# A Study of VoIP Performance in Anonymous Network - The Onion Routing (Tor)

Dissertation
for the award of the degree
**"Doctor rerum naturalium (Dr.rer.nat.)"**
of the Georg-August-University Göttingen

within the doctoral Programme in Computer Science (PCS)
of the Georg-August University School of Science (GAUSS)

submitted by
**Maimun Rizal**

from Indonesia (Aceh Utara)
Göttingen, 2014

**Thesis Committee**

Prof. Dr. Dieter Hogrefe
Institute of Computer Science, Georg-August-University Göttingen

Prof. Dr. Xiaoming Fu
Institute of Computer Science, Georg-August-University Göttingen

Dr. Somayeh Taheri
Institute of Computer Science, Georg-August-University Göttingen

**Members of the Examination Board:**

**Reviewer**

Prof. Dr. Dieter Hogrefe
Institute of Computer Science, Georg-August-University Göttingen

**Second Reviewer**

Prof. Dr. Xiaoming Fu
Institute of Computer Science, Georg-August-University Göttingen

**Further members of the Examination Board**

Prof. Dr. Eckart Modrow
Institute of Computer Science, Georg-August-University Göttingen

Prof. Dr. Carsten Damm
Institute of Computer Science, Georg-August-University Göttingen

Prof. Dr. Stephan Waack
Institute of Computer Science, Georg-August-University Göttingen

Prof. Dr. Konrad Rieck
Institute of Computer Science, Georg-August-University Göttingen

Date of the oral examination: 11th June 2014

*To My Family*

# ACKNOWLEDGEMENTS

First of all, I would like to express my deepest gratitude to my supervisor, Prof. Dr. Dieter Hogrefe for accepting me as his PhD student, for his support, helpful advice and guidance throughout my study. I would like to thank to my co-supervisor Prof. Dr. Xiaoming Fu, for his guidance, advice and input as well as during my study. And I am grateful to my second co-supervisor Dr. Somayeh Taheri, for her discussion and valuable feedback.

I would also like to extend my gratitude to all members of the examination board of my thesis: Prof. Dr. Eckart Modrow, Prof. Dr. Carsten Damm, Prof. Dr. Stephan Waack, and Prof. Dr. Konrad Rieck.

I would like to acknowledge all of the academic and administrative staff of the Institute of Computer Science at the Georg-August University of Göttingen for their guidance and encouragement. Especially, I would like to thank Prof. Dr. Jens Grabowski, Carmen Scherbaum and Annette Kadziora for their assistance during my study in the institute. And also my thanks are to Udo Burghardt, who provide technical support in my experimental research.

I would also express my great thank to former and current Telematics group members: Dr. Saleh Al-Shadly, Dr. Youssef Shehadeh, Dr. Angsgar Kellner, Dr. Parisa Memarmoshrefi, Dr. Roman Siebel, Sviatoslav Edelev, Hang Zhang, Salke Hartung, and Betty Mayeku, for knowledge sharing and discussion during my study in Germany. And also I would like to thank to members of computer security group.

I need to acknowledge the Aceh government for granting me a scholarship, and Educational Quality Assurance Institution – “Lembaga Penjaminan Mutu Pendidikan (LPMP)” Aceh, Ministry of Education and Culture Republic of Indonesia, for allowing me to take study leave overseas. I would like to acknowledge the Deutscher Akademischer Austausch Dients (DAAD) who gives me their best assistant during my

study in Germany. In particular, I would like to thank Tobias Ebel and Monika Gasienica-Jozkowy from DAAD for their help and support.

I would also express my thanks to few external people. I would like to thank Mr. Heiko Sommerfeldt, Kevin Bauer, and Vesselin Velickhov for sharing knowledge and discussion. Many thanks to Paul Syverson, Steven Murdoch, Karsten Loesing, Runa A. Sanvik, and all members of tor-talk@lists.torproject.org, for sharing knowledge about The Onion Routing (Tor).

I appreciate the Indonesian community for their help and warm friendship during my study in Göttingen. I would like to thank to all Acehnese student societies in Germany for their kind support. My sincerest thanks to my friends in Göttingen for making me feel like home, especially to Essy Harnelly, Ikhlas Ali Amran, Teuku Firman, Rezky Syahrezal, Ichsan, Angga Yudisthira, Achmad Fadilah, Febi Mutia, Nadya, Labudda, and Siti Maryam for their help and support during my study in Göttingen.

My deepest gratitude to my grandmother, mother, sister, parents in law and all my big family, for their love, support and prayer throughout my study. Last and not least, I would like to extend my profound gratitude to my wife, for her patience, love, and encouragement during my study.

Thank you very much for all and for everyone whom I know and I did not mention their name due to the limited space.

# ABSTRACT

Information technology is developing rapidly, especially in the field of computer science, telecommunications and information systems. In the early days of telecommunications, voice networks could not be combined with data networks. However, more recently it became possible to integrate data, audio and video services onto a single network, known as a multimedia network. The capabilities of the Internet are also rapidly increasing, and now, in addition to data packets (email, web browsing) the Internet transfers audio and video packets.

Voice over Internet Protocol (VoIP) is a multimedia service. This technology can be used to transmit audio, video, and data packets. Consequently, it is becoming the most preferred communications technology. VoIP will eventually replace the use of traditional telephony – Public Switched Telephone Networks (PSTN), although the switchover process is challenging. A chief concern it that, while PSTN transmits audio packets over a closed network, VoIP sends audio packets using an open network – the Internet. In PSTN, eavesdroppers must have direct access to physical network to obtain information from communications. Whereas with VoIP, eavesdroppers can monitor data packets from they are connected to the Internet. Hence, security and privacy are important considerations when switching to VoIP systems. One solution is anonymous systems, which can be used to implement security and privacy protocols. However, this typically reduces the Quality of Service (QoS) of the VoIP.

This empirical research study examines VoIP performance in an anonymous network – The Onion Routing (Tor). Two scenarios are implemented using the real Tor network to investigate three QoS metrics for VoIP: latency, jitter and packet loss. As recommended, latency in VoIP should be less than 400 ms, jitter should be less than 50 ms and packet loss should not be more than 5%. In addition, the research calculates the probability of attackers in the two scenarios implemented and evaluate Tor network forecasting.

Experiments were conducted in reference to two scenarios. The first scenario is VoIP calls routed through a Tor network with three Tor relays (default Tor). And the second scenario is VoIP calls routed through a Tor network with two Tor relays. Experiments were performed in three periods; December 2012, July 2013, and October 2013. During each experimental time period, a hundred calls were captured for each scenario. Experimental results show that the QoS of VoIP over the two Tor relays was better than the VoIP over the three Tor relays. However, a probability of attackers calculation found that the VoIP with three Tor relays returned better anonymity than that with two Tor relays.

The best results were returned over the experimental period in July 2013, when the acceptable calls using two Tor relays reached 64 calls at 5% packet loss. Meanwhile, the worst experimental results were returned in October 2013, when acceptable calls numbered 11 using three Tor relays at 1% packet loss. At the end of 2013, the actual data for relay numbers and bandwidth approached forecast result with time series analysis. Meanwhile, the numbers of Tor users differed from the Tor users forecast. In August 2013, Tor users increased dramatically; this is attributed to the fact that BotNet attack was using the Tor network to attack their target leading to a fivefold increase.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| AES | Advanced Encryption Standard |
| DoS | Denial of Services |
| DDoS | Distributed Denial of Services |
| DNS | Domain Name Services |
| GIPS | Global IP Solutions |
| HTTP | Hypertext Transfer Protocol |
| IETF | Internet Engineering Task Force |
| iLBC | Internet Low Bit Rate Codec |
| IP | Internet Protocol |
| ITU | International Telecommunication Union |
| JAP | Java Anonymity Proxy |
| LAN | Local Area Network |
| MAD | Mean Absolute Deviation |
| MCU | Multipoint Control Unit |
| MIKEY | Multimedia Internet Keying |
| MOS | Mean Opinion Score |
| MSE | Mean Squared Error |
| NAT | Network Address Translation |
| NRL | Naval Research Laboratory |
| NTP | Network Time Protocol |
| OR | Onion Routing |
| PSTN | Public Switched Telephone Networks |
| QoS | Quality of Services |
| RFC | Request for Comments |
| RTCP | Real Time Control Protocol |
| RTP | Real Time Protocol |
| RTT | Round Trip Time |
| SIP | Session Initiation Protocol |
| S/MIME | Secure Multipurpose Internet Mail Extensions |

| | |
|---|---|
| SRTP | Secure Real Time Protocol |
| SSL | Secure Socket Layer |
| TCP | Transmission Control Protocol |
| TLS | Transport Layer Security |
| Tor | The Onion Routing |
| UDP | User Datagram Protocol |
| UTC | Coordinated Universal Time |
| VoIP | Voice over Internet Protocol |
| VPN | Virtual Private Network |
| WAN | Wide Area Network |
| ZRTP | Zimmermann Real Time Protocol |
| ZTS | Zaitun Time Series |

# 1 INTRODUCTION

Computer science and information systems are increasingly being utilised. Therefore, it is also a very active area of research. An important technology that relates computer science and information systems is the Voice over Internet Protocol (VoIP). This technology supports the transference of voice, data, and images through packet-switched networks (Internet Protocol), and has provided significant changes to communications worldwide. Cost saving, enhanced functionality and ease of maintenance, are the principle reasons why VoIP has been among the most popular telecommunication technology growth areas. It is especially valuable for its utility supporting long distance communication and internal corporate networks. A corporation can increase its efficiency by combining voice and data networks in a single network.

However, incorporating VoIP has more attendant security risks than traditional communications because the voice packets are sent through an open and independent network, i.e. the Internet. Therefore, there is a need for those offering VoIP applications to provide users with greater insurance of privacy. As VoIP is a real-time technology multimedia application, it demands a good Quality of Services (QoS), that is a latency less than 400 ms, to provide acceptable performance. Providing user privacy in VoIP technology required that the voice packets be routed through a suitable anonymity system. However, the added cryptographic computations and random rerouting which guarantees the efficacy of anonymity systems, introduces delays.

This explains why the process of switching from Public Switched Telephone Networks (PSTN) to VoIP signifies a new data security challenge. Unlike PSTN, wherein the telephony networks are isolated from data networks and the Internet, the majority of VoIP communications are transferred over the public network – the Internet and so may easily become exposed to security threats, especially when the network is inappropriately designed [1].

In PSTN, eavesdropping usually requires direct access to tap a line or penetrate a switch. Thus, the intruder's risk of being discovered may increase when attempting physical access. In contrast, in VoIP, the opportunities for eavesdropping are tremendously greater, as it can be achieved simply by observing the different nodes on a packet network [2].

## 1.1 Research Problem

There have been many researches focused on VoIP security; for example adding encryption [3, 4], and steganography [5, 6] to VoIP communication to protect the message communicated between the sender and receiver. However, encryption techniques are as yet unable to conceal information about the caller and callee identity.

Nowadays, additional privacy techniques are needed with VoIP communication; to help to protect information shared between the caller and callee, and to insure it is not possible to discover who is calling to whom. However, combining security and privacy options together in a VoIP communication network increases end-to-end delays; thereby, degrading the quality of service (QoS).

In VoIP, guaranteeing privacy over an anonymity network is difficult to implement. Although there are many low latency networks, these are generally not designed to transfer real-time communications such as VoIP applications. Many anonymity networks are based on TCP streaming. Unlike VoIP, this transfers audio packets in the form RTP packet over an UDP based stream.

This research is based on an empirical model. The model and experiment are designed to transfer VoIP over an anonymous network – the Tor network. The main purpose of this research is then to investigate predictions of QoS performance in anonymity VoIP over an existing anonymous system – The Onion Routing (Tor). In other words, this research studies the feasibility of implementing VoIP over the Tor network and also focuses on Tor network forecasting. QoS performance analysis will be conducted to determine the advantages and the drawbacks of the Tor network as a support for real-time communication applications such as VoIPs.

## 1.2 Summary of Research Question

Four fundamental research questions are asked in this research:

- **Question 1:** How is the VoIP network integrated with an anonymous network – the Onion Routing (Tor) network?

Tor is a low latency anonymous network. It is only capable of transferring data packets, based on the TCP stream protocol. Voice packets usually use RTP, and are transferred with a UDP stream. Because VoIP uses a UDP stream, the Tor network cannot directly transfer voice packets onto it. There are several options available for transmitting voice packets through the Tor network. This research will discuss how best to integrate VoIP with the Tor network in chapter 3.

- **Question 2:** How are QoS performances affected by anonymising VoIP over the Tor network?

The original Tor network uses three relays, located between the sender and receiver. These are an entry relay, a middle relay and an exit relay. In implementation, although the Tor network is a low latency anonymous network, there is no guarantee that any data packet can be transferred with a latency of less than 1 second.

Latency is one of factors to consider when sending voice packets over the Tor network. This is a sensitive issue on VoIP, and based on ITU recommendations, end-to-end or one-way latency on VoIP should not exceed 400 ms. Therefore, this research modifies Tor relay use, so that it is shorter than that of the original Tor network. Besides this, using two relays results in a lower latency than using three relays.

Comparing QoS performances in VoIP using three or two Tor relays is discussed in chapter 4. Increasing QoS performances relative to the reduction of anonymity in the Tor network, is covered in relation to research question 3.

- **Question 3:** What is the difference in the probability of attackers afforded by the Tor network, when two Tor relays are used instead of three?

Differences in the number of relays used on the Tor network can cause a reduction in the anonymity. In other words, anonymity increases as more relays are used in the connection; however, the performance may decrease due to the path length of data connection. In this research, anonymity is based on the probability of attackers. In chapters 2 and 4, the details of probability of attackers are presented in relation to the number of relays and the path length of the Tor network connection.

- **Question 4:** Can the Tor network capabilities be predicted for the future to support VoIP systems?

Chapter 4 also discusses predictions about Tor network conditions. A time series analysis applying four trends related methods are used to forecast the progress on the Tor network in the future. Thus, the growth of the Tor network can be predicted, as can the ability of the Tor network to transfer VoIP calls.

## 1.3 Contributions

This research will make three contributions. First, it evaluates the performance of VoIP through an anonymous network – The Onion Routing (Tor) [7]. Second, it predicts the growth of users, bandwidth and relays in the Tor network using the time series analysis method. Third, it present the anonymity of the Tor network, based on the probability of attackers, as it related to the number of Tor relays and the path length of the Tor network.

## 1.4 Dissertation Structure

The remainder of this dissertation is organised as follow. In section 2, we provide an overview of the literature review on VoIP technology, anonymous system relevant to this study, the current attack on Tor, and related works. In section 3, we describe the research method used, which is discusses the research approach, research design, instruments, data collection procedures, and data analysis procedures. The results and analysis from the experiments are given in section 4. Finally, section 5 presents the conclusions, discussion, and generalizes the results and outlines the possibilities for future work.

# 2 LITERATURE REVIEW

This chapter presents the literature review and introduces VoIP architecture in anonymous network systems, including VoIP protocol, voice codec and QoS metrics. The chapter also introduces anonymous network systems, and the forecasting network method.

## 2.1 VoIP Protocol

At the present time, there are two common standards for signalling and controlling VoIP or Internet telephone calls; these are H.323 and Session Initiation Protocol (SIP). Both were developed in 1995 and were solutions for researchers when initiating communication between two computers in order to transfer voice and video media streams [8]. International Telecommunication Union (ITU) published the first H.323 standard in early 1996 and the Internet Engineering Task Force (IETF) published the SIP standard in draft form in 1996. H.323 provides specific QoS parameters, such as low end-to-end latency and packet loss, meanwhile SIP considers security [2, 9]. This research focuses on SIP, as this will be used in the experiments. Figure 2.1 presents general VoIP architecture with two computers and one VoIP server.

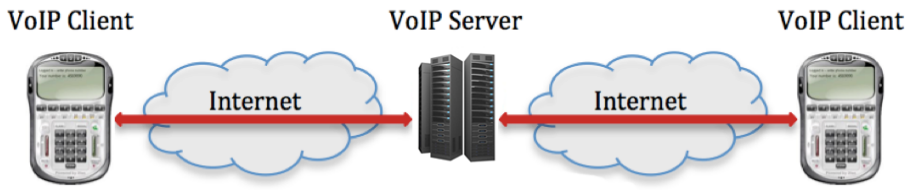


Figure 2.1 VoIP Architecture

### 2.1.1 H.323

In early 1996, the International Telecommunication Union (ITU) published H.323. It is designed to work with local and wide area networks with guaranteed QoS. It provides an establishment for transferring voice, video, and data communications

over an IP network. The H.323 protocol support Secured Real-Time Protocol (SRTP) and Multimedia Internet KEYing (MIKEY). SRTP acquires media confidentiality, while MIKEY is used for authentication (key exchange) [10]

The components in H.323 standard are Terminals, Gateways, Multipoint Control Unit (MCU), and Gatekeepers. Terminals are the end-user devices; these can be IP phones, softphones or Computer or smartphones. VoIP devices or terminals require a system control unit, media transport, media transmission, and packet-based network interface. Gateways are devices that handle communication between different networks with protocol translation and media conversion. The MCU handles conferencing with three or more terminals in a multipoint conference. The Gatekeeper manages a zone that includes terminal, gateways, and MCU. It is responsible for call routing and address resolution. It may also provide call control signaling, call authorisation, bandwidth management, and call management.

Implementing security in H.323 protocol is a complicated process. Using random ports in the H.323 protocol causes a security problem that effects firewalls. Since ports required for H.323 are not set, filtering the firewalls should open possible ports. Consequently, this condition will provide an opportunity for an attacker. The other problem in H.323 is Network Address Translation (NAT), because the IP and the port on the H.323 IP header do not match the NAT.

The H.235 standard [11] provides security for the H.323. Many security issues, such as, authentication, integrity, privacy and non-repudiation have been addressed in the H.235 standard. H.323 can also use a Secure Socket Layer (SSL) for transport layer security [11, 12]. Figure 2.2 presents signalling stack of H.323 and SIP.

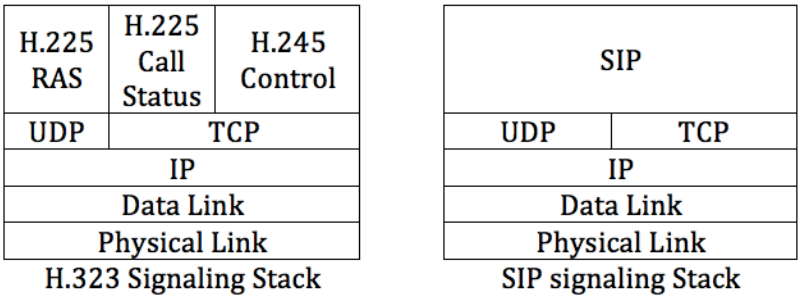


Figure 2.2 H.323 and SIP Signaling Stack [12]

### 2.1.2 Session Initiation Protocol (SIP)

Session Initiation Protocol (SIP) is an application-layer signalling protocol, developed by the Internet Engineering Task Force (IETF) in RFC 3261. It is used to setup, maintain, revise and control the multimedia communication for application layer. The protocol is well designed for easy implementation, ability and flexibility [13]. The primary function of SIP is session initiation, relying on RTP for media transfer [14].

Transport Layer security (TLS) is used to secure SIP hop-by-hop [15]. In hop-by-hop security, it is assumed that the caller and callee trust all proxy servers connecting them to inspect the message bodies in their message. End-to-End security in a SIP is obtained by Secure Multipurpose Internet Mail Extensions (S/MIME). The caller and callee do not trust proxy servers to check their message [16].

