JOHANNES KEPLER
UNIVERSITY LINZ

Submitted by
NAHEL FALHOUT

Submitted at
Institute of Networks and Security (INS)

Supervisor
Assoc.Prof. Mag. Dipl.-Ing. Dr. MICHAEL SONNTAG

04.2019

ASTERISK VOIP OVER TOR

Master Thesis

to obtain the academic degree of

Master of Science

in the Master’s Program

Networks and Security

ACKNOWLEDGEMENT

STATUTORY DECLARATION

I hereby declare that the thesis submitted is my own unaided work, that I have not used other than the sources indicated, and that all direct and indirect sources are acknowledged as references.
This printed thesis is identical with the electronic version submitted.

Date,

Signature

Abstract

VoIP (Voice over Internet Protocol), is a protocol that is used to transfer media like audio and video over networks. It became one of the most used protocols in telecommunication. In the long run, VoIP will replace the normal phone (PSTN) Public Switched Telephone Network. However, this replacement will face a lot of challenges. One of the big concerns are Eavesdroppers. While the traditional phone PSTN uses a private network, VoIP uses open networks like the internet. Consequently, eavesdropper in PSTN need direct-access to the network in order to obtain information, whereas in VoIP, an eavesdropper can obtain communication information from any place using internet connection. Therefore, privacy and security principles are crucial in the process of switching from the PSTN to the VoIP.

One possible solution is to use anonymous systems, which would provide privacy and security to the communication. However, the disadvantage that usually comes with this is a reduced service quality of VoIP.

This master thesis focusses on how to implement the VoIP protocol in the anonymous TOR (The Onion Routing) network and the consequential problems. Some of the issues are: How does the transfer from UDP packets through a TCP network work and the investigation of QoS which includes latency, Jitter and Packet loss. In the VoIP recommendation latency should not be more than 400ms, Jitter also less than 50ms and Packet loss less than 5%.

Within the frame of this thesis, an Elastix server was installed and configured as well as a Softphone in C# was programmed. Investigations were conducted in many scenarios and two scenarios were compared directly. The first packets were directly transferred between the callees (Direct RTP), while the second packets were routed through the Server (Non-Direct RTP).

Abstrakt

VoIP (Voice over Internet Protocol) ist ein Protokoll, das Medien wie Audio und Video über Netzwerke überträgt. Es ist eines der meistgenutzten Protokolle in der Telekommunikation. Obwohl VoIP langfristig das normale Public Switched Telephone (PSTN) erstzen wird, gibt es noch einige Hürden auf dem Weg dortin zu bewältigen. Ein Problem stellen Lauschangriffe dar. Traditionelle PSTN verwenden private Netzwerke, während VoIP ein offenes Netzwerk wie das Internet benützt. Das bedeutet, dass man für Lauschangriffe auf PSTN einen direkten Zugang zum Netzwerk benötigt, wohingegen ein Lauscher bei VoIP von jedem beliebigen Ort mit Internetverbindung Informationen erhalten kann. Beim Übergang von PSTN zu VoIP ist es daher unerlässlich Überlegungen zu Sicherheit und Datenschutz anzustellen.

Eine mögliche Lösung wäre es anonyme Systeme zu nutzen um so die Sicherheit zu erhöhen. Allerdings bringt dies meist eine Verschlechterung der Qualität des VoIP Services mit sich.

In dieser Masterarbeit wird der Fokus auf die Implementierung des VoIP Protokolls in das TOR (The Onion Routing) Netzwerk und die daraus entstehenden Probleme gelegt. Einige Kernpunkte sind: Wie funktioniert das Übertragen von UDP Paketen über ein TCP Netzwerk und die Ermittlung der QoS, Latenz, Jitter und Packet Loss. In der VoIP Empfehlung sollte die Latenz nicht mehr als 400ms sein, der Jitter weniger als 50ms und der Packet Loss weniger als 5% betragen.

Im Rahmen der Masterarbeit wurde ein Elastix server installiert und konfiguriert und ein Softphone in C# programmiert. Außerdem wurden Untersuchungen in verschiedenen Szenarien durchgeführt und es wurde ein Vergleich zweier Szenarien aufgestellt. Das erste Paket wurde direkt übertragen (Direct RTP), während das zweite über den Server geleitet wurde (Non-Direct RTP).

TABLE OF CONTENTS

LIST OF FIGURES

1. Introduction

Nowadays many research in the area of computer information and science system are very active. Computer information systems are rapidly utilized. One of the hot research areas is the Voice over IP protocol (VoIP). This protocol supports transferring voice, video, images, and data through public networks like the internet, and it has provided an outstanding change to the communication technology in the world. VoIP becomes the most popular communication service because of its advanced functionality and lower cost. It supports many different networks including internal or public networks providing long or short distance telecommunications.

However, VoIP has some security problems which might not exist in the traditional phone which has its own private network. VoIP uses public network i.e. the Internet, and there is a need to provide VoIP users with more security and privacy. VoIP is a real-time protocol, it needs a good QoS (Quality of Services), also latency should be less than 400ms to have a decent performance. To provide VoIP users with privacy packets should be routed to a secure sufficient anonymous system. Nonetheless, applying encryption or random routing between different anonymity systems will produce delays.

Many security challenges are facing the switch from traditional phone network PSTN to the VoIP. Different to PSTN which has its own network and this network is isolated. Most VoIP communications are done through public networks like the internet. The Internet has already many security threats and VoIP is not excluded, especially if the network has some designing issues (Ransome, 2005).

Eavesdropping on traditional phone networks needs direct physical access to the line, this access will increase the opportunity to discover it, basically, any attempt to access will be easy to be noticed. Unlike VoIP, Eavesdropper could be setting behind his computer and on another country. Open access to the same medium will increase the attack possibility (Richard Kuhn, 2005).

1.1 Research Problem

There is a lot of research regarding Voice Over IP security has been done. Many methods were tested like adding encryption to the VoIP (Liancheng Shan, 2009) (Chu, Huo, & Liu, 2011), also some other researches tried to apply Steganography to VoIP (R. Roselinkiruba, 2013). Although those implementations have suffered from many problems like hiding the ID and Information about callee and caller.

Protect the identity of the caller and the information of the call; like who is calling? where is he? And who is he calling? This needs more privacy and security techniques, but combining those techniques will cause some other problems. It will increase the delay between the callers and quality might be also an issue.

Ensure a one hundred present privacy in any system is hard to guarantee, besides it is difficult to be done. Anonymous networks are not designed to transfer real-time protocols like VoIP, also the latency is very high. Most of the Anonymous networks use the TCP protocol. VoIP transfers the audio packets using the Real-Time Transport Protocol (RTP) over a UDP protocol.

The aim of this master works is to increase privacy and achieve high security to the VoIP. This is empirical research which experiments and applies the concept of transferring VoIP packets over a TOR anonymous network. The main goal is to achieve the security and privacy principles with acceptable QoS and investigate this connection. Also, the implementation included a Softphone that has been designed to match the purpose of this research.

1.2 Research summary

This research will clear some points to achieve its purpose. Some of those points are:

- Transferring VoIP packets over the Tor- Onion Routing network:

The design of the Tor network allows only TCP packets to be transferred. VoIP protocol transfers RTP packets over UDP. How this problem has been solved and what has been done is discussed in chapter 2.

- Quality of service QoS performance and how is it affected by using the Tor network.
- How anonymous is VoIP through the Tor network?

Latency is the most important factor in VoIP QoS. The International Telecommunication Union (ITU) recommended that in VoIP the acceptable latency should be less than 400ms.

- How might the Tor attacks affect communication?

1.3 Literature review

In order to start the study, a few concepts should be cleared briefly. This chapter discusses the literature review and presents VoIP protocol and the other protocol used to transfer voice over a network like the internet. And VoIP construction in anonymous network systems which include VoIP protocol, audio codec, and metrics of QoS. The chapter also presents VoIP QoS, SIP, UDP, RTP, jitter, Packet loss and Latency.

1.3.1 VoIP Protocol

Presently, the most common standards for controlling and signaling VoIP or Internet telephone calls are the H.323 and Session Initiation Protocol (SIP). Both of these standards were made in the year 1995 and were used by researchers for finding solutions when starting voice and video communication between two computers (Jonathan Davidson, 2006). The first publication of H.323 was issued by the International Telecommunication Union (ITU) at the beginning of 1996. While the SIP standard was published by the Internet Engineering Task Force (IETF) in the year 1996. H.323 gives specific QoS parameters for instance low latency and packet loss. Nevertheless, SIP provides for security and privacy (Keromytis A. D., 2011). This study emphasis on SIP, as it will be used during the empirical study.

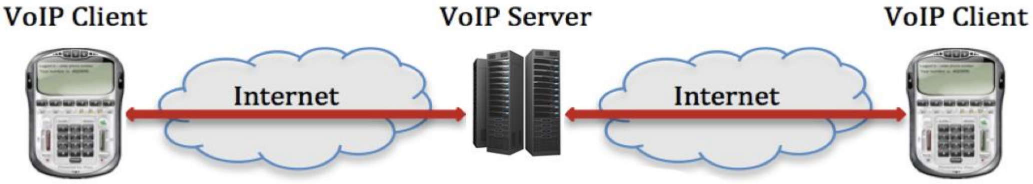


Figure 1: VoIP

1.3.2 H.323

In the earlier part of the year 1996, H.323 was published by the International Telecommunication Union (ITU). It was intended to work with both local and wide area with definite QoS. It gives a platform for transferring audio, video, and data communications through both LAN and WAN networks. The H.323 protocol supports Multimedia Internet Keying (MIKEY) and Secured Real-Time Protocol (SRTP). MIKEY is used for verification purpose while SRTP obtains media privacy. The H.323 standard is composed of the following parts: Gateways, Terminals, Gatekeepers and Multipoint Control Unit (MCU). Terminals are the devices used by end-user. These may be a smartphone, IP phones, or Computer. These terminals require a system control unit, a medium for transmission, and an interface which is based on packets. Gateways are instruments through which communication is carried out between different networks with media conversion and protocol translation. The MCU (Multipoint Control Unit) controls transmission between at least three terminals. The Gatekeeper oversees an entire zone that includes gateways, terminal, and MCU. Its function is directing calls and resolution of address. It may also control call signaling, bandwidth management, call authorization, and call management.

