocols to evaluate and their key metrics. Second the variables that can have an impact on these protocols. And third the appropriate methods and tools that will give valid and accurate results. The next chapter discusses these choices that need to be made for the design of the novel experimental framework.

CHAPTER 4

Design and Methodology

4.1 Introduction

This chapter outlines the design of a novel experimental framework for monitoring network traffic in order to evaluate end-to-end service performance. Since the focus of this project is IP telephony, evaluating the performance of different VoIP protocols is the aim. Based on the findings of the literature review, the choices made for the development of this framework are justified. In particular, these are the main requirements that were determined:

1. The VoIP protocols under scrutiny and their key metrics.
2. The security standards and underlying network mechanisms for evaluating their impact on VoIP.
3. The appropriate monitoring system for performing traffic measurements and extracting the results.

The key component of this framework is the Asterisk IP PBX which runs on the Linux operating system. A number of scholars, including Abbasi et al. (2005), Ahmed and Mansor (2008) and Montoro and Casilari (2009), validated Asterisk's viability and robustness for research purposes. Furthermore, due to its open source nature Asterisk is freely accessible and has a large community that can provide support. The following sections provide an overview of the framework as well as a description of the testbed that was designed and the methods that were followed in order to realise the experiments and gather the data for analysis.

4.2 Experimental framework overview

Like it was mentioned in the introduction, the first thing that was considered was the VoIP protocols under scrutiny. The literature review examined H.323, SIP and IAX. H.323 was found to be more complicated than SIP and also less extensible and scalable (Glasmann et al., 2003). Due to these reasons SIP has gained tremendous popularity over the years (Basicevic et al., 2008), so this was the out-of-band VoIP protocol which was evaluated. On the other hand IAX is the only existing in-band VoIP protocol and thus the one that was compared with SIP.

With respect to securing VoIP calls, various mechanism exist. Callegari et al. (2009) suggested that generic solutions like IPSec should be avoided, since they introduce big call setup times. SIPS (Session Initiation Protocol Secure) or SIP over TLS has been standardised by IETF (Salsano et al., 2002) and implemented by many VoIP platforms, including Asterisk. Regarding protection of the voice payload in out-of-band protocols, SRTP is the main protocol to use with ZRTP being its extension that defends against Man-in-the-Middle attacks in a more reliable way (Bresciani and Butterfield, 2009). IAX does not depend on any of these protocols, due to its embedded support for MD5, RSA and AES (Boucadair, 2009).

In addition, the network infrastructure that relayed voice was determined. Again, in the previous chapter various network mechanisms were reviewed. Apart from a pure IP network, technologies such as ATM, Frame Relay and MPLS were explored. According to the researchers the first two are not suitable for packet telephony, because they are unable to handle properly real-time data in case bursty traffic occurs (Samhat and Chahed, 2005). Cost of implementation is also another issue, especially for ATM (Kalmanek, 2002). Therefore, MPLS was chosen to be evaluated in comparison to a pure IP infrastructure.

Finally, the last thing which had to be specified was the appropriate method for gathering data in order to evaluate performance of all the above. As defined by Goode (2002) and Karapantazis and Pavlidou (2009) the key VoIP metrics are the one-way delay, jitter, packet loss and bandwidth utilisation that can be measured by using either active or passive probes. Fatemipour and Yaghmaee (2007) explained the advantages of NetFlow/IPFIX which is the industry standard for realising passive flow monitoring measurements, but Agrawal et al. (2007) suggested that active measurements can provide equally accurate results. Furthermore, Zangrilli and Lowekamp (2003) proposed a hybrid model in order to overcome the limitations of each method. Therefore, it was decided to perform the measurements by utilising a hybrid monitoring system based on NetFlow/IPFIX and active probes. Apart from the VoIP metrics, the processing and memory overheads of each protocol were also measured. Based on the above Figure 4.1 presents a diagram with the components that constitute this experimental framework.

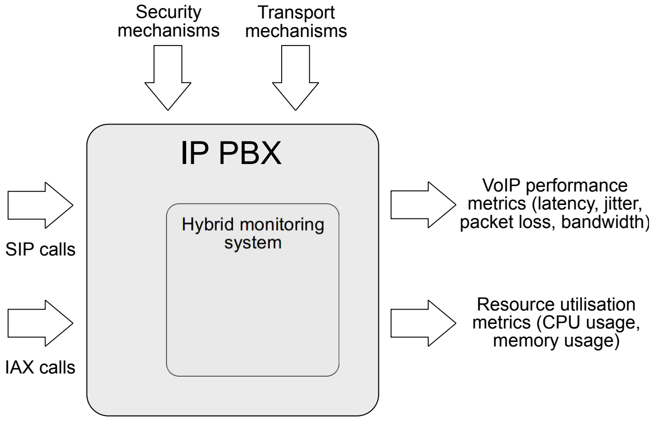

Security mechanisms
Transport mechanisms
IP PBX
Hybrid monitoring system
SIP calls
IAX calls
VoIP performance metrics (latency, jitter, packet loss, bandwidth)
Resource utilisation metrics (CPU usage, memory usage)

Figure 4.1: *Experimental framework diagram.*

4.3 Testbed description

The main objective of the designed testbed was to represent the architecture of a real-life VoIP network. Such a network consists of the following components: VoIP devices and CE (Customer Edge) routers from the customer's side and PE (Provider Edge) and core routers from the service provider's side. Figure 4.2 illustrates the topology of the testbed. It can be seen that there were two customer sites interconnected through the provider's WAN (Wide Access Area) cloud. All of the CE, PE and core routers were linked together by using their Ethernet interfaces. OSPFv2 (Open Shortest Path First version 2) was selected as the Layer 3 routing protocol, because it is an open standard IGP (Interior Gateway Protocol) defined by IETF and embraced by the network industry in its total. In first instance, experiments were realised with just OSPFv2 configured. A second run of the same experiments was carried out by setting up MPLS in the WAN cloud. A note which should be made here is that no traffic engineering or load balancing was used during the experiments. Thus, router R4 actually does not take any part in the whole routing process. It is presented here though in order to show the additional scenarios that can be investigated by exploiting the current experimental framework. Eventually, one should notice the presence of the Asterisk PBX servers installed in each site. Their functionality is explained in the next section.

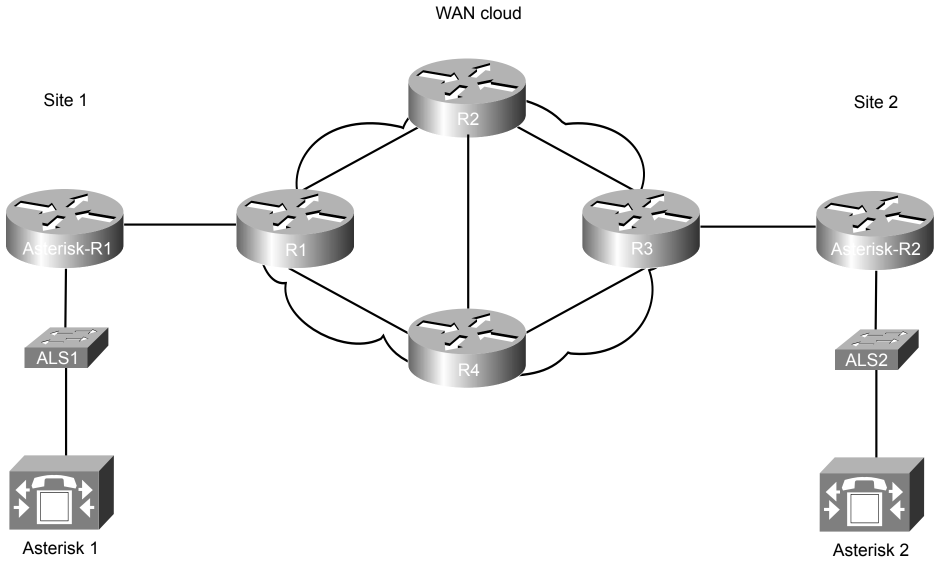


Figure 4.2: *Testbed topology.*

4.4 VoIP call generator

Asterisk was reviewed in the previous chapter and it was concluded that it is a robust VoIP platform. Yamamoto et al. (2008) and Zasepa et al. (2009) discussed cases of its deployment in big organisations. Apart from real-life examples though, as it was mentioned earlier Abbasi et al. (2005) supported the viability of Asterisk for research purposes as well. However, configuring Asterisk is a demanding task and requires an in-depth understanding of the way it implements things. Challenges that were faced during its configuration are discussed in Chapter 5.

In this project Asterisk was used as a VoIP call generator. Calls were generated from Asterisk 1 to Asterisk 2. Upon connection a pre-recorded message of about 2,5 minutes was played back in both directions. In this way real phone conversations were simulated. Montoro and Casilari (2009) followed a similar approach for the purposes of their study. The experiments were run for SIP/RTP and IAX with and without trunking under different conditions and different loads. In particular 30, 50, 90 and 120 concurrent calls were generated in order to stress the systems. In the literature review it was mentioned that up to 60 concurrent calls can be handled by Asterisk without any problems when a weak processor is used while when a powerful dual core processor is available then more than 600 concurrent calls are possible (Ahmed and Mansor, 2008). Since the PCs that were used in the testbed had a medium class processor, the maximum of 120 simultaneous calls was decided as a proper value for the experiments.

The tests were performed with and without the use of security mechanisms and repeated once for the OSPFv2 cloud and once for the MPLS cloud. Evaluating different codecs and QoS mechanisms such as DiffServ was out of scope, thus the European standard for any telephony system G.711 A-Law codec was selected and non VoIP traffic was not involved in any way during these tests.

4.5 Gathering data

The monitoring system was the last component of the experimental framework to consider. As it was explained previously a hybrid system based on NetFlow/IPFIX and active probes was decided to be used. For this reason, Argus flow monitoring software suite was installed in both of the Asterisk machines in order to capture flows, analyse and visualise measurement data regarding the four key VoIP metrics (one-way delay, jitter, packet loss, bandwidth utilisation) (McRee, 2007). Argus can also act as a NetFlow collector, therefore NetFlow was configured in one of the core routers for measuring the link utilisation in the WAN cloud. In addition, an active probe in the form of fping was setup in Asterisk 1 in order to measure latency, while Argus recorded the ICMP flows and did the one-way delay calculations. Finally, the resource utilisation metrics relevant to CPU and memory usage were measured by using the standard Linux resource statistics tool *dstat*[1].

4.6 Conclusions

This chapter described the methods that were followed in order to design a novel experimental framework for achieving the aim of the evaluation of different VoIP protocols and the factors that can affect their performance. As a generic framework it can be expanded in order to investigate further possibilities that are not limited to VoIP telephony. For example, peer-to-peer, traffic engineering, load balancing and QoS mechanisms are possible to be explored with few modifications regarding the configuration of the dif-

[1] http://dag.wieers.com/home-made/dstat/

ferent components involved in the testbed. In any case during the design phase the researcher needs to take decisions in respect to the various elements that constitute a similar experimental framework. Specifically, one should consider the following: The protocols to be tested, the various variables that may influence the performance of these protocols, such as the network infrastructure or the presence of encryption and background traffic, and the methods and tools to utilise in order to generate data, gather, analyse and visualise the results. The next chapter provides details on the implementation of the presented framework.

CHAPTER 5

Implementation

5.1 Introduction

The previous chapter presented the methods employed in order to design a novel experimental framework for the evaluation of VoIP signalling protocols. This chapter describes the implementation of this framework by providing snippets of the configuration code for its components. In particular, Cisco 2811 routers and 2950 switches, the Asterisk call generator and the NetFlow/Argus monitoring system had to be setup. Furthermore, the complications that were faced during the implementation phase are discussed.

5.2 Network configuration

The first part of the implementation phase was the configuration of the Cisco routers and switches. For the latter, no special treatment was needed. The simple network topology that was designed, did not require special switching features (e.g. VLANs). Therefore, only the routers had to be configured properly for routing the VoIP traffic. As it was explained earlier, OSPFv2 was selected for this purpose. Three different OSPF areas were setup (Stewart, 2007). This is illustrated in Figure 5.1.

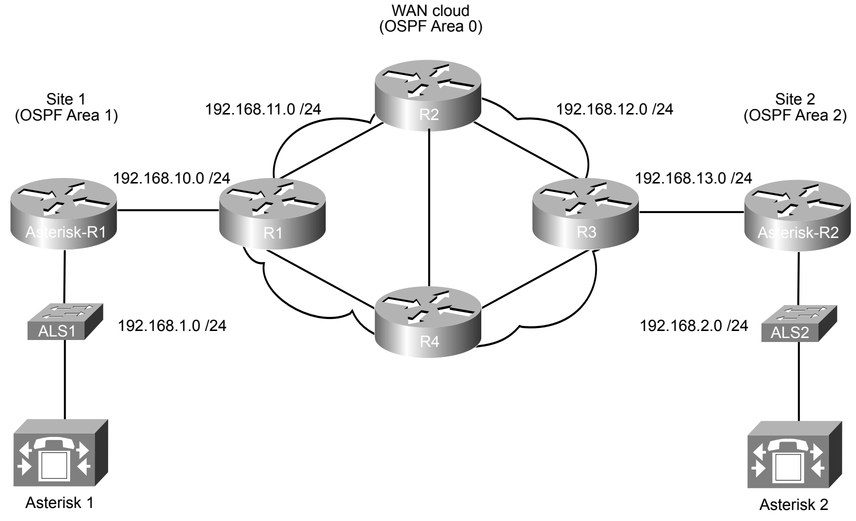


Figure 5.1: *Network topology.*

Figure 5.2 provides the OSPF configuration for the router Asterisk-R1 on Site 1.

```
interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 duplex auto
 speed auto
interface FastEthernet0/1
 ip address 192.168.10.1 255.255.255.0
 duplex auto
 speed auto
router ospf 100
 router-id 192.168.100.1
 log-adjacency-changes
 area 1 stub
 network 192.168.1.0 0.0.0.255 area 1
 network 192.168.10.0 0.0.0.255 area 1
```

Figure 5.2: *Router Asterisk-R1 configuration.*

The PE router R1 had to be configured as it is shown in Figure 5.3.

```
interface FastEthernet0/0
 ip address 192.168.10.2 255.255.255.0
 duplex auto
 speed auto
interface FastEthernet0/1
 ip address 192.168.11.1 255.255.255.0
 duplex auto
 speed auto
router ospf 100
 router-id 192.168.250.1
 log-adjacency-changes
 area 1 stub
 network 192.168.10.0 0.0.0.255 area 1
 network 192.168.11.0 0.0.0.255 area 0
```

Figure 5.3: *Router R1 configuration.*

The rest of the routers were configured accordingly. For the MPLS cloud, the appropriate command had to be applied to the corresponding interfaces. For example, for router R3:

```
interface FastEthernet0/0
 ip address 192.168.12.2 255.255.255.0
 duplex auto
 speed auto
 mpls ip
interface FastEthernet0/1
 ip address 192.168.13.2 255.255.255.0
 duplex auto
 speed auto
```

Figure 5.4: *Router R3 MPLS configuration.*

5.3 Call generator configuration

Like it was mentioned in the previous chapter, Asterisk was used as the call generator. For the experiments two different versions of Asterisk were compiled, installed and configured in each site: The first for testing SIPS/SRTP and the second for testing SIP/ZRTP. Unfortunately, none of those is officially supported yet. SRTP is implemented though in the Asterisk SVN trunk (which will soon be an official Asterisk 1.8 release). As for ZRTP, Phil Zimmermann's Zfone project provides a patch which is compatible only with the older Asterisk 1.4.23.1. In any case, the host operating system was Ubuntu Server Edition 10.04 installed in usual PCs equipped with Pentium IV @ 3GHz processors and 512MB RAM memory.