Three main components of the SIP system are User Agent (UA), servers, and Location Services (LS).

A **user agent** can be a SIP phone or SIP client software used from a computer or a mobile phone. It creates a SIP request to establish communication with other user agents and sends and receives packets (either video packets or audio packets). User Agent Client (UAC) and User Agent Server (UAS) are part of User Agents (UA). The responsibility of UAC is to initiate a request by sending a message INVITE to the intended recipient, while UAS must receive a request and generate responses to request that have been received [17].

There are three type of **servers** in a SIP system; namely: proxy server, redirect server, and registrar server. In the implementation, a SIP system requires all servers implement Transport Layer Security (TLS), and may also implement IPSec or other lower-layer security protocols [13]. A **proxy server** receives SIP requests from UA or another proxy server and then forwards a request to the destination. It is also responsible for user authentication and charging or billing for a SIP-Based VoIP network [18]. A **redirect server** maintains the database of SIP users. It supports user mobility as it is responsible for responding to requests associated with destination addresses. A **registrar server** saves information about SIP registration requests and

updates the user's location. A **location service** maintains the location database for registered UAs. It contains information about users, such as URIs, IP addresses, scripts, features and other preferences. Commonly, three servers are installed on a single SIP server.

There are three SIP routing methods [19]. The first method is a direct connection between caller and callee. When making a call, this method does not require a SIP proxy or VoIP provider. The identity of the SIP client is detailed in the IP address, allowing the caller would to dial IP address to communicate with the callee. This method is generally used on a homogeneous network, in which the caller and callee are on the same network such as a LAN, WAN or VPN. Figure 2.3 shows a SIP message transaction between caller and callee.

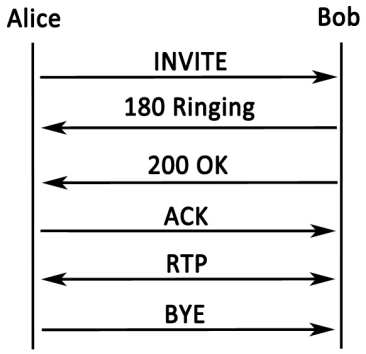


Figure 2.3 SIP method 1

The second method takes place during call setup; the caller communicates with the SIP proxy, which uses Location Services (LS) (integrated with the SIP server) to determine where the call should be routed. Then, the caller receives a message "302 Moved" from the SIP Server. After the above session, the SIP message exchanged in the second method is same as in the first method.

In the third method, the SIP proxy interacts with the location service to forward an INVITE message from caller to callee. In this case, the SIP proxy is responsible for determining the route from caller to callee. Once the caller and callee are connected, the entire packet voice is transferred through the SIP proxy. Typically, the third method is used on a heterogeneous network, in which the caller and callee are on a different network, sip proxy, or VoIP provider. Figures 2.4 and 2.5 describe the second and third SIP method.

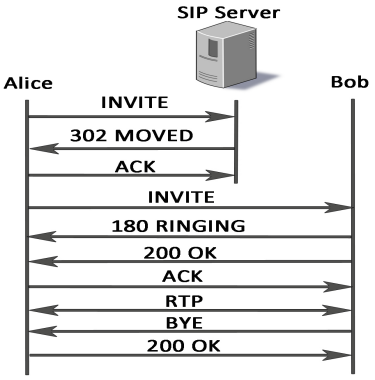


Figure 2.4 SIP method 2

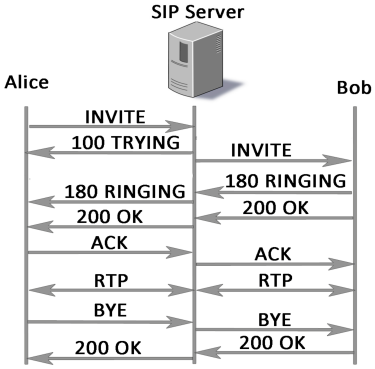


Figure 2.5 SIP method 3

The SIP request-response method is designed in the same way as Hypertext Transfer Protocol (HTTP) method. The SIP request message is described in table 2.1, and the SIP request is replied to with one of six SIP responses codes, as shown in table 2.2.

Table 2.1 SIP Request [20]

| SIP Request | Description |
|---|---|
| INVITE | Initiates a call signalling sequence |
| ACK | Confirms that the client has received a final response to an invitation |
| OPTIONS | Provides capabilities information, such as voice bit rates supported |
| BYE | Terminates a session / release a call |
| CANCEL | Cancel a pending request |
| REGISTER | Sends information about a user's location to the SIP registrar server |

Table 2.2 SIP Responses Codes [13]

| SIP Responses Codes | Description |
|---|---|
| 1xx | Provisional – request received, continuing to process the request |
| 2xx | Success – the action was successfully received, understood, and accepted |
| 3xx | Redirection – further action needs to be taken in order to complete the request |
| 4xx | Client error – the request contains bad syntax or cannot be fulfilled at this server |
| 5xx | Server error – the server failed to fulfill an apparently valid request |
| 6xx | Global failure – the request cannot be fulfilled at any server |

### 2.1.3 Real Time Protocol (RTP)

The Internet Engineering Task Force (IETF) developed a Real Time Protocol (RTP) in 1993 and first published this in 1996 as Request for comments (RFC) 1889; in 2003, this was superseded by RFC 3550. RTP is a common Internet application protocol, providing end-to-end network transport functionality, which supports interactive multimedia or transmission of real-time data such as telephone and video teleconferencing, and television services over multicast or unicast network service [21]. Two-way phone calls are multicast audio; therefore, RTP can be used for IP telephony or VoIP. In many applications, RTP is used with TCP, but not in VoIP, as RTP provides end-to-end streaming and delivery services over UDP [20].

RTP comprises two parts; these are the data and control part. The data parts are Real Time Protocol (RTP) and the control parts are Real Time Control Protocol (RTCP). The RTP conveys data with real-time properties. It includes timing reconstruction, loss detection, security, and content identification. Meanwhile, RTCP is mainly used to monitor the quality of services (QoS), to deliver information about the participants in on-going sessions and to manage synchronisation. It provides support for applications, such as real-time communications. Source identification, multicast-to-unicast translators, and different media stream synchronisation are supported by RTCP [21].

RTP provides services which include payload type identification, sequence numbering, time stamping and delivery monitoring [21]. **Payload type identification**

defines the type of RTP payload or indicates the kind of content being carried. Some payload type is static and can only be used for identification type; however, in a newer version, it can also be dynamic and used to assign a control protocol, such as payload type in SIP. **Sequence numbering** is used to synchronise a packet to sender and receiver. Sequence numbering is mainly used to detect losses or out-of-sequence packets. Sequence numbers increase by one for each RTP packet transmitted. **Time stamping** refers to the presentation time of the content being carried by the Protocol Data Unit (PDU). It is used to place incoming video or audio packets in the correct temporal order. It is most useful for video, but is also used for voice sampling rate. Time stamps increase in accordance with the time packet sent. **Delivery monitoring** – Clients (caller and callee) send Real Time Control Protocol (RTCP) packets in an RTP session to determine quality and network conditions if there are RTP packets, which are lost or contain errors.

There are five types of RTCP messages that RTP generates to report on the RTP session; these are, firstly: **Sender report** containing statistics from active senders, it can include transmission and reception statistics. Then **Receiver Report**, which is the statistical report received from those conference participants who are not the active sender. Afterwards, **Source description**, which contains information about the RTP source, including Domain Name Services (DNS) name. **Bye**, which is used to end a RTCP session. Lastly, **Application Specific** containing additionally information that the application would agree to share [20].

The Secured Real-Time Transport Protocol (SRTP) is a security profile for RTP and RTCP. It aims to provide confidentiality, message authentication, and reply protection to clear text RTP traffic [22, 23].

### 2.1.4 User Datagram Protocol (UDP)

The User Datagram Protocol (UDP) [24] and the Transport Transmission Protocol (TCP) [25] are the main Internet transport protocols. In 1980, David P. Reed designed UDP and defined RFC 768. UDP is a connectionless service that provides application-level procedures and an unreliable Internet transport protocol that sends any data packets without guarantee of data delivery and protection because of the

absence of a sending rate control. In addition, it is a simple protocol using minimal overheads, and hence data can be sent immediately [26]. Therefore, UDP has a low latency compared to TCP.

UDP is basically an application interface for IP [27]. It does not perform handshaking mechanism in the same way TCP does and it is focused purely on transmission. The purpose of UDP is to break upstream into a datagram, add a source and destination port information, a length, and a checksum. There are four UDP datagram fields: source port, destination port, length, and checksum.

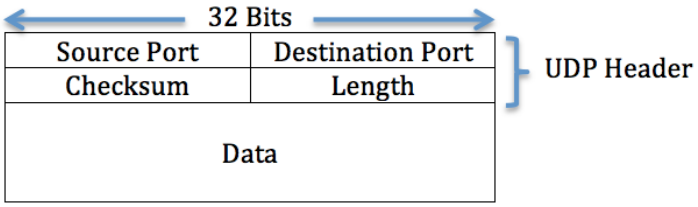


Figure 2.6 UDP header and Data

**Source port** indicates the sending port number used when sending any reply back to the source. The **Destination port** indicates a specific port of application services, such as port 53 for Domain Name Service (DNS). **Length** is the length of a datagram in bytes, including header and data. **Checksum**, UDP checksum is same algorithm as the IP checksum. It is provided as data integrity with minimal protection against transmission error. A Checksum in UDP is optional; if the UDP header does not use a checksum then the checksum should be set at 0. Figure 2.6 shows UDP header with data.

### 2.1.5 Transmission Control Protocol (TCP)

The Transmission Control Protocol (TCP) [25] is defined in RFC 793 from 1981, and provides highly end-to-end reliable connection in packet-switched computer communication networks. It employs flow control concerned with the user capability and congestion control that monitors capacity on the network [26]. TCP delivers data packets in order and ensures data is received completely at the receiver. Hence, TCP is called a connection-oriented protocol.

There are some features of TCP that effect those applications that use it [25, 27]:

**Stream data transfer**, TCP transfers a contiguous stream of bytes over a network. It groups data into TCP segments and transfers them to the destination through IP layer. **Reliability**, TCP uses a sequence number to assign each transmitted byte. Sender sends ACK to the destination and waits for a reply from the destination, if reply ACK is not received within a timeout interval, then the data from the sender is retransmitted. To avoid duplicate packets the TCP receiver rearranges the packet based on sequence number. **Flow control**, receiver TCP sends a reply ACK to sender, the receiver notifies the sender of the number of bytes that can be received without causing any problem (overrun and overflow) to the internal buffer. **Multiplexing** is allows multiple many processes within a single host to use TCP communication facilities simultaneously. TCP provides a set of addresses or ports within each host. A **Logical connection** is a combination of status information for each data stream, which includes sockets, sequence number, and windows size information. **Full Duplex**, TCP provides simultaneous data streams in both directions (sender to receiver or vice versa).

The three-way handshake is a method of TCP used to establish a connection between two participants. SYN, SYN-ACK and ACK are three packets on an established connection process. Host A (sender) sends TCP SYN packet to Host B (receiver). Host B receives Host A's SYN; then replies with TCP SYN-ACK packet. Host A receives Host B's SYN-ACK, then it sends ACK to Host B, then Host B replies with ACK after receiving an ACK packet from Host A. Finally, a TCP socket connection between Host A and Host B is established. Details of a TCP three-way handshake exchange are described in figure 2.7.

TCP protocol is unsuitable for data packets with delay sensitivity because TCP uses a three-way handshake to establish a connection between participants, and if there are packet loss or packet errors, the TCP will resend the packets again increasing delays.

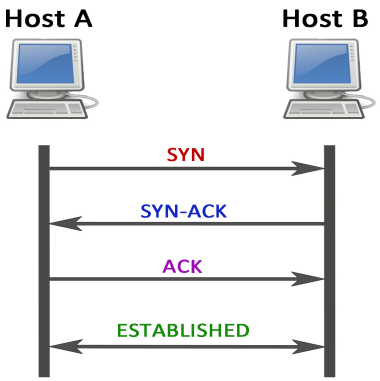


Figure 2.7 TCP three-way handshake

## 2.2 Voice Codec (Coder-Decoder)

In telecommunications technology, codec is an abbreviation for coder/decoder. It is a device or computer software component that compresses or converts analogue voice signals to digital data and vice versa. The purpose of codec is to reduce or compress the file size, so that it can be transferred faster and does not take up a large amount of disk space.

Nowadays, many of the codecs developed are suitable for VoIP, such as G.711, G.729, G.726, Speex and iLBC. In this research, we focus on one codec; the Internet Low Bit Rate Codec (iLBC). In 2004, the iLBC was a narrowband speech codec developed by Global IP Solutions (GIPS). It was a freeware codec with limited commercial value. Since 2011, iLBC has been acquired by Google.inc, and since then free software has been offered freely as open source. It is appropriate for VoIP applications, streaming audio, archival and messaging [28]. iLBC uses speech signals sampled at 8 kHz with frame lengths of 30 ms at 13.3 Kbps and 20 ms at 15.2 Kbps.

The iLBC algorithm deploys with controlled responses to packet losses. The iLBC codec enables graceful speech quality degradation when packet loss or delays occur in network connections. In addition, the iLBC codec delivers a better performance than ITU codecs such as G.729A and G.723.1. The results of iLBC performance can be seen in figure 2.8.

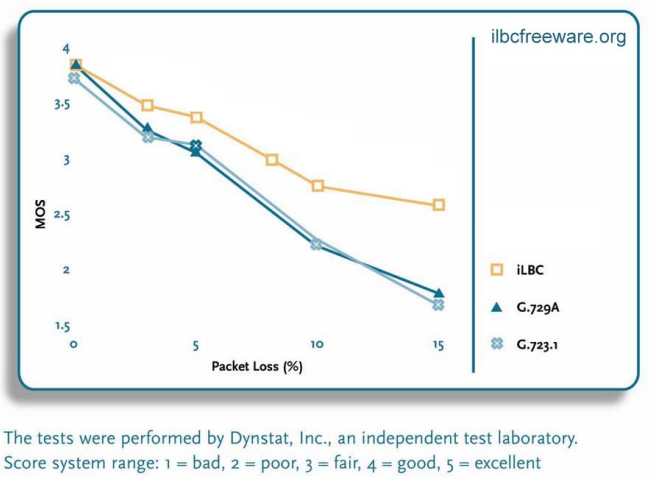


Figure 2.8 Performance (Mean Opinion Score (MOS)) comparison of iLBC, G.729 and G.723.1 [29]

## 2.3 Quality of Services of VoIP

Voice quality is very sensitive to three key performance criteria on the packet network. These are all common QoS factors measured in relation to networks, namely: latency, jitter and packet loss [30]. In 2003, the International Telecommunication Union - Telecommunication (ITU-T) announced recommendation G.114, for one-way transmission time. ITU-T recommends up to 250 ms one-way latency for interactive communications. Delays between 150 ms and 400 ms continue to be acceptable for long distance communications such as Berlin – Germany to New York – United States [31-33]. Moreover, average one-way jitter should be less than 30 ms [34, 35], and packet loss can be up to 5% [36, 37].

### 2.3.1 Latency

**Latency (end-to-end)** is the total time required by data packets or voice packets to reach their destination. There are several factors causing high latency in a network, such as distance from source to destination and the bandwidth of the network. Total latency or end-to-end latency includes propagation delay, transmission delay, queuing delay, codec processing delay, packetisation/depacketisation delay, and play-out buffer delay. There are brief explanations provided about delays on the network.

Table 2.3 ITU-T G.114 recommendation – propagation delay [33]

| Transmission or processing system | Contribution to one-way transmission time | Remarks |
|---|---|---|
| Terrestrial coaxial cable or radio-relay system: FDM and digital transmission | 4 μs/km | Allows for delay in repeaters and regenerators |
| Optical fibre cable system, digital transmission | 5 μs/km | |
| Submarine coaxial cable system | 6 μs/km | |
| Submarine optical fibre system:<br>Transmit terminal<br>Receiver terminal | <br>13 ms<br>10 ms | <br>Worst case |
| Satellite system:<br>400 km altitude<br>14000 km altitude<br>36000 km altitude | <br>12 ms<br>100 ms<br>260 ms | Propagation through space only (between earth stations) |

**Propagation delay** or media latency is the time taken to transfer a data packet from source to destination using media transmission. This depends on the link length (the physical distance of the communications path) and the propagation speed over the specific medium such coaxial cable (4 μs/km); optical fibre cable (5 μs/km) or satellite system (12 ms on 400 km altitude) [33]. Table 2.3 presents the one-way propagation delay in ITU-T G.114 recommendation.

**Transmission delay** is the time required to put all packets' bits into a link or network and is known as packetisation delay. It has nothing to do with the distance between sender and receiver. **Queuing delay** is the time spent by a packet in queues at input and output ports prior to processing. In other words, it is caused by queuing packets during the transferring process and is mainly due to congestion in the network. The **Codec processing delay** consists of codec's algorithmic delay and look-ahead delay. This delay is the time required for compressing and converting an analogue signal to a digital one. **Play-out buffer delay** is the time taken to reach the play-out buffer at the receiver end.

In the implementation, there are two ways to determine the end-to-end latency of the network. The first is by measuring the transference time of data by sending a packet and then waiting for a response from the destination. This is called round trip time (RTT) or two-way latency. This refers to the time of the delivery of data packets

from the source until the source receives a response from the destination. End-to-end latency is obtained by the following equation:

$$Latency_{end-to-end} = RTT/2 \quad (1)$$

The second approach is by capturing the sending time and receiving time at the source and destination. In this way, the time at source and destination must be synchronised; due to the accurate time at both places proper latency will be obtained. The end-to-end latency can be obtained as:

$$Latency_{end-to-end} = R_n - S_n \quad (2)$$

### 2.3.2 Jitter

**Jitter** is the variation in time between packets sent and packet arrival, it is caused by the difficulties that exist in a network such as the distance between sender and recipient, bandwidth and route changes. Jitter is a key measure of QoS in VoIP networks. According to Tim Szigeti, average one-way jitter should be less than 30 ms [34]. In the study one-way jitter is measured according to jitter calculation in Wireshark using the equation below:

$$D_n = \left(\left(R_n - R_{(n-1)}\right) - (S_n - S_{n-1})\right) \quad (3)$$

then,

$$J_n = J_{n-1} + \frac{|D_n| - J_{n-1}}{16} \quad (4)$$

where,

$D_n = \text{D}elay\ of\ n\ packet\ (ms)$;
$S_n = Sending\ time\ of\ n\ packet(ms)$;
$R_n = Receiving\ time\ of\ n\ packet\ (ms)$;
$J_n = Jitter\ of\ n\ packet\ (ms)$;

### 2.3.3 Packet Loss

**Packet Loss** is when packets arrive too late or do not arrive for processing at the destination [37]. Packet loss can be caused by many different factors such as overloaded links, physical media errors, low link quality and others. Voice quality lead to problems when packet loss exceeds 5% of the total packet [38].

In the case of a voice packet, packet loss is said to occur when the voice packets arrive at the destination exceeding the recommended maximum latency of 400 ms. Percentages of packet loss are calculated by dividing the number of packets lost with the total number of packets. An acceptable quality call is one that meets the recommendations of QoS.

## 2.4 Anonymous Systems

Pfitzmann and Kohntop introduced the most common definition of anonymity in an information community, in their paper [39] "Anonymity of a subject means that a subject is not identifiable within a set of subjects, the anonymity set."