Implementation of security and privacy in the H.323 protocol is a complex procedure. The H.323 protocol uses random ports which causes a security problem that impacts firewalls. Resultantly, this situation will give an opportunity for an intruder. Another 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 (Thomas Porter, 2011) gives protection to the H.323. Numerous security problems have been addressed in H.235 by H.235 standard. These problems include integrity, authentication, privacy, and non-repudiation. For transport layer security, H.323 may also use a Secure Socket Layer (SSL). (Ram Dantua, 2009)

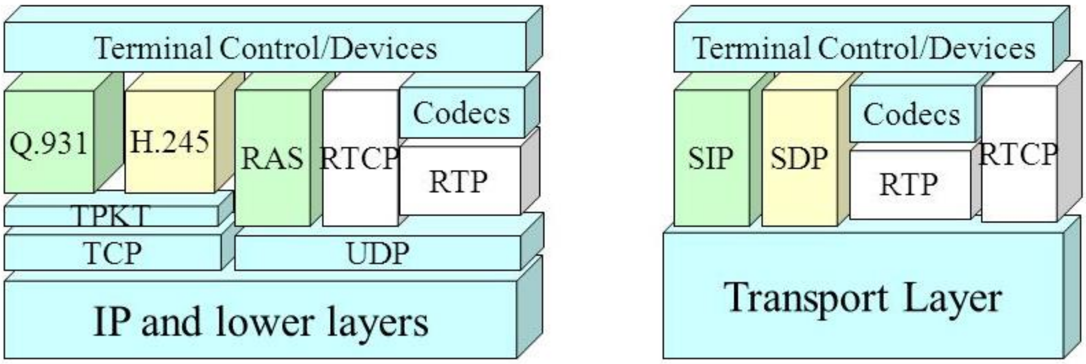


Figure 2: H.323, SIP Stack

1.3.3 Session Initiation Protocol (SIP)

SIP (Session Initiation Protocol) is a signaling protocol. SIP is an application layer protocol; it was developed by IETF (Internet Engineering Task Force) in the RFC 3261. Its function includes setup, maintenance, revision and control of the multimedia communication for the application layer (Johnston, 2004). The protocol is adequately developed for swift implementation and adaptability. The main purpose of the SIP is session initiation. It relies on RTP for media transfer. (Keromytis A. D., 2011)

In the SIP, hop-by-hop security uses the TLS (Transport Layer Security). In hop-by-hop security (Dierks Certicom, 1999), it is supposed that the caller and receiver trust all proxy servers which are connecting them. These proxy servers can check the communication data exchanged in their message. And in SIP caller and receiver security is obtained by the (S/MIME) Secure Multipurpose Internet Mail Extensions, the caller and receiver do not trust proxy servers. Thus, proxy servers cannot inspect their message. (Jiang)

Three main components of the SIP system are Location Services (LS), servers and User Agent (UA). A user agent normally refers to a Session Initiation Protocol phone or Session Initiation Protocol user software which is accessed from a phone or a computer. It builds a link between a user agent and other clients on the server to enable the transfer of data. Any User Agent should have two components first, UAS (User Agent Server) and the UAC (User Agent Client). The function of UAS is to initiate responses to requests received from User Agent Clients (H. Sinnreich, 2006).

Session Initiation Protocol consists of three types of servers which are registrar server, proxy server, and redirect server. Session Initiation Protocol in all servers requires minimal security level, thus a TLS, IPsec or any other security layer protocol should be implemented (Kolesnikov, 2010). First, the function of a registrar server is to store data about Session Initiation Protocol registration and location of the user. Second, the function of a proxy server is to receive requests of SIP from User Agents and then send this request to the end-user. A proxy server also preserves billing on a SIP. Lastly, the function of a redirect server is to keep the record of all Session Initiation Protocol users.

A location service keeps the record of the positions of all the User Agents who are registered. For instance, it involves saving data relating to the user which can be IP addresses, URIs, other preferences, scripts, and features. Generally, a Session Initiation Protocol comprises of three servers.

Sender and a receiver in SIP can establish the connection in three ways. The first way, the link between the sender and receiver is straightforward without any proxy. The sender contacts the receiver using the IP address. This way is normally utilized in WAN, VPN or LAN.

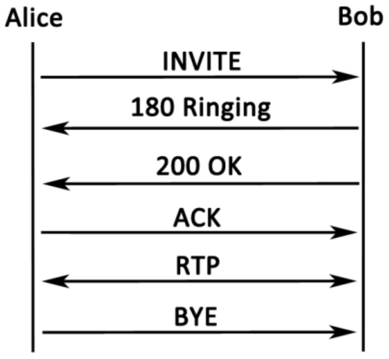


Figure 3: SIP Method 1

The second method happens during the setup of the call. Caller connect to the SIP proxy, it uses the LS (location services) to define the call routing (P. Ai-Chun, 2005).

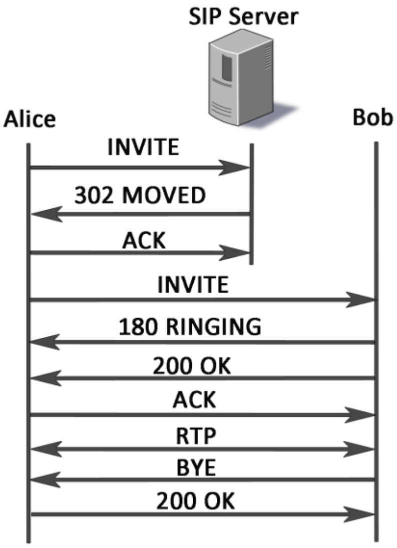


Figure 4: SIP method 2

Lastly, in the third method, the SIP proxy contacts the location service to send an invitation message to the receiver. This way is used in heterogeneous connections where the sender and the receiver are on separate networks.

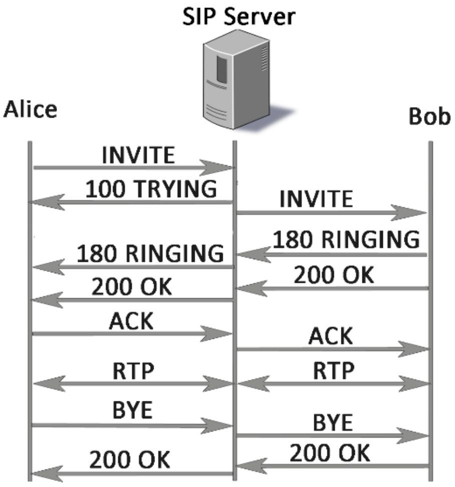


Figure 5: SIP method 3

1.3.4 VoIP in UDP and TCP

Chief internet protocols for transfer of data are the Transport Transmission Protocol (TCP) and the User Datagram Protocol (UDP) (Landström, 2008). UDP was defined as RFC 768 by David Reed in 1980. UDP is a protocol that transfers data online but without assurance of delivery. However, UDP transfers data quickly, i.e. it has low latency, due to fewer overheads. (Lydia Parziale, 2006). UDP is a connectionless protocol, it does not perform any sequencing or ensure data reliability.

The TCP protocol was developed in 1981 (Postel, 1981), and also defined as RFC 793. It gives a connection between two users that is secure and highly dependable. It monitors flow and congestion control. TCP delivers data packets in proper sequence and ensures receipt of data to the recipient. (Lydia Parziale, 2006).
TCP is connection-oriented protocol, it handles sequencing and error detection.
It ensures that a reliable stream of data is received on the destination.

The TCP high reliability in VoIP affects end user experience. Delays will happen every time a packet loss or an error occurs. This also will be translated into a high level of jitter and for the end user this is unacceptable in VoIP.

1.3.5 Quality of Service of VoIP

The three factors that determine the Quality of Service of VoIP are jitter, packet loss, and Latency. Based on ITU (the International Telecommunication Union) recommendations, the jitter lower than 30 ms (A. Duric S. A., 2004) (Bowei Xi, 2010). In one way transmission, the recommendation considered a latency with a maximum 250ms is acceptable. And for long destination (overseas connections) any delay between 150ms and 400ms still to be acceptable, packet loss of up to 5% (Gonia, 2004)

1.3.6 Real-time Transport Protocol (RTP, SIP)

Real-Time Protocol was made by The Internet Engineering Task Force (IETF) in 1993. Its maiden publishing was made in 1996 with the name of RFC 1889. Later on, RFC 3550 replaced it. Real-Time Protocol is an online protocol that specifies how programs manage the real time transmission of audio and video data over unicast or multicast network services. Also, it used in some Internet telephony systems or VoIP. In most applications, RTP uses TCP, but not in VoIP. (Schulzrinne, 2003)

RTP consists of two components namely the control and the data part. The control parts are known as Real-Time Control Protocol (RTCP) while the data parts are called Real Time Protocol (RTP). RTCP is chiefly utilized to manage synchronization, to monitor the quality of services (QoS) and to deliver information about the participants in on-going sessions. Multicast-to-unicast translators are supported by the RTCP, source identification and other synchronization.
The RTP carries data with real-time properties. These involve no change in loss detection, timing, content identification, and security.

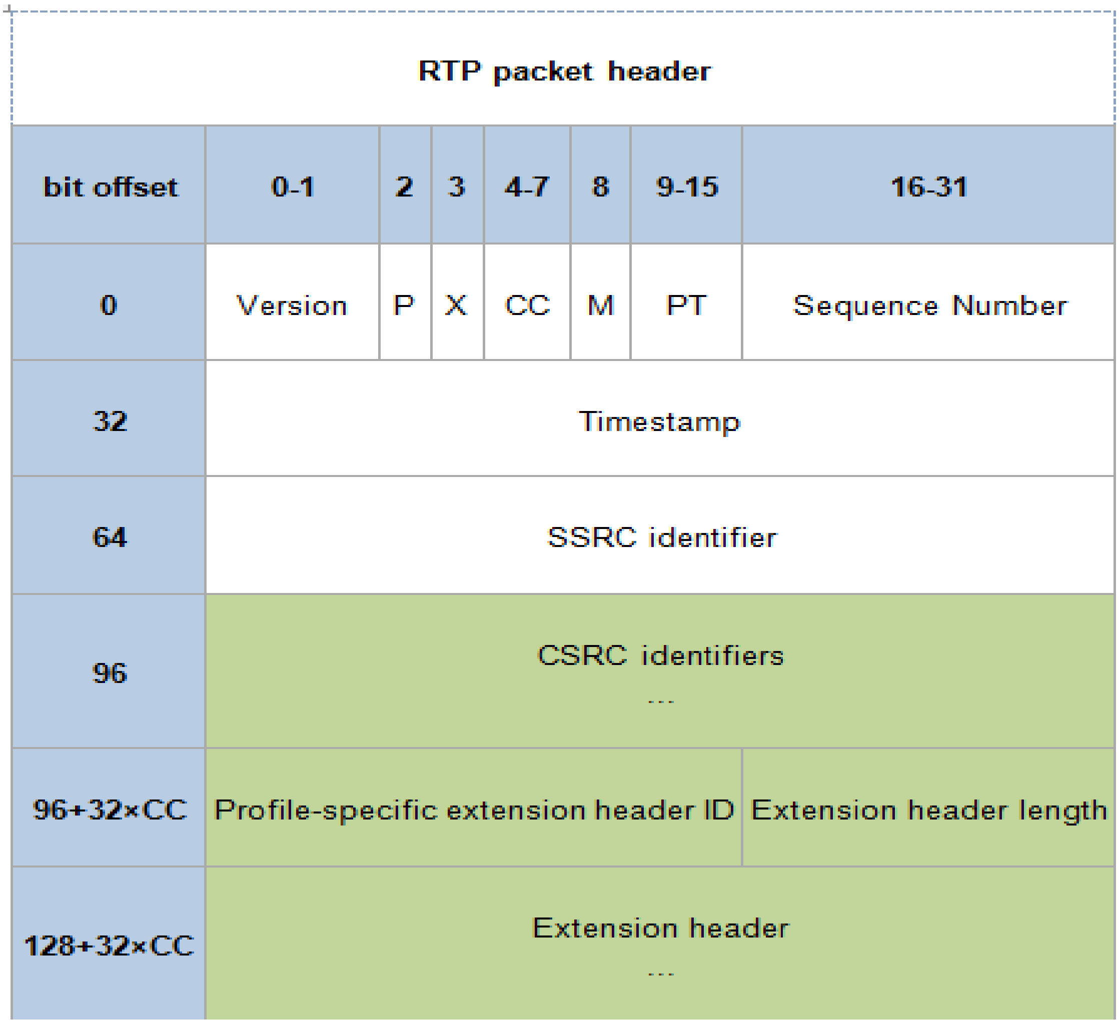
Real-Time Protocol gives multiple services which involve identification of payload, the numbering of sequence, stamping of time, monitoring of delivery. Identification of Payload tells the type of content in the communication as either static or dynamic. The numbering of Sequence is utilized to maintain synchronization of data between the caller and the recipient. For instance, it detects packets loss. Stamping of time is used to present content time that is transferred via PDU. Delivery monitoring, it sends RTCP packets in the RTP session from the sender to the receiver to determine the quality and conditions of the network in case of RTP packets loss or errors.