In order to create calls, there are two things that need to be defined in the Asterisk configuration files (Qadeer and Imran, 2008; Imran et al., 2009). First, SIP and IAX extensions need to be specified in *sip.conf* and *iax.conf* configuration files. Simply put, extensions are the phone numbers that one can dial. Second, a dialplan must be configured in *extensions.conf* configuration file. The dialplan designates the actions that will be taken by Asterisk when an extension is called. Configuring extensions is fairly simple, as it can be seen in Figure 5.4. For this project, 1000 and 3000 were the SIP and IAX extensions respectively in Asterisk 1. Accordingly for Asterisk 2, 2000 and 4000 were set up.

```
[1000]
type=friend
context=phones
defaultuser=1000
username=1000
secret=1000
host=dynamic
port=5060
```

Figure 5.5: *SIP extension in Asterisk 1.*

Since there were two Asterisk machines that needed to intercommunicate, before designing the dialplan, SIP and IAX trunks had to be defined between the two servers (Meggelen et al., 2007). This was also done in the sip.conf and iax.conf files. For example, the SIP trunk in Asterisk 1 was configured as follows:

```
[interboxserver2]
type=friend
host=192.168.2.2
context=callfromserver2
canreinvite=no
```

Figure 5.6: *SIP trunk in Asterisk 1.*

The final step was to design the dialplan. A sample configuration for Asterisk 1 is provided:

```
[globals]

[general]
autofallthrough=yes

[default]
include => phones

[incoming_calls]

[phones]
include => internal
include => calltoserver2

[internal]
exten => 1000,1,NoOp()
exten => 1000,n,Answer()
exten => 1000,n,Playback(demo-congratsX5)
exten => 1000,n,Hangup()

exten => 3000,1,NoOp()
exten => 3000,n,Answer()
exten => 3000,n,Playback(demo-congratsX5)
exten => 3000,n,Hangup()

[callfromserver2]
exten => _1XXX,1,NoOp(Call from server2)
exten => _1XXX,n,Dial(SIP/${EXTEN})
exten => _1XXX,n,Playback(demo-congratsX5)
exten => _1XXX,n,Hangup

exten => _3XXX,1,NoOp(Call from server2)
exten => _3XXX,n,Dial(IAX2/${EXTEN})
exten => _3XXX,n,Playback(demo-congratsX5)
exten => _3XXX,n,Hangup

[calltoserver2]
exten => _2XXX,1,NoOp(Call to server2)
exten => _2XXX,n,Dial(SIP/interboxserver2/${EXTEN})
exten => _2XXX,n,Hangup

exten => _4XXX,1,NoOp(Call to server2)
exten => _4XXX,n,Dial(IAX2/interboxserver2iax/${EXTEN})
exten => _4XXX,n,Hangup
```

Figure 5.7: *Dialplan configuration in Asterisk 1.*

For SIPS/SRTP and SIP/ZRTP there were some more actions that had to be taken. Since SIPS implies SIP over TLS, as it was seen earlier, digital certificates needed to be prepared. Furthermore, a few modifications were required to be made in the SIP configuration file and in the dialplan in order to activate the SRTP protocol. ZRTP on the other hand does not require any digital certificates. As it was analysed in the literature review, it introduces the concept of authentication within the media channel (Bresciani and Butterfield, 2009). Unfortunately, though none of the above worked. In respect to

SIPS/SRTP the calls were carried out without any encryption. As for ZRTP, the enrolment procedure between the two Asterisk servers was constantly failing. Unfortunately, since both of the solutions are not supported officially, the current documentation and troubleshooting information are extremely limited. However, the interested reader can refer to the Appendix C in order to study and try the configurations for these security mechanisms.

In addition to the above a note needs to be made for IAX. In the literature review the following features of IAX were discussed: Its capability for multiplexing many calls at the same stream (referred as trunking mode) and its integrated support for MD5, RSA and AES-128-bit (Boucadair, 2009). Regarding the latter, since SIPS/SRTP and SIP/ZRTP failed to work, it was decided to run the experiments only by using MD5, instead of RSA authentication and of course the AES-128bit encryption. The configuration of IAX trunking mode and encryption can be seen in Figure 5.8.

```
[interboxserver2iax]
type=friend
username=interboxserver1iax
secret=welcome
auth=md5
encryption=yes
host=192.168.2.2
context=callfromserver2
trunk=yes
```

Figure 5.8: *IAX in trunking and encryption mode.*

After configuring the two IP PBX servers, it was possible to generate calls between them. This was achieved by using Asterisk's powerful CLI (Command Line Interface). Asterisk can initiate calls from its CLI and connect two extensions (Meggelen et al., 2007). In order to automate the procedure simple bash scripts were created, like the one in Figure 5.9.

```
#!/bin/bash

for i in {1..120}
do
echo -n "$i"
asterisk -rx "channel originate local/3000@phones extension 4000@phones"
done
```

Figure 5.9: *Call generator bash script.*

5.4 Monitoring system configuration

The last thing that had to be configured was the monitoring system. The main component as it is already mentioned was Argus. Argus was compiled and installed in both of the Asterisk systems and run as a usual Unix service in daemon mode capturing network flows. NetFlow was also configured in the core router R2 in order to measure the link utilisation in the WAN cloud. Argus was functioning as the NetFlow collector. Unfortunately, NetFlow did not report correctly the traffic flows that were traversing through the WAN cloud, so the bandwidth usage measurements were based on the Argus flow records that were captured by the NIC cards on the Asterisk servers. Again the interested reader can check the NetFlow configuration in the Appendix B. It is important to remind that clock synchronisation is critical for accurate passive measurements. The literature review highlighted that requirement and mentioned NTP as one of the most common ways to succeed this (John et al., 2010). Therefore, core router R2 was selected to act as the NTP server for the testbed. For that to be possible the command *ntp master 5* had to be applied in the router using its CLI in the global configuration mode. Finally, Fping was used as the active probe from within Asterisk 1. In that way ICMP echo requests were sent to Asterisk 2 and captured by Argus in order to measure the latency. Fping was run without any special arguments. It has to be noted that the reader can refer to Appendix A for some sample screenshots which demonstrate the calls that were recorded by the monitoring system during the experiments.

5.5 Conclusions

This chapter discussed the implementation of the novel evaluation framework for the evaluation of different VoIP protocols and mechanisms that pose an impact on their performance. The use of global open standard technologies and tools made the implementation of this framework possible for achieving the aim of evaluating different VoIP architectures. The complications that were faced during this phase did not have any major impact, other than not being able to evaluate the security solutions for SIP/RTP. Thus, useful results were still extracted. The next chapter presents and analyses these results after the completion of the experiments.

CHAPTER 6
Evaluation

6.1 Introduction

This chapter presents the results of the tests which were carried out by using the novel experimental framework that was analysed in the Design and Implementation chapters. It should be reminded that the experiments were run for SIP/RTP and IAX under different conditions and loads. Specifically, Asterisk was used in order to generate 30, 50, 90 and 120 concurrent calls. Regarding IAX, the tests were performed with and without the use of its security mechanisms in the trunking and non-trunking mode. Unfortunately, configuring SIP security mechanisms failed, thus there are no results for SIPS/SRTP and SIP/ZRTP. In addition, the tests were run once for the OSPFv2 cloud and once for the MPLS cloud. It is important to highlight that there are very few comparative studies on SIP and IAX performance as it was seen in the literature review. Among these Montoro and Casilari (2009) focused only on bandwidth, CPU and memory utilisation of these protocols, while the current study examined also latency, jitter and packet loss.

6.2 One-way delay results

One-way delay calculations were based on the active probe that was pinging Asterisk 2. There are two possibilities in order to extract the results for this metric. The first one is to study the timestamps and durations of the ICMP echo requests and responses on the two VoIP servers. This method requires a strict clock synchronisation to the μsec between the machines. NTP was used for that, like it was already discussed, but this fine-grained clock synchronisation was not achieved as it is shown on Figure 6.1.

```
$ ra -u -r argus1.out -s stime dur -N 2 - echo
     Start time       Duration
 1279453866.858232   0.000405
 1279453867.865911   0.000405

$ ra -u -r argus2.out -s stime dur -N 2 - echo
     Start time       Duration
 1279453866.758261   0.000012
 1279453867.765945   0.000012
```

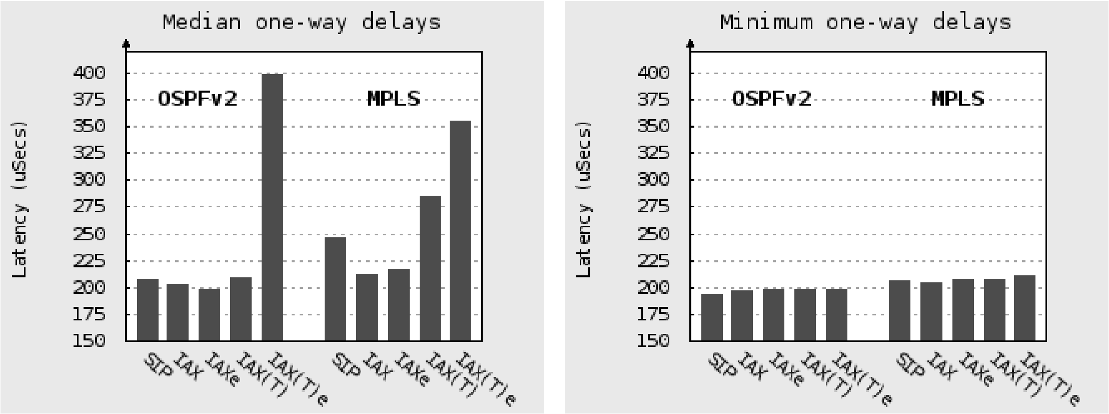
Figure 6.1: *Clock synchronisation.*

A difference of 100 μsecs in the clocks of the two Asterisk machines was observed in the start of the experiments. Notably, this difference was increased during time:

```
$ ra -u -r argus1.out -s stime dur -N 2 - echo
     Start time      Duration
 1279556029.374755   0.000424
 1279556030.382569   0.000421

$ ra -u -r argus2.out -s stime dur -N 2 - echo
     Start time      Duration
 1279556029.180989   0.000012
 1279556030.188809   0.000012
```

Figure 6.2: *Clock synchronisation over time.*

Therefore, the above method lacks accuracy. A safer method is to use the ICMP echo RTT values recorded on Asterisk 1 which does not depend on clock synchronisation. Figure 6.3 shows the one-way delay results that were calculated based on this approach.



(a) Median one-way delays

(b) Minimum one-way delays

Figure 6.3: *One-way delay results.*

The median and minimum one-way delay values are provided for comparison in Figures 6.3a and 6.3b respectively. The first observation here is that when using MPLS there was an increase in the one-way delay times. Imposing the MPLS labels between the IP and UDP headers introduced a small processing overhead for the routers in the WAN cloud. The second observation is that SIP and IAX in trunking and non-trunking mode are comparable. There was however an increase in the median one-way delays when IAX used both the AES encryption and the trunking mode. Probably, random delayed ICMP responses caused these increases. Nevertheless, they do not seem to reflect the real latencies, since in the rest of the cases the median values were closer to the minimum values. In any case though SIP and IAX proved to be extremely fast and way under the ideal 100 ms end-to-end delay (Karapantazis and Pavlidou, 2009). The two Asterisk servers were only 5 hops away one from the other, so delays of under 200 μsecs were succeeded. That is the reason why MPLS did not have a great impact on the performance of the VoIP flows. The same stands for the use of encryption in IAX which did not affect greatly latency, like Callegari et al. (2009) suggested. It must be noted that one-way delays were generally stable during each experiment and independent of the VoIP traffic load.

6.3 Jitter results

The next metric that was examined was the delay variation or jitter. Jitter can dramatically degrade VoIP calls if it is not constant throughout the communication of the involved parties. The results showed that jitter was the same for both the OSPFv2 cloud and the MPLS cloud. The next figure presents the jitter for SIP.

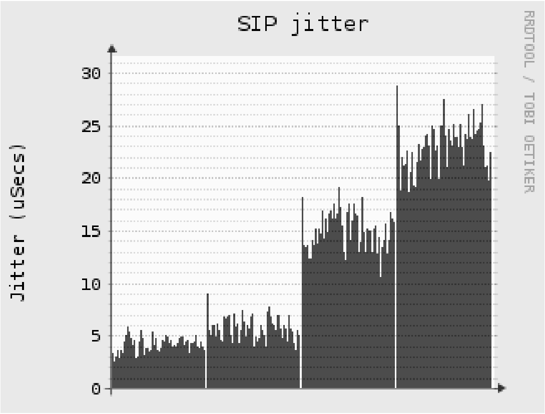
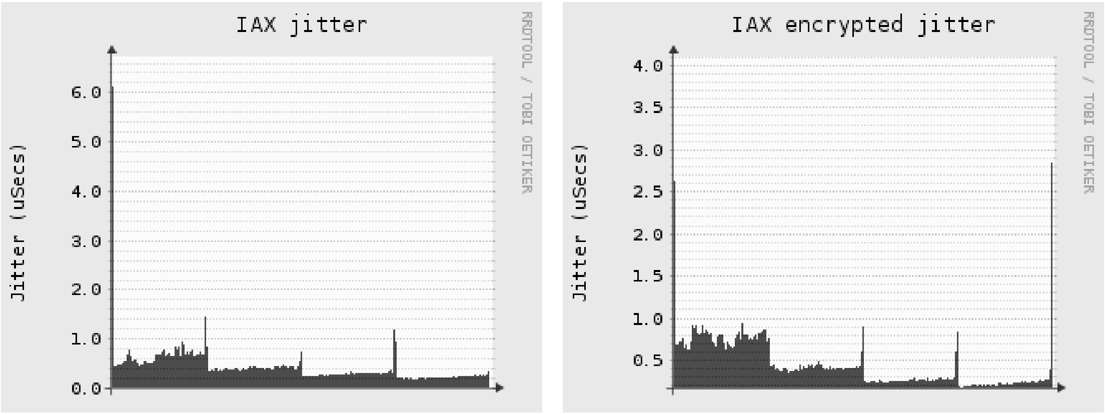


Figure 6.4: *SIP jitter results.*

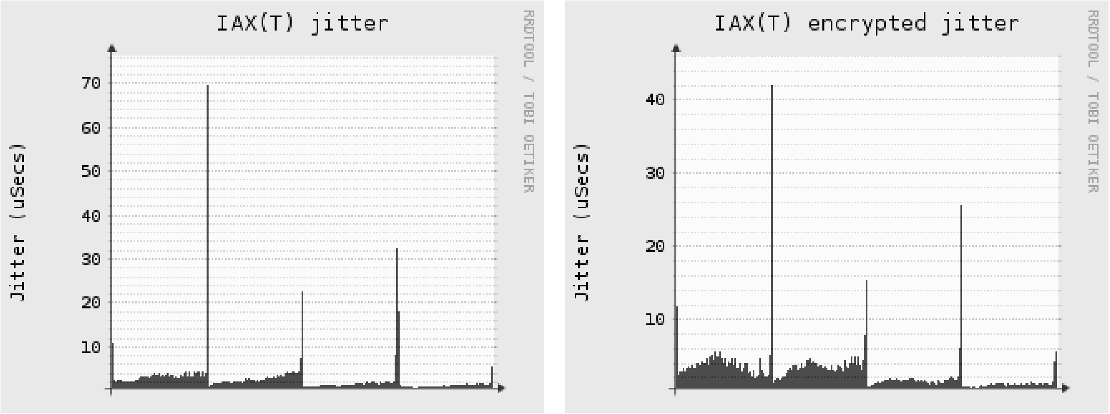
It can be observed that as the number of SIP calls increases, jitter also increases, but it does not fluctuate dramatically. It can also be seen that during all the experiments SIP jitter was very low. The highest value that was noticed was below 30 μsecs which certainly satisfies the requirement of 30 ms maximum jitter for exceptional voice quality (Karapantazis and Pavlidou, 2009). IAX performed even better as it is depicted in the next figures.



(a) IAX jitter

(b) IAX encrypted jitter

Figure 6.5: *IAX without trunking jitter results.*



(a) IAX(T) jitter

(b) IAX(T) encrypted jitter

Figure 6.6: *IAX with trunking jitter results.*

When calls were not multiplexed IAX achieved values of under 1 μsec without being affected by the number of simultaneous calls. For the trunking mode that was about 5 μsecs. The use of encryption practically did not pose any impact. A final note has to be made regarding the spikes that are present in all of the IAX jitter graphs. These are observed during the setup and tear down of the calls.