One of the advantages of an anonymous communication is for hiding information; it can be used to hide information about who is calling whom. Anonymous communication may conceal the identity of the caller or callee and the network address (relationship), such as the IP address from unauthorised surveillance.

There are three types of anonymity, namely sender anonymity, receiver anonymity and relationship anonymity. Sender anonymity is when information about the sender is hidden but that of the receiver may not be. Receiver anonymity is when information about the receiver is hidden. Relationship anonymity, also called unlink-ability, is when the connection between the sender and the receiver cannot be tracked or identified. Even where information about sender and receiver is known, the fact that they are communicating with each other cannot be detected [40].

Privacy protection in SIP is divided into four classes: (1) where the caller's absolute anonymity, the identity of the caller is hidden to all network components such as caller and callee providers and even to callee; (2) where the caller's

eponymity only to the callee – the identity of the caller is hidden to the callee, (3) where the caller's eponymity only to his/her provider – the identity of the caller is hidden to his/her provider, and (4) where caller's eponymity only to callee's provider – the identity of the caller is hidden to callee's provider [41].

In most cases, the anonymous system is divided into two classes. The first is an anonymous system with high latency and the second is a low latency anonymous network [42]. For instance, crowds is one of the high latency anonymous system, whereas, JAP, Tor, PipeNet, and Anonymizer are low latency anonymous systems. Below is a brief outline of the characteristics of high and low latency anonymous system.

### 2.4.1 Crowds

In 1998, Michael K. Reiter and Aviel D. Rubin introduced a new anonymity system for web transactions, called Crowds. The Crowds aims to protect users' privacy when accessing websites; assuring web browsing anonymity, by preventing websites from identifying users by concealing each user as a member of the Crowds [41, 43, 44]. However, use of Crowds does not provide anonymity from global eavesdroppers [45] and nor can it defend against denial of service (DoS) attacks by rogue crowds members [46]. The basic idea of The Crowds is "blending into a crowd" – a web transaction will hide with other crowds members [46, 47]. Since then, the Crowds became one of references on anonymity system. Since its origination, Crowds has become a well-known anonymity system, and the Crowds concept can be understood by viewing the following figure.

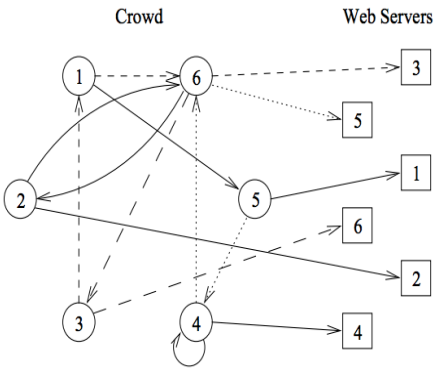


Figure 2.9 Crowds architecture [46]

Crowds consist of client and servers as central crowds. All clients or nodes within in crowds are called “jondos”. The word “jondo” is derived from “John Doe” which emphasizes the anonymity of the network users [48, 49]. Each jondos is connected with a central server, where it receives a list of the crowds members. Maintaining anonymity in crowds, is achieved because each jondos forwards a web request from other randomly selected jondos in the crowds. That jondos sends the request direct to the destination website or forwards it on to other jondos. In this last case, the step can be repeated by forwarding to the next crowd’s member (jondos). This method prevents an adversary, or even other crowds members from determining the identity of the origin initiator [46-48]. Once a path is chosen, all communication from the sender to the receiver will use that same path within a 24-hour period [44]. Messages between jondos are encrypted with private keys. A private key is created for each jondos when the jondos establishes a connection with the central server.

### 2.4.2 Java Anon Proxy (JAP)

Java Anon Proxy (JAP) also called JonDonym was developed under the auspices of a project under taken at Dresden Technical University, Regensburg University and Schleswig-Holstein Privacy Commission. JAP is a proxy system with a single static IP address used by many JAP clients/users. It makes web browsing untraceable. The idea of JAP is a Mixes network. An anonymous group provides a Mixes cascade run by independent organisations. This is different from peer-to-peer based anonymous networks, such as The Onion Routing (Tor) whose relays are themselves anonymous. Figure 2.10 shows entire anonymous service system used by JAP. The network consists of JAP (installed on the user’s computer), mix-server (anonymizing intermediaries), cache proxy, and an info service [50].

Maximum anonymity in JAP is achieved if there are many JAP users on the cascade or JAP server. However, numerous users on a JAP server will decrease the bandwidth and transfer rate available for each user, meaning that the latency on the network will also increase [47, 51].

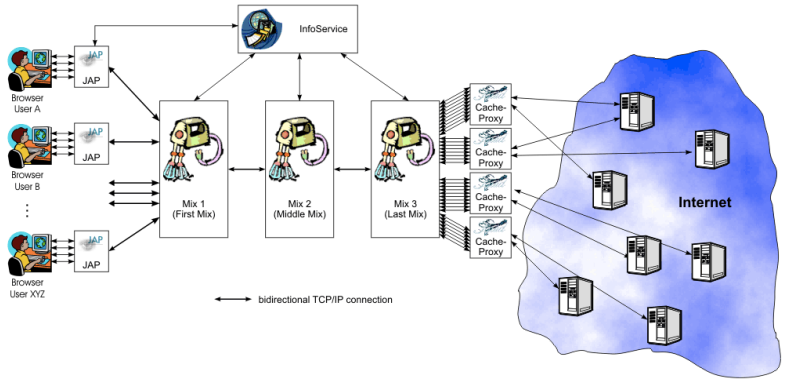


Figure 2.10 The concept of anonymous service [50]

### 2.4.3 The Onion Routing (Tor)

The Onion Routing (OR) research began in late 1995 at the Naval Research Laboratory (NRL) [52]. OR is a low-latency anonymous system that is resistant to eavesdropping and traffic analysis [53]. It aims to conceal communication between the sender and destination. The sender communicates with the destination via several routers. This means that the eavesdropper has no information about who is calling whom.

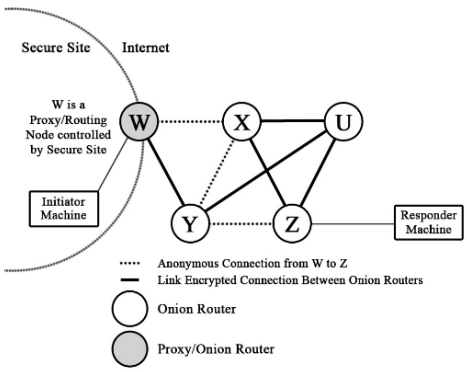


Figure 2.11 Onion routing topology [54] [48].

At the beginning of the OR development, a single malicious relay on the OR network could record traffic between the initiator/sender and the destination/receiver, and may then use it to decrypt the traffic. Also at that time, the OR allowed intermediate relays to create their own onion routers to the next relay on the route, as selected by a sender when there is no direct connection to the next relay available

[55]. In first-generation onion routing, it was necessary to acquire a separate proxy for each application [51]. Figure 2.11 illustrates the topology of an Onion Routing (OR).

The Onion Routing (Tor) – the second-generation of OR – is a circuit-based low-latency anonymous communication service that only supports TCP streams over the Internet. It aims to thwart attackers from identifying single or multiple communication links to or from single user [51]. It is a Socket Secure (SOCKS) server supporting SOCKS 5, which hides the client from their destination. The Tor network has been added perfect forward secrecy, congestion control, directory servers, integrity checking, configurable exit policies, and practical designs for location-hidden services via rendezvous points, when all features are not found by Onion Routing (OR). The Tor operates on the real-world Internet. It does not have a requirement for special privileges or kernel modifications, and require little synchronization or coordination between nodes, providing a reasonable trade-off between anonymity, usability, and efficiency [51]. Nowadays, Tor is a free software P2P network most widely used to achieve anonymity on the Internet [56]. It is the most popular anonymous communication network, and has an estimated over 500,000 users, occupying more than 3000 network relays, and about 2000 MBps of total bandwidth in July 2013 [57, 58]. Furthermore, the Tor network is well supported by Tor project forum [59].

Tor encrypts data multiple times and it is decrypted as it travels over the network a layer at a time: much like peeling an onion [60]. Tor clients send data packets to volunteer proxy routers worldwide, to hide the location of the sender and the recipient from anyone conducting traffic analysis or network observations.

The Tor client receives the relay list from the Tor directory server. It then selects three relays: an entry relay, a middle relay, and an exit relay in an unpredictable manner. Data from the sender will then be encrypted using a private relay key, as has been selected. The first data is encrypted using a key from the exit relay, then by using the middle relay's key the last encryption is performed using the entry relay's key. After this, data from the Tor client is sent to the entry relay. On arrival at the entry relay the data is decrypted using a private entry relay key. Therefore, on entering the relay, data is secured with two private keys (middle and exit relay's keys). Then the entry relay forwards the data to the middle relay. In the

middle relay, the data is decrypted using the middle relay's key and then the data is transmitted to the exit relay. At the exit relay, the data is sent without encryption to its final destination. So the exit relay is the sender from the perspective of the actual destination (receiver). Figure 2.11 depicts Tor network architecture.

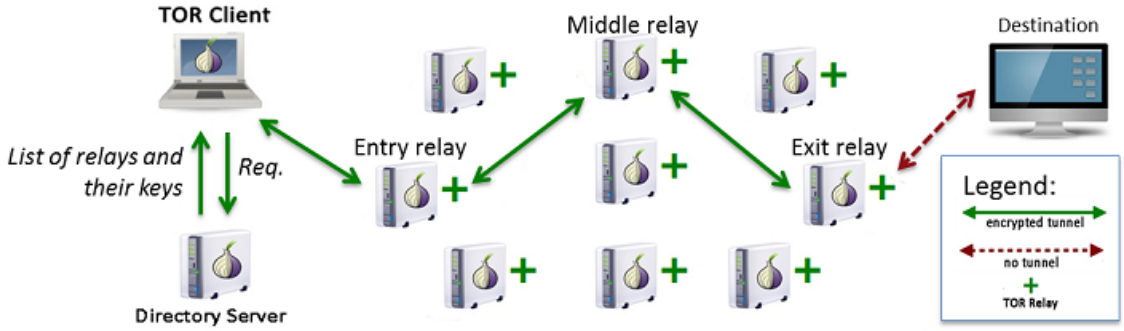


Figure 2.12 Tor network architecture

Tor is a network of volunteer-operated routers that enables users to communicate privately in the presence of eavesdroppers who have local (non-global) views of the Internet [61]. It maintains anonymity by selecting connection relays at random, and also replaces relay connections every 10 minutes.

### 2.4.4 PipeNet

PipeNet [62] was described by Wei Dai in 2000. It is a simple anonymous protocol that provides private protection against traffic analysis by anonymous packet forwarders. It uses three or four intermediate nodes to establish a connection between sender and receiver. The basic idea of PipeNet is a virtual link encryption. This establishes a rerouting pathway to deliver the packet [44].

PipeNet is similar to onion routing, and is a low latency anonymous system that heightens anonymity. It is an ideal anonymous architecture system. However, a single user is able to disconnect from the network by not forwarding messages [51]. In terms of implementation, PipeNet has never been deployed on a large scale network such as on the Internet, as the packet loss of PipeNet is extremely large [63].

### 2.4.5 Anonymizer

Anonymizer [64] is a simple proxy-based service which uses a single centralised anonymous proxy; it acts as an intermediary and offers privacy protection for a client's computer from the rest of the Internet. Therefore, it has a relatively low delay and also low anonymity level compare to sophisticated anonymous network; the end-to-end relationship is not anonymous with regard to Anonymizer itself [65]. Clients use Anonymizer for many reasons, such as bypassing censorship applied in some countries, preventing identity theft or protecting data when browsing the Internet.

Unfortunately, at the moment, anonymizer servers are only available in the U.S. Therefore, latency is high for communication between continents such as communication between a caller in U.S and callee in Germany.

## 2.5 Probability of Attackers

Anonymity is an essential requirement for many applications, which are transferred to open network; Internet protocols. It protects the user's identity with in a variety of ways; in particular sender anonymity (protect the identity of the sender), receiver anonymity (protect the identity of the receiver) and relationship/unlinkability anonymity (protects the link between the sender and the receiver).

Several papers have described the anonymity degree of anonymous network systems, such as the anonymity degree in MIX and Crowds network [66], peer-to-peer networks (chord) [67], and anonymous communication systems [44]. In general, a degree of anonymity calculation is based on the Shannon Entropy. In 1948, Claude Elwood Shannon introduced Shannon entropy - a formula of probability in his journal "A Mathematical Theory of Communication" [68]. The anonymity degree calculation aims to determine whether the attackers can identify the initiator or sender of a message on the network. However, each anonymous system provides a different degree of anonymity and to measure the degree of anonymity is a complicated task. The formula for calculating Shannon Entropy is:

$$H(X) = -\sum_{i=1}^{N} p_i \, log_2(p_i) \qquad (5)$$

Let X be the discrete random variable, $H(X)$ is entropy of X event and $N$ is the number of honest relays in the network (anonymity set), *pi* is the probability that the attacker will break anonymity. Let $H_M$ be the maximum of entropy for the system to be measured, $H_M$ can be calculated as:

$$H_M = log_2(N) \quad (6)$$

The attackers learns the possibility of attacking the system, and it can be expressed as the maximum entropy of the system ($H_M$) decreased by the entropy of the system after attacks ($H(X)$). We can normalise the values by dividing by the maximum entropy ($H_M$). Afterwards, the degree of anonymity in the system can be defined as:

$$d = 1 - \frac{H_M - H(X)}{H_M} = \frac{H(X)}{H_M} \quad (7)$$

The main question concerns whether the entropy model can be used to measure anonymity in the Tor network. According to Paul Syverson [69], the degree of anonymity in the Tor network cannot be measured using the Shannon Entropy method. This entropic method has failed to communicate capabilities to adversaries, regarding how much information can be acquired from the Tor network; thus, the entropic conception of anonymity leads to the assumption of an anonymous system and adversary model as irrational in practice [69]. Another reason for this is that the anonymity of the Tor network is difficult to measure using the entropy method because the actual number of Tor users in a certain time frame is unknown; such that, available information only estimates the number of Tor users.

Defining the capabilities of an adversary is one of the challenges when designing anonymous communication systems. An adversary may be an observer capable of observing a connection incapable of initiating connections (e.g a sniffer on an Internet connection). Another adversary capability is as a disruptor; i.e. delaying or even corrupt traffic on a link. The adversary may also be a hostile user who initiates or destroys connections. The adversary controls relays (compromised relays) used as connections between the source and destination. It can manipulate the connections as well as create new connections [70].

The adversary on a Tor network is a compromised relay. The Tor network is particularly vulnerable to the Global Passive Adversary (GPA) [51, 70]. The GPA model can observe all traffic on a link in the system. Therefore, the capabilities of GPA are too strong for the Tor network to realistically handle attacks [69]. Therefore, we have assumptions about the adversary on the Tor network. The adversary on a Tor network should compromise all the relays on a network that connects the sender and receiver, then the adversary can know who is talking to whom (relationship anonymity). Therefore, if one of the relays used to connect the sender and receiver cannot be controlled by the adversary, then an anonymous Tor network is maintained. Figure 2.13 show the path link on the Tor network has been controlled by the adversary,

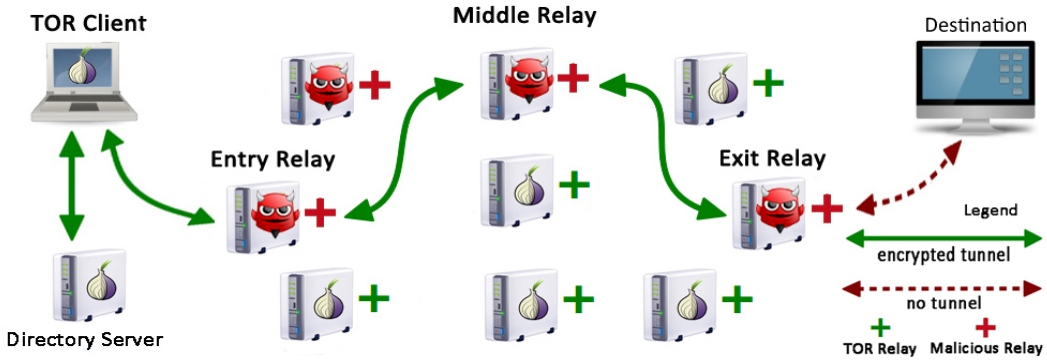


Figure 2.13 The adversary controls the path link on the Tor network

If we assume that attackers will know the original initiator/sender, original destination/receiver, and that they will also discover whether the sender and receiver are communicating between themselves, if the attackers take control of all the relays being used as a transmission media. Then the probability of attackers at the entry relays is $\frac{N_c}{N_r}$, for subsequent relays, the attackers probability will be obtained by $\frac{N_c-(l-1)}{N_r-(l-1)}$, so the probability of attackers on Tor network within two relays ($l = 2$) and three relays ($l = 3$) are:

$$P_{a2} = \frac{N_c}{N_r} \times \frac{N_c-1}{N_r-1} = \frac{{N_c}^2-N_c}{{N_r}^2-N_r} \quad (8)$$

$$P_{a3} = \frac{N_c}{N_r} \times \frac{N_c-1}{N_r-1} \times \frac{N_c-2}{N_r-2} = \frac{{N_c}^3-3{N_c}^2-2N_c}{{N_r}^3-3{N_r}^2-2N_r} \quad (9)$$

Where, $P_{a2}$ is the probability of attackers on a Tor network within two Tor relays, and $P_{a3}$ is the probability of attackers on a Tor network with three relays. Comparisons between the probability of attackers with two relays and three relays are described in chapter 4.

## 2.6 Open Virtual Private Network (OpenVPN)

A Virtual Private Network (VPN) is a network that uses a public communication network such as the Internet, to provide a secure network tunnelled through another network. VPNs are commonly used for implementing secure point-to-point communications. In our experimentation, we use OV as a VPN. This has three functions, the first applies the OV as an encapsulation method; the OV uses TCP streams to transfer UDP streams over the Tor network. The seconds, OV is used for identification VoIP users (identity of VoIP users are OV's IP address), and another function is that the OV uses an encrypted tunnel from sender to destination, thus communications have end-to-end security.

### 2.6.1 Encryption

VoIP over OV can insure security. OV uses the OpenSSL library to encrypt both data and control channels. All voice traffic is encrypted and then sent over the OV. The OV server and client will create an encrypted communication network (encrypted tunnel). After the tunnel has been established, the VoIP client will communicate with other VoIP clients through the encrypted tunnel. In this scenario, the VoIP client and the OV client will be on the same box/machine and the OV server will be installed on the VoIP server machine.

### 2.6.2 Authentication

Authentication in OV can be established in several ways, such as using pre-shared keys and certificate-based authentication. Pre-shared secret key authentication has the benefit of simplicity, being the easiest authentication method in OV. Certificate-based authentication is the most secure form of authentication in OV, and

it relies on RSA certificates and keys. It is built using OpenSSL command. It is included in OpenSSL distribution. Moreover, the common name and email address of the certificate holder is another field secured field by RSA certificates.

### 2.6.3 Security

VoIP over OV achieves 3 security goals. These are confidentiality, integrity and availability.

**Confidentiality** – the data that is transferred over OV should only be available to the authorised person. Initially, when the OV server and the client were configured, both were supplied with each other's keys. On the client side, there is the certificate from the server, the client's certificate and a key. Hence, only a client with a certificate and a key can communicate with the OV server.

**Integrity** – the data transferred must not be altered between the sender and the receiver. Since a secured tunnel is created, data transfer will not be affected.