In RTP Session there are five types of Real Time Control Protocol (RTCP) messages. The first, Sender report which includes statistics about transmission and receptions from the senders. The second, Receiver Report which is also a statically report from receiver to the passive participants. The third type of messages is Source description, it has information about the source of the RTP like DNS name, email address. The fourth type is Goodbye (BYE), it shuts down the session. Finally, the APP (Application Specific Message), it provides information about specific application. (Schulzrinne, 2003)

Security in RTCP and RTP is obtained by Secured Real-Time Transport Protocol (SRTP). The by Secured Real-Time Transport Protocol provides the RTP with a security profile, which adds authentication to the message, confidentiality and replays protection to RTP. (M. Baugher, 2004)

1.3.7 RTP packet Header

The RTP header consists of at least 12 bytes and can add several additional bytes attached to the stream, and after the header comes the payload which represents the data. (Schulzrinne, 2003)



RTP packet header

bit offset	0-1	2	3	4-7	8	9-15	16-31
0	Version	P	X	CC	M	PT	Sequence Number
32	Timestamp						
64	SSRC identifier						
96	CSRC identifiers ...						
96+32×CC	Profile-specific extension header ID						Extension header length
128+32×CC	Extension header ...						

Figure 6: RTP Packet Header

Version: It consists of 2 bits which specify the protocol version. The current version used is version 2. (Adeel Ahmed, 2010)

P: The padding consists of 1 bit and is used to see if there are additional bytes at the end of the packet sent. These additional bytes are might be required if cryptographic algorithms are used.

X: The Extension flag consists of 1 bit and is used to see if there are additional extension attached to the RTP header or not.

CC: The CSRC Count is composed of 4 bits. It contains the number of CSRC identifiers that follow the RTP header.

M: Marker consists of 1 bit and is used to enable the inclusion of information on the limits of the frame in the packet sent.

PT: The Payload Type user consists of 7 bits and is used to identify the Payload format.

Sequence Number: takes a random number consisting of 16 bits and then increments by one at each transmission. It is used to know if there is data loss or arrival in the wrong order at the receiver (Kolesnikov, 2010)

Timestamp: consists of 32 bits and is used to enable file viewing with a specific Sampling rate.

SSRC: consists of 32 bits and carries a random number representing the source used in synchronization between streams.

CSRC List: Consists of 32 bits and identifies data sources in the Payload field when data is transferred from more than one source. The number of these sources.

1.3.8 Jitter

Jitter measured time difference in packet arrival time in the sender and receiver. Generally, Jitter is produced by multiple factors which include network bandwidth, changes in route and the distance between the caller and the callee. In a VoIP, jitter should not be more than 30ms. (Tim Szigeti, 1994)

1.3.9 Packet Loss

Packet Loss occurs when data does not arrive at all or not on time at the receiver's end (Ransome, 2005). Mostly Packet loss occurs due to physical media errors, overloaded links, or low link quality. Recommended packet loss in a network is 5% (Network, 2006). When packet loss is greater than 5%, the voice quality is affected. Packet loss is measured by dividing the number of packets lost with the total number of packets. (Network, 2006)

1.3.10 Latency

Latency is defined as the time taken by some data between the sender and the recipient. High latency can be caused by multiple factors which include network bandwidth and path length between the sender and the receiver. Latency is determined by taking a total of transmission delay, queuing delay, propagation delay, playout buffer delay. Codec processing delay and packetization/depacketization delay. (ITU, 2003)

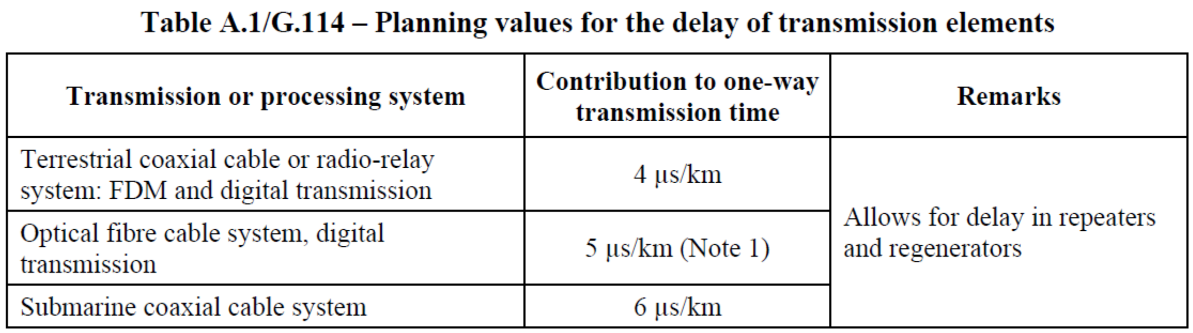


Table A.1/G.114 – Planning values for the delay of transmission elements

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 (Note 1)	
Submarine coaxial cable system	6 µs/km	

Figure 7: ITU-T Recommendation G.114 *(ITU, 2003)*

Propagation delay is defined as the required time to transfer a data packet from sender to receiver using media transmission. Propagation speed based on the medium figure. 7 and the physical distance might affect the delay.

Transmission delay is the required time to pull all data packets into the network also called Packetization delay. Distance does not affect the transmission delay.

Queuing delay is the required time for packets in queues at input and output ports until it can be executed. Codec processing delay time required to compress and convert the analogue signal to digital signal. Playout buffer delay time needs to get to the buffer (playout buffer) of the receiver.

Latency in a network is measured by two methods. First, by measuring the arriving time at the sender and the receiver we can find the latency. The latency is then calculated by taking the difference of the arrival times. In the second method, two-way latency i.e. transmission time of information is captured. Thus, the latency in the second case is calculated by the difference of time of the delivery of response from the original recipient.

1.3.11 Pipe Net

Pipe Net (Dai, n.d.)was designed in 2000 by Wei Dai. Basically, it is an anonymous protocol that allows security against traffic analysis. It makes use of three or four transitional nodes to create a link between the caller and the recipient. The fundamental idea behind Pipe Net virtual link encryption which creates a rerouting pathway to transfer the data (Yong Guan, 2002).

Pipe Net is like onion routing as it has an anonymous network having low latency. It is a perfectly anonymous system. But the problem arises when a user is allowed to disconnect by not sending messages. In practical use, Pipe Net has experienced failure because it has extremely large packets loss because of the Pipenet architecture, beside it does not support security services like IPSec and VPN. (Ronggong Song, 2002)

1.3.12 Anonymizer

Anonymizer is another centralized proxy network which is quite easy to use (LLC, n.d.). It functions as an intermediate connection between the internet and the client's PC for privacy protection. Hence, it has a comparatively low anonymity level and a low delay compared to other anonymous networks. In Anonymizer, the end-to-end connection is not private. However, Customers make use of Anonymizer due to many causes, such as thwarting identity theft, evading censorship in few countries or securing data while Internet use.

Unluckily, nowadays, Anonymizer servers are still merely accessible in the U.S. Thus, latency is higher when communicating from one continent to another, for example, a connection established from the U.S to Germany has very high latency.

1.4 The Onion Routing (Tor)

The research on Onion Routing (OR) started at the Naval Research Laboratory (NRL) (Paul, 2011) in 1995. OR is a low-latency anonymous system that is resistant to traffic analysis and eavesdropping (M.G. Reed, 1998). Its main purpose is to maintain privacy during communication between the caller and receiver. The caller sends a signal to the receiver through numerous routers. As a result, the eavesdropper has no data on the users calling. In first-generation onion routing, a single infected relay on the OR network could save traffic data between the caller and the recipient which may be used to maneuver the traffic later (Roger Dingledine N. M., 2004). Likewise, at the start of the OR design, it was essential to obtain a distinct proxy for each application.

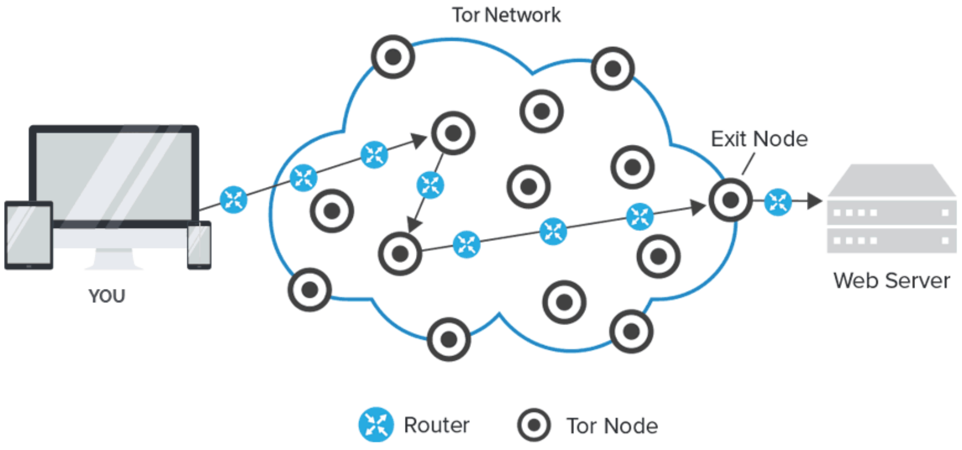


Figure 8: TOR

1.4.1 Tor Network

The second-generation of OR - The Onion Routing (Tor) - is a circuit-based low-latency anonymous communication. It only handles TCP streams over an open network i.e. the Internet. It focuses on preventing the intruders from detecting connections in the communication like eavesdropping and traffic analysis, which conceals the identity of the customer from its receiver. It supports SOCKS 5 (Roger Dingledine N. M., 2004). The Tor network has been developed for higher congestion control, forward privacy, integrity checking, organizable exit rules. It does not demand special administrative rights or kernel alterations, and need slight coordination between the caller and the receiver, giving a practical trade-off between anonymity, effectiveness, and usability. Tor is available as free software and is used very popularly for browsing privately on the Internet (Panchenko, Lanze, & Engel, 2012). Its popularity can be gauged from the fact that it has more than a half million users worldwide, using about 2 Gbps of total bandwidth in July of 2013 and more than 3000 network relays (Project, Tor Metrics, 2009–2018).

Tor encrypts information three times before sending it. Then, it decrypts information layer-by-layer as it transfers through the network. Tor customers communicate via global proxy networks, to conceal their location along with the identity of the one whom they are contacting.

The Tor customer obtains the list of relays from the directory at the Tor server. After that, it randomly picks three relays which are one entry relay, one middle relay, and another for the exit. Information from the caller is subsequently encrypted by means of a private relay key, which has been previously chosen. A key from the exit relay is used to encrypt the initial data, after that by making use of the key of the middle relay the encryption of the later part is done using the key of the entry relay. Then, data packets from the Tor customer is transferred to the entry relay. By making use of a private entry relay key, the data packets are decrypted after it enters in the entry relay. Thus, on arriving the relay, data packets are protected with two private keys (the key of exit and middle relay). After that, the data is sent by the entry relay to the middle relay where it is decrypted by making use of the key of the middle relay. Then the data packets are transferred to the exit relay from where it forwarded with no encryption to its termination point.

TOR browser uses HTTPs encryption, it uses the server's public key to encrypt the data exchange. Only target server can decrypt and read the data. The message will be transferred from node to another until reaching the target server with own session key. The server encrypts the response with the session key and only sender will able to decrypt the response from the server (Wikipedia).