6.4 Packet loss

The third metric that was measured is relevant to packet loss. Surprisingly packet loss was not faced at any of the tests. For example for SIP in OSPFv2, this can be confirmed in the following figure. The number of packets and bytes at both directions of the flows were exactly the same.

```
$ racluster -r argus1.out -w - - rtp | ra -N 5
  SrcAddr    Sport    DstAddr  Dport SrcPkts DstPkts SrcBytes DstBytes
192.168.1.2.24244 192.168.2.2.10188   3522    6989   753708  1495646
192.168.1.2.18324 192.168.2.2.18372   6970    6989  1491580  1495646
192.168.1.2.18368 192.168.2.2.7238    6974    6989  1492436  1495646
192.168.1.2.30840 192.168.2.2.25546     27    6989     5778  1495646
192.168.1.2.12792 192.168.2.2.18964   6970    6989  1491580  1495646

$ racluster -r argus2.out -w - - rtp | ra -N 5
  SrcAddr    Sport    DstAddr  Dport SrcPkts DstPkts SrcBytes DstBytes
192.168.1.2.24244 192.168.2.2.10188   3522    6989   753708  1495646
192.168.1.2.18324 192.168.2.2.18372   6970    6989  1491580  1495646
192.168.1.2.18368 192.168.2.2.7238    6974    6989  1492436  1495646
192.168.1.2.30840 192.168.2.2.25546     27    6989     5778  1495646
192.168.1.2.12792 192.168.2.2.18964   6970    6989  1491580  1495646
```

Figure 6.7: *Packets and bytes exchanged.*

One can thus conclude that the involved devices were able to process all the transactions without dropping packets due to network congestion or other processing overload.

6.5 Bandwidth utilisation results

Bandwidth utilisation is the last metric related to VoIP. As it was mentioned in the implementation chapter NetFlow failed to work properly, so the measured bandwidth utilisation is based on the captured flows by Argus using the NICs of the servers. It should be mentioned that the results presented in the next figures are in respect to the upstream, but they are the same for the downstream as well, since the voice flows in both directions were identical.

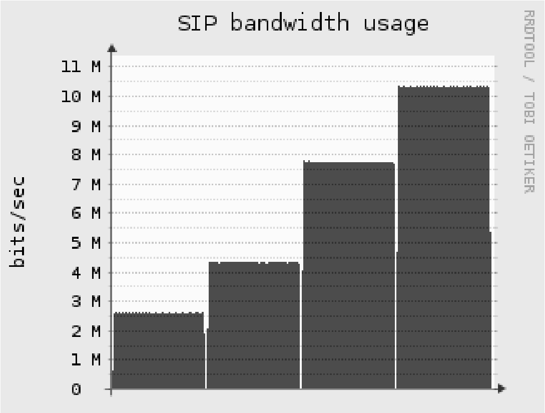
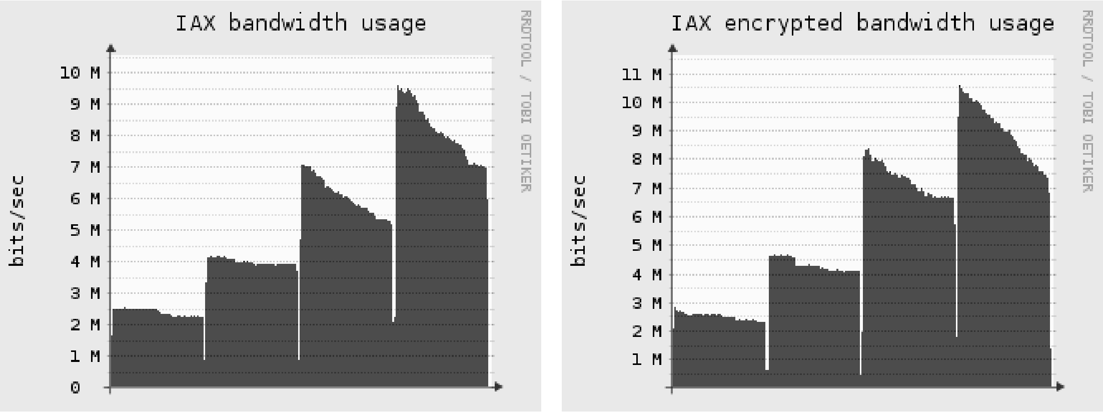


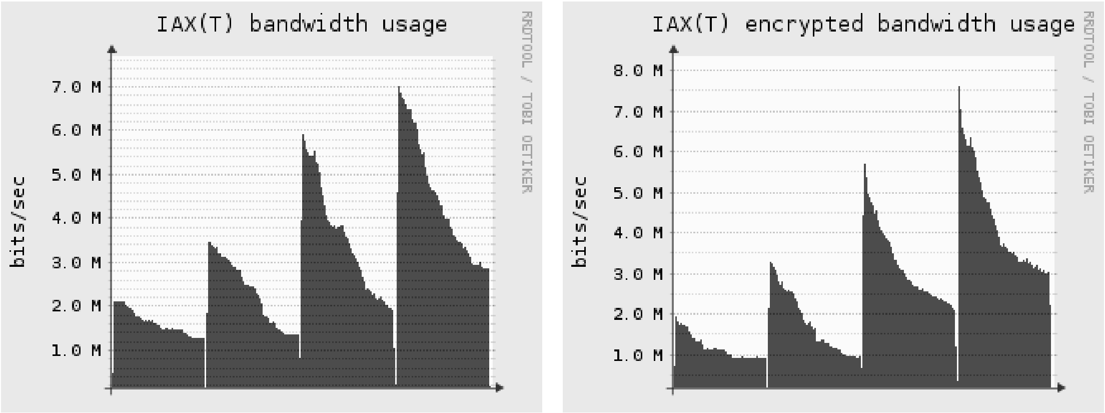
Figure 6.8: *SIP bandwidth utilisation.*

The SIP bandwidth utilisation results are in total agreement with those measured by (Montoro and Casilari, 2009). They also correspond to the theoretical value which is 85,6 Kbps per call. Notably, a maximum of 10,3 Mbps was required for the 120 calls. It should also be mentioned that SIP scaled directly with the number of calls. This is not the case for IAX which tried to use bandwidth more efficiently. This is shown in Figures 6.9 and 6.10.



(a) IAX bandwidth utilisation (b) IAX encrypted bandwidth utilisation

Figure 6.9: *IAX without trunking bandwidth utilisation results.*



(a) IAX bandwidth utilisation

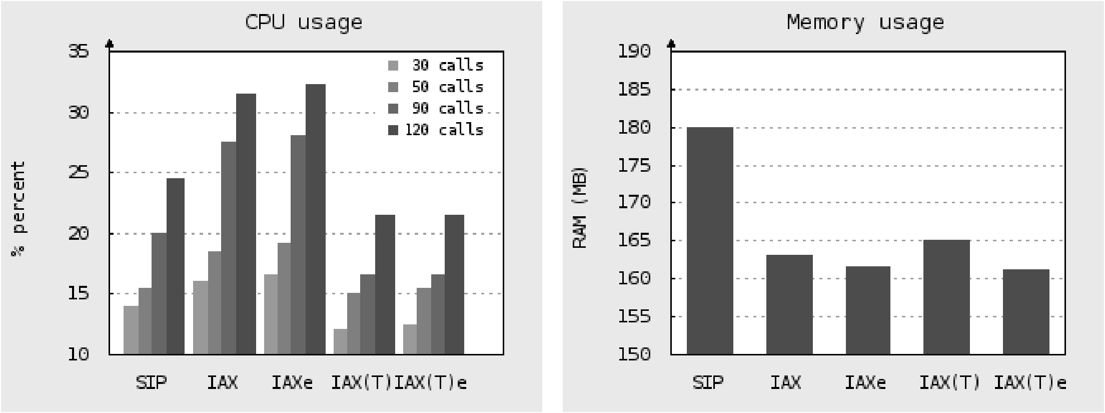
(b) IAX encrypted bandwidth utilisation

Figure 6.10: *IAX with trunking bandwidth utilisation results.*

The interesting fact about IAX is that it did not have the linear behaviour of SIP and this was more evident when calls were multiplexed in the trunking mode. This was not observed by Montoro and Casilari (2009). Average bandwidth consumption per call was about 75 Kbps which differs from the theoretical 82,4 Kbps. For the encrypted IAX calls the above figures are 80 Kbps and 87,2 Kbps respectively. Significant bandwidth savings were made when calls were multiplexed like it was highlighted by Karapantazis and Pavlidou (2009) and proved by Montoro and Casilari (2009). In this case it was not easy to calculate the average bandwidth consumption per call, since this metric depends on the number of calls being multiplexed. It can be seen though that when trunking was enabled a maximum of about 7 Mbps was required even when encryption was used as opposed to 9.5 Mbps without encryption and 10.5 Mbps with encryption in the non-trunking mode. Montoro and Casilari (2009) did not measure bandwidth utilisation of encrypted IAX calls, so there cannot be a comparison here that either confirms or not their measurements. In any case it can be concluded that AES-128 security mechanism of IAX did not introduce a big overhead.

6.6 CPU and memory utilisation results

Finally, CPU and memory utilisation were measured. The results are presented in Figure 6.11. It can be seen that IAX had a bigger processing overhead than SIP, unless the trunking mode was enabled. It can also be seen that while the number of simultaneous calls was increased then also the CPU usage was increased. The maximum values for the 120 concurrent calls were 24.5% for SIP and 31.5% and 32.3% for IAX without and with encryption and 21.5% in the trunking mode in both cases. The opposite stands for the memory usage which was stable throughout the experiments. However, it can be observed that SIP used about 15MB RAM more memory than IAX, but this essentially did not stress the system. Regarding the CPU usage the results agree with those presented by Montoro and Casilari (2009), but with respect to memory usage the authors stated that they could not come to an accurate conclusion. Finally, the use of encryption had a minor impact when IAX was on the non-trunking mode and no impact at all when in the trunking mode. This confirms the findings of Callegari et al. (2009) as presented in the literature review. The authors stated that VoIP specific security solutions should be preferred over more generic ones.



(a) CPU utilisation

(b) Memory utilisation

Figure 6.11: *CPU and memory utilisation results.*

6.7 Conclusions

The results produced from the novel evaluation framework which was presented in the Design and Implementation chapters revealed interesting information regarding in-band and out-of-band VoIP signalling protocols. Specifically, it was proved that IAX outperforms SIP in respect to jitter and bandwidth utilisation especially when in the trunking (multiplexing) mode, but in respect to one-way delay they are comparable and practically equally fast. Packet loss did not pose a problem for both of the protocols, since it was not faced at all during the experiments. SIP and IAX were also similar regarding the CPU and memory usage, with SIP having less processing overhead and more memory overhead compared to IAX in the non-trunking mode. On the other hand when IAX utilised its trunking mode then it outperformed SIP in both the CPU and memory usage, even when its embedded AES-128 encryption mechanism was used which generally did not have a great impact on all of the experiments and metrics. It can be concluded that IAX is highly optimised for VoIP communications and thus it should be considered when deploying a VoIP infrastructure.

In addition to the above, it was seen that MPLS in its simple form, which does not utilise its traffic engineering and QoS mechanisms, is not useful. The processors of the modern routers are extremely powerful and they are able to perform the routing table lookups in a very fast way. That was a big problem for the older routers and one of the reasons that MPLS was developed. However, like it was proved by using the novel evaluation framework this is not an issue any more and thus the MPLS labels imposed between Layer 2 and Layer 3 headers introduce a small overhead. Therefore, MPLS should be considered mainly for its traffic engineering and QoS mechanisms that were described in the literature review.

Generally, it can be concluded that the novel evaluation framework produced credible and solid results that confirmed the findings of other researchers mentioned in the literature review. Its main limitation is the clock synchronisation that was not able to give more accurate results regarding one-way delays. A good picture though of the whole system was possible to be extracted. The next chapter discusses this and the overall conclusions of this MSc project.

CHAPTER 7

Conclusions

7.1 Introduction

The aim of this project was to evaluate different VoIP signalling protocols in terms of their key performance metrics and the impact of security and packet transport mechanisms on them by using a novel evaluation framework. The previous chapter showed that the designed evaluation framework produced valid results therefore meeting this aim.

This chapter discusses how the objectives of this project were met along with the final conclusions regarding the in-band and out-of-band VoIP signalling protocols and the factors that affect their performance. In addition, a critical analysis assess the work undertaken for the purpose of this project. Finally, future work is proposed in order to explore further possibilities in the field of VoIP by using the designed evaluation framework.

7.2 Meeting the objectives

The first chapter determined the following objectives:

1. Critically review the main VoIP protocols and their security mechanisms, the different network protocols which are used in order to transport voice and the main methods for performing traffic measurements.
2. Design a flexible and applicable to real-life networks novel evaluation framework based on the findings of the literature review in order to evaluate VoIP protocols and the security and packet transport mechanisms that can affect their performance by utilising open standard methods and tools.
3. Implement the framework by setting up the selected methods and tools and evaluate the VoIP protocols based on the results produced by carrying out a number of experiments as set in the design by using this framework.

The first objective was met by carrying out an in-depth literature review in the field of VoIP. In particular the architectures of in-band and out-of-band VoIP signalling protocols were critically analysed together with the various existing mechanisms that provide protection to the VoIP calls. Performance issues of these security mechanisms were also discussed. Furthermore, different packet transport techniques that relay voice were studied. In addition, the main methods, frameworks and tools that perform traffic measurements were assessed. Finally, the Asterisk open source IP PBX was reviewed. The literature review concluded that there is a need for a flexible experimental framework that can be used to evaluate IP telephony. In order to produce this framework the following requirements were determined:

1. The VoIP protocols under scrutiny and their key metrics.
2. The security standards and underlying network mechanisms for evaluating their impact on VoIP.
3. The appropriate monitoring system for performing traffic measurements and extracting the results.

The second objective was met by designing a novel evaluation framework based on the above requirements and findings of the literature review. The framework was composed of different components. The first component of the framework was a small network consisted of two sites interconnected through a WAN cloud. In this way a real-life network was simulated. The second component was the VoIP call generator. Finally, the third component was the network monitoring system that performed traffic measurements. The choices for these components were justified as all of them were based on open standard methods and tools which were critically examined in the literature review.

The third objective was met by implementing the experimental framework and using it in order to evaluate the VoIP protocols by analysing the results produced after performing various experiments as set in the design phase. The implementation involved the configuration of the networking devices, VoIP call generator and monitoring system. The main issue in this phase was failing to configure properly the VoIP call generator in respect to the security mechanisms of SIP/RTP. The second issue was failing to configure one of the parts that composed the monitoring system which was NetFlow in one of the routers of the WAN cloud. Another issue with the implementation was the clock synchronisation that had to be ultra precise to the μsec in order to get more accurate results regarding one-way delays, but this did not happen. However, none of these issues hindered the successful realisation of the experiments. Valid results were still produced that helped to achieve the aim of this project. The conclusions of these results are summarised in the next section.

7.3 Conclusions

After performing the experiments by using the novel experimental framework, this study proved that IAX is highly optimised for VoIP communications. IAX showed exceptional performance especially when its trunking mode was employed. Of special significance is the fact that when in that mode bandwidth consumption and CPU utilisation were extremely reduced compared to SIP. These results agree with the ones presented by Montoro and Casilari (2009). Furthermore, the use of AES-128 encryption embedded in IAX did not pose any major impact. Therefore, VoIP specific security mechanisms should be used whenever is possible, since they introduce a minor overhead as suggested by Callegari et al. (2009).

However, IAX has not gained great popularity in the VoIP market (Karapantazis and Pavlidou, 2009). On the contrary, SIP dominates the world of packet telephony and it is no question why it was chosen as the core VoIP signalling protocol by 3GPP for its IMS (Basicevic et al., 2008). Its generic design allows it to carry tasks that are no limited to VoIP (for example, instant messaging). It is thus left to the VoIP engineer to decide the protocol of his preference in new VoIP deployments. The opposite stands for the use or not of MPLS. The results proved that when its traffic engineering and QoS capabilities are not utilised then MPLS is outperformed by OSPF just by a fraction of a msec. Therefore, using MPLS is meaningful only for QoS reasons.

7.4 Critical analysis

The aim and objectives of this project have been achieved, therefore the final remarks of the work undertaken have to be made. The first remark is relevant to the evaluation of the different VoIP protocols and various parameters that affect their performance as stated in the aim. This project presented the notion of in-band and out-of-band VoIP signalling protocols. In particular, SIP was evaluated against IAX. The literature review showed that there is limited bibliography on this field. Among other researchers Abbasi et al. (2005) and Montoro and Casilari (2009) studied the aforementioned protocols. However, they defined different metrics and followed heterogeneous methods in order to carry out their experiments. Between the two, the current study is closer to Montoro and Casilari (2009), but went further by examining all the key VoIP metrics as defined by Goode (2002) and Karapantazis and Pavlidou (2009). Furthermore, two of the parameters that can affect the performance of the VoIP signalling protocols were also examined: Security mechanisms and packet transport mechanisms.