**Availability** – the data transferred must be available as needed. After a connection is established between the OV client and the server, data traffic can be transferred [71].

## 2.7 Time Series Analysis

Forecasting is never accurate, but it is an important step to ensuring the continuity of activity as well as network planning. One of the methods applied to predicting the future is time series analysis. A time series analysis is a method employed to predict an event by observing a response variable at regular time periods (e.g, hourly, daily, weekly, monthly, quarterly or annually) and variables measured over time should be sequential. A time series aims to acquire the expected patterns to predict the future events of the variables.

Basically, time series data has three fundamental behaviours; these are a trend, seasonal, and random variation. **Trend** shows the long-term movements in the data; it might involve a higher or lower value over a longer period of time [72] such as

population increases or decreases, change in incomes, development of technology, and/or changing consumer preferences. **Seasonal** can be identified by analysing daily, weekly, monthly and even movements over multiple years in historical data. Patterns can be recognised when they repeat over time, such as transaction activities in a supermarket, restaurant, or daily traffic volume. **Random** variations refer to residual variations, which are unpredictable and difficult to identify; such as prediction of a major strike or a war.

Various disciplines employ time series data as forecasting method, including mathematical statistics, finance, meteorology, communications and computer science, etc. In Tor network, forecasting uses the trends of time series analysis for predicting Tor network capacity, including Tor users, relays and bandwidth. In this research, Tor network forecasting was performed by applying a linear trend and three non-linear trends (quadratic, cubic and exponential).

A linear trend is a simple function that forms a straight line based on historical data. A straight line is used to give future predictions and the line might then be a straight upward or downward line. In general, the form of a linear trend is as presented in equation 10. In some cases, time series data cannot be analysed by observing a linear trend; such cases occur because time series data initially has a different gradient with subsequent data. Thus, these cases are better analysed according to non-linear rather than linear trends. Below are the formulas for all non-linear trends; formula 11 is quadratic, formula 12 is a general formula describing a cubic trend and formula 13 is an equation for an exponential trend.

$$T_t = a + b.Y_t \tag{10}$$

$$T_t = a + b.Y_t + c.Y_t^2 \tag{11}$$

$$T_t = a + b.Y_t + c.Y_t^2 + d.Y_t^3 \tag{12}$$

$$T_t = ab^y \tag{13}$$

where,

$T_t = Trend\ value\ of\ period\ t$;
$a = Constant\ of\ trend\ value\ at\ base\ period$;
$b, c, d = Coefficient\ of\ trend\ line\ direction$;

$Y_t = an\ independent\ variable\ (represents\ time\ variable)$;

Accuracy and control of forecasting is important to minimise forecast error. A forecasting method can be selected by evaluating forecast accuracy using the actual time series data. The two commonly used measures to insure forecast accuracy are Mean Absolute Deviation (MAD) and Mean Squared Error (MSE). Forecasting using the lowest values of MAD and MSE provides the best-forecast accuracy. According to Stevenson [72], formulas used to compute MAD and MSE are:

$$\text{MAD} = \frac{\sum_{t=1}^{n} |y_t - F_t|}{n} \quad (14)$$

$$MSE = \frac{\sum_{t=1}^{n} (y_t - F_t)^2}{n-1} \quad (15)$$

Where,

$y_t = Actual\ data\ of\ a\ given\ time\ period$;
$F_t = Forecasted\ data\ of\ a\ given\ time\ period$;
$n = Numbers\ of\ data$;

## 2.8 Current Attacks on Tor Network

There are several common attacks that occur on an anonymous network, such as Denial of Services (DoS), replay attack, message coding attack, collusion attack, packet volume attack, packet counting attack, message delaying attack, flooding attack, intersection attack, and timing/latency attack [73]. In August 2013, the Tor network was used by the community to attack a target, the attack used was a BotNet attack. It is classified as a Distributed Denial of Services (DDoS) attack.

BotNet is a combination of two words; robot (Bot) and network (Net); it is a malware-compromised machine, which is one of the most serious security threats. BotNets are used for various purposes [74]; either for legitimate or illegitimate activities. In the case of legitimate activities, botnets are used by several IRC bots that have been linked to and set the channel modes on other bots, leading users to protect IRC channels from unwanted participants. Whereas in illegitimate activities, botnets

are used to send spamming mails, stealing personal information, phishing, disseminating malware, and Distributed Denial of Services (DDoS) injection [75].

Botnets are a collection of hundreds, and sometimes even thousands, of the compromised computers from independent networks controlled by a botnet originator, also known as a "bot herder", or a "bot master". In early August 2013, BotNet used Tor network to attack their target or their destination; therefore, the numbers of Tor users increased significantly from fewer than one million users to more than five million users. This resulted in the degradation of QoS across the Tor network. Latency in the Tor network also increased extremely, and this was very detrimental to VoIP users of the Tor network.

## 2.9 Related Work

Although there have been numerous research studies done on VoIP, there has been very limited research into how to anonymise VoIP over Tor. Liberatore et. al. [32] investigated the quality of service performance on anonymous VoIP (aVoIP). The aVoIP [32] proposes a means to provide user privacy in VoIP. It has a similar architecture to the Tor network, but it uses UDP instead of TCP. Thus, aVoIP can be called "Private Tor".

aVoIP is installed and tested on a large distributed overlay network, PlanetLab. PlanetLab is a platform spread throughout the continent that are used to perform test-bed operations and deployed for large-scale networks. aVoIP experiments were performed with 40 proxies in Asia, 49 proxies in America and 121 proxies in Europe. Their experimental results showed that quality of calls with proxies was at a level of 46% acceptability in Asia, 71% in Europe and 86% in America [32]. This aVoIP research described how many relays or proxies were used, but did not mention the total number of aVoIP users or the bandwidth provided by "Private Tor". Therefore, the researchers do not know the ratio between users, relays/proxies and bandwidth.

TORFone was designed to communicate voices via the Internet (make a call). It is similar to Skype, but has some fundamental differences, which are:

It is an open source project, so there is no “backdoor” and bugs are found and fixed immediately. TORFone is decentralised; therefore, it does not require an external server or identity to register (username or number). TORFone’s developer claimed that it provides full confidentiality by using the Diffie Hellman key exchange method with 4096 bits and voice traffic is secured using encryption method AES-256-OCB; it also uses PKDF2+HMAC for authentication. Thus, the attackers cannot listen to a conversation unless they can access participants’ computers. Implementation of TORFone results in up to 2-4 seconds of voice latency, because voice traffic will pass through several relays located around the world [76].

Another anonymous VoIP based on the Tor network is the 1985phone. In June 2013, the 1985phone concept was presented by Jonathan Corbett [77]. It is a peer-to-peer protocol. The 1985phone concept was similar to the Tor concept in that the voice traffic data is transferred via several relays before reaching the destination. 1985phone users were to become relays for the other users. Therefore, the biggest challenge in the implementation of this concept is the limitation of resources, such as the lack of batteries, mobile phone capabilities, and also limited bandwidth for data usage on a cell phone.

# 3 RESEARCH METHODS

As validity and consistency are required when conducting research, a number of orderly steps were designed to meet these requirements. The research methodology used in this dissertation is based on an empirical research template. Empirical research is a class of research methods in which empirical observations or data are collected to answer a particular research question. While primarily used in academic research, it can also be useful for answering practical questions. The aim of this empirical study was to investigate the QoS performance of VoIP in the anonymous network – The Onion Routing (Tor).

This chapter discusses how research was conducted to gather relevant data according to research objectives, in order to answer the research questions stated in Chapter I. In this chapter, the researcher will discuss the approach to the research, the research design, instruments, data collection procedures, and the data analysis processes.

## 3.1 Research Approach

The research approach in empirical research can be divided into two categories of methods, i.e. quantitative research and qualitative research. The **quantitative research approach** involves collecting data in number form and using statistical modelling for data analysis. While, the **qualitative research approach** involves collecting qualitative data such as text, image and sounds with observations, interviews, and documentary evidence [78].

The most common quantitative approaches are experimental, survey or historical data. In experimental research, the researcher applies a treatment and then measures the results (before and/or after) of this treatment. This method can be used to demonstrate a casual relationship between variables. Alternatively, survey research commonly utilizes a set of questions to be asked in a face-to-face interview, using the telephone, mail or via Internet in order to assess thoughts, opinions or the feelings of

the research respondent. Historical data uses existing data, which is then analysed in order to have a pattern in historical data [78].

The most common qualitative research methods are case study and action research. A case study is the collection of observational data in a real world setting, such as on a software development project. While action research is the implementation of the research idea in practice, evaluation of results, and modification of the idea (a combination of an experimental and a case study) [78].

This research uses empirical research methods employing both approaches – qualitative research and quantitative research approaches. Experiments, action research and historical data have been used to gather the data and analyse results. Experiments and action research have been conducted in two scenarios and over three different periods of time. Historical data was also used for Tor network forecasting. The details of experiments and action research are presented in the research design.

## 3.2 Research Design

The research focused on VoIP over the Tor network, referring to relays used only in Europe. Experiments and action research were conducted in two network scenarios. These were VoIP calls experiment, through a Tor network with three relays and VoIP calls through a Tor network with two relays. The experiments were conducted in December 2012, July and October 2013. Both scenarios used OpenVPN which encapsulates UDP and TCP, so that audio packets can be transmitted over the Tor network. The detailed architecture of the experimental scenario can be seen in figure 3.1.

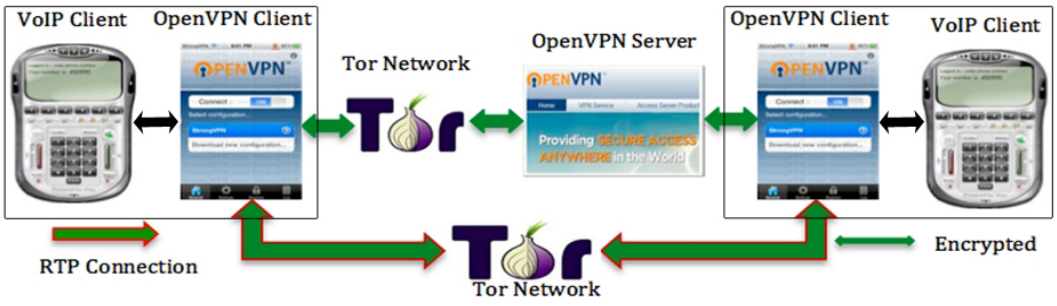


Figure 3.1 VoIP through OpenVPN over Tor network.

One hundred calls were placed to acquire the results for each scenario at different periods of time during the experiment. Figures 3.2 and 3.3 depict Tor network architecture with three and two Tor relays.

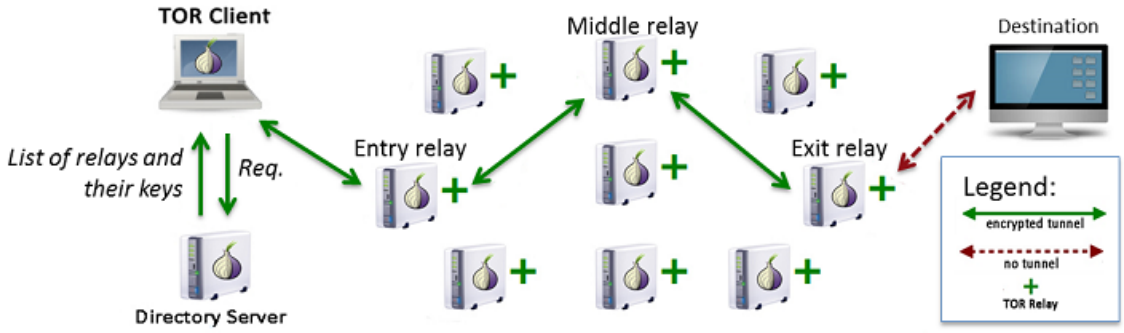


Figure 3.2 Architecture of Tor network with three relays

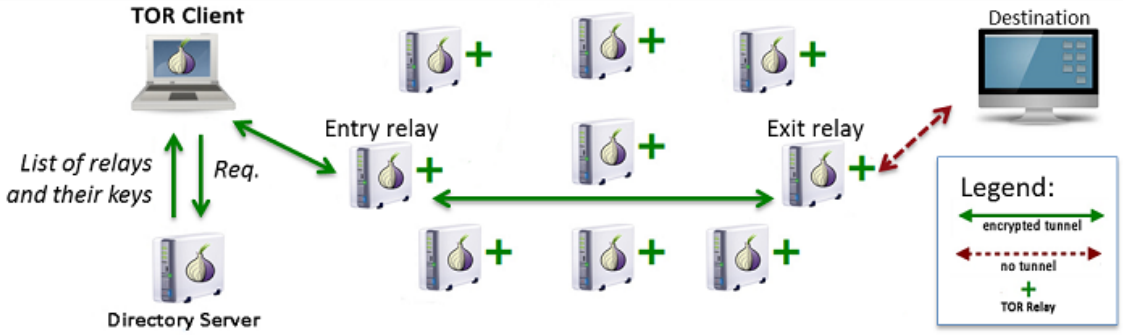


Figure 3.3 Architecture of Tor network with two relays

Figure 3.4 illustrates Alice's communication with Bob through pipeline multi layers; the first layer is a pipe for OpenVPN (OV). In this pipeline, UDP is wrapped with TCP; thus, the RTP packets can be transmitted through Tor network. OV connects Alice and Bob with a private key held by them alone. This ensures the end-to-end security of the communication between them. The second to fourth pipes are the Tor network with three relays and each Tor relay has a private key. Using a Tor network communication between Alice and Bob will be anonymous, because each relay only knows where the messages come from, and where the messages are sent (not a real sender and receiver).

The differences between a Tor network using three relays and one using two relays are the number of pipe layers that encapsulate the UDP packets. When using three relays, there are four pipe layers that wrap UDP packets, whereas in two Tor

relays there are three pipe layers. Figure 3.5 shows Alice's communication with Bob using OV through two Tor relays.

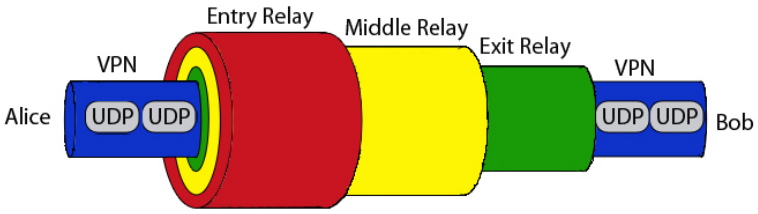


Figure 3.4 VoIP over VPN through three Tor relays

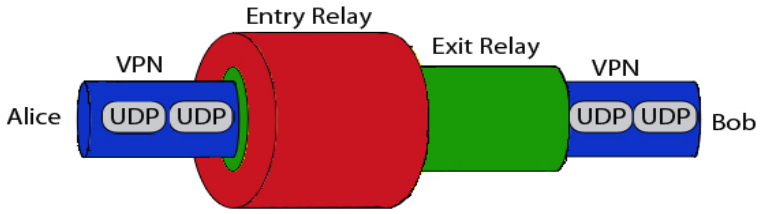


Figure 3.5 VoIP over VPN through two Tor relays

## 3.3 Instruments

In this research, two computers were used as VoIP and OpenVPN clients, and one computer was a VoIP and OpenVPN server. Tor relays in Europe were selected with a bandwidth of more than 2 MBps (high bandwidth).

Specifically, two computers used as VoIP clients, had an Intel quad core processor Q8300 (2.5 GHz) with 4 GB RAM (memory) and Intel Core 2 Duo (2.4 GHz) with 5 GB RAM (memory). Meanwhile, the other computer, which was used as a VoIP and OpenVPN server had an Intel Pentium 4 (3 GHz) with 4 GB RAM (memory).

The software used during this research was the Tor client, Privoxy, Network Time Protocol (NTP), OpenVPN server and client, PhonerLite as VoIP client, Zaitun Time Series (ZTS). We modified the relays with Tor open source code.

Vidalia 0.2.15 as a **Tor client** that was installed onto the computer to connect the VoIP client with the global Tor network. The Tor client acquired a list of Tor relays from the Tor directory server. This meant, the Tor client could randomly select the guard/entry relay, middle relay and exit relay as a pathway. The Tor client had a

SOCKS5 proxy serving as an intermediary for any application to the Tor network. Socket Secure (SOCKS) proxy is an open source socket, which in this setup served as a tool for transferring applications that enable communication with the global Tor networks. Figure 3.6 depicts the connection between the VoIP client and the Tor network.

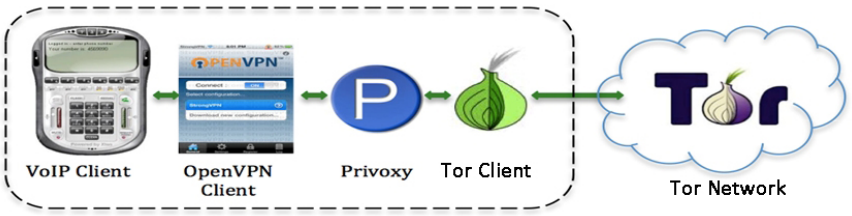


Figure 3.6 Relationship between VoIP client and Tor network

Privoxy is free software and licensed under the GNU GPLv2. It is a SOCKS proxy application that supports SOCKS4 and SOCKS5. It connects the OV client with the Tor client. Privoxy's other functions include: capabilities in advanced filtering to enhance privacy, modifying http headers and web pages, controlling access and removing ads, and other internet junk. Privoxy's configuration is flexible and can be customised according to need, and can also be used as stand-alone or multi-user network [79].

Network Time Protocol (NTP) is a protocol used to synchronise the time on the Internet network. In early 1980, David Mills developed NTP, which has become an Internet standard. The latest NTP standard is the IETF standard, which is set forth in RFC 5905. The latest NTP guarantees the potential accuracy to the tenth of a microsecond in modern network technology with modern workstations and fast LANs connections [80]. It uses Coordinated Universal Time (UTC) to synchronise clock time. Time accuracy on a network is important in situations, such as air traffic control that require accuracy to the microsecond, and in communication systems that need accuracy in terms of time to calculate latency in the network.

Currently Tor networks only transfer TCP in stream-based forms, and in general, audio packets in real-time communication are using UDP stream. Therefore, we cannot directly transmit audio packets over the Tor network. There are several ways to transmit voice packets via the Tor network. One of these is by using encapsulation method. OpenVPN was used in this research as an encapsulation tool. It

has the capability of covering UDP with a TCP stream. In this way, the VoIP client can communicate with other VoIP clients using a direct method call. The virtual IP address of the OV client is used as the identity for the VoIP client. Another advantage of using OV in this research is the addition of end-to-end security, because of the AES encryption method used in the virtual network communication between OV clients.

PhonerLite is a free softphone made by Heiko Sommerfeld, which is easy to use and user friendly. And also, Phonerlite already support many voice codecs such as GSM, G. 711, Speex and iLBC. It can be used for peer-to-peer VoIP calls that IP address used as users identity. Currently, Phonerlite can only run on Windows operating system, and it currently supports encryption methods such as Transport Layer Security (TLS), Secure Real Time Protocol (SRTP), and the Zimmermann Real Time Protocol (ZRTP) [81]. Figure 3.7 shows PhonerLite.