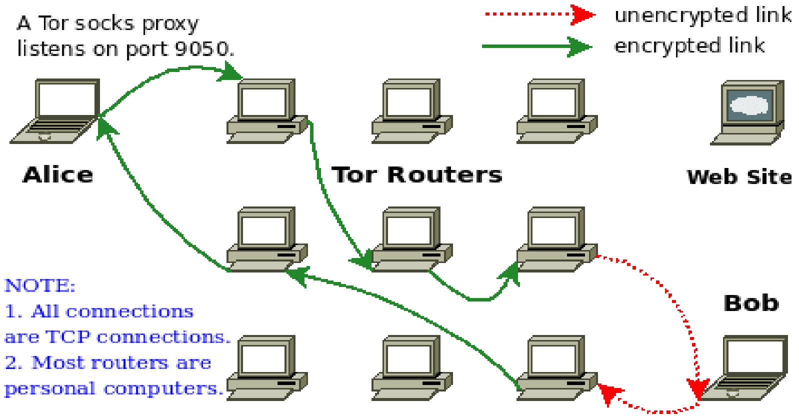


Figure 9: Tor Architecture *(Ramzi A. Haraty, 2017)*

The client can connect to anonymous network though the router's network which is Tor. (Kevin Bauer, 2012) It maintains privacy by choosing the relays of connection at random; Moreover, after every 10 minutes, it relays networks to ensure privacy.

1.4.2 Tor Relay Condition

The condition of relays is important when defining the Quality of Service of VoIP through the Tor. Relays are the primary link between the caller and the recipient. Thus, in order to transmit voice packets without delay, it is necessary that the bandwidth in any of the relays shall not be overloaded. Latency is also caused by overloading of the relays. This may increase the latency of the Tor network above the limit of 400ms (ITU, 2003).

1.4.3 Possibility of Attackers

In the internet communication world, anonymity becomes a necessary condition. It secures the customer's information in many ways. Some of these security measures are:

- Maintaining the anonymity of the sender.
- Hiding the identity of the receiver.
- Concealing the links of the connection points.

One of the Tor attacks is the end-to-end confirmation attack. The attacker is able to monitor both ends on communication, he finds matched patterns between the outgoing and incoming data. Those patterns will help to deanonymize the user ID, name or voice packets. in this case, attacker does not need to decrypt the packets. This sort of attacks is achieved by comparing packets transmission time to associated the transmitted data. Correlation attacks like this are problematic for low-latency anonymity networks like Tor.
According to Tor Project Tor is not designed to protect against attacks in which an attacker can monitor or measure the traffic that is going on the tor network (Project, One cell is enough to break Tor's anonymity, 2009).

Many types of research have defined the anonymity degree, and provided it based on different anonymous networks, like the degree of anonymity in P2P networks, MIX and Crowds network, and anonymous communication systems (Claudia Diaz, 2003). Generally, the degree of anonymity is calculated through the Shannon Entropy which was defined by Claude Elwood Shannon in 1948. It is the formula of possibility which Shannon presented in the thesis "A Mathematical Theory of Communication" (Shannon, 1948). The calculation of the degree of anonymity is aimed to determine the possibility that the attackers can identify the sender or caller of communication on the connection. But every system that is anonymous gives a distinct degree of privacy. Thus, to calculate and measure the anonymity degree is a very difficult task, according to (Claudia Diaz, 2003) in Onion Routing.

$$d = H(X)/Hm = log2(S)/log2(N)$$

d: Degree of anonymity.

N: The size of the anonymity set.

S: The size of the subset of the anonymity set.

The basic question behind this study is that whether the entropy model can be utilized to find the degree of anonymity in an anonymous network like The Onion routing network. Research by Paul Syverson (Syverson, 2013) suggests that the Shannon Entropy method fails to calculate the anonymity degree in the Tor network. The Shannon Entropy method has not been successful to address and present abilities to the adversaries concerning the data attained from the Tor network. Therefore, the idea of entropic anonymity makes an assumption of an adversary model and anonymous system as impractical. Another reason why the anonymity degree of The Onion Routing (Tor) network cannot be calculated using the entropy method is that the real number of Tor clients at a definite time is not known.

When designing anonymous communication systems, it is difficult to determine the abilities of an adversary. An adversary can be an observer who is able to view a connection but is not able to initiate links (For example a sniffer on the Internet). The adversary can also be a disruptor that is suspending traffic on a connection. Another ability of an adversary is being a hostile user who starts or terminates links. The adversary maintains relays which are used as links between the sender and the receiver. It can redirect the links in addition to start new links. (Paul Syverson, 2001)

The Tor network is mainly susceptible to Global Passive Adversary (GPA). The Global Passive Adversary (GPA) model can view all the traffic on a connection in the network. Thus, the abilities of GPA are much stronger for the Tor network to practically avoid intrusion. Hence, on a Tor network, the adversary must compromise each relay that links the caller and recipient to identify the caller and the callee. Thus, if the adversary fails to control one of the relays then the Tor network remains anonymous.

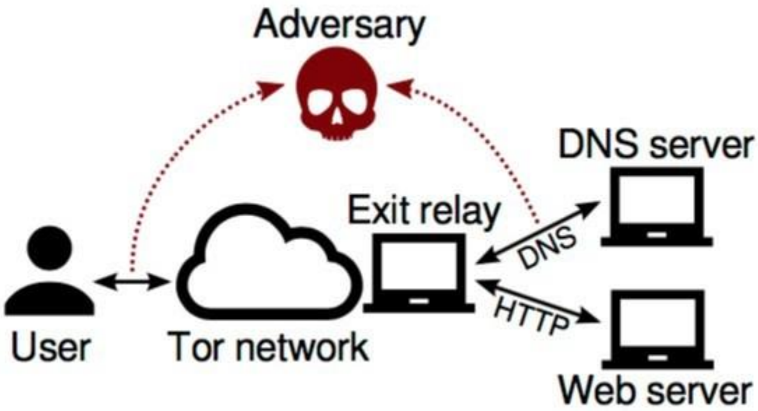


Figure 10: Attack on TOR

1.4.4 Attacks on Tor Network

Some types of attacks that usually arise while using a network are replay attack, Denial of Services (DoS), packet counting attack, collusion attack, packet volume attack, message coding attack, flooding attack, message delaying attack, timing/latency attack, and intersection attack (Volker Fusenig, 2008). In summer 2013, a post on the Tor mailing list took the attention on a huge usage of the network and users in a very short amount of time. At the beginning, no one was able to tell why this happened, but after a while, the researchers found out that it was caused by a very big botnet that suddenly switched to Tor. The HTTP protocol over Tor with centralized structure what was the botnet using. It uses a pre-configured old version of Tor to connect to the network. (Munson, 2013) (ProtACT Team, 2013)

Botnets which is controlled by a “bot master” is a collection of hundreds or thousands of the computers which have been compromised. In August of 2013, the Tor network was used by Botnet to attack its target. Upon this increase in Tor users from one million to more than five million, the Quality of Service (QoS) decreased rapidly in Tor. Also, this resulted in a higher latency in the Tor network.

1.5 Open Virtual Private Network (Open VPN)

A Virtual Private Network (VPN) is a connection method that adds security and privacy to private and public networks. It allows anonymous connection between two or more networks via the internet. Virtual Private Networks (VPNs) are mainly utilized for securing communications which are point-to-point. In the empirical study, Open VPN is used as a VPN. There are three reasons behind choosing it. First, the Open VPN can be used for the encapsulation method. It makes use of TCP streams to transmit UDP streams through the Tor network. Secondly, Open VPN is identifyees the VoIP clients (each with OpenVPN IP address). Lastly, Open VPN uses a secured channel for sending data from caller to the receiver. Therefore, communications on Open VPN have end-to-end protection.

1.5.1 Encryption

Using Open VPN for sending VoIP will increase data protection and privacy. The OpenSSL library in Open VPN is used to encrypt the data and communication medium. All traffic is first encrypted before forwarding it to the Open VPN. The communication network between the Open VPN server and the user is encrypted. Once the connection has been created, the VoIP user will come in contact with another user of VoIP via an encrypted connection. Thus, the Open VPN user and the VoIP user are connected to the Open VPN Server.

1.5.2 Authentication

There are many methods to ensure authentication in Open VPN. For instance, certificate-based and Preshared keys authentication. Certificate-based authentication depends on cryptosystem RSA (Rivest–Shamir–Adleman) certificates and keys. Among all the authentications in Open VPN., it is the safest form. It is developed using OpenSSL command. Also, it is included in OpenSSL distribution. Also, the certificate holders name and email address are other fields which are secured by RSA certificates. Pre-shared secret key authentication many benefits: It is a simple, easy and flexible authentication method in Open VPN.

1.6 Asterisk

Asterisk is an open source telephone platform built to run on Linux frameworks and has a large number of integrated communications applications to reflect the experience in telephony. Asterisk's power lies in its customizable nature to suit the needs of all small or large enterprises and this feature is not available as a free business solution.

Asterisk includes a range of standard applications such as voice mail, voice and video conferencing, call center management software, call forwarding and many more. In addition to its flexibility in integration and compatibility with other technologies used in business. A wide range of interfaces designed to manage Asterisk and after checking the most appropriate and most useful and stable is the Elastix interface. Also, Elastix is a leveraged Freepbx and added many applications like call center which make managing large call groups more effective which is not supported in Asterisk. (Alcantara, 2017)

1.6.1 Elastix

Elastix is an integrated communication suite based on Asterisk that combines voice over IP, mail, instant messaging, video conferencing, a fax server and more. (Li, Li, Wang, & Nan, 2011)

Elastix enables switching from a traditional telephone system to own communications system, which meets all the needs of companies and organizations to establish telephone communications on their servers without reference to the telephone exchange system. Elastix provides telephony and other communication technologies to make a more productive and efficient organizational environment.

Elastix combines the following basic components:

- Asterisk platform (version 1.4)
- Flash Operator Panel.
- Hylafax integrated digital fax system.
- Instant Messaging is an Openfire system.
- Application to manage audio conferences.
- Interface to manage the freePBX settings.
- Integrated communications reporting system.
- OSLEC.
- Integrated email server integrated with Postfix system.
- An interface for email via the Round Cube webmail browser.
- CentOS operating system, a Linux and business-oriented model.

Elastix programmers have set up a web interface that allows easy access to all of the components.

Elastix 4 is used in this documentation, it is an open source licensed. Later releases starting from Elastix 5 are released under the terms of the 3CX license.

1.7 Anonymous

1.7.1 Introduction

VoIP may be a technology gives the opportunity for people to make phone calls through the internet instead of Public Switched Phone Network (PSTN).
Because VoIP offers money savings with additional versatile and advanced options over Plain Telephone System (POTS), a lot of voice calls nowadays are done through VOIP. (Shiping Chen, 2006)

For privacy reasons, people typically wish their phone communication to be anonymous and don't wish people understand that they need even talked over the phone.
The use of VoIP has made it easier to attain anonymous voice call, particularly once VoIP calls between computers. this can be as a result of VoIP calls between peer computers haven't any phone numbers related to them, and that they might simply be protected by end-to-end coding and transfer through anonymized networks like (Tor, Onion Routing, Freedom).

People intuitively suppose their pc to pc VoIP calls might stay anonymous if they're encrypted end-to-end and routed through an anonymizing network.

Our goal is to research sensible techniques for the effective chase of anonymous VoIP calls on the web and give some examples of weakness of a number of the current anonymous network systems.