The above would not be possible without the novel experimental framework that was designed specifically in order to achieve the aim of this project. The novelty of this framework stems from the fact that standard methods and tools were employed for its design and implementation. All of the components that were used, including the Asterisk IP PBX, OSPF, MPLS, NetFlow/IPFIX and Argus flow monitoring system define global open standards in the field of VoIP and networking. Therefore, they are easily accessible and affordable to anyone. For example, in this study Cisco routers were utilised, but if someone has no access to this expensive equipment, then it can easily turn a Linux box into a router by using the OSPF routing protocol.

The major limitation of this framework was the clock synchronisation, which is one of the most significant issues as it was pointed out by John et al. (2010). Despite the use of NTP a fine-grained clock synchronisation was not achieved and this is due to the chip electronic parts of the PCs that cannot provide a high level of accuracy which is required for extremely detailed latency measurements. The problem was overcome partially with the use of the fping active probe. Otherwise, extremely costly monitoring cards or GPS equipment should be acquired. Generally, the field of network traffic measurements is huge. In this study only a few of the possibilities that can be achieved with NetFlow/IPFIX and Argus were presented. Especially for the latter McRee (2007) stated that “it is easy to use but hard to master”.

7.5 Future work

The current study covered only a few aspects of VoIP telephony. Using the designed novel evaluation framework further parameters that can affect the quality of VoIP services can be examined. MPLS with its impressive traffic engineering and QoS features was presented, therefore the future researcher can explore the numerous possibilities in the field of load balancing, QoS, and so on. Again in respect to the routing issues further studies can be carried out by examining other IGP protocols against their impact on VoIP in comparison to OSPF such as IS-IS (Intermediate System To Intermediate System) or it would be possible to study evolving technologies like IPv6 that is expected to replace IPv4.

Apart from the above issues which are relevant to packet transport mechanisms, the future researchers could investigate the various voice codecs and their impact on the key VoIP metrics (one-way delay, jitter, packet loss, bandwidth utilisation). The sound quality offered by VoIP is not only affected by the signalling protocols themselves, but also by the algorithms used in order to digitise voice. Usually high quality voice codecs require more bandwidth, but this comes at the expense of lower network efficiency, therefore the presented evaluation framework could be used in order to measure these overheads. Finally, the Asterisk VoIP platform could be evaluated in terms of security (for example DoS or Man-in-the-Middle attacks).

References

Abbasi, T., Prasad, S., Seddigh, N., and Lambadaris, I. (2005). A comparative study of the SIP and IAX VoIP protocols. In *Proceedings of the 2005 Canadian Conference on Electrical and Computer Engineering*, pages 179–183, Saskatoon, Canada.

Agrawal, S., Kanthi, C. N., Naidu, K. V. M., Ramamirtham, J., Rastogi, R., Satkin, S., and Srinivasan, A. (2007). Monitoring infrastructure for converged networks and services. *Bell Labs Technical Journal*, 12(2):63–77.

Agrawal, S., Narayan, P., Ramamirtham, J., Rastogi, R., Smith, M., Swanson, K., and Thottan, M. (2006a). VoIP service quality monitoring using active and passive probes. In *Proceedings of the 1st International Conference on Communication System Software and Middleware*, pages 1–10, Delhi, India.

Agrawal, S., Ramamirtham, J., and Rastogi, R. (2006b). Design of active and passive probes for VoIP service quality monitoring. In *Proceedings of the 12th International Telecommunications Network Strategy and Planning Symposium*, pages 1–6, New Delhi, India.

Ahmed, M. and Mansor, A. M. (2008). CPU dimensioning on performance of asterisk VoIP PBX. In *Proceedings of the 11th Communications and Networking Simulation Symposium*, pages 139–146, Ottawa, Canada. ACM.

Aire, E., Maharaj, B., and Linde, L. (2004). Implementation considerations in a SIP based secure voice over IP network. In *Proceedings of the 7th AFRICON Conference in Africa*, volume 1, pages 167–172, Gaborone, Botswana.

Alcock, S., Lawson, D., and Nelson, R. (2007). Extracting application objects from TCP packet traces. In *Proceedings of the 2007 Australasian Telecommunication Networks and Applications Conference*, pages 151–156, Christchurch, New Zealand.

Alexander, A., Wijesinha, A., and Karne, R. (2009). An evaluation of secure Real-Time transport protocol (SRTP) performance for VoIP. In *Proceedings of the 3rd International Conference on Network and System Security*, pages 95–101, Gold Coast, Australia.

Anderson, S., Niccolini, S., and Hogrefe, D. (2009). SIPFIX: a scheme for distributed SIP monitoring. In *Proceedings of the 2009 International Symposium on Integrated Network Management*, pages 382–389, Long Island, New York, USA.

Andersson, L. and Bryant, S. (2008). The IETF multiprotocol label switching standard: The MPLS transport profile case. *Internet Computing, IEEE*, 12(4):69–73.

Baharlooei, Z. and Hashemi, M. (2009). A low cost VoIP architecture for private networks. In *Proceedings of the 2009 International Conference on Future Networks*, pages 8–12, Bangkok, Thailand.

Barakovic, J., Bajric, H., and Husic, A. (2006). Multimedia traffic analysis of MPLS and non-MPLS network. In *48th International Symposium ELMAR-2006 focused on Multimedia Signal Processing and Communications*, pages 285–288, Zadar, Croatia.

Barbieri, R., Bruschi, D., and Rosti, E. (2002). Voice over IPsec: analysis and solutions. In *Proceedings of the 18th Annual Computer Security Applications Conference*, pages 261–270, Las Vegas, Nevada, USA.

Basicevic, I., Popovic, M., and Kukolj, D. (2008). Comparison of SIP and H.323 protocols. In *Proceedings of the 3rd International Conference on Digital Telecommunications*, pages 162–167, Bucharest, Romania.

Bassil, C., Serhrouchni, A., and Rouhana, N. (2005). Critical analysis and new perspective for securing voice networks. In Lorenz, P. and Dini, P., editors, *Networking - ICN 2005*, volume 3421 of *Lecture Notes in Computer Science*, pages 810–818. Springer Berlin / Heidelberg.

Boucadair, M. (2009). *Inter-Asterisk Exchange (IAX): Deployment Scenarios in SIP-Enabled Networks*. Wiley-Blackwell, West Sussex, UK.

Bresciani, R. and Butterfield, A. (2009). A formal security proof for the ZRTP protocol. In *Proceedings of the 4th International Conference for Internet Technology and Secured Transactions*, pages 1–6, London, UK.

Bross, J. and Meinel, C. (2008). Can VoIP live up to the QoS standards of traditional wireline telephony? In *Proceedings of the 4th Advanced International Conference on Telecommunications*, pages 126–132, Athens, Greece.

Brownlee, N. (1998). Network management and realtime traffic flow measurement. *Journal of Network and Systems Management*, 6:223–228.

Brownlee, N. (2001a). IP streams, flows and torrents: Measuring stream distributions in real time. In *Proceedings of the 2001 Workshop on Passive and Active Measurements*, Amsterdam, Holland.

Brownlee, N. (2001b). Using NeTraMet for production traffic measurement. In *Proceedings of the 2001 International Symposium on Integrated Network Management*, pages 213–226, Seattle, Washington, USA.

Brownlee, N., Christ, P., Jaehnert, J., Liang, Y., Srinivasan, K., and Zhou, J. (2003). MobyDick FlowVis - using NeTraMet for distributed protocol analysis in a 4G network environment. In *Proceedings of the 3rd IEEE Workshop on IP Operations and Management*, pages 55–60, Kansas City, Missouri, USA.

Brownlee, N. and Fulton, R. (2000). Kawaihiko and the third-quartile day [traffic management]. *Communications Magazine, IEEE*, 38(8):162–168.

Callegari, C., Garroppo, R. G., Giordano, S., and Pagano, M. (2009). Security and delay issues in SIP systems. *International Journal of Communication Systems*, 22(8):1023–1044.

Cao, F. and Malik, S. (2006). Vulnerability analysis and best practices for adopting IP telephony in critical infrastructure sectors. *Communications Magazine, IEEE*, 44(4):138–145.

Chaudet, C., Fleury, E., Lassous, I. G., Rivano, H., and Voge, M. (2005). Optimal positioning of active and passive monitoring devices. In *Proceedings of the 2005 ACM conference on Emerging Network Experiment and Technology*, pages 71–82, Toulouse, France. ACM.

Chava, K. and How, J. (2007). Integration of open source and enterprise IP PBXs. In *Proceedings of the 3rd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*, pages 1–6, Orlando, FL, USA.

Chin, J. (2004). *Cisco Frame Relay Solutions Guide: Implement Frame Relay Solutions on Cisco Networks with This Definitive Resource*. Cisco Press, Indianapolis, IN, USA, 2nd edition.

Choi, B. and Bhattacharyya, S. (2005). Observations on cisco sampled NetFlow. *SIGMETRICS Perform. Eval. Rev.*, 33(3):18–23.

Choi, B. and Zhang, Z. (2006). Adaptive random sampling for traffic volume measurement. *Telecommunication Systems*, 34(1-2):71–80.

Chong, H. M. and Matthews, H. (2004). Comparative analysis of traditional telephone and voice-over-Internet protocol (VoIP) systems. In *Proceedings of the 2004 IEEE International Symposium on Electronics and the Environment*, pages 106–111, Phoenix , AR, USA.

Claffy, K., Fomenkov, M., Katz-Bassett, E., Beverly, R., Cox, B. A., and Luckie, M. (2009). The workshop on active internet measurements (AIMS) report. *SIGCOMM Computer Communication Review*, 39(5):32–36.

Dalgic, I. and Fang, H. (1999). Comparison of H.323 and SIP for IP telephony signaling. volume 3845, pages 106–122. SPIE.

Davoli, F., Meyer, N., Pugliese, R., and Zappatore, S., editors (2009). *Grid Enabled Remote Instrumentation*. Springer US, Boston, MA, USA.

De, B., Joshi, P., Sahdev, V., and Callahan, D. (2003). End-to-end voice over IP testing and the effect of QoS on signaling. In *Proceedings of the 35th Southeastern Symposium on System Theory*, pages 142–147, Virginia, Morgantown, WV, USA.

Drago, I., R. R. Barbosa, R., Sadre, R., Pras, A., and Schönwälder, J. (2010). Report of the second workshop on the usage of NetFlow/IPFIX in Network Management. *Journal of Network and Systems Management*, pages 1–7.

Estan, C., Keys, K., Moore, D., and Varghese, G. (2004). Building a better NetFlow. In *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 245–256, Portland, Oregon, USA. ACM.

Fatemipour, F. and Yaghmaee, M. (2007). Design and implementation of a monitoring system based on IPFIX protocol. In *Proceedings of the 3rd Advanced International Conference on Telecommunications*, page 22, Mauritius.

Finamore, A., Mellia, M., Meo, M., Munafò, M., and Rossi, D. (2010). Live traffic monitoring with tstat: Capabilities and experiences. In Osipov, E., Kassler, A., Bohnert, T., and Masip-Bruin, X., editors, *Wired/Wireless Internet Communications*, volume 6074 of *Lecture Notes in Computer Science*, pages 290–301. Springer Berlin / Heidelberg.

Foukarakis, M., Antoniades, D., Antonatos, S., and Markatos, E. P. (2007). Flexible and high-performance anonymization of NetFlow records using anontool. In *Proceedings of the 3rd International Conference on Security and Privacy in Communications Networks*, pages 33–38, Nice, France.

Francis-Cobley, P. P. and Coward, A. D. (2004). Voice over IP versus Voice over Frame Relay. *International Journal of Network Management*, 14(4):223–230.

Ganguly, S. and Bhatnagar, S. (2008). *VoIP: Wireless, P2P and New Enterprise Voice over IP*. Wiley, West Sussex, UK.

Glasmann, J., Kellerer, W., and Muller, H. (2003). Service architectures in H.323 and SIP: a comparison. *Communications Surveys and Tutorials, IEEE*, 5(2):32–47.

Goode, B. (2002). Voice over internet protocol (VoIP). *Proceedings of the IEEE*, 90(9):1495–1517.

Groom, F. M. and Groom, K. M. (2006). *The Basics of Voice Over Internet Protocol*. International Engineering Consortium.

Gupta, P. and McKeown, N. (2001). Algorithms for packet classification. *Network, IEEE*, 15(2):24–32.

Gupta, P. and Shmatikov, V. (2007). Security analysis of Voice-over-IP protocols. In *Proceedings of the 20th IEEE Computer Security Foundations Symposium*, pages 49–63, Venice, Italy.

Hartpence, B. (2007). Curricular response to the real time data and VoIP tidal wave. *Journal of Computing Sciences in Colleges*, 22(6):37–42.

Imran, A. and Qadeer, M. (2009). Conferencing, paging, voice mailing via Asterisk EPBX. In *Proceedings of the International Conference on Computer Engineering and Technology*, volume 1, pages 186–190, Singapore.

Imran, A., Qadeer, M., and Khan, M. (2009). Asterisk VoIP private branch exchange. In *Proceedings of the 2009 International Conference on Multimedia, Signal Processing and Communication Technologies*, pages 217–220, Aligarh, Uttar Pradesh, India.

Iselt, A., Kirstadter, A., Pardigon, A., and Schwabe, T. (2004). Resilient routing using MPLS and ECMP. In *Proceedings of the 2004 Workshop on High Performance Switching and Routing*, pages 345–349, Phoenix, AZ, USA.

Ishibashi, K., Kanazawa, T., Aida, M., and Ishii, H. (2004). Active/passive combination-type performance measurement method using change-of-measure framework. *Computer Communications*, 27(9):868–879.

Jajor, J., Mazurek, C., and Procyk, W. (2009). Grids and networks Monitoring–Practical approach. In *Grid Enabled Remote Instrumentation*, pages 173–186. Springer Berlin / Heidelberg.

Jeliazkova, N., Iliev, L., and Jeliazkov, V. (2006). UPerfsonarUI - a standalone graphical user interface for querying perfSONAR services. In *Proceedings of the IEEE 2006 John Vincent Atanasoff International Symposium on Modern Computing*, pages 77–81, Sofia, Bulgaria.

John, W., Tafvelin, S., and Olovsson, T. (2010). Passive internet measurement: Overview and guidelines based on experiences. *Computer Communications*, 33(5):533–550.

Kalmanek, C. (2002). A retrospective view of ATM. *SIGCOMM Computer Communication Review*, 32(5):13–19.

Kapov, M. and Dlaka, D. (2006). IP telephony network saving capacity due to substitution of PSTN by IP network. In *Proceedings of the 2006 International Conference on Software, Telecommunications and Computer Networks*, pages 336–341, Split - Dubrovnik, Croatia.

Karapantazis, S. and Pavlidou, F. (2009). VoIP: a comprehensive survey on a promising technology. *Computer Networks*, 53(12):2050–2090.

Kasdirin, H. and Rahman, R. A. (2003). The process flow and analysis of voice over ATM in common communication network. In *Proceedings of 4th National Conference on Telecommunication Technology*, pages 250–256, Shah Alam, Malaysia.

Kazemi, N., Wijesinha, A. L., and Karne, R. (2010). Evaluation of IPsec overhead for VoIP using a bare PC. In *Proceedings of the 2nd International Conference on Computer Engineering and Technology*, volume 2, pages 586–589, Chengdu, China.

Kim, C., Shin, S., Ha, S., Yoon, K., and Han, S. (2004). Architecture of end-to-end QoS for VoIP Call Processing in the MPLS network. In Solé-Pareta, J., Smirnov, M., Mieghem, P. V., Domingo-Pascual, J., Monteiro, E., Reichl, P., Stiller, B., and Gibbens, R. J., editors, *Quality of Service in the Emerging Networking Panorama*, volume 3266 of *Lecture Notes in Computer Science*, pages 44–53. Springer Berlin / Heidelberg.

King, A. and Hunt, R. (2000). Protocols and architecture for managing TCP/IP network infrastructures. *Computer Communications*, 23(16):1558–1572.

Kocak, C., Erturk, I., and Ekiz, H. (2009). MPLS over ATM and IP over ATM methods for multimedia applications. *Computer Standards & Interfaces*, 31(1):153–160.