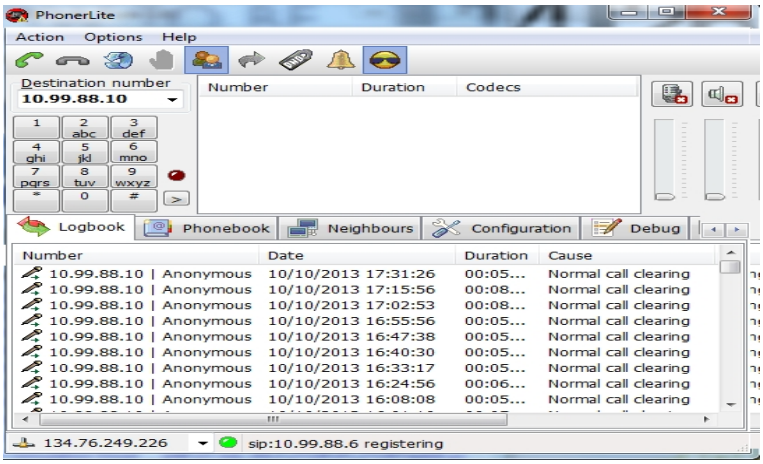


Figure 3.7 PhonerLite – VoIP client

Wireshark was used to capture audio packets on the sender and receiver ends. Wireshark is free and open source software that is useful as a network analyser. The original version of Wireshark was known as Ethereal; however, due to a trademark issue, Ethereal was renamed Wireshark in May 2006. Currently, Wireshark is used in various sectors, such as, education, network analyser, network troubleshooting, and communication protocol development. In terms of functionality, Wireshark works much like tcpdump; however, it has a graphical front-end and is integrated with additional options, such as packet filtering [82]. Figure 3.8 which follows, is a screenshot of the Wireshark network analyser.

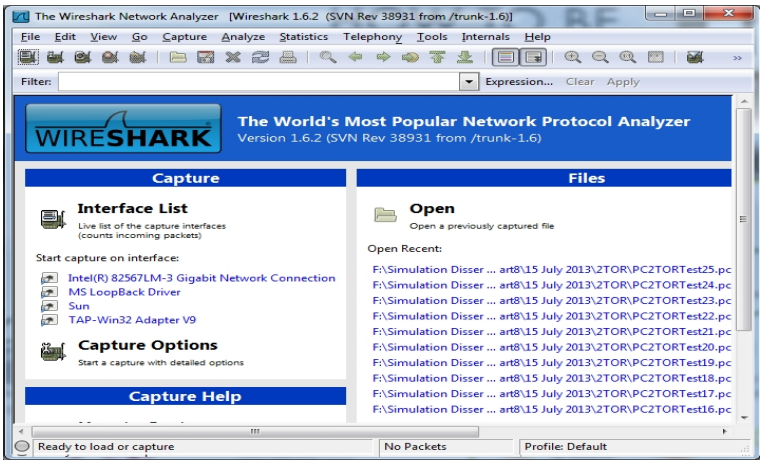


Figure 3.8 The Wireshark Network Analyzer

For Tor network forecasting, Zaitun Time Series Analysis software was used to predict the future of the Tor network. It is free open source software designed to statistically analyse time series data. It was developed by students at Statistics Institute of Indonesia [83].

## 3.4 Data Collection Procedures

Experiments on VoIP calls over Tor networks were tested over three periods - December 2012, July 2013 and October 2013. These calls were tested during the weekday from 9 am to 5 pm. Voice packets were captured 5 minutes after each test. We collected the VoIP packet timestamp on the sender and receiver end. Network Time Protocol (NTP) was used in this research to establish time accuracy between the sender and receiver.

The Tor client was reset for each new VoIP call in order to obtain different relays for each call. Therefore, the path connections used for each call experiment differed from one another. The relays used were randomly selected by the Tor client which was installed on the sender.

Voice packets were captured with Wireshark at the sender and the receiver ends. Then, the data obtained was used to calculate latency on the network. Voice packets, which had latency exceeding the provisions of the ITU, i.e. more than 400 ms, would be assumed to be packet loss. The latency of each packet is then used to

calculate the jitter. In each experiment, a time a call was made and the relays used for the path connection was recorded.

## 3.5 Data Analysis Procedures

In this research, data analysis was conducted to compare the experimental results for each period and analyse the possible causes of variations in the results. Raw data obtained from Wireshark was used to calculate QoS performance in VOIP through the Tor network. Three QoS metrics were calculated – latency, jitter and packet loss. Latency in the network is obtained from packet time as it arrives at the receiver, decreased by packet time transmitted in the sender. Furthermore, average jitter is derived from the variation of latency for each audio packet sent, and packet loss can be obtained if the latency of the packet exceeds the permitted latency of 400 ms.

Statistical analysis procedures were used for prediction about the Tor network prior to December 2013. Four methods of analysis were used in a time series; reporting linear, quadratic, cubic and exponential trends. Mean Absolute Deviation (MAD) and Mean Squared Error (MSE) were used to insure accurate forecasting. In theory, the time series method with the smallest values for MAD or MSE is the method with the highest accuracy in the real data. These data analysis procedures were presented in detail in chapter 4, in the section detailing the results and analysis of Tor forecasting.

# 4 RESULTS AND ANALYSIS

The objectives of this chapter are to analyse and interpret the data collected in the empirical experiment to answer the research questions given in chapter 1. This chapter presents the findings obtained from the VoIP calls over the Tor network with three and two Tor relays and a discussion of Tor network forecasting, and the probability of attackers.

## 4.1 VoIP over Tor Network

In this research, VoIP calls were conducted over the Tor network with two scenarios; i.e. VoIP over a Tor network with three relays (default Tor network) and VoIP over a Tor network with two relays. Experimental data were collected in three time periods; in December 2012, July 2013, and October 2013 on weekday, between 9 am and 5 pm. A hundred calls were captured for each scenario and each of the calls lasted five minutes.

The experiments for VoIP over a Tor network with two relays has the advantage of a lower latency compared to that over three Tor relays, because implementing VOIP over two Tor relays may shorten the length of the voice packets route.

In December 2012, the average total numbers of relays on the Tor network numbered 2,978; from these 177 relays in Europe with a bandwidth of more than 2 MBps were selected. By comparison, in July 2013, there were 3,965 relays on the Tor network; from these 231 relays in Europe with a bandwidth of more than 2 MBps were selected. By, October 2013, the total number of Tor relays was 4,453 relays; from these 298 relays in Europe with a bandwidth of more than 2 MBps were selected. The total bandwidth of the relays captured (or measured) in December 2012 and July 2013 was more than 900 and 1,300 MBps respectively. Meanwhile, the bandwidth of the relays measured in October 2013 reached 1,500 MBps. These results indicated that the total number of relays and bandwidth within the different periods

had undergone a significant increase. Specifically, the overall relays went up by 30% in 7 months, while the number for total bandwidth also rose by 44% over the same period.

Of the average Tor users, in December 2012, there were 802,243 users, and in July 2013 the Tor users declined slightly to 785,903 users. Meanwhile in October 2013, the average number of Tor users increased drastically compared to that over the two previous periods. In this period, the number of Tor users exceeded four million. The increasing number of average Tor users was not proportional to the increase in the number of relays or bandwidth over the experimental periods. Table 4.1 shows the details of Tor network conditions in December 2012, July 2013 and October 2013.

Table 4.1 Tor network condition

| Tor network condition | Dec. 2012 | Jul. 2013 | Oct. 2013 |
|---|---|---|---|
| Average number of Tor relays | 2978 | 3965 | 4453 |
| Average estimate number of Tor users | 802,243 | 785,903 | 4,753,768 |
| Average total Tor bandwidth (MBps) | 2,143.08 | 2,553.85 | 2,661.23 |
| Relays with > 2MBps in Europe | 177 | 231 | 298 |
| Total bandwidth of relays in use (MBps) | 965.66 | 1,391.19 | 1,550.63 |

### 4.1.1 Three Tor Relays

This research measured three metrics of QoS, namely latency, jitter and packet loss. Wireshark was used to capture the audio packets at the sender and receiver ends. The qualities of VoIP calls were considered acceptable if they met the requirements of VoIP recommendation, which included the following criteria: less than 400 ms latency, up to 30 ms jitter and less than 5% packet loss. VoIP calls with three Tor relays were made in December 2012, July 2013 and October 2013.

According to the experiment with VoIP calls with three Tor relays conducted in December 2012, 36% acceptable calls at a 5% packet loss and 21% acceptable calls at 1% packet loss were acquired. Average latency obtained on these experiments was 137.55 ms and 156.98 ms for 1% and 5% packet losses. Meanwhile, the result for average jitter obtained for 1 up to a 5% packet loss differed slightly by less than 1 ms. Table 4.2 below shows the details of the VoIP call experiments performed in December 2012 using three Tor relays.

Table 4.2 The experimental results for December 2012 using three Tor relays

| Tolerated packet loss | 5% | 4% | 3% | 2% | 1% |
|---|---|---|---|---|---|
| Acceptable quality calls (%) | 36 | 33 | 30 | 24 | 21 |
| Av. Latency (ms) | 156.98 | 153.72 | 152.69 | 145.17 | 137.55 |
| Av. Jitter (ms) | 15.46 | 15.03 | 15.15 | 15.13 | 14.97 |

Hundreds of calls were also collected using the three Tor relays in July 2013. The results of the experiment in this period show 20% to 38% acceptable calls for 1% and 5% packet loss. The average obtained for 5% and 1% packet loss was 157.16 ms and 138.93 ms. The average jitter obtained for each packet loss (1% up to 5% packet loss) indicates a slight difference, ranging between 18-20 ms. Table 4.3 shows the details of the experimental results for July 2013.

Table 4.3 The experimental results for July 2013 with three Tor relays

| Tolerated packet loss | 5% | 4% | 3% | 2% | 1% |
|---|---|---|---|---|---|
| Acceptable quality calls (%) | 38 | 34 | 31 | 28 | 20 |
| Av. Latency (ms) | 157.16 | 155.14 | 154.5 | 144.96 | 138.93 |
| Av. Jitter (ms) | 18.58 | 18.97 | 19.22 | 18.82 | 19.23 |

The results obtained from hundred VoIP calls in October 2013 were 11% and 24% acceptable calls with 1% and 5% packet losses. The experiment resulted in an average latency of 179.96 ms at 5% packet loss and 135.23 ms at 1% packet loss. Average jitters in this experiment were 18.59 ms and 17.51 ms at 5% and 1% packet loss. Detailed results for the VoIP calls experiments in October 2013 are presented in table 4.4 below.

Table 4.4 The experimental results for October 2013 with three Tor relays

| Tolerated packet loss | 5% | 4% | 3% | 2% | 1% |
|---|---|---|---|---|---|
| Acceptable quality calls (%) | 24 | 20 | 19 | 15 | 11 |
| Av. Latency (ms) | 179.96 | 169.25 | 166.93 | 143.09 | 135.23 |
| Av. Jitter (ms) | 18.59 | 18.04 | 17.94 | 17.26 | 17.51 |

The results of the VoIP call through three Tor relays show the best results were obtained in July 2013. In this period, the highest number of acceptable calls was 38% at 5% packet loss; this is followed by results for December 2012, when there were 36% calls with 5% packet loss. The experimental results for October 2013 returned the worst results for the three different periods. This was because the number

of Tor users in October 2013 was the highest for the three periods of data collection (experiments). The increase in Tor users in this period was very significant.

In October 2013, Tor network had more than four million users on average. Whereas, in December 2012 and July 2013, there were fewer than one million users. The number of average Tor relays in October 2013 was 4,453 relays. In fact, this number represents an increase in the average number since Tor relays in July 2013, when there were 3,965 relays. Meanwhile in the first experimental periods, the total average number of Tor relays was just 2,978 relays. The average bandwidth in October 2013 increased slightly compare to that in July 2013, which ranged from 2.55 to 2.66 GBps. In December 2012, the average bandwidth was just 2.14 GBps. Table 4 shows a comparison of the Tor network conditions within the three experimental periods. The details for acceptable calls, latency and jitter in the three periods of the experiment are shown in figures 4.1 to 4.3.

Based on the Tor network conditions and the results of the experiments in the three different periods, it is apparent that a significant increase in number of Tor users took place in October 2013; this was not accompanied by a significant increase in the total amount of bandwidth and so may consequently result in high latency on the Tor network. These conditions are unfavourable for VOIP users who use the Tor network to make anonymous calls.

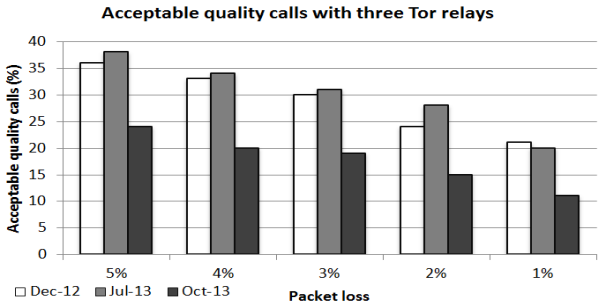


Figure 4.1 Acceptable quality calls with three Tor relays

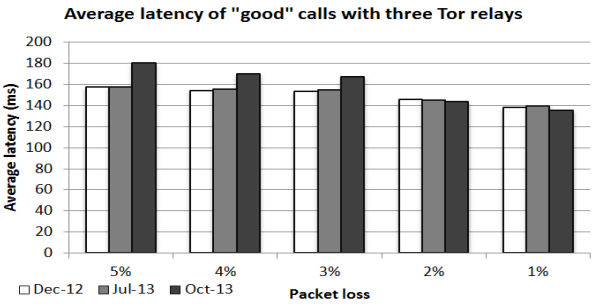


Figure 4.2 Average latency of "good" calls with three Tor relays

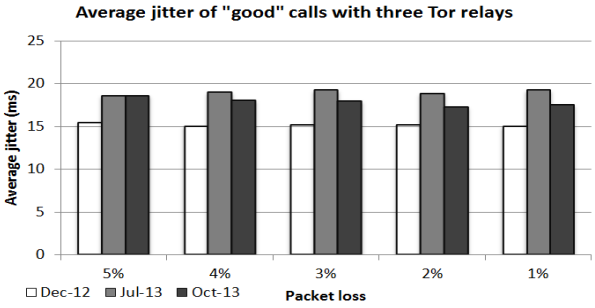


Figure 4.3 Average jitter of "good" calls with three Tor relays

### 4.1.2 Two Tor Relays

The second scenario is VoIP over two Tor relays. Modifying the number of Tor relays as a medium for transferring voice packets from three relays to two relays will provide lower latency. However, using two relays instead of three can reduce the degree of anonymity. Experiments on two Tor relays were performed at the same time of day as experiments with three Tor relays, i.e. in December 2012, July 2013 and October 2013.

The experiments were conducted in December 2012, resulting in 36% acceptable calls at 1% packet loss and 54% calls at 5% packet loss. The average latency for 1% packet loss was 98.83 ms, while that for 5% packet loss was 122.90 ms. Average jitter for 1% and 5% packet loss was in the range of 13 to 16 ms. Table 4.5 shows the details of experimental results in December 2012.

Table 4.5 The experimental results for December 2012 with two Tor relays

| Tolerated packet loss | 5% | 4% | 3% | 2% | 1% |
|---|---|---|---|---|---|
| Acceptable quality calls (%) | 54 | 50 | 48 | 44 | 36 |
| Av. Latency (ms) | 122.90 | 117.08 | 110.71 | 103.55 | 93.83 |
| Av. Jitter (ms) | 15.83 | 15.45 | 15.14 | 14.85 | 13.87 |

Furthermore, the results of the experiment performed in July 2013 show there were 65% acceptable calls with 5% packet loss, but only 45% acceptable calls were obtained with 1% packet loss. Next, the average latencies acquired for 1% and 5% packet loss were 100.01 ms and 117.70 ms respectively. Finally, the average jitter obtained for 1 up to 5% packet loss ranged from 13 to 16 ms. Table 4.6 below provides the details of the experimental data on VoIP over the two relays captured in July 2013.

Table 4.6 The experimental results for July 2013 with two Tor relays

| Tolerated packet loss | 5% | 4% | 3% | 2% | 1% |
|---|---|---|---|---|---|
| Acceptable quality calls (%) | 65 | 64 | 59 | 51 | 45 |
| Av. Latency (ms) | 117.70 | 117.34 | 109.94 | 103.95 | 100.01 |
| Av. Jitter (ms) | 15.45 | 15.45 | 15.08 | 14.43 | 13.62 |

In the study conducted in October 2013, it was found that there were 28% acceptable calls at 5% packet loss and 19% calls at 1% packet loss. Further analysis of the data reveals the average latency for acceptable calls was 5% and 1%, and packet losses were 120.94 ms and 92.25 ms. Regarding jitter, the experimental data shows that the average jitter obtained range from 15 to 17 ms. The results of the experiment for the second scenario in October 2013 returned the worst results for all scenarios in all time periods. Similarly, in the first scenario the worst results also occurred in October 2013. Table 4.7 presents the details of the experimental result with the second scenario in October 2013.

Table 4.7 The experimental results for October 2013 with two Tor relays

| Tolerated packet loss | 5% | 4% | 3% | 2% | 1% |
|---|---|---|---|---|---|
| Acceptable quality calls (%) | 28 | 24 | 22 | 20 | 19 |
| Av. Latency (ms) | 120.94 | 107.15 | 98.23 | 94.44 | 92.25 |
| Av. Jitter (ms) | 16.10 | 16.24 | 15.87 | 15.60 | 15.29 |

In comparison, the experimental results of the second scenario were the same as the results for the first scenario, in which the best results were those for July 2013. These experimental results indicated there were 54% acceptable calls in December 2013, 65% calls in July 2013 and only 28% calls in October 2013, with 5% packet loss. The average latency in December 2012 amounted to 122.90 ms for 5% packet loss and 93.83 ms for 1% packet loss. The average latency obtained for the experiment in July 2013 amounted to 117.70 ms, obtained for 5% packet loss and

100.01 ms for 1% packet loss. Meanwhile the average latency in October 2013 was 120.94 ms for a 5% packet loss and 92.25 ms for a 1% packet loss. For average jitter, there was little difference in each experimental period which ranged from 13-16 ms. A comparison of the results obtained from the experiments on VoIP over two Tor relays in December 2012, July 2013 and October 2013 are shown in figures 4.4 to 4.6.

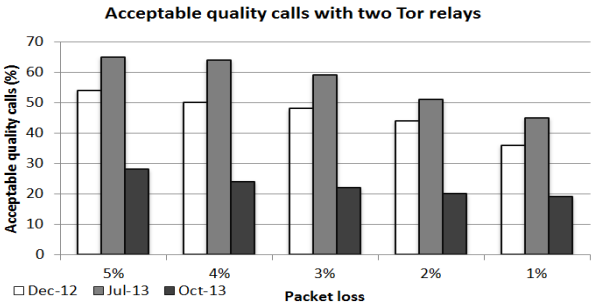


Figure 4.4 Acceptable quality calls with two Tor relays

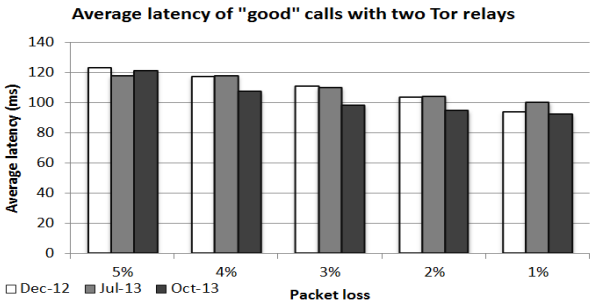


Figure 4.5 Average latency of "good" calls with two Tor relays

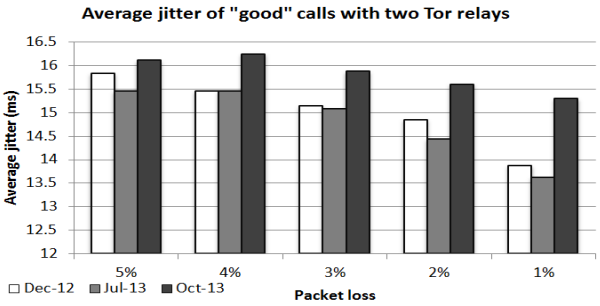


Figure 4.6 Average jitter of "good" calls with two Tor relays

The QoS comparison for the first and the second scenarios revealed that VOIP for two Tor relays delivered a better QoS than VoIP for three Tor relays. This applies to all time periods.