For example, the findnot.com is an internet anonymizer which supports IP transport protocol, and can be used in Skype P2P VoIP (UDP) calls through the anonymous VPN which is provided by findnot.com. Skype offers free pc to pc VoIP calls supported KaZaa peer-to-peer technology (Phillip Kisembe, August 2017).
Several properties of Skype have created it a very good candidate for anonymous VoIP calls on the Internet:

- It is free and widely used
- Skype traffic is encrypted from end-to-end by 256-bit AES encryption.
- Skype tries to reduce latency by finding dynamic routes and encrypted calls through many peers
- It uses a P2P signaling protocol to initiate the VoIP calls.
- Skype can automatically cut through most firewalls and NAT gateways with using the intermediate peers.

Almost all of Skype calls are UDP, it is hard directly use anonymizing systems (Onion Routing, Tor or any other service), who do not support anonymization of all UDP flows, to anonymize Skype VoIP calls.
Anonymizing Voice over IP is to some degree hard to achieve still possible.it is not only the idea of hiding the IP address, which can be easily done, but it is also more voice recognition and latency of the Tor network.
For individuals behind Tor, who know one another, it is easy to hide the fact that they are having a call with each other from their ISP, man in the middle, etc. But this will not make the call anonymous, just the face they know each other.

1.7.2 Anonymous Systems

Kohntop and Pfitzmann defined Anonymity as making a person unidentifiable from others in a network. There can be many reasons for using anonymous communication in transferring information. For instance, to conceal identity from the recipient of the data or from a possible future attack (Andreas Pfitzmann, 2009). Anonymity can be categorized as relationship anonymity, sender anonymity, and receiver anonymity. Relationship anonymity can be defined as the anonymity in which the link between the caller and the callee is hidden or unidentifiable. The communication between the sender and the receiver is hard to be identified in such anonymity. Sender anonymity refers to anonymity in which the identity of the sender is concealed but the recipient’s data is not hidden. Receiver anonymity can be achieved when the identity of the recipient is concealed.

In SIP, security is achieved by any of the following four methods. First is called absolute anonymity in which the data of the caller is concealed from everyone on the network. In the second method, the identity of the caller is hidden from only the recipient. Third, the identity of the caller is hidden from the caller’s communication network provider. Fourth the identity of the caller is hidden from the recipient's communication network provider. (Lokesh Bhoobalan, 2011)

The anonymous systems are of two types. First, an anonymous network with low latency and a second anonymous network with high latency. Some examples of the low-latency network are Anonymizer, JAP, Pipe Net, and Tor. Meanwhile, an example of a high-latency network is Crowds.

1.7.3 Crowds

Crowds was developed by Aviel Rubin and Michael Reiter in 1998. The Crowds helps in achieving user anonymity in the following cases: guaranteeing web-browsing anonymity, by thwarting websites from identifying users by hiding each user as a member of the Crowds and accessing websites (George Danezis, 2010). One of the major shortfalls of Crowds is that it does not ensure protection from denial of service (DoS) attacks by intruders nor security against worldwide eavesdroppers (Xu Jing, 2010). Nowadays, Crowds is one of the most used anonymous networks in the world.

Users and servers are members of the Crowds network. Every user in Crowds is called "jondos" which means anonymity of the users of the network. Jondos is taken from the word John Doe. Every user, jondos is linked with Crowds network from where it communicates with other users on the Crowds network (Rubin, 1997) (Jie Wu, 2010).

The privacy in Crowds is achieved by hiding the identity of users (jondos) when they send data online at random. Thus, every user is not able to recognize with whom he is contacting on the network. However, the path of communication which a user adopts to communicate with others is valid for 24 hours after which it is altered in the same random procedure. Each message from the sender to the receiver in a Crowds network is protected using a key which is created when a person establishes a link on Crowds network. An example of Crowds concept is clearer in Figure 11.

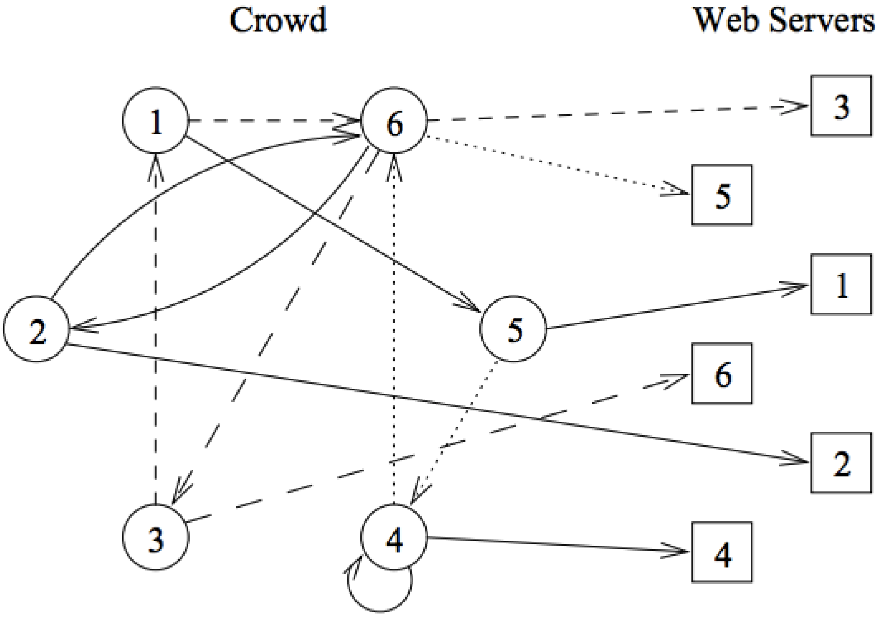


Figure 11: Paths in Crowds *(Rubin, 1997)*

1.7.4 Java Anon Proxy (JAP)

Dresden Technical University, Regensburg University, and Schleswig-Holstein Privacy Commission jointly introduced the Java Anon Proxy (JAP). JAP is a proxy system with a single static IP address used by every user on the JAP network

JAP can be used for anonymous web browsing. Connection in JAP goes through many intermediaries which called Mixes. JAP has a predefined sequence of Mixes called Mix Cascade. The client can choose between those different Cascades.
JAP has a different structure to TOR that consists of relays and they are anonymous themselves. (Prof. Dr. Hannes Federrath, n.d.)

To achieve the highest privacy in JAP, it is necessary to have a maximum number of clients on the JAP network. But, the latency on the network increases as the JAP clients use less bandwidth or increase the transmission rate of the data.

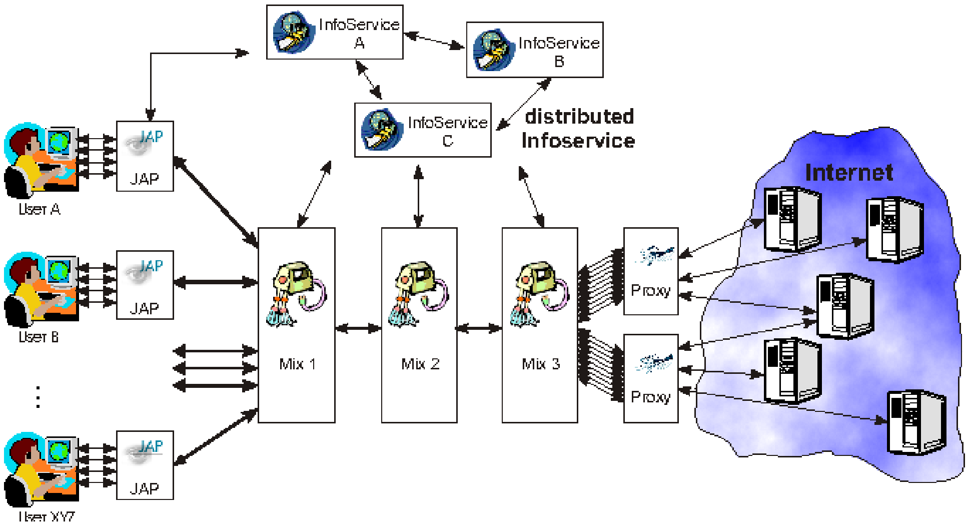


Figure 12: AN.ON system architecture

1.7.5 Related Work

There has been little research on the topic of how to run VoIP in an anonymous network such as The Onion Routing network. Some of these researches are mentioned here. (Marc Liberatore, 2011)

Liberatore studied the performance of VoIP in an anonymous network. He suggested a network through which a user can browse privately in VoIP. His method has similar features to the Tor network, however, it utilized UDP stream rather than a TCP stream. Hence, his network was also known as “Private Tor”. His network was deployed and tested on PlanetLab. It is a multicontinental network where large networks are installed and tested. Liberatore’s network performance was checked in 40, 49, and 121 proxies in Asia, America, and Europe respectively. The result of the performance depicted QoS acceptability of 46 % in Asia, 86 % in America and 72 % in Europe. The detailed research report mentions the number of proxies or relays used; however, it does not show the total bandwidth or the number of VoIP users in the network. Thus, the research data of Liberatore is insufficient to check the relationship between relays/proxies, bandwidth, and users.

To transfer audio packets via the internet, TORFone was developed. Its function is very much like Skype apart from a few basic distinctions. These distinctions are:

1. Unlike Skype, TORFone is not centralized. Thus, username and other registration formalities are not required in it.

2. TORFone application results in a latency of 2 to 4 seconds. of voice latency, as the data passes over many relays which are situated worldwide. (Gegel, 2012)

1985phone is one more proxy network for VoIP which works similar to the Tor network. Jonathan Corbett developed the 1985phone in June of 2013. The similarity between the 1985phone and the Tor network is that in both networks the data is transmitted to the target via many relays. Users of 1985phone worked as relays for other users. Thus, 1985phone failed to effectively implement due to limited resources, such as a shortage of mobile phone capabilities, the lack of batteries, and also limited bandwidth for using data on a mobile phone.

1.7.6 Research Methods and Design

As discussed earlier, the purpose of this research was to find out the Quality of Service of the VoIP in private systems such as The Onion Routing.

This section elaborates the research method which was utilized to find out the answers to the research questions which were mentioned in section 1. Apart from this, various instruments, data analysis techniques and data collection procedure adopted in this research are also discussed in this section.

The aim of the investigation was to find out the performance of VoIP through the Onion Routing network. Transferring UPD packets over VPN and through TOR network. Like Onion, TCP channel tunneling Tor, Tor channel tunneling a new TCP protocol stack using VPN, the new TCP tunneling VoIP UDP.

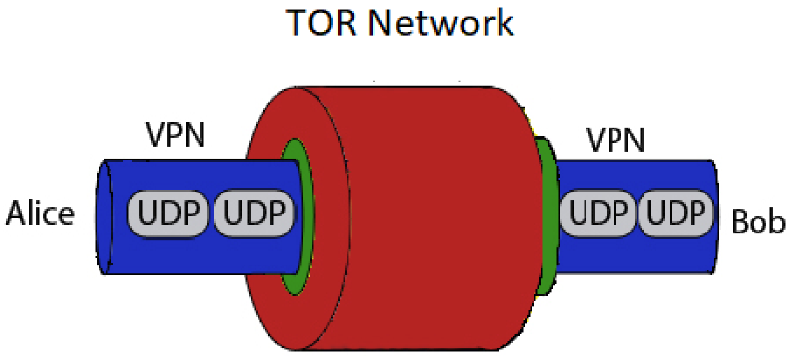


Figure 13: VoIP over VPN through Tor

Mainly the configuration split the data transmission, Clients will have 2 IPs. VoIP packets will use the TOR and VPN connection, any other connection to the internet will be kept connected to the normal internet. Socks proxy for the port 9050 were used. PC will connect to TOR and the remote server, which is running Elastix Server.

1.8 Implementation

In the implementation, three Virtual Machines (Oracle VM) computers were used, all of them were connected to the internet. Two were used as Clients, they had OpenVPN Client and MySIP softphone.
The third computer was the OpenVPN Server, VoIP (Elastix) Server and it is configured to become a TOR node with an Onion address.