Konstantoulakis, G. and Sloman, M. (2007). Call management policy specification for the asterisk telephone private branch exchange. In *Proceedings of the 8th IEEE International Symposium on Policies for Distributed Systems and Networks*, pages 251–255, Bologna, Italy.

Lee, T. (2005). Multiprotocol laber switching (MPLS) and differentiated services (DS) as quality of service (QoS) solutions. In *Proceedings of the 7th International Conference on Advanced Communication Technology*, volume 2, pages 1039–1043, Phoenix Park, Gangwon-Do, Korea.

Liu, H. and Mouchtaris, P. (2000). Voice over IP signaling: H.323 and beyond. *Communications Magazine, IEEE*, 38(10):142–148.

Loiseau, P., Goncalves, P., Guillier, R., Imbert, M., Kodama, Y., and Primet, P. (2009). Metroflux: A high performance system for analysing flow at very fine-grain. In *Proceedings of the 5th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*, pages 1–9, Washington DC, USA.

Mainwaring, K. (2000). A role for ATM in telephony and IP networks. *Computer Networks*, 34(3):439–454.

Malis, A. (2006). Converged services over MPLS. *Communications Magazine, IEEE*, 44(9):150–156.

Materna, B. (2006). VoIP insecurity. *Communications Engineer*, 4(5):39–42.

McGlone, J., Marshall, A., and Woods, R. (2008). A real-time flow monitor architecture encompassing on-demand monitoring functions. In *Proceedings of the 2008 Network Operations and Management Symposium*, pages 871–874, Salvador da Bahia, Brazil.

McQuerry, S. (2008). *Interconnecting Cisco Network Devices, Part 2 (ICND2): (CCNA Exam 640-802 and ICND Exam 640-816).* Cisco Press, Indianapolis, IN, USA, 3rd edition.

McRee, R. (2007). Argus - auditing network activity. *Information Systems Security Association Journal*, 2007(November):40–42.

Meggelen, J. V., Smith, J., and Madsen, L. (2007). *Asterisk: The Future of Telephony*. O'Reilly Media, Sebastopol, CA, USA, 2nd edition.

Mellia, M., Carpani, A., and Cigno, R. L. (2003). TStat: TCP STatistic and analysis tool. In *Proceedings of the 2nd International Workshop on Quality of Service in Multiservice IP Networks*, pages 145–157, Milano, Italy.

Mellia, M., Cigno, R. L., and Neri, F. (2005). Measuring IP and TCP behavior on edge nodes with tstat. *Computer Networks*, 47(1):1–21.

Mellia, M. and Meo, M. (2010). Measurement of IPTV traffic from an operative network. *European Transactions on Telecommunications*, 21(4):324–336.

Minoli, D. (2002). *Voice Over MPLS : Planning and Designing Networks*. McGraw-Hill Professional, New York, NY, USA.

Molina, M., Chiosi, A., D'Antonio, S., and Ventre, G. (2006). Design principles and algorithms for effective high-speed IP flow monitoring. *Computer Communications*, 29(10):1653–1664.

Montoro, P. and Casilari, E. (2009). A comparative study of VoIP standards with Asterisk. In *Proceedings of the 4th International Conference on Digital Telecommunications*, pages 1–6, Colmar, France.

Mortada, I. and Probst, W. (2001). Internet telephony signaling. *Telematics and Informatics*, 18(2-3):159–194.

Nasr, M. (2003). New technique for improving voice quality degraded by cell loss in ATM networks. In *Proceedings of the 14th International Symposium on Micro-NanoMechatronics and Human Science*, volume 2, pages 674–677 Vol. 2, Nagoya, Japan.

Park, N. (2010). Adoption and use of Computer-Based voice over internet protocol phone service: Toward an integrated model. *Journal of Communication*, 60(1):40–72.

Pasqualini, S., Iselt, A., Kirstädter, A., and Frot, A. (2004a). Mpls protection switching versus ospf rerouting. In Solé-Pareta, J., Smirnov, M., Mieghem, P. V., Domingo-Pascual, J., Monteiro, E., Reichl, P., Stiller, B., and Gibbens, R. J., editors, *Quality of Service in the Emerging Networking Panorama*, volume 3266 of *Lecture Notes in Computer Science*, pages 174–183. Springer Berlin / Heidelberg.

Pasqualini, S., Iselt, A., Kirstädter, A., and Frot, A. (2004b). MPLS protection switching versus OSPF rerouting. In *Quality of Service in the Emerging Networking Panorama*, pages 174–183. Springer Berlin / Heidelberg.

Petraschek, M., Hoeher, T., Jung, O., Hlavacs, H., and Gansterer, W. (2008). Security and usability aspects of Man-in-the-Middle attacks on ZRTP. *Journal of Universal Computer Science*, 14(5):673–692.

Pezaros, D. P., Georgopoulos, K., and Hutchison, D. (2010). High-speed, in-band performance measurement instrumentation for next generation IP networks. *Computer Networks*, In Press, Corrected Proof.

Pilli, E. S., Joshi, R., and Niyogi, R. (2010). Network forensic frameworks: Survey and research challenges. *Digital Investigation*, In Press, Corrected Proof.

Porter, T., Jr, J. K., and Baskin, B. (2006). *Practical VoIP Security*. Syngress, Rockland, MA, USA.

Pras, A., Sadre, R., Sperotto, A., Fioreze, T., Hausheer, D., and Schönwälder, J. (2009). Using NetFlow/IPFIX for network management. *Journal of Network and Systems Management*, 17(4):482–487.

Qadeer, M. and Imran, A. (2008). Asterisk voice exchange: An alternative to conventional EPBX. In *Proceedings of the 2008 International Conference on Computer and Electrical Engineering*, pages 652–656, Phuket Island, Thailand.

Ram, A., DaSilva, L., and Varadarajan, S. (2002). Assessment of voice over IP as a solution for voice over ADSL. In *Proceedings of the 2002 Global Telecommunications Conference*, volume 3, pages 2463–2467 vol.3, Taipei, Taiwan.

Ranganathan, M. K. and Kilmartin, L. (2003). Performance analysis of secure session initiation protocol based VoIP networks. *Computer Communications*, 26(6):552–565.

Ranjbar, A. (2007). *CCNP ONT Official Exam Certification Guide*. Cisco Press, Indianapolis, IN, USA.

Rao, S. S. (2008). Internet telephony in India. *Telematics and Informatics*, 25(2):57–71.

Robert, L., Darko, B., Simon, M., Darko, M., Igor, F., and Nada, C. (2008). To VoIP or not to VoIP, is this really the question now? *Computer Communications*, 31(17):4136–4137.

Rong, B., Lebeau, J., Bennani, M., Kadoch, M., and Elhakeem, A. K. (2005). Traffic aggregation based SIP over MPLS network architecture. In *Proceedings of the 19th International Conference on Advanced Information Networking and Applications*, pages 827–832, Tamkang University, Taiwan. IEEE Computer Society.

Rossi, D. and Mellia, M. (2006). Real-Time TCP/IP analysis with common hardware. In *Proceedings of the 2006 IEEE International Conference on Communications*, volume 2, pages 729–735, Istanbul, Turkey.

Rossi, D. and Valenti, S. (2010). Fine-grained traffic classification with netflow data. In *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, pages 479–483, Caen, France. ACM.

Roughan, M. (2010). A case study of the accuracy of SNMP measurements. *Journal of Electrical and Computer Engineering*, 2010:7.

Rubio-Loyola, J., Sala, D., and Ali, A. (2008). Accurate real-time monitoring of bottlenecks and performance of packet trace collection. In *Proceedings of the 33rd IEEE Conference on Local Computer Networks*, pages 884–891, Montreal, Canada.

Rudinsky, J. (2007). VoIP-PSTN interoperability by Asterisk and SS7 signalling. In Bestak, R., Simak, B., and Kozlowska, E., editors, *Personal Wireless Communications*, volume 245 of *IFIP International Federation for Information Processing*, pages 169–173. Springer Boston.

Salah, K. (2006). On the deployment of VoIP in ethernet networks: methodology and case study. *Computer Communications*, 29(8):1039–1054.

Salsano, S., Veltri, L., and Papalilo, D. (2002). SIP security issues: the SIP authentication procedure and its processing load. *Network, IEEE*, 16(6):38–44.

Samhat, A. and Chahed, T. (2005). IP versus AAL2 for transport in the UMTS radio access network. *Computer Communications*, 28(5):477–484.

Schulzrinne, H. and Rosenberg, J. (1998). A comparison of SIP and H.323 for Internet Telephony. In *The 8th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, Cambridge, UK.

Shah, N. B. and Mohapatra, S. K. (2005). Integrating and managing converged multiservice networks. *Bell Labs Technical Journal*, 10(1):139–156.

Soares, V., Neves, P., and Rodrigues, J. (2008). Past, present and future of IP telephony. In *Proceedings of the 2008 International Conference on Communication Theory, Reliability, and Quality of Service*, pages 19–24, Bucharest, Romania.

Sommers, J., Barford, P., and Willinger, W. (2007). Laboratory-based calibration of available bandwidth estimation tools. *Microprocessors and Microsystems*, 31(4):222–235.

Song, W.-C. and Choi, D.-J. (2006). Experiences in end-to-end performance monitoring on KOREN. In Kim, Y.-T. and Takano, M., editors, *Management of Convergence Networks and Services*, volume 4238 of *Lecture Notes in Computer Science*, pages 383–392. Springer Berlin / Heidelberg.

Spinsante, S., Gambi, E., and Bottegoni, E. (2008). Security solutions in VoIP applications: State of the art and impact on quality. In *Proceedings of the 2008 International Symposium on Consumer Electronics*, pages 1–4, Algarve, Portugal.

Stewart, B. (2007). *CCNP BSCI Official Exam Certification Guide*. Cisco Press, Indianapolis, IN, USA, 4th edition.

Trammell, B., Boschi, E., Mark, L., and Zseby, T. (2007). Requirements for a standardized flow storage solution. In *Proceedings of the 2007 International Symposium on Applications and the Internet*, page 84, Hiroshima, Japan.

Ubik, S. and Zejdl, P. (2008). Passive monitoring of 10 gb/s lines with pc hardware. In *Proceedings of the 2008 Terena Networking Conference*, Brugge, Belgium.

Uzunalioglu, H., Houck, D. J., and Wang, Y. T. (2006). Call admission control for voice over IP. *International Journal of Communication Systems*, 19(4):363–380.

van den Nieuwelaar, M. and Hunt, R. (2004). Real-time carrier network traffic measurement, visualisation and topology modelling. *Computer Communications*, 27(1):128–140.

Varshney, U., Snow, A., McGivern, M., and Howard, C. (2002). Voice over IP. *Commun. ACM*, 45(1):89–96.

Vázquez, E., Álvarez-Campana, M., and García, A. B. (2004). Network convergence over MPLS. In *High Speed Networks and Multimedia Communications*, pages 290–300. Springer Berlin / Heidelberg.

Wang, R. and Hu, X. (2004). VoIP development in China. *Computer*, 37(9):30–37.

Wintermeyer, S. and Bosch, S. (2009). *Practical Asterisk 1.4 and 1.6: From Beginner to Expert*. Addison-Wesley Professional, Boston, MA, USA.

Wright, D. (2002). Voice over MPLS compared to voice over other packet transport technologies. *Communications Magazine, IEEE*, 40(11):124–132.

Yamamoto, R., Iseki, F., and Kim, M. W. (2008). Validation of VoIP system for university network. In *Proceedings of the 10th International Conference on Advanced Communication Technology*, volume 3, pages 1836–1841, Phoenix Park, Gangwon-Do, Korea.

Yeryomin, Y., Evers, F., and Seitz, J. (2008). Solving the firewall and NAT traversal issues for SIP-based VoIP. In *Proceedings of the 16th International Conference on Telecommunications*, pages 1–6, Marrakech, Morocco.

Zahariadis, T. and Spanos, S. (2004). The clearest voice [SIP vs H.323]. *Communications Engineer*, 2(2):14–17.

Zang, H. and Nucci, A. (2009). Traffic monitor deployment in IP networks. *Computer Networks*, 53(14):2491–2501.

Zangrilli, M. and Lowekamp, B. (2003). Comparing passive network monitoring of grid application traffic with active probes. In *Proceedings of the 4th International Workshop on Grid Computing*, pages 84–91, Phoenix, AZ, USA.

Zasepa, M., Sekalski, P., Sakowicz, B., and Mazur, P. (2009). Implementation of cost-effective VoIP network. In *Proceedings of the 16th International Conference on Mixed Design of Integrated Circuits and Systems*, pages 159–162, Łódź, Poland.

Zhang, C., Liu, B., Su, X., Alvarez, H., and Ibarra, J. (2007). An experimental approach to integrating NetFlow flow-level records and NLANR packet-level traces. In *Proceedings of the 2nd International Conference on Internet Monitoring and Protection*, pages 22–25, Silicon Valley, CA, USA.

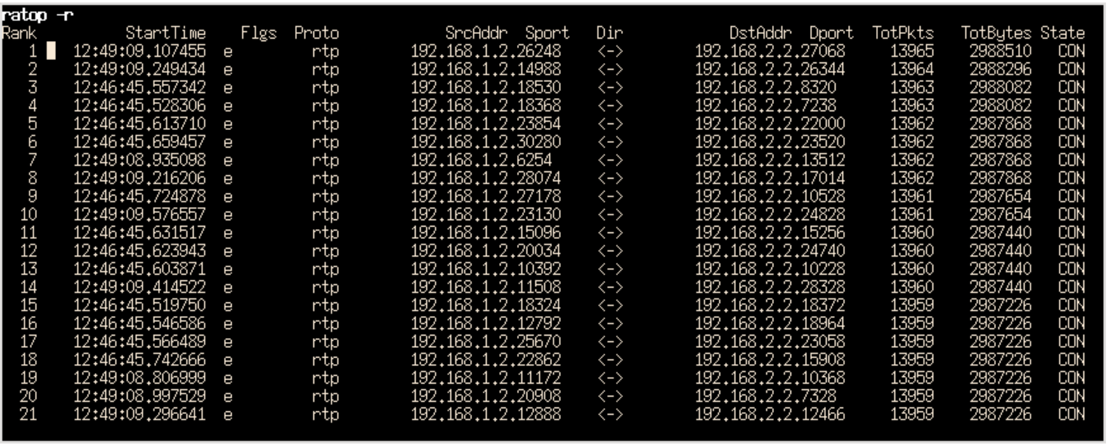
Zubairi, J. (2008). Voice transport techniques over MPLS. In *Proceedings of the 2008 International Symposium on High Capacity Optical Networks and Enabling Technologies*, pages 25–29, Penang, Malaysia.

APPENDIX A

Monitoring VoIP calls

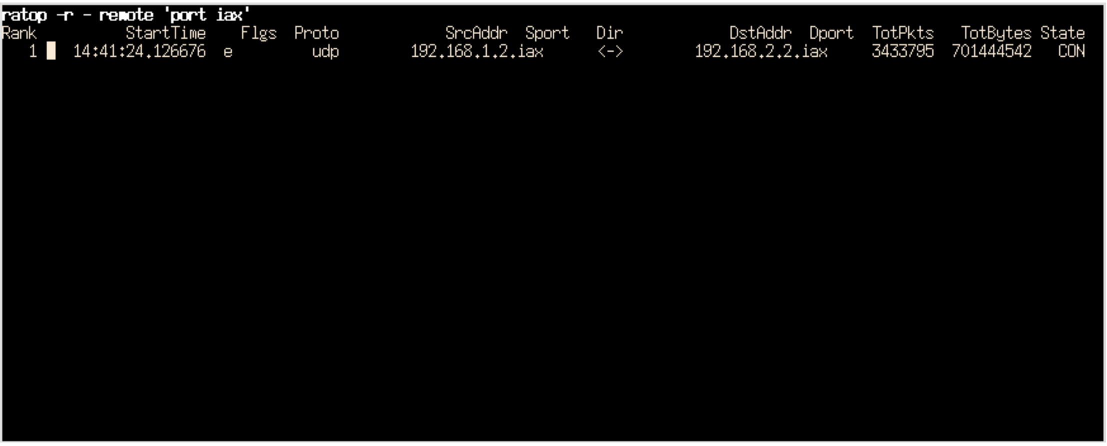
A.1 Asterisk calls

Following are some sample screenshots which illustrate the calls that were generated by Asterisk during the experiments and captured by Argus and Wireshark.