The experiment on VoIP with two relays, which was performed in December 2012, shows 54% acceptable calls for 5% packet loss, while the results obtained for

the VoIP experiment over three relays performed in the same period showed 36% acceptable calls. Likewise, for average latency, using two Tor relays as a medium transmission gained low latency compared to the three Tor relays. The average latency with three Tor relays was 156.98 ms for 5% packet loss, whereas using two Tor relays results in a slightly lower figure of 122.90 ms for the same packet loss. Average jitter obtained for VoIP over two and three relays showed only a little difference, ranging between 13 and 16 ms. Comparative results for the two scenarios in December 2012 can be seen in figures 4.7 to 4.9.

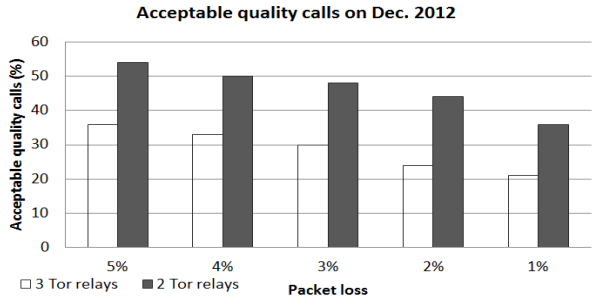


Figure 4.7 Acceptable quality calls in December 2012

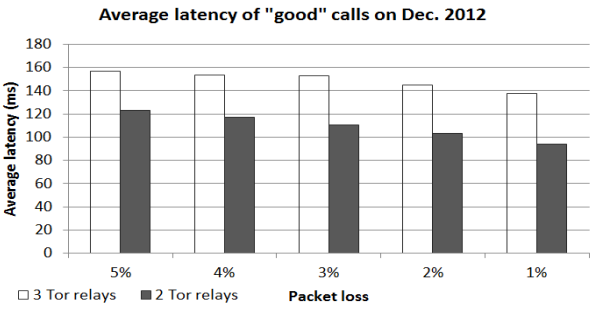


Figure 4.8 Average latency of “good” calls in December 2012

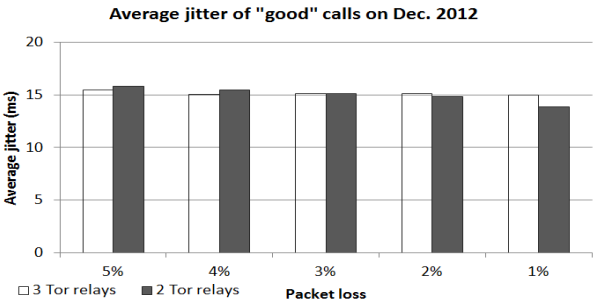


Figure 4.9 Average jitter of “good” calls in December 2012

In July 2013, the experimental results showed that VoIP over two Tor relays had a significant number of acceptable calls compared to that over three Tor relays. Acceptable calls obtained for the second scenario numbered 65%, while the

alternative scenario only led to 38% calls at 5% packet loss. For average latency, it was found that the average latency of the two Tor relays was lower by 40 ms than that for the three Tor relays. At 5% packet loss, VoIP with three Tor relays had 157.16 ms of average latency, and 117.70 ms when using two Tor relays. The differences in average jitter in both scenarios average 5 ms. Average jitter obtained for both scenarios ranged between 13 and 19 ms. Figures 4.10 to 4.12 show the details of the experimental results, for tests conducted in July 2013.

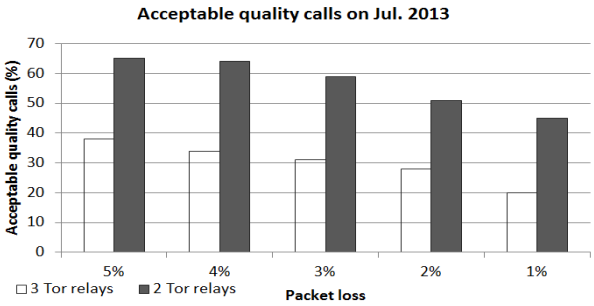


Figure 4.10 Acceptable quality calls in July 2013

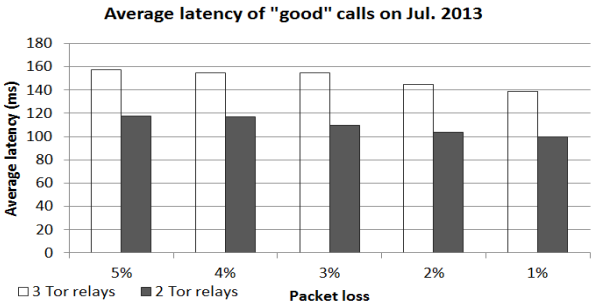


Figure 4.11 Average latency of "good" calls in July 2013

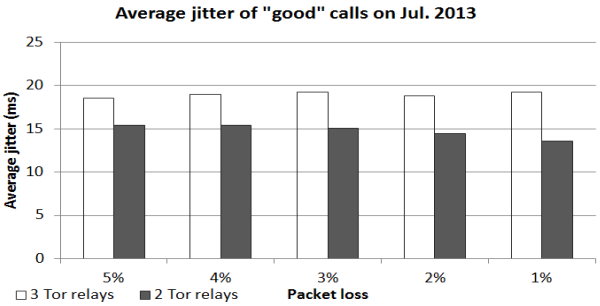


Figure 4.12 Average jitter of "good" calls in July 2013

The results of the experiment performed in October 2013 revealed that VoIP over two Tor relays led to more acceptable calls than the alternative scenario. At 5% packet loss, it had 28% calls and VoIP over three Tor relays led to only 24% calls. Average latency at 5% packet loss was 179.96 ms using the default Tor network and

120.94 ms using two Tor relays. Whereas, average jitter for both scenarios was around 14 to 19 ms. The details of the experimental results for October 2013 are shown in figures 4.13 to 4.15.

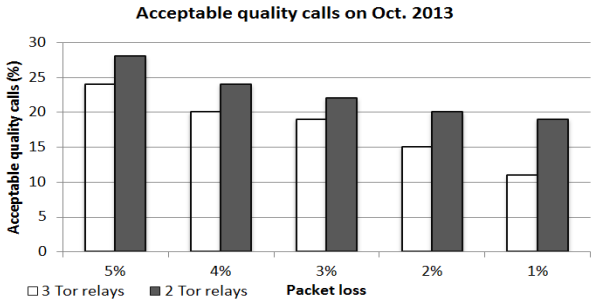


Figure 4.13 Acceptable quality calls in October 2013

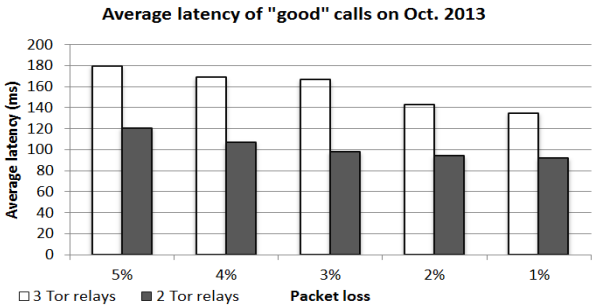


Figure 4.14 Average latency of "good" calls in October 2013

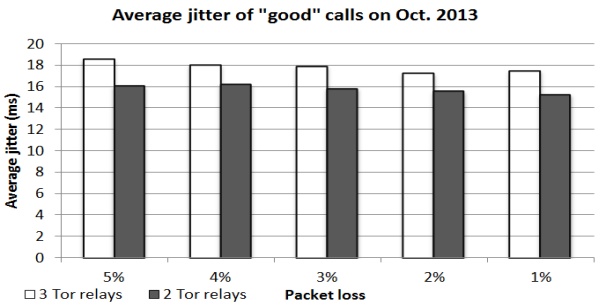


Figure 4.15 Average jitter of "good" calls in October 2013

The results for the experiment performed in October 2013 revealed the worst results compared to those for December 2012 and July 2013. In mid-August 2013, the Tor network received Botnet attacks, which led to an extraordinary increase in the number users; the number of users increased eight fold compared to July 2013. Total users per-day increased from mid-August 2013 until October 2013, exceeding more than four million. However, relays and bandwidth only increased slightly. The tremendous growth in the number of Tor users led to overloaded bandwidth at each

Tor relay which caused high latency on the Tor network greatly affecting the QoS performance of VoIP over the Tor network.

In this research, the experimental results were unsatisfactory, such that the best results were in July 2013 with only 65 calls of 100 calls tested, thereby meeting the requirements of the VoIP standard. This condition a consequence of multiple factors, one of these was Tor relays condition used to connect the caller and callee. The Tor relays conditions when conducting the experiments were unknown. This is due to security reasons, since providing information on relays conditions can increase the possibility of an attacker using relays condition information to attack the Tor network. Thus, a VoIP user on a Tor network does not know the condition of the Tor relays in the pathway; i.e. whether the Tor relays have enough bandwidth to transfer the audio packet or whether the relay usage has already been exceeded (overload).

## 4.2 Tor Network Forecasting

Tor network forecasting was conducted in early August 2013. Four time series analysis methods were used for Tor network forecasting, namely: linear, quadratic, cubic, and exponential trend. The data used for the Tor network forecasting was obtained from www.metrics.torproject.org [57] and included users, relays and bandwidth data from 1$^{st}$ January 2012 to 31$^{st}$ July 2013. Two methods were used to determine which forecasting methods had best accuracy; the Mean Absolute Deviation (MAD) and Mean Squared Error (MSE) calculation. The smallest values for MAD and MSE obtained from the forecasting method were considered to be the best for forecasting accuracy.

Figure 4.16 shows the details of the Tor users forecasting. From this graph, it can be seen that according to the two forecasting models (linear and exponential) Tor users were expected to increase at the end of 2013. On the contrary, the quadratic and cubic forecasting method show that there will be a reduction in the number of Tor users by the end of 2013. In fact, forecasting with a cubic model returned the lowest MAD and SSE thus it appears to be the best method for forecasting Tor user numbers. The details MAD and SSE are shown in table 4.8. Based on forecasting results, it can

be concluded that the number of Tor users is expected to reach between 400 thousand to one million users by the end of 2013.

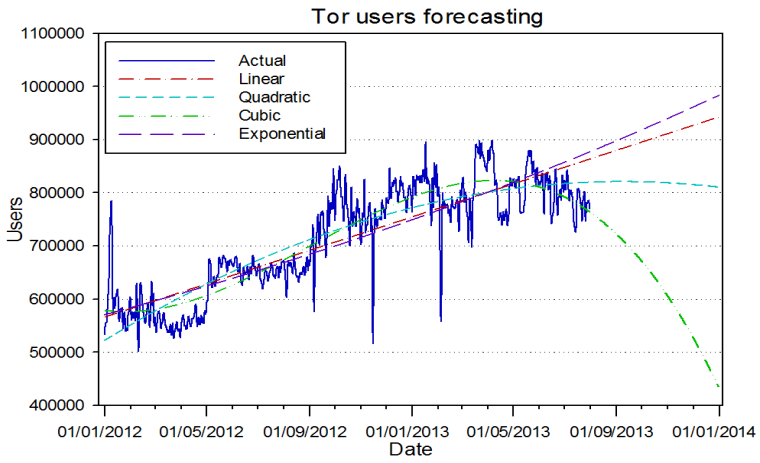


Figure 4.16 Tor users forecasting

Table 4.8 Users of Tor network accuracy

| Tor users forecasting accuracy | | | | |
|---|---|---|---|---|
| Methods | Linear | Quadratic | Cubic | Exponential |
| MAD | 41,259 | 37,767 | 33,110 | 43,099 |
| MSE | 2,778,415,742 | 2,390,678,338 | 1,932,286,675 | 3,013,107,630 |

Figure 4.17 shows the details of the Tor relays forecasting data, which indicates an increase in the average number of Tor relays at the end of 2013 for each forecasting method. However, extreme increases in the average number of Tor relays was indicated by the cubic forecasting method. Based on the result of the MAD and MSE, the best forecasting method, i.e. that which most closely approaches accuracy, is the cubic method then followed by the quadratic method. According to the cubic prediction, the number of Tor relays is expected to reach about 6,000 by the end of 2013. In contrast, the cubic method predicts there will be more than 4,500 Tor relays by the end of December 2013. Whereas, the estimations for the other two forecasting methods (linear and exponential) are almost similar to those for the quadratic, as they predict the number of Tor relays will be around 3,900 – 4,000 relays. The details for the accuracy of the supposed average number of Tor relays accuracy is shown in table 4.9.

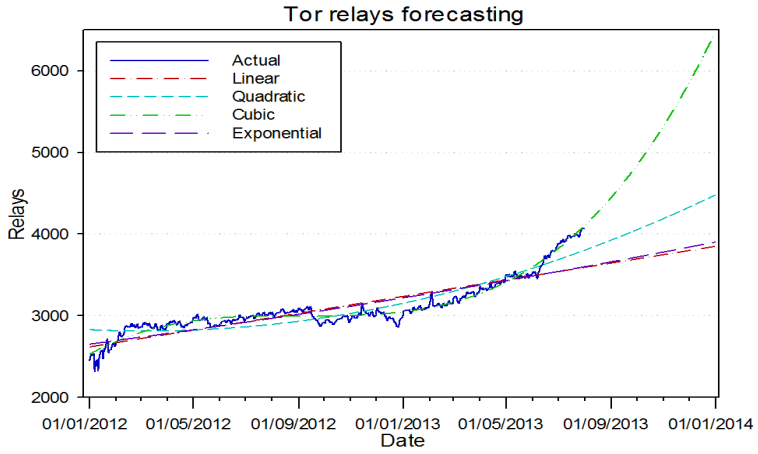


Figure 4.17 Tor relays forecasting

Table 4.9 Relays of Tor network accuracy

| Tor relays forecast accuracy | | | | |
|---|---|---|---|---|
| Methods | Linear | Quadratic | Cubic | Exponential |
| MAD | 126 | 106 | 46 | 121 |
| MSE | 25,138 | 16,305 | 3,499 | 23,241 |

By the end of 2013, all four forecasting methods predict that Tor bandwidth will have been increased. The highest bandwidth prediction was returned by the cubic forecast method; it reached almost 3.5 GBps. In contrast, the lowest prediction was returned when using quadratic forecast method, in which the bandwidth was only about 2.7 GBps. The details when forecasting the Tor relays can be seen in figure 4.18. Forecasting accuracy is shown with the lowest values for MAD and MSE, as is indicated by the cubic forecast method, which returns the lowest value for both accuracy methods (MAD and MSE). Table 4.10 shows the results of the forecast accuracy method for the four forecasting methods.

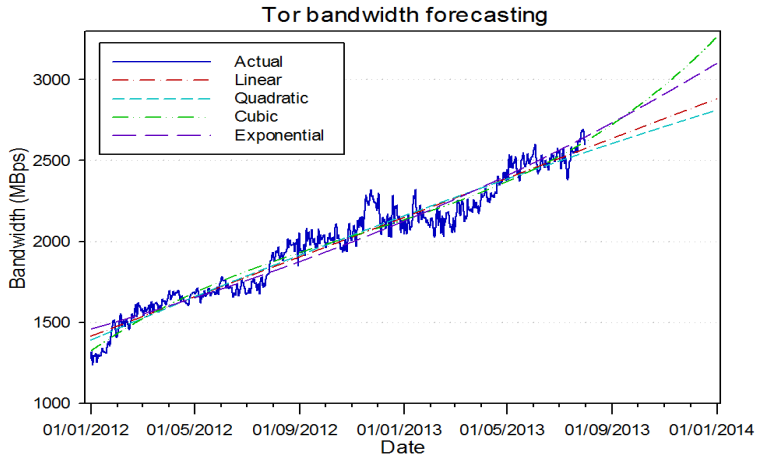


Figure 4.18 Tor bandwidth forecasting

Table 4.10 Bandwidth of Tor network accuracy

| Tor bandwidth forecast accuracy | | | | |
|---|---|---|---|---|
| Methods | Linear | Quadratic | Cubic | Exponential |
| MAD | 62.77 | 63.12 | 58.47 | 67.07 |
| MSE | 6,137.39 | 6,024.68 | 5,367.65 | 7,358.20 |

## 4.3 Tor Network Forecasting Validation

Tor network forecasting was performed in July 2013. There were three Tor network components forecasted, namely: Tor users, relays, and bandwidth. In mid-January 2014, researchers validated the results of Tor network forecasting. This was to determine whether the forecast results approached or were in accordance with the actual data. Forecasting was done to predict the three Tor network components in the period August to December 2013. The Tor network forecasting validation is presented in the graph from January to December 2013.

Figure 4.19 to 4.21 shows the validation results for the Tor network forecasting. It can be seen that the forecasting validation of Tor users deviates from the forecast results. This is because there were unexpected events beginning in early August 2013 (as stated above, the BotNet community used the Tor network as a medium to attack their targets). Therefore, the numbers of Tor users in August 2013 exceeded two million users, peaking in September 2013, at which point there were nearly six million users. After this, the number of Tor users slowly decreased until the end of 2013 when it approached three million users. Tor network forecasting took place in July 2013, and all four methods predicted the numbers of Tor users would only approach one million users by the end of December 2013. The details of the Tor user forecasting validation are presented in figure 4.19.

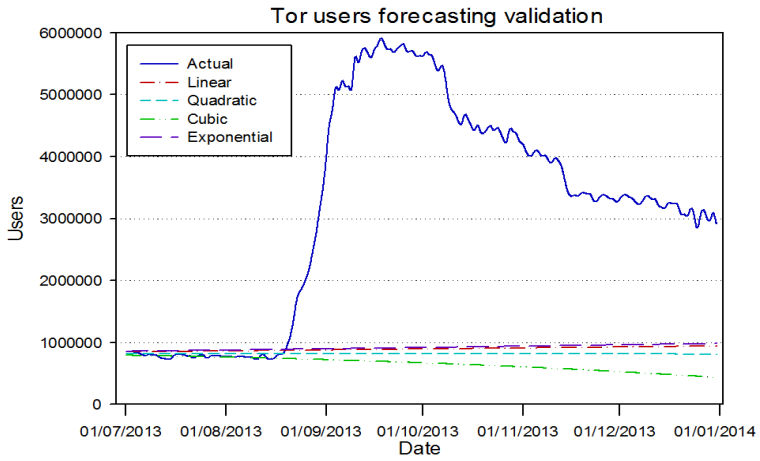


Figure 4.19 Tor users forecasting validation

Figure 4.20 shows the Tor relays forecasting validation. At the end of 2013, the number of Tor relays reached more than 4900 relays. As mentioned previously, the most accurate forecasting method was cubic, since it had the lowest MSE and MAD value. The forecasting results returned by this method estimated that there would be more than six thousand relays by the end of December 2013. Thus, the total of over 4900 is close to the predicted result obtained using the quadratic method. The results of the forecasting validation for the Tor relays are shown in figure 4.20 in detail.