The Clients VMs were running Windows 7 and Bridged to the Host PC network adapter. The server was Elastix (4.0.76) x86 based on CentOS Linux release 7.0.1406 (CORE).

1.8.1 Elastix Server configuration

Elastix server must have some extra Packages for the EPEL(Enterprise Linux) repo are required. And OpenVPN by default is not included in the CentOS repositories. The EPEL repo is the other repositoriers are managed by the Fedora Project, which also contains non-standard popular packages.

```
yum install epel-release
```

- Getting OpenVPN installed: First OpenVPN should be installed. Also, to generate the SSL key pairs Easy RSA should be installed, which will secure the VPN connections.

```
yum install openvpn easy-rsa -y
```

- OpenVPN configuration: OpenVPN has already a sample of configuration files in its documentation directory. This sample could be used and copy the sample server-sample.conf file as a starting point for own configuration file.

```
cp /usr/share/doc/openvpn-*/sample/sample-config-files/server.conf

/etc/openvpn/server-sample.conf
```

Reconfigure the file to only listen on localhost (127.0.0.1). Using text editor vi (visual editor) in Linux Also, Port 1194 TCP is used because TOR supports only TCP.

```
vi /etc/openvpn/server.conf
local 127.0.0.1
port 1194
proto tcp
dev tun
```

Later when keys are generated, the default Diffie-Hellman encryption length for Easy RSA will be 2048 bytes, so filename should be pointed to dh.pem.

```
dh /etc/openvpn/keys/dh.pem
```

Configuring the server and provide VPN subnet for OpenVPN to draw addresses from. The server has IP: 10.10.0.1, the other IPs will be available for users. Each client will be able to connect the server IP address 10.10.0.1. And allow different clients to be able to "see" each other. So, they can make asterisk call and answer.

```
server 10.10.0.0 255.255.0.0
client-to-client
```

Generating certificates will be done later on.

```
ca /etc/openvpn/keys/ca.crt
cert /etc/openvpn/keys/server.crt
key /etc/openvpn/keys/server-nopass.key
```

The parameter keepalive mandate causes ping-like messages to be sent forward and backward over the connection with the goal that each side knows when the opposite side has gone down.

Ping every 10 seconds accept that remote companion is down if no ping got amid in a 120 second time period.

```
keepalive 10 120
```

A cryptographic cipher has been set. This should be the same in the client's config file.

```
cipher BF-CBC        # Blowfish (default)
;cipher AES-128-CBC   # AES
;cipher DES-EDE3-CBC  # Triple-DES
;cipher AES-256-CBC   # AES
```

AES-128-CBC: provides more than enough security for VPN. And it is not broken. Also, considered one of the best for embedded OpenVPN devices that do not support the more modern symmetric-key cryptographic block ciphers GCM (Galois/Counter Mode) standard.

For compression compatible with older clients' comp-lzo has been used. Also, this should be enabled or disabled by both the server and clients.

```
comp-lzo
```

OpenVPN should keep running without any privileges once it has begun, so it needs to keep running with a user and group of nobody. This can be done by uncommenting these lines:

```
user nobody
group nobody
```

The persist options will try not to access some certain resources on a restart that may never again be available due to the privilege downgrade.

```
persist-key
persist-tun
```

Set status and logfile and level of log file verbosity, 9 is extremely verbose.

```
status /var/log/openvpn-status.log
log         /var/log/openvpn.log
log-append  /var/log/openvpn.log
verb 4
```

The final file should be like

```
local 127.0.0.1
port 1194
proto tcp
dev tun
ca /etc/openvpn/keys/ca.crt
cert /etc/openvpn/keys/server.crt
key /etc/openvpn/keys/server-nopass.key
dh /etc/openvpn/keys/dh.pem
server 10.10.0.0 255.255.0.0
client-to-client
keepalive 10 120
# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.
cipher BF-CBC        # Blowfish (default)
;cipher AES-128-CBC   # AES
;cipher DES-EDE3-CBC  # Triple-DES
;cipher AES-256-CBC   # AES
comp-lzo
user nobody
group nobody
persist-key
persist-tun
status /var/log/openvpn-status.log
log         /var/log/openvpn.log
log-append  /var/log/openvpn.log
verb 4
duplicate-cn
```

- Generating Keys and Certificates: After the server is configured, keys and certificates should be generated. Easy RSA installs one script to generate these keys and certificates. A key directory should be created, Key and Certificate generation should be also done in the same directory.

```
mkdir -p /etc/openvpn/easy-rsa
cp -rf /usr/share/easy-rsa/3/* /etc/openvpn/easy-rsa
cd /etc/openvpn/easy-rsa/
```

First, *Init pki* env and then Build the certificate authority, the CA PEM password, and CA Common Name here is *„freepbx"* and *„Anonymous CA"*. */etc/openvpn/easy-rsa/pki/ca.crt*

```
./easyrsa init-pki
./easyrsa build-ca
```

Now for generating the Diffie-Hellman key exchange file. This command takes a while to complete. The file: */etc/openvpn/easy-rsa/pki/dh.pem*

```
./easyrsa gen-dh
```

Also Server Key and Certificate need to be generated. prompting the server PEM password and the CA PEM password *„freepbx"*.
And the Server Key */etc/openvpn/easy-rsa/pki/issued/server.crt and /etc/openvpn/easy-rsa/pki/private/server.key*

```
./easyrsa build-server-full server
```

All of the clients will also need certificates to be able to authenticate. These keys and certificates will be shared with clients. Although, it is possible to generate separate keys and certificates for each client but in the following scenario same shared certificate and key for all clients.
/etc/openvpn/easy-rsa/pki/issued/client.crt and /etc/openvpn/easy-rsa/pki/private/client.key

```
cd /etc/openvpn/easy-rsa
./easyrsa build-key client
```

And copy the file to the OpenVPN */etc/openvpn/keys*

```
mkdir -p /etc/openvpn/keys
cp /etc/openvpn/easy-rsa/pki/ca.crt /etc/openvpn/keys
cp /etc/openvpn/easy-rsa/pki/dh.pem /etc/openvpn/keys
cp /etc/openvpn/easy-rsa/pki/issued/server.crt /etc/openvpn/keys
cp /etc/openvpn/easy-rsa/pki/private/server.key /etc/openvpn/keys
```

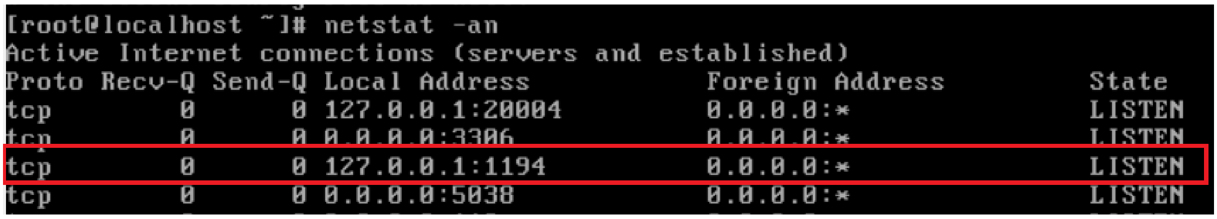
Finally, Server-key should have nopassword.

```
cd /etc/openvpn/keys
openssl rsa -in server.key -out server-nopass.key
```

- Starting OpenVPN: Open VPN is ready and the service can be started.

```
/usr/sbin/openvpn --config /etc/openvpn/server.conf --daemon
```

Checking the running services, OpenVPN is just listening on 127.0.0.1:1194. This will be used in the Hidden Service



```
[root@localhost ~]# netstat -an
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:20004         0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:1194          0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:5038            0.0.0.0:*               LISTEN
```

Figure 14: OpenVPN Service

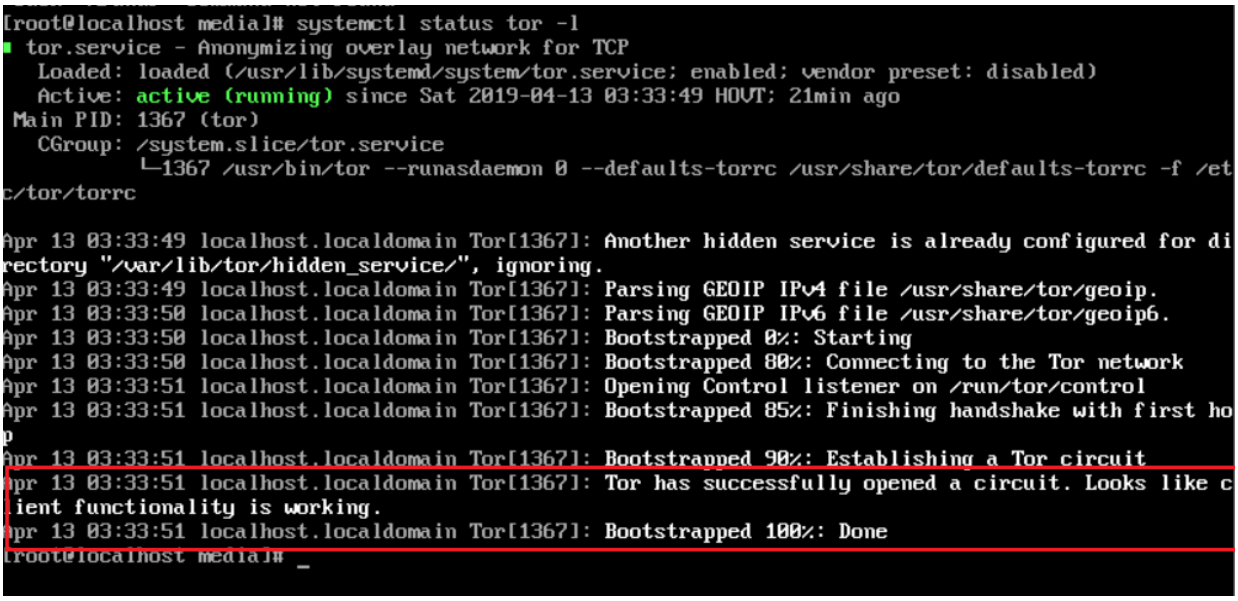
- TOR installation: TOR should be also installed.

```
yum -y install tor
```

- Configuring Tor Hidden service: In this point, OpenVPN should run under the Hidden Service or TOR. Only the TOR network will be able to connect to the OpenVPN Server.

```
cat<<EOF>>/etc/tor/torrc
HiddenServiceDir /var/lib/tor/hidden_service/
HiddenServicePort 1194 127.0.0.1:1194
EOF
```

- Start TOR Service: After configuring the OpenVPN, TOR Service is ready to start.