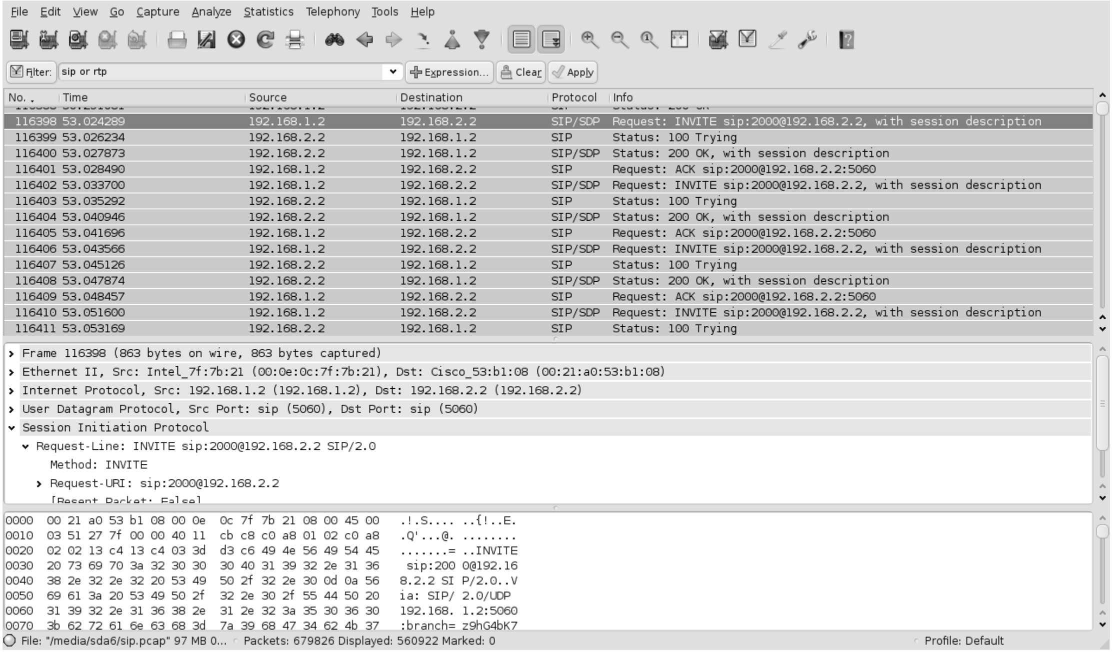
```
ratop -r
Rank        StartTime    Flgs  Proto          SrcAddr  Sport    Dir          DstAddr  Dport  TotPkts   TotBytes State
   1     12:49:09.107455  e      rtp      192.168.1.2.26248     <->     192.168.2.2.27068     13965    2988510   CON
   2     12:49:09.249434  e      rtp      192.168.1.2.14988     <->     192.168.2.2.26344     13964    2988296   CON
   3     12:46:45.557342  e      rtp      192.168.1.2.18530     <->     192.168.2.2.8320      13963    2988082   CON
   4     12:46:45.528306  e      rtp      192.168.1.2.18368     <->     192.168.2.2.7238      13963    2988082   CON
   5     12:46:45.613710  e      rtp      192.168.1.2.23854     <->     192.168.2.2.22000     13962    2987868   CON
   6     12:46:45.659457  e      rtp      192.168.1.2.30280     <->     192.168.2.2.23520     13962    2987868   CON
   7     12:49:08.935098  e      rtp      192.168.1.2.6254      <->     192.168.2.2.13512     13962    2987868   CON
   8     12:49:09.216206  e      rtp      192.168.1.2.28074     <->     192.168.2.2.17014     13962    2987868   CON
   9     12:46:45.724878  e      rtp      192.168.1.2.27178     <->     192.168.2.2.10528     13961    2987654   CON
  10     12:49:09.576557  e      rtp      192.168.1.2.23130     <->     192.168.2.2.24828     13961    2987654   CON
  11     12:46:45.631517  e      rtp      192.168.1.2.15096     <->     192.168.2.2.15256     13960    2987440   CON
  12     12:46:45.623943  e      rtp      192.168.1.2.20034     <->     192.168.2.2.24740     13960    2987440   CON
  13     12:46:45.603871  e      rtp      192.168.1.2.10392     <->     192.168.2.2.10228     13960    2987440   CON
  14     12:49:09.414522  e      rtp      192.168.1.2.11508     <->     192.168.2.2.28328     13960    2987440   CON
  15     12:46:45.519750  e      rtp      192.168.1.2.18324     <->     192.168.2.2.18372     13959    2987226   CON
  16     12:46:45.546586  e      rtp      192.168.1.2.12792     <->     192.168.2.2.18964     13959    2987226   CON
  17     12:46:45.566489  e      rtp      192.168.1.2.25670     <->     192.168.2.2.23058     13959    2987226   CON
  18     12:46:45.742666  e      rtp      192.168.1.2.22862     <->     192.168.2.2.15908     13959    2987226   CON
  19     12:49:08.806999  e      rtp      192.168.1.2.11172     <->     192.168.2.2.10368     13959    2987226   CON
  20     12:49:08.997529  e      rtp      192.168.1.2.20908     <->     192.168.2.2.7328      13959    2987226   CON
  21     12:49:09.296641  e      rtp      192.168.1.2.12888     <->     192.168.2.2.12466     13959    2987226   CON
```

Figure A.1: *Argus monitoring SIP calls.*



```
ratop -r - remote 'port iax'
Rank        StartTime    Flgs  Proto          SrcAddr  Sport    Dir          DstAddr  Dport  TotPkts   TotBytes State
   1     14:41:24.126676  e      udp      192.168.1.2.iax       <->     192.168.2.2.iax     3433795  701444542   CON
```

Figure A.2: *Argus monitoring IAX calls.*



Filter: sip or rtp

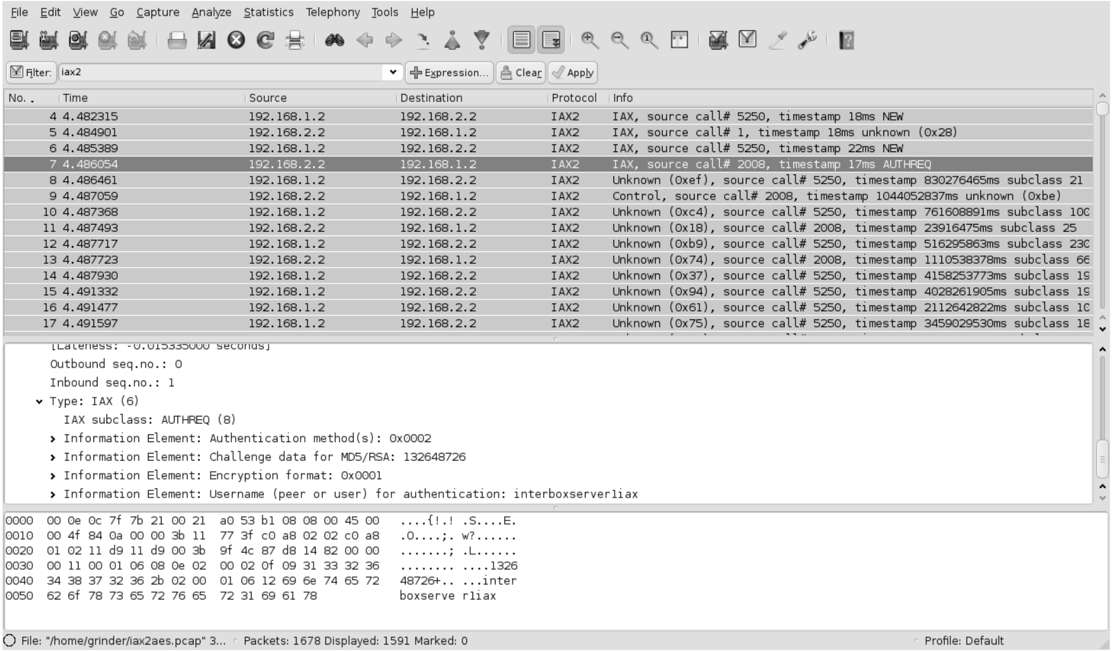
```
No. .  Time          Source          Destination     Protocol  Info
116398 53.024289     192.168.1.2     192.168.2.2     SIP/SDP   Request: INVITE sip:2000@192.168.2.2, with session description
116399 53.026234     192.168.2.2     192.168.1.2     SIP       Status: 100 Trying
116400 53.027873     192.168.2.2     192.168.1.2     SIP/SDP   Status: 200 OK, with session description
116401 53.028490     192.168.1.2     192.168.2.2     SIP       Request: ACK sip:2000@192.168.2.2:5060
116402 53.033700     192.168.1.2     192.168.2.2     SIP/SDP   Request: INVITE sip:2000@192.168.2.2, with session description
116403 53.035292     192.168.2.2     192.168.1.2     SIP       Status: 100 Trying
116404 53.040946     192.168.2.2     192.168.1.2     SIP/SDP   Status: 200 OK, with session description
116405 53.041696     192.168.1.2     192.168.2.2     SIP       Request: ACK sip:2000@192.168.2.2:5060
116406 53.043566     192.168.1.2     192.168.2.2     SIP/SDP   Request: INVITE sip:2000@192.168.2.2, with session description
116407 53.045126     192.168.2.2     192.168.1.2     SIP       Status: 100 Trying
116408 53.047874     192.168.2.2     192.168.1.2     SIP/SDP   Status: 200 OK, with session description
116409 53.048457     192.168.1.2     192.168.2.2     SIP       Request: ACK sip:2000@192.168.2.2:5060
116410 53.051600     192.168.1.2     192.168.2.2     SIP/SDP   Request: INVITE sip:2000@192.168.2.2, with session description
116411 53.053169     192.168.2.2     192.168.1.2     SIP       Status: 100 Trying
```

```
> Frame 116398 (863 bytes on wire, 863 bytes captured)
> Ethernet II, Src: Intel_7f:7b:21 (00:0e:0c:7f:7b:21), Dst: Cisco_53:b1:08 (00:21:a0:53:b1:08)
> Internet Protocol, Src: 192.168.1.2 (192.168.1.2), Dst: 192.168.2.2 (192.168.2.2)
> User Datagram Protocol, Src Port: sip (5060), Dst Port: sip (5060)
v Session Initiation Protocol
  v Request-Line: INVITE sip:2000@192.168.2.2 SIP/2.0
      Method: INVITE
    > Request-URI: sip:2000@192.168.2.2
      [Resent Packet: False]
```

```
0000  00 21 a0 53 b1 08 00 0e  0c 7f 7b 21 08 00 45 00   .!.S.... ..{!..E.
0010  03 51 27 7f 00 00 40 11  cb c8 c0 a8 01 02 c0 a8   .Q'...@. ........
0020  02 02 13 c4 13 c4 03 3d  d3 c6 49 4e 56 49 54 45   .......= ..INVITE
0030  20 73 69 70 3a 32 30 30  30 40 31 39 32 2e 31 36    sip:200 0@192.16
0040  38 2e 32 2e 32 20 53 49  50 2f 32 2e 30 0d 0a 56   8.2.2 SI P/2.0..V
0050  69 61 3a 20 53 49 50 2f  32 2e 30 2f 55 44 50 20   ia: SIP/ 2.0/UDP
0060  31 39 32 2e 31 36 38 2e  31 2e 32 3a 35 30 36 30   192.168. 1.2:5060
0070  3b 62 72 61 6e 63 68 3d  7a 39 68 47 34 62 4b 37   :branch= z9hG4bK7
```

File: "/media/sda6/sip.pcap" 97 MB 0... Packets: 679826 Displayed: 560922 Marked: 0 Profile: Default

Figure A.3: *Wireshark capturing SIP calls.*



Filter: iax2

```
No. .  Time        Source        Destination   Protocol  Info
     4 4.482315    192.168.1.2   192.168.2.2   IAX2      IAX, source call# 5250, timestamp 18ms NEW
     5 4.484901    192.168.2.2   192.168.1.2   IAX2      IAX, source call# 1, timestamp 18ms unknown (0x28)
     6 4.485389    192.168.1.2   192.168.2.2   IAX2      IAX, source call# 5250, timestamp 22ms NEW
     7 4.486054    192.168.2.2   192.168.1.2   IAX2      IAX, source call# 2008, timestamp 17ms AUTHREQ
     8 4.486461    192.168.1.2   192.168.2.2   IAX2      Unknown (0xef), source call# 5250, timestamp 830276465ms subclass 21
     9 4.487059    192.168.2.2   192.168.1.2   IAX2      Control, source call# 2008, timestamp 1044052837ms unknown (0xbe)
    10 4.487368    192.168.1.2   192.168.2.2   IAX2      Unknown (0xc4), source call# 5250, timestamp 761608891ms subclass 10C
    11 4.487493    192.168.2.2   192.168.1.2   IAX2      Unknown (0x18), source call# 2008, timestamp 23916475ms subclass 25
    12 4.487717    192.168.1.2   192.168.2.2   IAX2      Unknown (0xb9), source call# 5250, timestamp 516295863ms subclass 23C
    13 4.487723    192.168.2.2   192.168.1.2   IAX2      Unknown (0x74), source call# 2008, timestamp 1110538378ms subclass 66
    14 4.487930    192.168.1.2   192.168.2.2   IAX2      Unknown (0x37), source call# 5250, timestamp 4158253773ms subclass 1S
    15 4.491332    192.168.1.2   192.168.2.2   IAX2      Unknown (0x94), source call# 5250, timestamp 4028261905ms subclass 1S
    16 4.491477    192.168.1.2   192.168.2.2   IAX2      Unknown (0x61), source call# 5250, timestamp 2112642822ms subclass 1C
    17 4.491597    192.168.1.2   192.168.2.2   IAX2      Unknown (0x75), source call# 5250, timestamp 3459029530ms subclass 18
```

```
      [Lateness: -0.015335000 seconds]
      Outbound seq.no.: 0
      Inbound seq.no.: 1
    v Type: IAX (6)
        IAX subclass: AUTHREQ (8)
      > Information Element: Authentication method(s): 0x0002
      > Information Element: Challenge data for MD5/RSA: 132648726
      > Information Element: Encryption format: 0x0001
      > Information Element: Username (peer or user) for authentication: interboxserver1iax
```

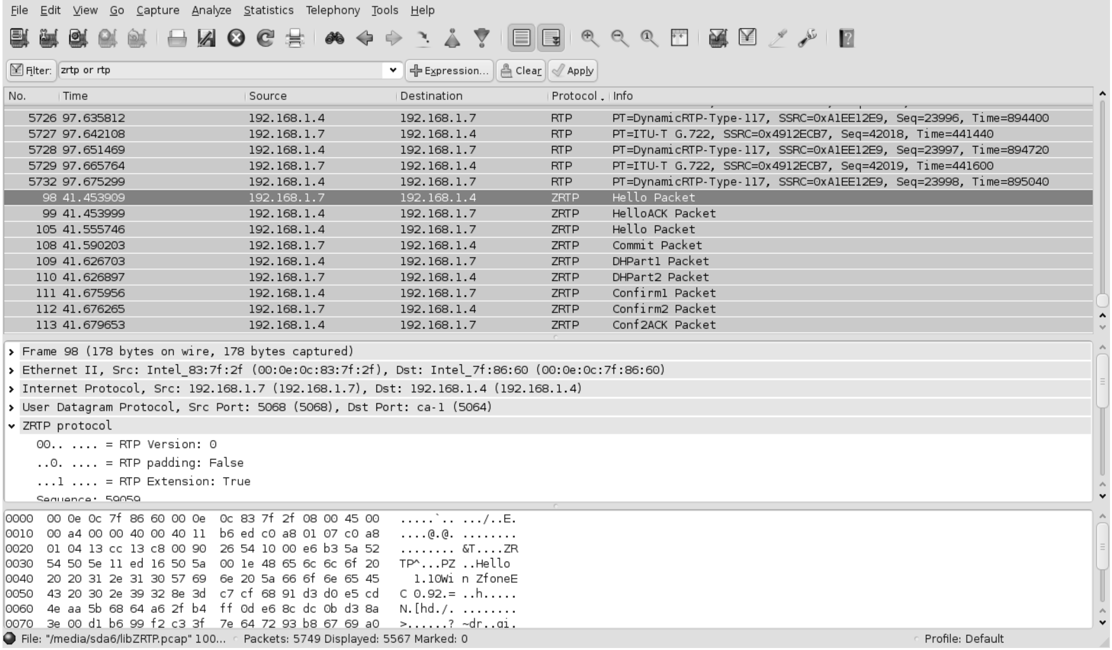
```
0000  00 0e 0c 7f 7b 21 00 21  a0 53 b1 08 08 00 45 00   ....{!.! .S....E.
0010  00 4f 84 0a 00 00 3b 11  77 3f c0 a8 02 02 c0 a8   .O....;. w?......
0020  01 02 11 d9 11 d9 00 3b  9f 4c 87 d8 14 82 00 00   .......; .L......
0030  00 11 00 01 06 08 0e 02  00 02 0f 09 31 33 32 36   ........ ....1326
0040  34 38 37 32 36 2b 02 00  01 06 12 69 6e 74 65 72   48726+.. ...inter
0050  62 6f 78 73 65 72 76 65  72 31 69 61 78            boxserve r1iax
```

File: "/home/grinder/iax2aes.pcap" 3... Packets: 1678 Displayed: 1591 Marked: 0 Profile: Default

Figure A.4: *Wireshark capturing IAX calls.*

A.2 SIP/ZRTP calls

As discussed in Chapters 5 and 6 generating SIP/ZRTP calls with Asterisk failed. Some sample calls were generated though using Phil Zimmermann's Zfone and the SIP Communicator[1] Java VoIP and instant messaging client that supports ZRTP in order to examine the operation of this protocol. These calls were captured by Wireshark as it is shown in the following figures.