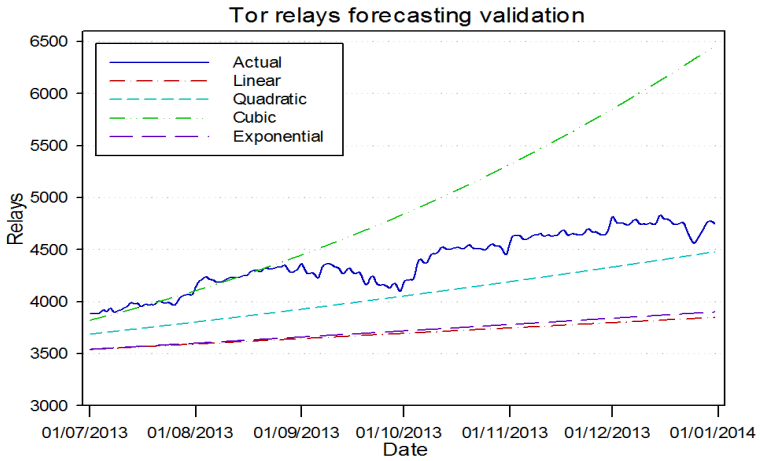


Figure 4.20 Tor relays forecasting validation

Tor forecasting in July 2013 shows the cubic method was the best forecasting method for predicting the growth of Tor bandwidth. Based on the forecast results, by the end of December 2013, the Tor bandwidth was 3200 MBps and in reality the actual Tor bandwidth found by the end of December 2013 perfectly match cubic's prediction.

Based on Tor network forecasting, we found that the growth in number of Tor users was extreme, while the growth in Tor relays and bandwidth was not proportional to the increase in Tor users. This impacted on the QoS performance of VoIP over the Tor network. This was indicated by the results of experiments conducted in December 2012, July, and October 2013. In October 2013, Tor users numbered above four million and the number of Tor relays had slightly increased, but bandwidth had decreased over the period of the experiment.

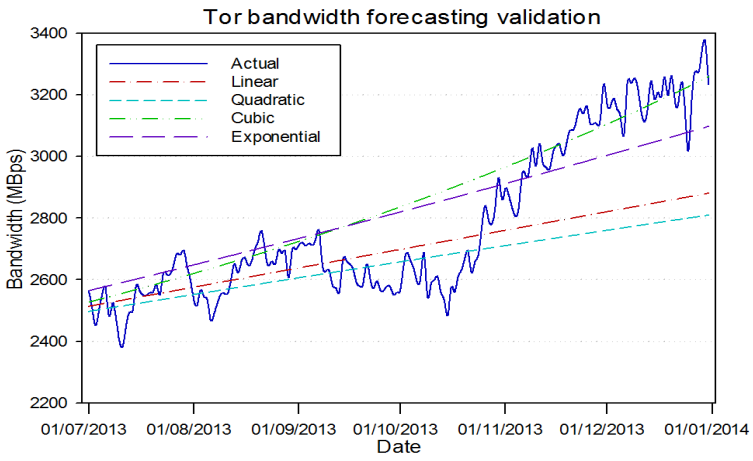


Figure 4.21 Tor bandwidth forecasting validation

## 4.4 Tor Relays Condition

The QoS of VoIP over the Tor network is inseparable from the condition of the relays which is used as a link between the sender and the receiver. The condition of each relay is used as a link to transfer the audio packet, but which must not be in the overloaded bandwidth. If there are one or more relays with the overloaded bandwidth, there will be a delay or latency caused by the queuing packets. This delay is usually called a queuing delay. The delay varies but is generally not advantageous for VoIP users who require end-to-end latency in a network of less than 400 ms. This condition will be difficult to achieve if the Tor relays have an overloaded bandwidth. The results for good latency and jitter from the experiment in July 2013 are shown in figures 4.22 and 4.23. The results of these experiments show an average latency of about 200 ms with a 0% packet loss and an average jitter of about 18 ms.

Figures 4.24 and 4.25 show bad latency and jitter resulted from one of the experiments conducted in July 2013. The average latency obtained in this experiment

exceeded 400 ms and the peak of the packet latency reached nearly 4,000 ms. Based on these figures, it can be ascertained that the audio packets from the sender will involve a long delay before they reach the receiver. The figures also show the packet loss on this experiment exceeded 50%, with an average jitter of less than 30 ms. This means the audio quality was good, but that there was a long delay between sender and receiver.

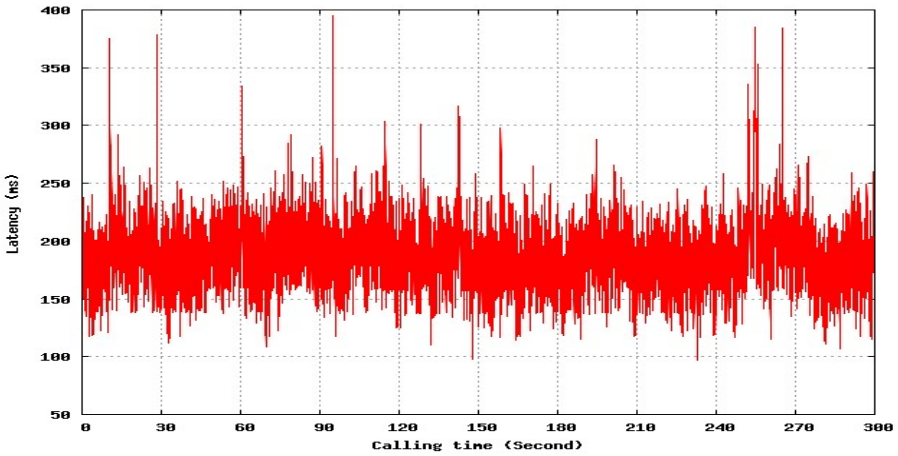


Figure 4.22 Good latency in experiment

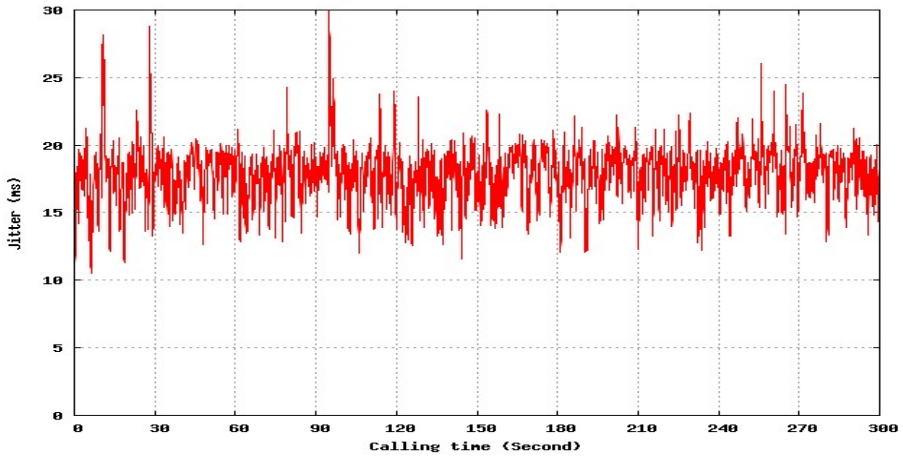


Figure 4.23 Good jitter in experiment

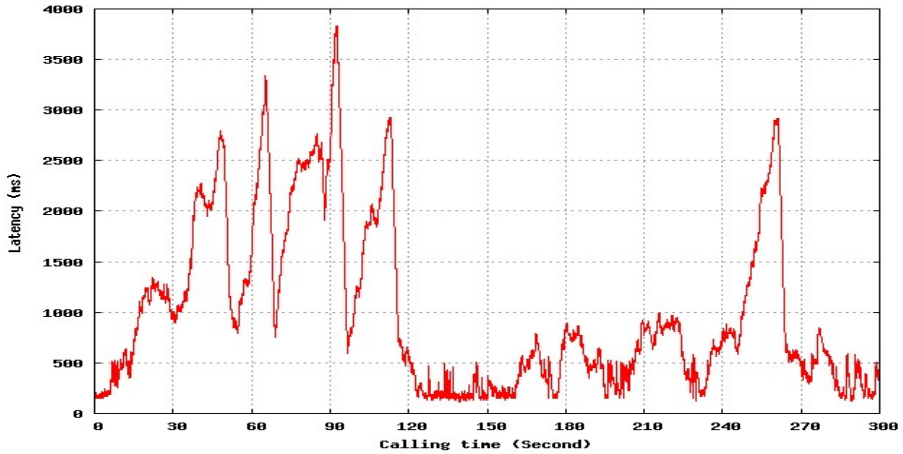


Figure 4.24 Latency on bad QoS calls in the experiment

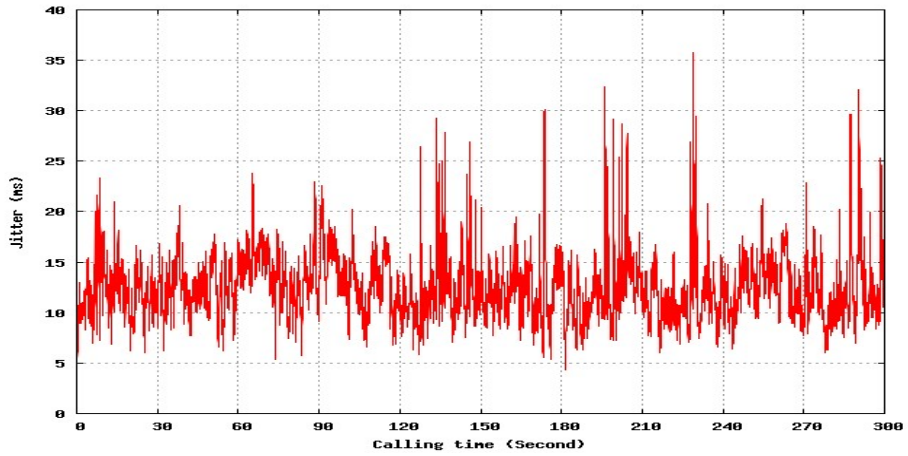


Figure 4.25 Jitter on bad calls in the experiment

## 4.5 Probability of Attackers

Attacker probability measurements aim to discern the probability of attackers according to the experimental scenarios, which were VoIP calls over two Tor relays and VoIP calls over three Tor relays. Based on the results of the experiments, it appears that VoIP calls through two Tor relays provide better performance than VoIP calls over three Tor relays. Attackers probability on the Tor network is described in section 2.5.

Currently, Tor users have already exceeded one million users, and the Tor network has more than 5,000 relays scattered around the world. Therefore, for attacker probability we assume that the number of Tor relays is 5,000.

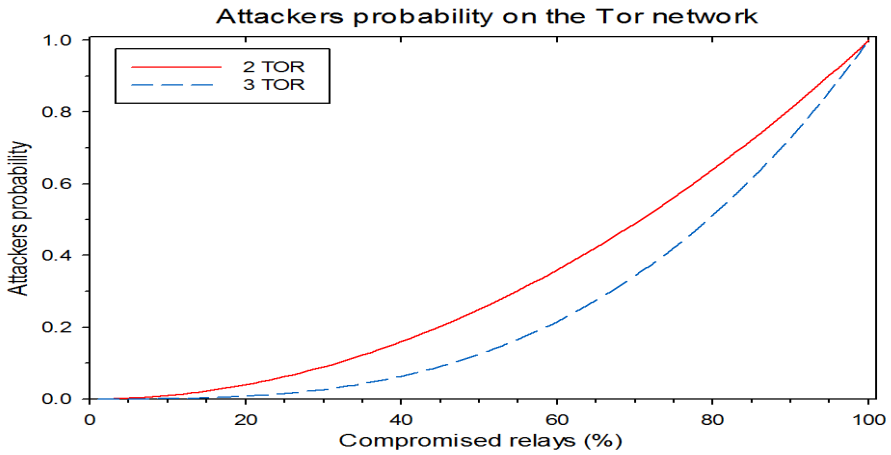


Figure 4.26 Attackers probability of Tor network

With a very large numbers of Tor relays and path lengths used, the attackers probability is very small. Figure 4.26 shows the attacker probability between VoIP calls with two Tor relays and three Tor relays. Based on the results for these calculations, the probability of attackers on the Tor network with two relays is nearly 0.2, and the compromised relays are about 40%. However, the probability of an attacker is the same as three relays will be effected when number of compromised relays is about 60%. Based on these results, we can be infer the level of attackers probability within three Tor relays are less than within two Tor relays.

# 5 CONCLUSION

In this chapter, the researcher present a conclusion to the study based on the research finding. The research will also include recommendations for future work and challenges.

## 5.1 Conclusion

This dissertation, which is based on empirical research, aims to obtain the value of QoS performance in VoIP over a Tor network. In summary, the present study has proven that although the Tor network is not designed to transfer audio packets in real time, it was capable of transmitting audio packets with reduced positive results in terms of QoS.

Furthermore, four fundamental research questions are fulfilled. In regard to the first research question, this research has shown that VoIP can be integrated with the Tor network by means of encapsulation. By doing so, the audio packets (UDP stream based) can be transmitted over the Tor network (TCP stream based). In this research, the audio packets were encapsulated by using openVPN. Communications between the caller and callee via OV include peer-to-peer communication, which is a virtual IP address for their identity.

The second question is the main research question. After integrating VoIP with the Tor network, the QoS performance when of anonymising VOIP was calculated. The best QoS performance for VoIP over Tor network were the experimental results for July 2013, when 65% of calls were acceptable, and the worst result was in October 2013, when only 24% of calls were acceptable for a 5% packet loss. These results were not good, when compared to the QoS performance of PSTN, which reached 99.999%.

Tor is designed to assure the anonymity of data packets being sent from the sender to a destination. Existing anonymity on the Tor network is guaranteed by unlink-ability; where the pathway used by the sender is unknown to others. Even if

there are eavesdroppers who capture a data packet they still do not know the sender or receiver's identity. By default, three Tor relays are used as a pathway to connect the sender and receiver. Each Tor relay has a private key using a 128 bits Advanced Encryption Standard (AES) encryption; thus, the security on each relay is maintained. The Tor network does not have end-to-end security, because the data from the exit relay to the destination is sent without encryption.

The Tor network is a low latency network designed to disperse data such as email, chat, and websites. By using TCP, the Tor network sends data without errors. The Tor network is not suitable for real time communication, because it has no guarantee of latency. According to ITU standard requirements, latency on audio real time communication should be not more than 400 ms (one way latency). Based on the research results, it is found to be difficult to acquire a good QoS performance on VoIP over a Tor network if there is insufficient bandwidth. This is apparent from the results for each experimental period. If the ratio of Tor bandwidth and Tor users average is high, then acceptable calls will return good results. Thus, the limitations of VoIP over the Tor network describe the Tor network conditions, such as the number of Tor users and Tor bandwidth (to have good quality calls) and number of relays (to achieve a good anonymity).

Currently, the Tor network is reliable, trustworthy, and updated periodically. By using the Tor network, VoIP users can engage in anonymous communication. However, selecting high anonymity levels results in a degradation of QoS performance in VoIP over the Tor network. This was proven by the experimental results, in which three Tor relays were found to have higher anonymity than two Tor relays; although the QoS performances for three Tor relays as much lower than for two Tor relays. Due to the Tor network's anonymous connections, on August 2013, the BotNet community began using it to attack their target. This led to a sudden and dramatic increase in the number of Tor users. In July 2013, there were about 900 thousand users on the Tor network, but in August 2013, Tor users numbered over five million.

Although Tor has not been designed for transmitting voice, this research has found that a significant number of calls show the good quality necessary to meet ITU

recommendations (G.114). In addition, our research also quantifies the trade-off between call quality and the probability of attackers.

## 5.2 Future Work

This dissertation is based on empirical research. It has limitations, such as the fact that many of the components of the Tor network (numbers of users, numbers of relays, and bandwidth) cannot be adjusted. The researcher was only able to investigate the relays to be used as a pathway in the experiment. This means that the researcher only used the data acquired during the periods when the experiments were conducted.

This research focuses on a QoS performance analysis of VoIP over the Tor network. In the future, research could be done to analyse the QoS performance in VoIP on other anonymous networks; such as VoIP over JAP/ANON, Crowds, P5, and Anonymizer and/or building an anonymous network specifically for VoIP.

This research could also be followed by integration of an encapsulation tool with the VoIP client. Therefore, in the future, OpenVPN will no longer be required, because the VoIP client will be able to use Tor client identity for the VoIP dialling process. This will result in full anonymity in VoIP calls, because there is no third party acting to support communication between the caller and callee.

Currently, the Tor network does not implement a minimum limit on bandwidth usage; therefore, if Tor user numbers increase, then the bandwidth will be shared equally among all Tor users. This may then lead to a drop in the quality of the Tor network. Thus, the Tor network should prioritise existing users when bandwidth is overloaded, by limiting the minimum bandwidth dedicated to Tor users. The Tor network could then reject new users when the limit for minimum bandwidth has been reached. In this way, the Tor network could maintain the necessary QoS to support VoIP calls.

# REFERENCES

[1] X. Yang, R. Dantu, and D. Wijesekera, "Security Issues in VoIP Telecommunication Networks," *Handbook on Securing Cyber-Physical Critical Infrastructure: Foundations and Challenges,* pp. 763-789, 2012.

[2] Nist and E. Aroms, *NIST 800-58 Security Considerations For Voice Over IP Systems*: CreateSpace, 2012.

[3] S. Yoon, H. Jung, and K.-S. Lee, "A Study on the Interworking for SIP-Based Secure VoIP Communication with Security Protocols in the Heterogeneous Network," in *Security Technology*. vol. 58, D. Ślęzak, T.-h. Kim, W.-C. Fang, and K. Arnett, Eds., ed: Springer Berlin Heidelberg, 2009, pp. 165-175.

[4] K. Ono and S. Tachimoto, "SIP signaling security for end-to-end communication," in *The 9th Asia-Pacific Conference on Communications (APCC)*, 2003, pp. 1042-1046 Vol.3.

[5] Z. Yu, C. Thomborson, C. Wang, J. Fu, and J. Wang, "A Security Model for VoIP Steganography," presented at the Proceedings of the 2009 International Conference on Multimedia Information Networking and Security - Volume 01, 2009.

[6] W. Mazurczyk and Z. Kotulski, "New security and control protocol for VoIP based on steganography and digital watermarking," presented at the Informatyka - Badania i Zastosowania (IBIZA), Kazimierz Dolny, 2006.

[7] M. Rizal, S. Taheri, and D. Hogrefe, "Empirical Performance Analysis of Anonymizing VoIP over The Onion Router (TOR) Network," in *Proc. The IEEE international Conference on Privacy and Security in Mobile Systems (PRISMS)* Atlantic City, NJ, USA, 2013.

[8] A. Kumar, "An Overview of Voice over Internet Protocol (VoIP)," *Rivier College Online Academic Journal,* vol. 2, Spring 2006.

[9] W. Mazurczyk and Z. Kotulski, "Covert Channel for Improving VoIP Security," in *Advances in Information Processing and Protection*, J. Pejaś and K. Saeed, Eds., ed: Springer US, 2008, pp. 271-280.

[10] E. Coulibaly and L. Lian Hao, "Security of VoIP networks," in *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, 2010, pp. V3-104-V3-108.

[11] I. T. Union, "Recommendation H.235: Security and encryption for H-Series (H.323 and other H.245-based) multimedia terminals," ed, 1998.

[12] R. Dantu, S. Fahmy, H. Schulzrinne, and J. Cangussu, "Issues and challenges in securing VoIP," *Computers & Security,* vol. 28, pp. 743-753, 11// 2009.

[13] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks*, et al.*, "RFC3261 - SIP: Session Initiation Protocol," 06// 2002.