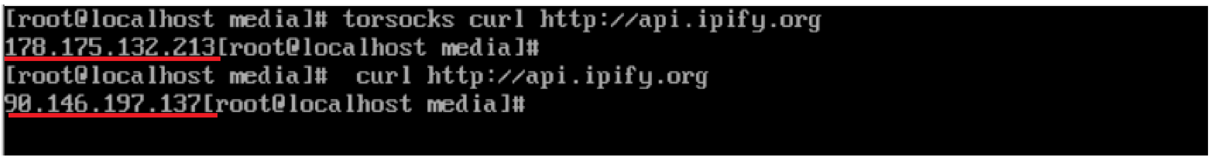


```
[root@localhost media]# systemctl status tor -l
● tor.service - Anonymizing overlay network for TCP
   Loaded: loaded (/usr/lib/systemd/system/tor.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2019-04-13 03:33:49 HOVT; 21min ago
 Main PID: 1367 (tor)
   CGroup: /system.slice/tor.service
           └─1367 /usr/bin/tor --runasdaemon 0 --defaults-torrc /usr/share/tor/defaults-torrc -f /et
c/tor/torrc

Apr 13 03:33:49 localhost.localdomain Tor[1367]: Another hidden service is already configured for di
rectory "/var/lib/tor/hidden_service/", ignoring.
Apr 13 03:33:49 localhost.localdomain Tor[1367]: Parsing GEOIP IPv4 file /usr/share/tor/geoip.
Apr 13 03:33:50 localhost.localdomain Tor[1367]: Parsing GEOIP IPv6 file /usr/share/tor/geoip6.
Apr 13 03:33:50 localhost.localdomain Tor[1367]: Bootstrapped 0%: Starting
Apr 13 03:33:50 localhost.localdomain Tor[1367]: Bootstrapped 80%: Connecting to the Tor network
Apr 13 03:33:51 localhost.localdomain Tor[1367]: Opening Control listener on /run/tor/control
Apr 13 03:33:51 localhost.localdomain Tor[1367]: Bootstrapped 85%: Finishing handshake with first ho
p
Apr 13 03:33:51 localhost.localdomain Tor[1367]: Bootstrapped 90%: Establishing a Tor circuit
Apr 13 03:33:51 localhost.localdomain Tor[1367]: Tor has successfully opened a circuit. Looks like c
lient functionality is working.
Apr 13 03:33:51 localhost.localdomain Tor[1367]: Bootstrapped 100%: Done
[root@localhost media]# _
```

Figure 15: Tor Service

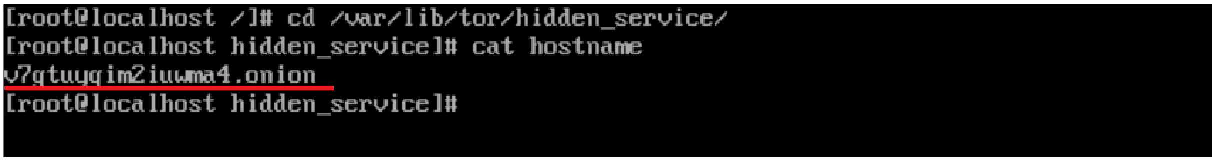
Tor hidden service is running, and also a test shows that too.



```
[root@localhost media]# torsocks curl http://api.ipify.org
178.175.132.213[root@localhost media]#
[root@localhost media]#  curl http://api.ipify.org
90.146.197.137[root@localhost media]#
```

Figure 16: Server 2 IPs

- Get Tor onion address: the server now has an Onion address one the TOR networks.



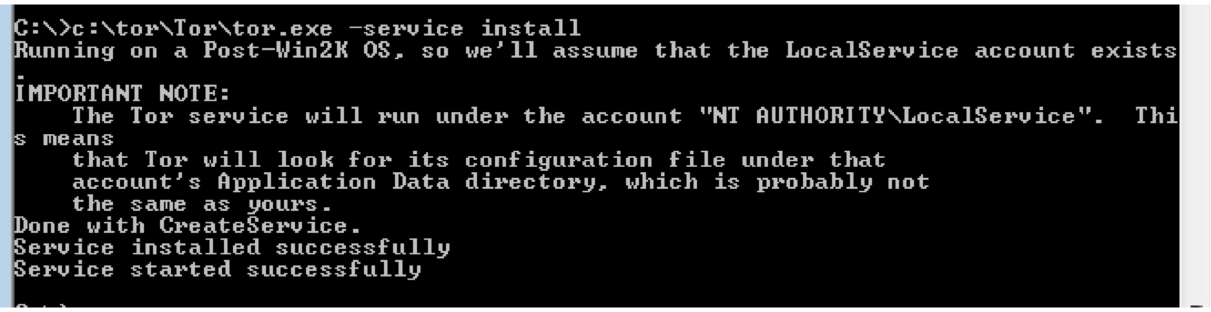
```
[root@localhost /]# cd /var/lib/tor/hidden_service/
[root@localhost hidden_service]# cat hostname
v7gtuyqim2iuwma4.onion
[root@localhost hidden_service]#
```

Figure 17: Server Onion Address

1.8.2 Windows Client configuration

- Installing Tor Client: In order for a client to connect TOR, Expert Bundle Client should be installed. One important thing Tor Geoip (geoip, geoip6) should be copied to the *c:\windows\system32.*
  The Tor service will run under the account "NT AUTHORITY\LocalService".

```
c:\> c:\tor\Tor\tor.exe –service install
```



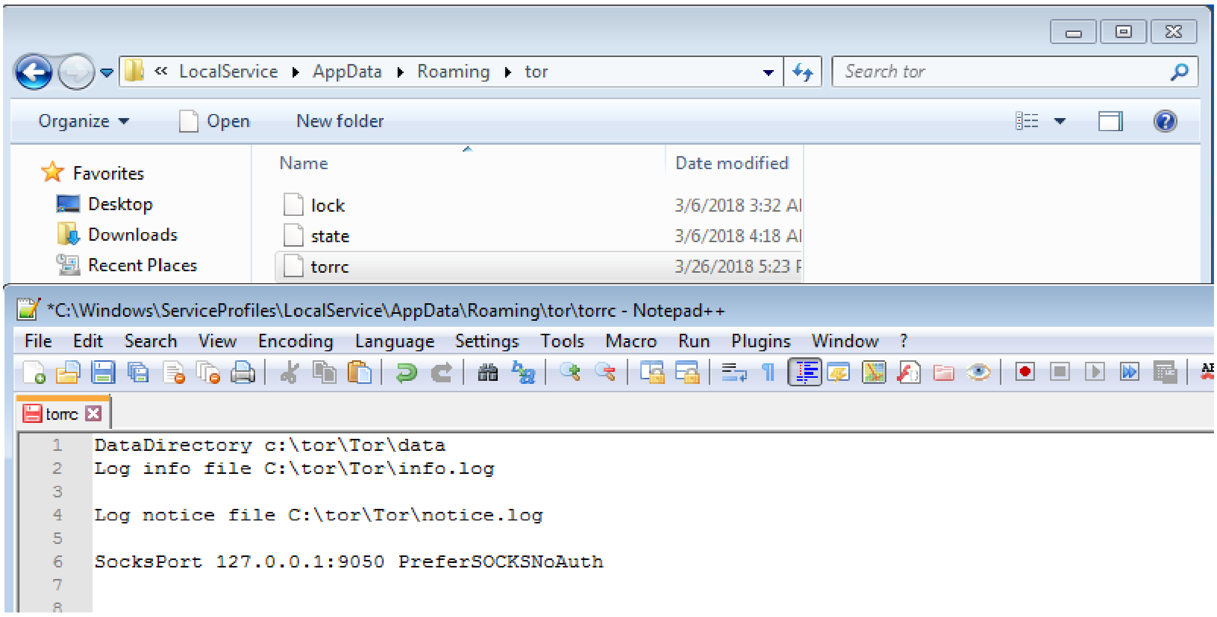
```
C:\>c:\tor\Tor\tor.exe -service install
Running on a Post-Win2K OS, so we'll assume that the LocalService account exists
.
IMPORTANT NOTE:
    The Tor service will run under the account "NT AUTHORITY\LocalService".  Thi
s means
    that Tor will look for its configuration file under that
    account's Application Data directory, which is probably not
    the same as yours.
Done with CreateService.
Service installed successfully
Service started successfully
```

Figure 18: Client's Tor Service

The Tor client uses a Socket Secure (SOCKS) to transfer any communication between the instruments and the Tor with user-specific config is in *torrc* file. Which will let Tor make the connection through SOCKS at 127.0.0.1:9050. (Roger Dingledine N. M.)

Service profile should be also configured. In the *C:\Windows\ServiceProfiles\LocalService\AppData\Roaming\tor*

torrc file has been created with Socks port, data directory, and Logfile configuration.


« LocalService ▸ AppData ▸ Roaming ▸ tor
Search tor
Organize ▾
Open
New folder
Favorites
Desktop
Downloads
Recent Places
*C:\Windows\ServiceProfiles\LocalService\AppData\Roaming\tor\torrc - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
torrc

Name	Date modified
lock	3/6/2018 3:32 AI
state	3/6/2018 4:18 AI
torrc	3/26/2018 5:23 F

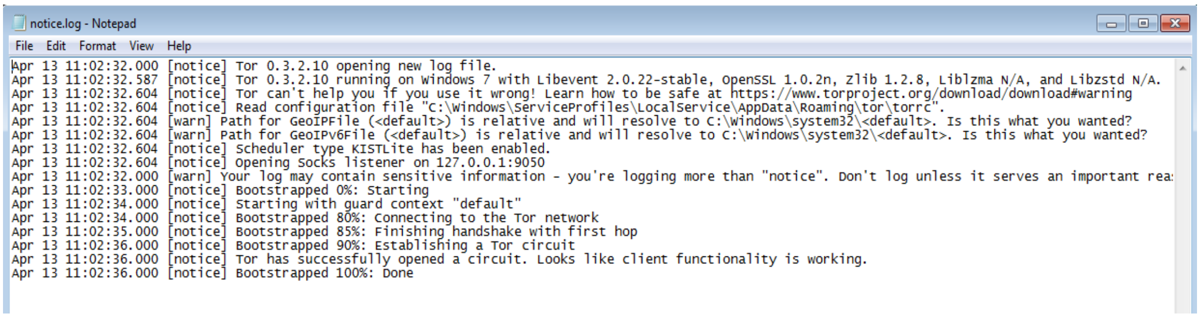
```
DataDirectory c:\tor\Tor\data
Log info file C:\tor\Tor\info.log

Log notice file C:\tor\Tor\notice.log

SocksPort 127.0.0.1:9050 PreferSOCKSNoAuth
```

Figure 19: Client's Tor configuration

After starting the service, Logfile shows connection status.

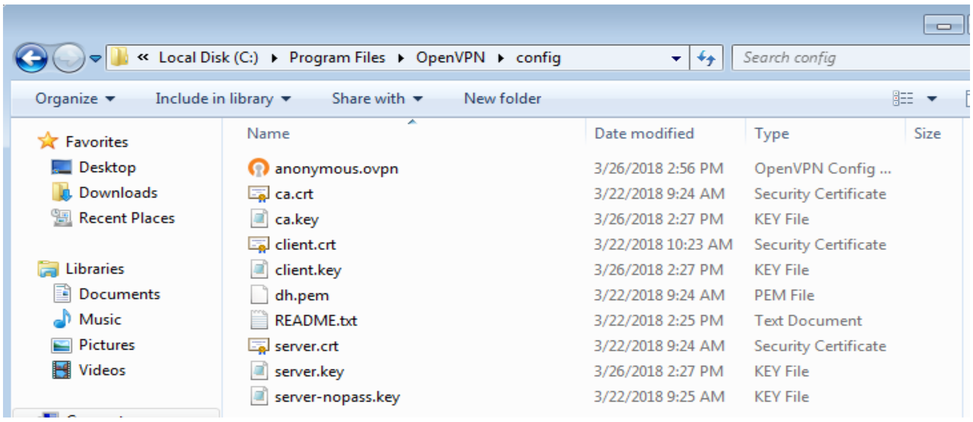


```
Apr 13 11:02:32.000 [notice] Tor 0.3.2.10 opening new log file.
Apr 13 11:02:32.587 [notice] Tor 0.3.2.10 running on Windows 7 with Libevent 2.0.22-stable, OpenSSL 1.0.2n, Zlib 1.2.8, Liblzma N/A, and Libzstd N/A.
Apr 13 11:02:32.604 [notice] Tor can't help you if you use it wrong! Learn how to be safe at https://www.torproject.org/download/download#warning
Apr 13 11:02:32.604 [notice] Read configuration file "C:\Windows\ServiceProfiles\LocalService\AppData\Roaming\tor\torrc".
Apr 13 11:02:32.604 [warn] Path for GeoIPFile (<default>) is relative and will resolve to C:\Windows\system32\<default>. Is this what you wanted?
Apr 13 11:02:32.604 [warn] Path for GeoIPv6File (<default>) is relative and will resolve to C:\Windows\system32\<default>. Is this what you wanted?
Apr 13 11:02:32.604 [notice] Scheduler type KISTLite has been enabled.
Apr 13 11:02:32.604 [notice] Opening Socks listener on 127.0.0.1:9050
Apr 13 11:02:32.000 [warn] Your log may contain sensitive information - you're logging more than "notice". Don't log unless it serves an important rea:
Apr 13 11:02:33.000 [notice] Bootstrapped 0%: Starting
Apr 13 11:02:34.000 [notice] Starting with guard context "default"
Apr 13 11:02:34.000 [notice] Bootstrapped 80%: Connecting to the Tor network
Apr 13 11:02:35.000 [notice] Bootstrapped 85%: Finishing handshake with first hop
Apr 13 11:02:36.000 [notice] Bootstrapped 90%: Establishing a Tor circuit
Apr 13 11:02:36.000 [notice] Tor has successfully opened a circuit. Looks like client functionality is working.
Apr 13 11:02:36.000 [notice] Bootstrapped 100%: Done
```