File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter:
Expression...
Clear
Apply
Profile: Default

Filter: zrtp or rtp

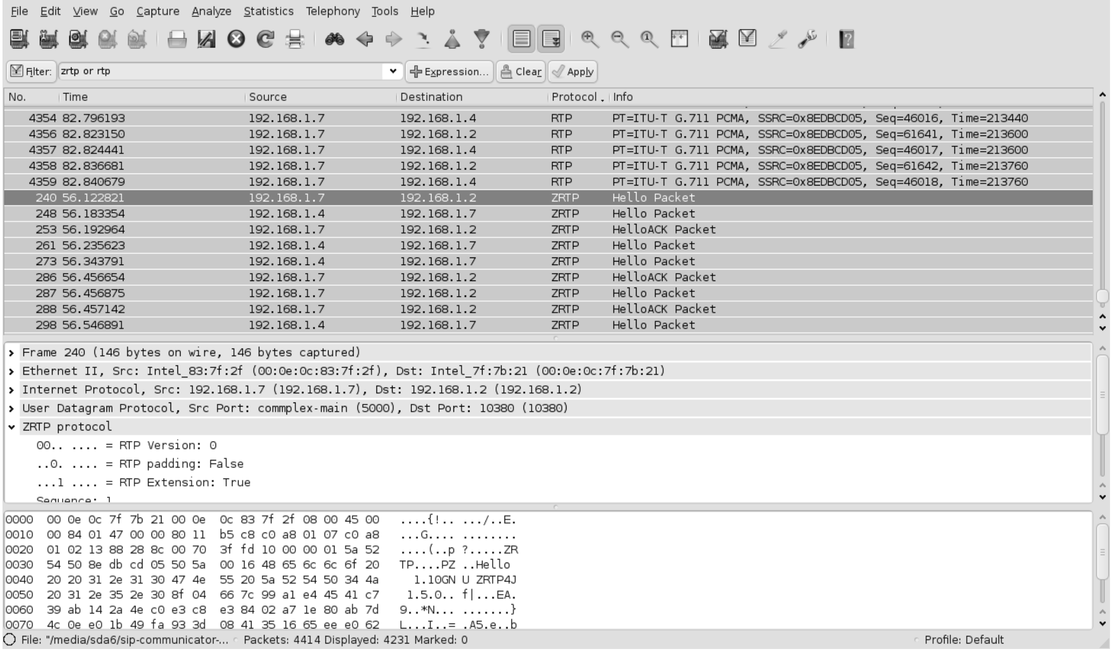
```
No.   Time       Source        Destination   Protocol  Info
5726 97.635812   192.168.1.4   192.168.1.7   RTP       PT=DynamicRTP-Type-117, SSRC=0xA1EE12E9, Seq=23996, Time=894400
5727 97.642108   192.168.1.7   192.168.1.4   RTP       PT=ITU-T G.722, SSRC=0x4912ECB7, Seq=42018, Time=441440
5728 97.651469   192.168.1.4   192.168.1.7   RTP       PT=DynamicRTP-Type-117, SSRC=0xA1EE12E9, Seq=23997, Time=894720
5729 97.665764   192.168.1.7   192.168.1.4   RTP       PT=ITU-T G.722, SSRC=0x4912ECB7, Seq=42019, Time=441600
5732 97.675299   192.168.1.4   192.168.1.7   RTP       PT=DynamicRTP-Type-117, SSRC=0xA1EE12E9, Seq=23998, Time=895040
  98 41.453909   192.168.1.7   192.168.1.4   ZRTP      Hello Packet
  99 41.453999   192.168.1.4   192.168.1.7   ZRTP      HelloACK Packet
 105 41.555746   192.168.1.4   192.168.1.7   ZRTP      Hello Packet
 108 41.590203   192.168.1.7   192.168.1.4   ZRTP      Commit Packet
 109 41.626703   192.168.1.4   192.168.1.7   ZRTP      DHPart1 Packet
 110 41.626897   192.168.1.7   192.168.1.4   ZRTP      DHPart2 Packet
 111 41.675956   192.168.1.4   192.168.1.7   ZRTP      Confirm1 Packet
 112 41.676265   192.168.1.7   192.168.1.4   ZRTP      Confirm2 Packet
 113 41.679653   192.168.1.4   192.168.1.7   ZRTP      Conf2ACK Packet
```

```
> Frame 98 (178 bytes on wire, 178 bytes captured)
> Ethernet II, Src: Intel_83:7f:2f (00:0e:0c:83:7f:2f), Dst: Intel_7f:86:60 (00:0e:0c:7f:86:60)
> Internet Protocol, Src: 192.168.1.7 (192.168.1.7), Dst: 192.168.1.4 (192.168.1.4)
> User Datagram Protocol, Src Port: 5068 (5068), Dst Port: ca-1 (5064)
v ZRTP protocol
    00.. .... = RTP Version: 0
    ..0. .... = RTP padding: False
    ...1 .... = RTP Extension: True
    Sequence: 59059
```

```
0000  00 0e 0c 7f 86 60 00 0e  0c 83 7f 2f 08 00 45 00   .....`.. .../..E.
0010  00 a4 00 00 40 00 40 11  b6 ed c0 a8 01 07 c0 a8   ....@.@. ........
0020  01 04 13 cc 13 c8 00 90  26 54 10 00 e6 b3 5a 52   ........ &T....ZR
0030  54 50 5e 11 ed 16 50 5a  00 1e 48 65 6c 6c 6f 20   TP^...PZ ..Hello
0040  20 20 31 2e 31 30 57 69  6e 20 5a 66 6f 6e 65 45     1.10Wi n ZfoneE
0050  43 20 30 2e 39 32 8e 3d  c7 cf 68 91 d3 d0 e5 cd   C 0.92.= ..h.....
0060  4e aa 5b 68 64 a6 2f b4  ff 0d e6 8c dc 0b d3 8a   N.[hd./. ........
0070  3e 00 d1 b6 99 f2 c3 3f  7e 64 72 93 b8 67 69 a0   >......? ~dr..gi.
```

File: "/media/sda6/libZRTP.pcap" 100... Packets: 5749 Displayed: 5567 Marked: 0

Figure A.5: *SIP/ZRTP calls with Phil Zimmermann's Zfone.*


File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter:
Expression...
Clear
Apply
Profile: Default

Filter: zrtp or rtp

```
No.   Time       Source        Destination   Protocol  Info
4354 82.796193   192.168.1.7   192.168.1.4   RTP       PT=ITU-T G.711 PCMA, SSRC=0x8EDBCD05, Seq=46016, Time=213440
4356 82.823150   192.168.1.7   192.168.1.2   RTP       PT=ITU-T G.711 PCMA, SSRC=0x8EDBCD05, Seq=61641, Time=213600
4357 82.824441   192.168.1.7   192.168.1.4   RTP       PT=ITU-T G.711 PCMA, SSRC=0x8EDBCD05, Seq=46017, Time=213600
4358 82.836681   192.168.1.7   192.168.1.2   RTP       PT=ITU-T G.711 PCMA, SSRC=0x8EDBCD05, Seq=61642, Time=213760
4359 82.840679   192.168.1.7   192.168.1.4   RTP       PT=ITU-T G.711 PCMA, SSRC=0x8EDBCD05, Seq=46018, Time=213760
 240 56.122821   192.168.1.7   192.168.1.2   ZRTP      Hello Packet
 248 56.183354   192.168.1.4   192.168.1.7   ZRTP      Hello Packet
 253 56.192964   192.168.1.7   192.168.1.2   ZRTP      HelloACK Packet
 261 56.235623   192.168.1.4   192.168.1.7   ZRTP      Hello Packet
 273 56.343791   192.168.1.4   192.168.1.7   ZRTP      Hello Packet
 286 56.456654   192.168.1.7   192.168.1.2   ZRTP      HelloACK Packet
 287 56.456875   192.168.1.7   192.168.1.2   ZRTP      Hello Packet
 288 56.457142   192.168.1.7   192.168.1.2   ZRTP      HelloACK Packet
 298 56.546891   192.168.1.4   192.168.1.7   ZRTP      Hello Packet
```

```
> Frame 240 (146 bytes on wire, 146 bytes captured)
> Ethernet II, Src: Intel_83:7f:2f (00:0e:0c:83:7f:2f), Dst: Intel_7f:7b:21 (00:0e:0c:7f:7b:21)
> Internet Protocol, Src: 192.168.1.7 (192.168.1.7), Dst: 192.168.1.2 (192.168.1.2)
> User Datagram Protocol, Src Port: commplex-main (5000), Dst Port: 10380 (10380)
v ZRTP protocol
    00.. .... = RTP Version: 0
    ..0. .... = RTP padding: False
    ...1 .... = RTP Extension: True
    Sequence: 1
```

```
0000  00 0e 0c 7f 7b 21 00 0e  0c 83 7f 2f 08 00 45 00   ....{!.. .../..E.
0010  00 84 01 47 00 00 80 11  b5 c8 c0 a8 01 07 c0 a8   ...G.... ........
0020  01 02 13 88 28 8c 00 70  3f fd 10 00 00 01 5a 52   ....(..p ?.....ZR
0030  54 50 8e db cd 05 50 5a  00 16 48 65 6c 6c 6f 20   TP....PZ ..Hello
0040  20 20 31 2e 31 30 47 4e  55 20 5a 52 54 50 34 4a     1.10GN U ZRTP4J
0050  20 31 2e 35 2e 30 8f 04  66 7c 99 a1 e4 45 41 c7    1.5.0.. f|...EA.
0060  39 ab 14 2a 4e c0 e3 c8  e3 84 02 a7 1e 80 ab 7d   9..*N... .......}
0070  4c 0e e0 1b 49 fa 93 3d  08 41 35 16 65 ee e0 62   L...I..= .A5.e..b
```

File: "/media/sda6/sip-communicator-... Packets: 4414 Displayed: 4231 Marked: 0

Figure A.6: *SIP/ZRTP calls with SIP Communicator.*

[1] http://sip-communicator.org/

APPENDIX B

Router configurations

B.1 Asterisk-R1

```
hostname Asterisk1
!
ip cef
!
ip dhcp pool 1
   network 192.168.1.0 255.255.255.0
   default-router 192.168.1.1
!
ip dhcp pool 0
   host 192.168.1.2 255.255.255.0
   hardware-address 000e.0c7f.7b21
!
ip dhcp pool 2
   host 192.168.1.3 255.255.255.0
   hardware-address 000e.0c7f.8a43
!
interface Loopback0
 ip address 192.168.100.1 255.255.255.0
!
interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 192.168.10.1 255.255.255.0
 duplex auto
 speed auto
!
router ospf 100
 log-adjacency-changes
 area 1 stub
 network 192.168.1.0 0.0.0.255 area 1
 network 192.168.10.0 0.0.0.255 area 1
!
scheduler allocate 20000 1000
ntp clock-period 17179730
ntp server 192.168.11.2
!
end
```

Figure B.1: *Asterisk-R1 router configuration.*

B.2 Asterisk-R2

```
hostname Asterisk2
!
ip cef
!
ip dhcp pool 1
   network 192.168.2.0 255.255.255.0
   default-router 192.168.2.1
!
ip dhcp pool 0
   host 192.168.2.2 255.255.255.0
   hardware-address 000e.0c83.7df8
!
interface Loopback0
 ip address 192.168.200.1 255.255.255.0
!
interface FastEthernet0/0
 ip address 192.168.13.1 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 192.168.2.1 255.255.255.0
 duplex auto
 speed auto
!
router ospf 100
 log-adjacency-changes
 area 2 stub
 network 192.168.2.0 0.0.0.255 area 2
 network 192.168.13.0 0.0.0.255 area 2
!
scheduler allocate 20000 1000
ntp clock-period 17179862
ntp server 192.168.12.1
!
end
```

Figure B.2: *Asterisk-R2 router configuration.*

B.3 R1

```
hostname R1
!
ip cef
!
interface FastEthernet0/0
 ip address 192.168.10.2 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 192.168.11.1 255.255.255.0
 duplex auto
 speed auto
 mpls ip
!
router ospf 100
 router-id 192.168.250.1
 log-adjacency-changes
 area 1 stub
 network 192.168.10.0 0.0.0.255 area 1
 network 192.168.11.0 0.0.0.255 area 0
!
scheduler allocate 20000 1000
ntp clock-period 17179718
ntp server 192.168.11.2
!
end
```

Figure B.3: *R1 router configuration.*

B.4 R2

```
hostname R2
!
ip cef
interface FastEthernet0/0
 ip address 192.168.11.2 255.255.255.0
 ip route-cache flow
 duplex auto
 speed auto
 mpls ip
!
interface FastEthernet0/1
 ip address 192.168.12.1 255.255.255.0
 duplex auto
 speed auto
 mpls ip
!
router ospf 100
 router-id 192.168.250.2
 log-adjacency-changes
 network 192.168.11.0 0.0.0.255 area 0
 network 192.168.12.0 0.0.0.255 area 0
!
ip flow-export source FastEthernet0/0
ip flow-export destination 192.168.1.2 9996
!
scheduler allocate 20000 1000
ntp master 5
!
end
```

Figure B.4: *R2 router configuration.*

B.5 R3

```
hostname R3
!
ip cef
!
interface FastEthernet0/0
 ip address 192.168.12.2 255.255.255.0
 duplex auto
 speed auto
 mpls ip
!
interface FastEthernet0/1
 ip address 192.168.13.2 255.255.255.0
 duplex auto
 speed auto
!
router ospf 100
 router-id 192.168.250.3
 log-adjacency-changes
 area 2 stub
 network 192.168.12.0 0.0.0.255 area 0
 network 192.168.13.0 0.0.0.255 area 2
!
scheduler allocate 20000 1000
ntp clock-period 17179628
ntp server 192.168.12.1
!
end
```

Figure B.5: *R3 router configuration.*

APPENDIX C

Asterisk configurations

C.1 Asterisk 1

```
[general]
defaultexpirey=1800
maxexpirey=3600
pedantic=yes
srvlookup=no
tlsbindaddr=0.0.0.0
tlscertfile=/etc/asterisk/cert/asterisk1.pem
tlsdontverifyserver=yes
tlscafile=/etc/asterisk/cert/ca1.crt

disallow=all
allow=alaw

[interboxserver2]
type=friend
host=192.168.2.2
context=callfromserver2
canreinvite=no
transport=tls
srtpcapable=yes

[1000]
type=friend
context=phones
defaultuser=1000
username=1000
secret=1000
host=dynamic
port=5060
nat=yes
qualify=yes

[1001]
type=friend
context=phones
defaultuser=1001
username=1001
secret=1001
host=dynamic
port=5060
```