[14] A. D. Keromytis, "A Comprehensive Survey of Voice over IP Security Research," *Communications Surveys & Tutorials, IEEE,* vol. 14, pp. 514-537, 2012.

[15] T. Dierks and C. Allen, *The TLS Protocol Version 1.0*: RFC Editor, 1999.

[16] W. Jiang, "A lightweight Secure SIP Model for End-to-End Communication," presented at the In Proceeding the 10th International Symposium on Broadcasting Technology (ISBT '05), Beijing, China, 2005.

[17] H. Sinnreich and A. B. Johnston, *Internet Communication Using SIP: Delivering VoIP and Multimedia Services with Session Initiation Protocol*, Second ed. Indianapolis, Indiana: Wiley Publishing, Inc., 2006.

[18] P. Ai-Chun, L. Chih-Hsiao, L. Shu Ping, and H. Hui-Nien, "A study on SIP session timer for wireless VoIP," in *Wireless Communications and Networking Conference, 2005 IEEE*, 2005, pp. 2306-2311 Vol. 4.

[19] I. L. Cincunegui, "Quality of Service for VoIP in Wireless Communications," Doctor of Philosophy Thesis Electrical Electronic and Computer Engineering, Newcastle Univerity, Newcastle, 2011.

[20] Goralsky and Walter, *The Illustrated Network: How Tcp/ip Works in a Modern Network*. Amsterdam: Elsevier/Morgan Kaufmann Publisshers, 2009.

[21] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, *RTP: A Transport Protocol for Real-Time Applications (RFC 3550)*: RFC Editor, 2003.

[22] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norman, "The Secured Real-Time Transport Protocol (SRTP) - RFC 3711," 2004.

[23] V. K. Gurbani and V. Kolesnikov, "A secure and lightweight scheme for media keying in the session initiation protocol (SIP): work in progress," presented at the Principles, Systems and Applications of IP Telecommunications, Munich, Germany, 2010.

[24] J. Postel, "RFC 768: User Datagram Protocol," p. 3, 1980.

[25] J. Postel, "RFC 793: Transmission Control Protocol," ed, 1981, p. 85.

[26] S. Landström, "TCP/IP Technology for Modern Network Environments," Doctoral Thesis, Department of Computer Science and Electrical Engineering, Division of Systems and Interaction, Luleå University of Technology, Sweden, 2008.

[27] L. Parziale, D. T. Britt, C. Davis, J. Forrester, W. Liu, C. Matthews, *et al.*, *TCP/IP Tutorial and Technical Overview*, Eighth Edition ed.: International Business Machines Corporation - IBM, 2006.

[28] RFC3952, "Real-time Transport Protocol (RTP) Payload Format for internet Low Bit Rate Codec (iLBC) Speech," in *Network Working Group*, ed: The Internet Society, 2004.

[29] *Internet Low Bitrate Codec (iLBC)*. Available: http://www.ilbcfreeware.org/

[30] P. Drew and C. Gallon, "Next-Generation VoIP Network Architecture," Multiservice Switching Forum, California2003.

[31] M. Hassan, A. Nayandoro, and M. Atiquzzaman, "Internet telephony: Services, technical challenges, and products," *Ieee Communications Magazine,* vol. 38, pp. 96-103, Apr 2000.

[32] M. Liberatore, B. Gurung, B. N. Levine, and M. Wright, "Empirical tests of anonymous voice over IP," *Journal of Network and Computer Applications,* vol. 34, pp. 341-350, 1// 2011.

[33] "ITU-T Recommendation G.114: One-Way Transmission Time," 05 2003.

[34] T. Szigeti and C. Hattingh, *End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs (Networking Technology)*: Cisco Press, 2004.

[35] B. Xi, H. Chen, W. S. Cleveland, and T. Telkamp, "Statistical analysis and modeling of Internet VoIP traffic for Network Engineering," *Electronic Journal of Statistics,* vol. 4, pp. 58-116, 2010.

[36] K. Gonia, "Latency and QoS for Voice over IP," *SANS institute,* 2004.

[37] G. S. Tucker, "Voice over Internet Protocol (VoIP) and Security," *GIAC Security Essentials Certification (GSEC). SANS Institute,* 2005.

[38] "Implementing VoIP Service over Wireless Network," Alvarion, White Paper, 2006.

[39] A. Pfitzmann and M. Köhntopp, "Anonymity, Unobservability, and Pseudonymity — A Proposal for Terminology," in *Designing Privacy Enhancing Technologies*. vol. 2009, H. Federrath, Ed., ed: Springer Berlin Heidelberg, 2001, pp. 1-9.

[40] L. K. Bhoobalan and P. Harsh, "An Experimental Study and Analysis of Crowds based Anonymity," *The 2011 International Conference on Internet Computing,* 2011.

[41] L. Kazatzopoulos, C. Delakouridis, and G. F. Marias, "Providing anonymity services in SIP," in *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*, 2008, pp. 1-6.

[42] N. Komal and S. Shriniwas, "A New Approach towards The Onion Router Network Using An Attack Dependent on Cell-Counting," *International Journal of Emerging Technology and Advanced Engineering,* vol. 3, 2013.

[43] Danezis, George, C. Diaz, and P. Syverson, "Systems for anonymous communication," *Handbook of Financial Cryptography and Security, Cryptography and Network Security Series,* pp. 341-389, 2009.

[44] Y. Guan, X. Fu, R. Bettati, and W. Zhao, "An Optimal Strategy for Anonymous Communication Protocols," presented at the Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02), 2002.

[45] J. Xu, Z. Wang, L. Zhang, and Q. Wang, "Recipient Anonymity: An Improved Crowds Protocol Based on Key Sharing," in *Information Engineering (ICIE), 2010 WASE International Conference on*, 2010, pp. 60-64.

[46] M. K. Reiter and A. D. Rubin, "Crowds: anonymity for Web transactions," *ACM Transactions on Information and System Security,* vol. 1, pp. 66-92, 1998.

[47] J. Ren and J. Wu, "Survey on anonymous communications in computer networks," *Computer Communications,* vol. 33, pp. 420-431, Mar 1 2010.

[48] B. Humphreys, "Multimedia Performance of Anonymous Systems," *4th Annual Multimedia Systems, Electronics and Computer Science, University of Southampton,* 2003.

[49] M. K. Reiter and A. D. Rubin, "Anonymous Web transactions with crowds," *Communications of the Acm,* vol. 42, pp. 32-38, Feb 1999.

[50] H. Federrath and S. Köpsel. (2000-2011). *JAP: Anonymity and Privacy*. Available: https://anon.inf.tu-dresden.de

[51] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: the second-generation onion router," presented at the Proceedings of the 13th conference on USENIX Security Symposium - Volume 13, San Diego, CA, 2004.

[52] P. Syverson, "A peel of onion," presented at the Annual Computer Security Applications Conference (ACSAC) 2011, Orlando, Florida, 2011.

[53] M. G. Reed, P. F. Syverson, and D. M. Goldschlag, "Anonymous connections and onion routing," *IEEE Journal on Selected Areas in Communications,* vol. 16, 1998.

[54] D. M. Goldschlag, M. G. Reed, and P. F. Syverson, "Hiding Routing Information," in *Workshop on Information Hiding*, Cambridge, United Kingdom, 1996.

[55] P. Syverson. (2005). *Onion Routing*. Available: http://www.onion-router.net

[56] A. Panchenko, F. Lanze, and T. Engel, "Improving performance and anonymity in the Tor network," in *Performance Computing and Communications Conference (IPCCC), 2012 IEEE 31st International*, 2012, pp. 1-10.

[57] K. Loesing. *Tor Metrics Portal*. Available: https://metrics.torproject.org/

[58] J. B. Kowalski and K. Gabert. (2006-2007). *Tor Network Status*. Available: https://torstatus.blutmagie.de/

[59] R. Dingledine, I. Goldberg, N. Mathewson, F. Rieger, C. Bowden, M. Hoban-Dunn*, et al. Tor Project*. Available: www.torproject.org

[60] N. S. Evans and C. Grothoff, "Deanonymizing Tor."

[61] K. Bauer, M. Sherr, D. McCoy, and D. Grunwald, "ExperimenTor: a testbed for safe and realistic tor experimentation," presented at the Proceedings of the 4th conference on Cyber security experimentation and test, San Francisco, CA, 2011.

[62] W. Dai. (2000). *PipeNet 1.1*. Available: http://www.weidai.com/pipenet.txt

[63] R. Song and L. Korba, "Anonymous Internet Communication Based on IPSec," presented at the Proceedings of the IFIP 17th World Computer Congress - TC6 Stream on Communication Systems: The State of the Art, 2002.

[64] L. Cottrel. *The Anonymizer*. Available: http://www.anonymizer.com

[65] M. Rennhard, S. Rafaeli, and L. Mathy, "Design, Implementation, and Analysis of an Anonymity Network for Web Browsing," Swiss Federal Institute of Technology, Computer Engineering and Network Laboratory, Technical Report TIK-Nr. 129, 2002.

[66] C. Diaz, S. Seys, J. Claessens, and B. Preneel, "Towards measuring anonymity," presented at the Proceedings of the 2nd international conference on Privacy enhancing technologies, San Francisco, CA, USA, 2003.

[67] N. Borisov and J. Waddle, "Anonymity in Structured Peer-to-Peer Networks," Computer Science Division (EECS), University of California, Berkeley, California UCB/CSD-05-1390, 2005.

[68] C. E. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical Journal,* vol. 27, pp. 379-423, 623-656, July, October 1948.

[69] P. Syverson, "Why I'm Not an Entropist," in *Security Protocols XVII*. vol. 7028, B. Christianson, J. Malcolm, V. Matyáš, and M. Roe, Eds., ed: Springer Berlin Heidelberg, 2013, pp. 213-230.

[70] P. Syverson, G. Tsudik, M. Reed, and C. Landwehr, "Towards an Analysis of Onion Routing Security," *Designing Privacy Enhancing Technologies,* vol. 2009, pp. 96-114, 2001.

[71] M. Feilner and N. Graf, "Beginning OpenVPN 2.0.9: Build and Integrate Virtual Private Networks using OpenVPN," *Packt Publishing, Birmingham, UK,* December 2009.

[72] W. J. Stevenson, *Operations Management*, Eleventh ed.: McGraw-Hill, 2012.

[73] V. Fusenig, D. Spiewak, and T. Engel, "Anonymous Communication in Multihop Wireless Networks," *Journal of Research and Practice in Information Technology,* vol. 40, pp. 207-225, 2008.

[74] P. Correia, E. Rocha, A. Nogueira, and P. Salvador, "Statistical Characterization of the Botnets C&C Traffic," *Procedia Technology,* vol. 1, pp. 158-166, // 2012.

[75] C. Hyunsang, L. Hanwoo, L. Heejo, and K. Hyogon, "Botnet Detection by Monitoring Group Activities in DNS Traffic," in *Computer and Information Technology, 2007. CIT 2007. 7th IEEE International Conference on*, 2007, pp. 715-720.

[76] V. Gegel. (2012). *TOR Fone - p2p secure and anonymous VoIP tool*. Available: http://torfone.org/

[77] J. Corbett. (2013). *1985 Phone - Peer-to-peer Encrypted Phone Calls to Avoid NSA Wiretapping*. Available: http://www.1985phone.com/

[78] D. Moody. (2002). *Empirical Research Methods*. Available: http://www.itu.dk/~oladjones/semester%203/advanced%20it%20mgt%20and%20software%20engineering/project/materials/what%20is%20empirical%20research1.pdf

[79] P. Developers. (2001-2013). *Privoxy*. Available: http://www.privoxy.org/

[80] D. Mill, E. J. Martin, J. Burbank, and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithm Specification," Internet Engineering Task Force (IETF) RFC 5905, 2010.

[81] H. Sommerfeldt. (2012). *PhonerLite*. Available: http://www.phonerlite.de/index_en.htm

[82] G. Combs. (1998). *Wireshark*. Available: www.wireshark.org

[83] R. Z. A. Fathony, S. H. Wibowo, K. Anas, and L. Amelia. (2008). *Zaitun Time Series - Time Series Analysis and Forecasting Software*. Available: http://www.zaitunsoftware.com/home

# CURRICULUM VITAE

europass

## Europass Curriculum Vitae

### Personal information

| | |
|---|---|
| First name(s) / Surname(s) | **Maimun Rizal** |
| Address(es) | Robert Koch Strasse 38/ App. 224, 37075 – Göttingen, Germany |
| Telephone(s) | +49 551 39 172027 (office) Mobile: +49 176 45000 367 |
| E-mail | maimun.rizal@gmail.com / maimun.rizal@cs.uni-goettingen.de |
| Nationality | Indonesia |
| Place and date of birth | Aceh Utara, 02 May 1980 |
| Gender | Male |

### The reason for Obtaining a Doctoral Degree

Currently, Indonesia still requires improvement in all areas of human resources. Hence, it is necessary to improve knowledge for betterment of the nation in the future. Hopefully, one day, I can give contribution to Indonesia or my province (Aceh Province) according to the field that I am studying (Computer Science and Information Systems).

### Desired employment / Occupational field

**Information Systems Technology**

### Work experience

| | |
|---|---|
| Dates | June 2009 to February 2010 |
| Occupation or position held | Facilitation of Educational Resources Staff |
| Main activities and responsibilities | Analyse teacher quality and facilitate them to improve educational quality in Province of Aceh, Indonesia. |
| Name and address of employer | Lembaga Penjaminan Mutu Pendidikan – LPMP (Educational Quality Assurance Institution)<br>Jalan Banda Aceh – Medan Km. 12.5, Desa Niron, Kec. Suka Makmur, Aceh Besar - Indonesia |
| Type of business or sector | Education sector |

| | |
|---|---|
| Dates | October 2005 to December 2005 |
| Occupation or position held | Air Movement Assistant (AMA) |
| Main activities and responsibilities | Managing airplane schedule for humanitarian, loading and unloading passenger, and collecting all activities airplane report (fuel consumption, airplane operation time, weather report for airplane crews) |
| Name and address of employer | United Nations Humanitarian Air Services (UN-HAS) – United Nations World Food Program (UN-WFP)<br>Sultan Iskandar Muda Military Airport – Blang Bintang, Aceh – Indonesia<br>Humanitarian sector |
| Type of business or sector | Humanitarian sector |
| | |
| Dates | June 2004 to September 2005 |
| Occupation or position held | Data and Information Staff |
| Main activities and responsibilities | Supporting data and information to other section in LPMP Aceh |
| Name and address of employer | Lembaga Penjaminan Mutu Pendidikan – LPMP (Educational Quality Assurance Institution)<br>Jalan Banda Aceh – Medan Km. 12.5, Desa Niron, Kec. Suka Makmur, Aceh Besar - Indonesia |
| Type of business or sector | Education sector |
| | |
| Dates | June 2000 to Mein 2004 |
| Occupation or position held | Assistant Lecturer |
| Main activities and responsibilities | Assisting training at Controlling System Laboratory |
| Name and address of employer | Controlling System Laboratory, Electrical engineering Department, Faculty of Engineering, Syiah Kuala University (UNSYIAH), Darussalam, Banda Aceh – Indonesia. |
| Type of business or sector | Education sector |
| | |
| Dates | August 2002 to October 2002 |
| Occupation or position held | Student on the Job Training |
| Main activities and responsibilities | Doing research in Power Line Communication (PLC) |
| Name and address of employer | Network and Broadband Laboratory, Division of Research and Information Technology (DivRisTI), PT. TELKOM Indonesia, Tbk., Bandung, Jawa Barat - Indonesia |
| Type of business or sector | Telecommunications sector |

## Education

| | |
|---|---|
| Dates | April 2010 – June 2014 |
| Title of qualification awarded | PhD Candidate |
| Principal subjects | Computer Science |
| Name and type of organisation providing education and training | Institute of Computer Science, Faculty of Mathematics and Computer Science, Georg – August University, Göttingen, Germany |

| | |
|---|---|
| Dates | March 2008 |
| Title of qualification awarded | Master of Science (M.Sc) |
| Principal subjects/occupational skills covered | Information Security (InfoSec) / Security in Voice over Internet Protocol (VoIP) – Implementation and Analysis |
| Name and type of organisation providing education and training | Center for Advanced Software Engineering(CASE), Faculty of Computer Science and Information Systems, Technology University of Malaysia (UTM), Johor Bahru – Malaysia. |
| CGPA | 3.46 / 4.00 |
| Dates | February 2004 |
| Title of qualification awarded | Bachelor of Engineering (B.Eng.) |
| Principal subjects/occupational skills covered | Telecommunication / Simulation of cell breathing on CDMA cellular technology |
| Name and type of organisation providing education and training | Electrical department, Faculty of Engineering, Syiah Kuala University (UNSYIAH), Banda Aceh - Indonesia |
| CGPA | 3.05 / 4.00 |
| Dates | June 1995 – June 1998 |
| Name of school | Public Senior High School 3, Banda Aceh - Indonesia |
| Dates | June 1992 – June 1995 |
| Name of school | Public Junior High School 2, Banda Aceh - Indonesia |
| Dates | June 1987 – June 1992 |
| Name of school | Public Elementary School 61, Banda Aceh - Indonesia |

## Seminar and Training

| | |
|---|---|
| Dates | 24-27 June 2013 |
| Presented paper | Empirical Performance Analysis of Anonymizing VoIP over the Onion Router (TOR) Network |
| Principal subjects/occupational skills covered | Privacy and Security in Mobile Systems |
| Name and type of organisation providing education and training | International Conference on Global Wireless Summit (GWS) 2013 |
| Level in national or international classification | International |
| Place | Atlantic City, New Jersey, USA |
| Dates | March – June 2009 |
| Principal subjects/occupational skills covered | Intensive English Course – IELTS Preparation |
| Name and type of organisation providing education and training | English Language Centre (ELC) language centre, Kuala Lumpur, Malaysia |
| Dates | December 2008 – February 2009 |
| Principal subjects/occupational skills covered | Intensive German Course |
| Name and type of organisation providing education and training | German Malaysian Institute (GMI), Kajang, Malaysia |

| | |
|---|---|
| Dates | 18 – 21 September 2006 |
| Principal subjects/occupational skills covered | Deep Knowledge Security Conference |
| Name and type of organisation providing education and training | Hack in the Box Security Conference (HITBSecConf) 2006, Kuala Lumpur, Malaysia |
| Level in national or international classification | International |
| Dates | 24 November 2005 |
| Principal subjects/occupational skills covered | HIV/AIDS in the Workplace Training |
| Name and type of organisation providing education and training | United Nations World Food Program (UN-WFP), Banda Aceh, Indonesia |
| Level in national or international classification | National |
| Dates | 2 November 2005 |
| Principal subjects/occupational skills covered | Earthquake and Tsunami Safety Training |
| Name and type of organisation providing education and training | United Nations World Food Program (UN-WFP), Banda Aceh, Indonesia |
| Level in national or international classification | National |
| Dates | July 2005 |
| Principal subjects/occupational skills covered | Computer Skill and Information Management Workshop |
| Name and type of organisation providing education and training | Vocational Education Development Centre (VEDC), Malang, Jawa Timur, Indonesia |
| Level in national or international classification | National |

## Publication

| | |
|---|---|
| Conference Proceeding | M. Rizal, S. Taheri, and D. Hogrefe, “Empirical Performance Analysis of Anonymizing VoIP over The Onion Router (TOR) Network,” in Proc. The IEEE International Conference on Privacy and Security in Mobile Systems (PRISMS) Atlantic City, New Jersey, USA, 2013 |