Figure C.1: *sip.conf.*

```
[general]

autokill=yes
calltokenoptional=0.0.0.0/0.0.0.0
maxcallnumbers=16382
bandwidth=high
trunktimestamps=yes

minregexpire=180
maxregexpire=180

[interboxserver2iax]
type=friend
username=interboxserver1iax
secret=welcome
auth=md5
encryption=yes
host=192.168.2.2
context=callfromserver2
trunk=yes

[3000]
type=friend
context=phones
secret=3000
host=dynamic

[3001]
type=friend
context=phones
secret=3001
host=dynamic
transfer=no
```

Figure C.2: *iax.conf.*

```
[general]

staysecure=yes
cache_path=/var/lib/asterisk/zrtp_cache.dat
cache_saving_period=900
saving period in seconds

ATL=HS80
SAS=R256
CIPHER=AES3
PKTYPE=EC384P
PKTYPE=EC256P

DIR=/etc/asterisk/zrtp
sas_replay_dtmf=123
automatic_play_sas=no
automatic_play_sas_intro=no
insert_zrtp_sdp_tag=no
```

Figure C.3: *zrtp.conf.*

```
[globals]

[general]
autofallthrough=yes

[default]
include => phones

[incoming_calls]

[phones]
include => internal
include => calltoserver2

[internal]
exten => 1000,1,NoOp()
exten => 1000,n,Set(_SIPSRTP=${SIPPEER(${EXTEN},srtpcapable)})
exten => 1000,n,Set(_SIPSRTP_CRYPTO=enable)
exten => 1000,n,Set(_SIP_SRTP_SDES=enable)
exten => 1000,n,Answer()
exten => 1000,n,Playback(demo-congratsX5)
exten => 1000,n,Hangup()

exten => 3000,1,NoOp()
exten => 3000,n,Answer()
exten => 3000,n,Playback(demo-congratsX5)
exten => 3000,n,Hangup()

[callfromserver2]
exten => _1XXX,1,NoOp(Call from server2)
exten => _1XXX,n,Set(_SIPSRTP=${SIPPEER(${EXTEN},srtpcapable)})
exten => _1XXX,n,Set(_SIPSRTP_CRYPTO=enable)
exten => _1XXX,n,Set(_SIP_SRTP_SDES=enable)
exten => _1XXX,n,Dial(SIP/${EXTEN})
exten => _1XXX,n,Playback(demo-congratsX5)
exten => _1XXX,n,Hangup

exten => _3XXX,1,NoOp(Call from server2)
exten => _3XXX,n,Dial(IAX2/${EXTEN})
exten => _3XXX,n,Playback(demo-congratsX5)
exten => _3XXX,n,Hangup

[calltoserver2]
exten => _2XXX,1,NoOp(Call to server2)
exten => _2XXX,n,Set(_SIPSRTP=${SIPPEER(${EXTEN},srtpcapable)})
exten => _2XXX,n,Set(_SIPSRTP_CRYPTO=enable)
exten => _2XXX,n,Set(_SIP_SRTP_SDES=enable)
exten => _2XXX,n,Dial(SIP/interboxserver2/${EXTEN})
exten => _2XXX,n,Hangup

exten => _4XXX,1,NoOp(Call to server2)
exten => _4XXX,n,Dial(IAX2/interboxserver2iax/${EXTEN})
exten => _4XXX,n,Hangup
```

Figure C.4: *extensions.conf.*

C.2 Asterisk 2

```
[general]
defaultexpirey=1800
maxexpirey=3600
pedantic=yes
srvlookup=no
tlsbindaddr=0.0.0.0
tlscertfile=/etc/asterisk/cert/asterisk2.pem
tlsdontverifyserver=yes
tlscafile=/etc/asterisk/cert/ca2.crt

disallow=all
allow=alaw

[interboxserver1]
type=friend
host=192.168.1.2
context=callfromserver1
canreinvite=no
transport=tls
srtpcapable=yes

[2000]
type=friend
context=phones
defaultuser=2000
username=2000
secret=2000
host=dynamic
port=5060
nat=yes
qualify=yes

[2001]
type=friend
context=phones
defaultuser=2001
username=2001
secret=2001
host=dynamic
port=5060
```

Figure C.5: *sip.conf.*

```
[general]

autokill=yes
calltokenoptional=0.0.0.0/0.0.0.0
maxcallnumbers=16382
bandwidth=high
trunktimestamps=yes

minregexpire=180
maxregexpire=180

[interboxserver1iax]
type=friend
username=interboxserver2iax
secret=welcome
auth=md5
encryption=yes
host=192.168.1.2
context=callfromserver1
trunk=no

[4000]
type=friend
context=phones
secret=4000
host=dynamic

[4001]
type=friend
context=phones
secret=4001
host=dynamic
```

Figure C.6: *iax.conf.*

```
[general]

staysecure=yes
cache_path=/var/lib/asterisk/zrtp_cache.dat
cache_saving_period=900
saving period in seconds

ATL=HS80
SAS=R256
CIPHER=AES3
PKTYPE=EC384P
PKTYPE=EC256P

DIR=/etc/asterisk/zrtp
sas_replay_dtmf=123
automatic_play_sas=no
automatic_play_sas_intro=no
insert_zrtp_sdp_tag=no
```

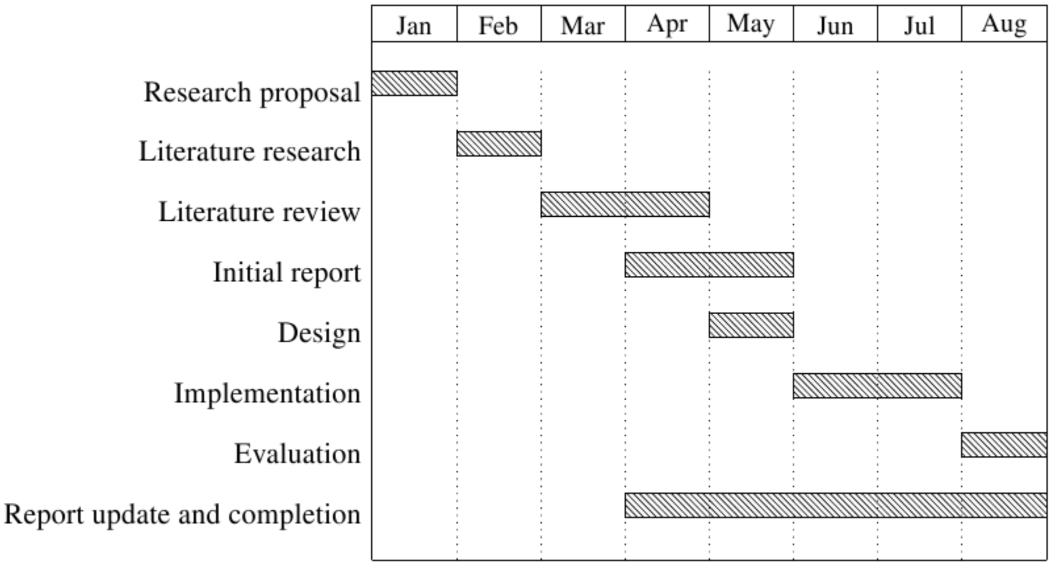
Figure C.7: *zrtp.conf.*

```
[globals]

[general]
autofallthrough=yes

[default]
include => phones

[incoming_calls]

[phones]
include => internal
include => calltoserver1

[internal]
exten => 2000,1,NoOp()
exten => 2000,n,Set(_SIPSRTP=${SIPPEER(${EXTEN},srtpcapable)})
exten => 2000,n,Set(_SIPSRTP_CRYPTO=enable)
exten => 2000,n,Set(_SIP_SRTP_SDES=enable)
exten => 2000,n,Answer()
exten => 2000,n,Playback(demo-congratsX5)
exten => 2000,n,Hangup()

exten => 4000,1,NoOp()
exten => 4000,n,Answer()
exten => 4000,n,Playback(demo-congratsX5)
exten => 4000,n,Hangup()

[callfromserver1]
exten => _2XXX,1,NoOp(Call from server1)
exten => _2XXX,n,Set(_SIPSRTP=${SIPPEER(${EXTEN},srtpcapable)})
exten => _2XXX,n,Set(_SIPSRTP_CRYPTO=enable)
exten => _2XXX,n,Set(_SIP_SRTP_SDES=enable)
exten => _2XXX,n,Dial(SIP/${EXTEN})
exten => _2XXX,n,Playback(demo-congratsX5)
exten => _2XXX,n,Hangup

exten => _4XXX,1,NoOp(Call from server2)
exten => _4XXX,n,Dial(IAX2/${EXTEN})
exten => _4XXX,n,Playback(demo-congratsX5)
exten => _4XXX,n,Hangup

[calltoserver1]
exten => _1XXX,1,NoOp(Call to server1)
exten => _1XXX,n,Set(_SIPSRTP=${SIPPEER(${EXTEN},srtpcapable)})
exten => _1XXX,n,Set(_SIPSRTP_CRYPTO=enable)
exten => _1XXX,n,Set(_SIP_SRTP_SDES=enable)
exten => _1XXX,n,Dial(SIP/interboxserver1/${EXTEN})
exten => _1XXX,n,Hangup

exten => _3XXX,1,NoOp(Call to server2)
exten => _3XXX,n,Dial(IAX2/interboxserver1iax/${EXTEN})
exten => _3XXX,n,Hangup
```

Figure C.8: *extensions.conf.*

APPENDIX D

Project management



APPENDIX E

Project proposal

1. Project outline details

Please suggest a title for your proposed project. If you have worked with a supervisor on this proposal, please provide the name. NB you are strongly advised to work with a member of staff when putting your proposal together.

1. **Title of the proposed project:** A comparative study of in-band and out-of-band VoIP protocols in Layer 3 and Layer 2.5 environments.
2. **Name of supervisor:** Prof. William Buchanan

2. Brief description of the research area - background

Please provide background information on the broad research area of your project in the box below. You should write in narrative (not bullet points). The academic/theoretical basis of your description of the research area should be evident through the use of references. Your description should be between half and one page in length.

During the last ages the Internet has thrived as one of the main communication means by introducing a variety of new applications and services. One of the most important is VoIP (Voice over Internet Protocol). VoIP emerges as a significant alternative to conventional circuit switched telephone networks. It can be observed through the years that telephone service providers as well as large enterprises install VoIP softswitches in order to replace their old PABXs (Private Automatic Branch eXchanges) that connect to the PSTN (Public Switched Telephone Network). This new technology offers them great cost savings, ease of maintenance as well as deployment of new services for their customers and end users (Karapantazis and Pavlidou, 2009).

But although packet switched telephony has so many advantages against classic circuit switched telephony, the voice quality that it offers, still remains a key issue. Turning the analog sound into digital is one factor that affects voice quality, but not the only one. Due to their nature, VoIP systems are extremely sensitive to delay, jitter (variation of delay) and packet loss, especially in the modern converged networks, that carry both voice and data. Various signalling protocols have been proposed and developed through the years, that try to overcome the above critical issues: Out-of-band (that don't carry voice payload) and in-band (that do carry voice payload). Abbasi et al. (2005) and

Montoro and Casilari (2009) in their research have shown that in-band signalling protocols perform better than the out-of-band, when their security and encryption features, that cause extra overhead, are not in use.

The critical issue of performance does not only depend on the VoIP protocols themselves, but also on the underlying routing mechanisms. While Fortz et al. (2002) states that traditional Layer 3 link state routing protocols can be configured to perform really well, Kocak et al. (2009) argues that Layer 2.5 protocols must be preferred, particularly for multimedia applications. On the other hand, Rong et al. (2005) claims that such technologies can bring into effect long call setup delays.

In this study, out-of-band and in-band VoIP protocols are going to be evaluated in terms of features and performance and how security mechanisms can affect the latter one. Also, the study will attempt to determine how Layer 2.5 protocols compare to classic Layer 3 protocols in regards to the efficiency of the VoIP systems.

3. Project outline for the work that you propose to complete

Please complete the project outline in the box below. You should use the emboldened text as a framework. Your project outline should be between half and one page in length.

The idea for this research arose from:

Working in the R&D industry for circuit switched communications which highlighted the need for easy deployment and maintenance of services as well as inexpensive telephony.

The aims of the project are as follows:

- To provide a critical analysis of VoIP architectures, protocols, evaluationframeworks, secure methods and routing infrastructures, with a focus on defining the key evaluation metrics involved.
- To conduct an evaluation of VoIP for in-band and out-of-band protocols in Layer 3 and Layer 2.5 environments, especially focusing on test traffic generation, security integration and evaluation metrics (such as delay, jitter, packet loss and bandwidth usage).
- To provide an analysis of the results and suggest recommendations based on findings and areas for further research on VoIP technologies.

Personal aims include:

- Completion of the compulsory component for the award of an MSc Advanced Networking
- Publishing of a paper based on the MSc Project.

- Preparation for employment in the networking and system administration field.

The main objective of the research project is:

- To design a VoIP architecture and an evaluation framework, in order to assess the key performance metrics related to VoIP for in-band and out-of-band protocols.

The main research questions that this work will address include:

- To what extent do in-band or out-of band VoIP protocols perform better?
- To what extent does encryption affect performance of VoIP protocols?
- To what extent do Layer 2.5 protocols affect performance of VoIP systems?

The project will involve the following research/field work/experimentation/evaluation:

- Research into different VoIP technologies and architectures.
- Research into methods for the evaluation of VoIP systems and routing mechanisms.
- Evaluation of the proposed framework. This will include measurements in a variety of experimental conditions and evaluation of the results as well as of the tools and methods used.

Resources:

Computing and library facilities at Napier University will be used extensively in support of the project. No specialist resources will be required.

This work will require the use of specialist software:

Asterisk PBX, Nagios, Argus, softphones

This work will require the use of specialist hardware:

Cisco routers and switches, normal PCs

The project is being undertaken in collaboration with:

None available at the moment

4. References

Please supply details of all the material that you have referenced in sections 6 and 7 above. You should include at least three references, and these should be to high quality sources such as refereed journal and conference papers, standards or white papers. Please ensure that you use a standardised referencing style for the presentation of your references, e.g. APA, as outlined in the yellow booklet available from the School of Computing office and http://www.dcs.napier.ac.uk/~hazelh/gen_ho/apa.pdf

Abbasi, T., Lambadaris, I., Prasad S. and Seddigh N. (2005). A comparative study of the SIP and IAX VoIP protocols. In *Proceedings of the 2005 Canadian Conference on Electrical and Computer Engineering.* pp. 179-183.

Ahson, S.A. and Ilyas, M. (2008). *VoIP Handbook: Applications, Technologies, Reliability, and Security.* CRC Press, Boca Raton, FL, USA.

Conway, A.E. (2000). A performance monitoring system for VoIP gateways. In *Proceedings of the 2nd International Workshop on Software and Performance.* Ottawa, Ontario, Canada: ACM, pp. 38-43.

Fortz, B., Rexford, J. and Thorup, M. (2002). Traffic engineering with traditional IP routing protocols. *Communications Magazine, IEEE*, 40(10), 118-124.

Karapantazis, S. and Pavlidou, F. (2009). VoIP: A comprehensive survey on a promising technology. *Computer Networks*, 53(12), 2050-2090.

Kocak, C., Erturk, I. and Ekiz, H. (2009). MPLS over ATM and IP over ATM methods for multimedia applications. *Computer Standards & Interfaces*, 31(1), 153-160.

Lima, A.F.M., Carvalho, L.S.G., Souza,J.N. and Souza Mota, E. (2007). A framework for network quality monitoring in the VoIP environment. *International Journal of Network Management*, 17(4), 263-274.

Meggelen, J.V., Madsen, L. and Smith, J.W. (2005).*Asterisk: The Future of Telephony*, O'Reilly Media, Sebastopol, CA, USA. 2nd edition.

Minoli, D. (2002). *Voice over MPLS : Planning and Designing Networks*, McGraw-Hill Professional, New York City, NY, USA.

Montoro, P. and Casilari, E. (2009). A Comparative Study of VoIP Standards with Asterisk. In *Proceedings of the 2009 Fourth International Conference on Digital Telecommunications*. IEEE Computer Society, pp. 1-6.

Rong, B., Lebeau, J., Bennani, M., Kadoch, M. and Elhakeem, A.K. (2005). Traffic Aggregation Based SIP over MPLS Network Architecture. In *Proceedings of the 19th International Conference on Advanced Information Networking and Applications.* IEEE Computer Society, pp. 827-832.