Figure 20: Tor Logs

To coordinate internet time, the Network Time Protocol (NTP) is utilized. NTP was developed by David Mills in 1980s. It has now achieved the status of standard for the Internet. The newest NTP standard is the Internet Engineering Task Force (IETF). It was made in RFC 5905 (D. Mills, June 2010). It ensures the accuracy of up to one-thousandth of a second. To coordinate the time of the clock, it utilizes UTC. In scenarios such as communication systems, a higher degree of accuracy of time is necessary to find out the latency of the network at a particular point.

- Getting OpenVPN on Client: OpenVPN Client from OpenVPN website (https://openvpn.net/community-downloads/) has been installed. Client and Server certificates were copied from the OpenServer to the OpenVPN Client config folder



Name	Date modified	Type	Size
anonymous.ovpn	3/26/2018 2:56 PM	OpenVPN Config ...	
ca.crt	3/22/2018 9:24 AM	Security Certificate	
ca.key	3/26/2018 2:27 PM	KEY File	
client.crt	3/22/2018 10:23 AM	Security Certificate	
client.key	3/26/2018 2:27 PM	KEY File	
dh.pem	3/22/2018 9:24 AM	PEM File	
README.txt	3/22/2018 2:25 PM	Text Document	
server.crt	3/22/2018 9:24 AM	Security Certificate	
server.key	3/26/2018 2:27 PM	KEY File	
server-nopass.key	3/22/2018 9:25 AM	KEY File	

Figure 21: Client's OpenVPN Configuration

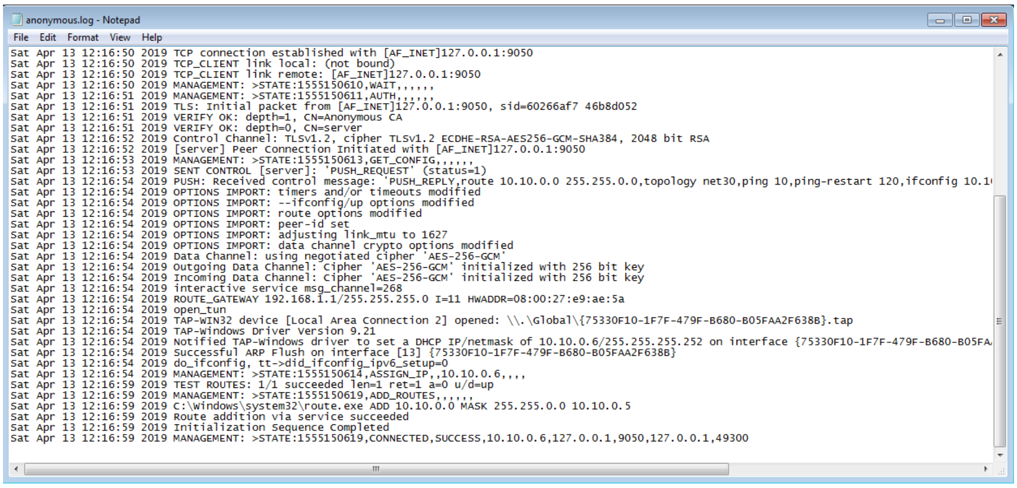
OpenVPN anonymous.ovpn is also configured.

```
client
dev tun
remote v7gtuyqim2iuwma4.onion 1194
proto tcp
resolv-retry infinite
nobind
persist-key
ca ca.crt
cert client.crt
key client.key
comp-lzo
```

```
keepalive 10 120
verb 3
socks-proxy 127.0.0.1 9050
```

Using the remote server address the client will connect over TOR network to this node.

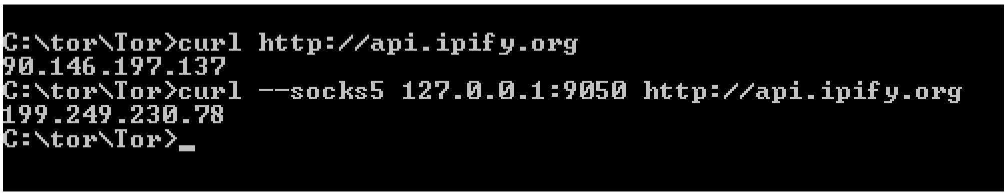
OpenVPN Client will connect to the remote Server which is the Elastix server.



```
Sat Apr 13 12:16:50 2019 TCP connection established with [AF_INET]127.0.0.1:9050
Sat Apr 13 12:16:50 2019 TCP_CLIENT link local: (not bound)
Sat Apr 13 12:16:50 2019 TCP_CLIENT link remote: [AF_INET]127.0.0.1:9050
Sat Apr 13 12:16:50 2019 MANAGEMENT: >STATE:1555150610,WAIT,,,,,,
Sat Apr 13 12:16:51 2019 MANAGEMENT: >STATE:1555150611,AUTH,,,,,,
Sat Apr 13 12:16:51 2019 TLS: Initial packet from [AF_INET]127.0.0.1:9050, sid=60266af7 46b8d052
Sat Apr 13 12:16:51 2019 VERIFY OK: depth=1, CN=Anonymous CA
Sat Apr 13 12:16:51 2019 VERIFY OK: depth=0, CN=server
Sat Apr 13 12:16:52 2019 Control Channel: TLSv1.2, cipher TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 2048 bit RSA
Sat Apr 13 12:16:52 2019 [server] Peer Connection Initiated with [AF_INET]127.0.0.1:9050
Sat Apr 13 12:16:53 2019 MANAGEMENT: >STATE:1555150613,GET_CONFIG,,,,,,
Sat Apr 13 12:16:53 2019 SENT CONTROL [server]: 'PUSH_REQUEST' (status=1)
Sat Apr 13 12:16:54 2019 PUSH: Received control message: 'PUSH_REPLY,route 10.10.0.0 255.255.0.0,topology net30,ping 10,ping-restart 120,ifconfig 10.1
Sat Apr 13 12:16:54 2019 OPTIONS IMPORT: timers and/or timeouts modified
Sat Apr 13 12:16:54 2019 OPTIONS IMPORT: --ifconfig/up options modified
Sat Apr 13 12:16:54 2019 OPTIONS IMPORT: route options modified
Sat Apr 13 12:16:54 2019 OPTIONS IMPORT: peer-id set
Sat Apr 13 12:16:54 2019 OPTIONS IMPORT: adjusting link_mtu to 1627
Sat Apr 13 12:16:54 2019 OPTIONS IMPORT: data channel crypto options modified
Sat Apr 13 12:16:54 2019 Data Channel: using negotiated cipher 'AES-256-GCM'
Sat Apr 13 12:16:54 2019 Outgoing Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
Sat Apr 13 12:16:54 2019 Incoming Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
Sat Apr 13 12:16:54 2019 interactive service msg_channel=268
Sat Apr 13 12:16:54 2019 ROUTE_GATEWAY 192.168.1.1/255.255.255.0 I=11 HWADDR=08:00:27:e9:ae:5a
Sat Apr 13 12:16:54 2019 open_tun
Sat Apr 13 12:16:54 2019 TAP-WIN32 device [Local Area Connection 2] opened: \\.\Global\{75330F10-1F7F-479F-B680-B05FAA2F638B}.tap
Sat Apr 13 12:16:54 2019 TAP-Windows Driver Version 9.21
Sat Apr 13 12:16:54 2019 Notified TAP-Windows driver to set a DHCP IP/netmask of 10.10.0.6/255.255.255.252 on interface {75330F10-1F7F-479F-B680-B05FA
Sat Apr 13 12:16:54 2019 Successful ARP Flush on interface [13] {75330F10-1F7F-479F-B680-B05FAA2F638B}
Sat Apr 13 12:16:54 2019 do_ifconfig, tt->did_ifconfig_ipv6_setup=0
Sat Apr 13 12:16:54 2019 MANAGEMENT: >STATE:1555150614,ASSIGN_IP,,10.10.0.6,,,,
Sat Apr 13 12:16:59 2019 TEST ROUTES: 1/1 succeeded len=1 ret=1 a=0 u/d=up
Sat Apr 13 12:16:59 2019 MANAGEMENT: >STATE:1555150619,ADD_ROUTES,,,,,,
Sat Apr 13 12:16:59 2019 C:\windows\system32\route.exe ADD 10.10.0.0 MASK 255.255.0.0 10.10.0.5
Sat Apr 13 12:16:59 2019 Route addition via service succeeded
Sat Apr 13 12:16:59 2019 Initialization Sequence Completed
Sat Apr 13 12:16:59 2019 MANAGEMENT: >STATE:1555150619,CONNECTED,SUCCESS,10.10.0.6,127.0.0.1,9050,127.0.0.1,49300
```

Figure 22: OpenVPN logs

The client has 2 IP addresses one on the TOR network and another IP address on the Internet.



```
C:\tor\Tor>curl http://api.ipify.org
90.146.197.137
C:\tor\Tor>curl --socks5 127.0.0.1:9050 http://api.ipify.org
199.249.230.78
C:\tor\Tor>_
```

Figure 23: Client's 2 IPs

At present, TCP streams are only transmitted by the Onion routing networks. Generally, the UDP stream is used by voice packets. Thus, voice packets cannot be transmitted through the Onion routing network. In order to achieve this goal, various other methods are available. OpenVPN encapsulation was used to convert the UDP stream to the TCP stream. Consequently, it was made possible for VoIP clients to communicate directly with the Tor network. OpenVPN client’s IP address was used to represent each VoIP client. Moreover, OpenVPN also provided end-to-end security of the communication network.

To record the voice of the caller and the callee, Wireshark was utilized. It is free to use software and is presently being used in multiple fields like communication protocol development, network analyzer, network troubleshooting, and education. Wireshark also has packet filtering capability.

1.8.3 Elastix Users configuration

Elastix Server supports a Web GUI, this Interface is accessible in the local network through any remote web browser. In the URL the Elastix server IP address will give access to the GUI on the web.

Web GUI address could be found in the Server using the „*ifconfig*" command, the username and Password are configured at the installation.

Server credentials. User: root, Pass: freepbx

Mysql. User: root, Pass: freepbx

Web Access. User: admin, Pass: freepbx

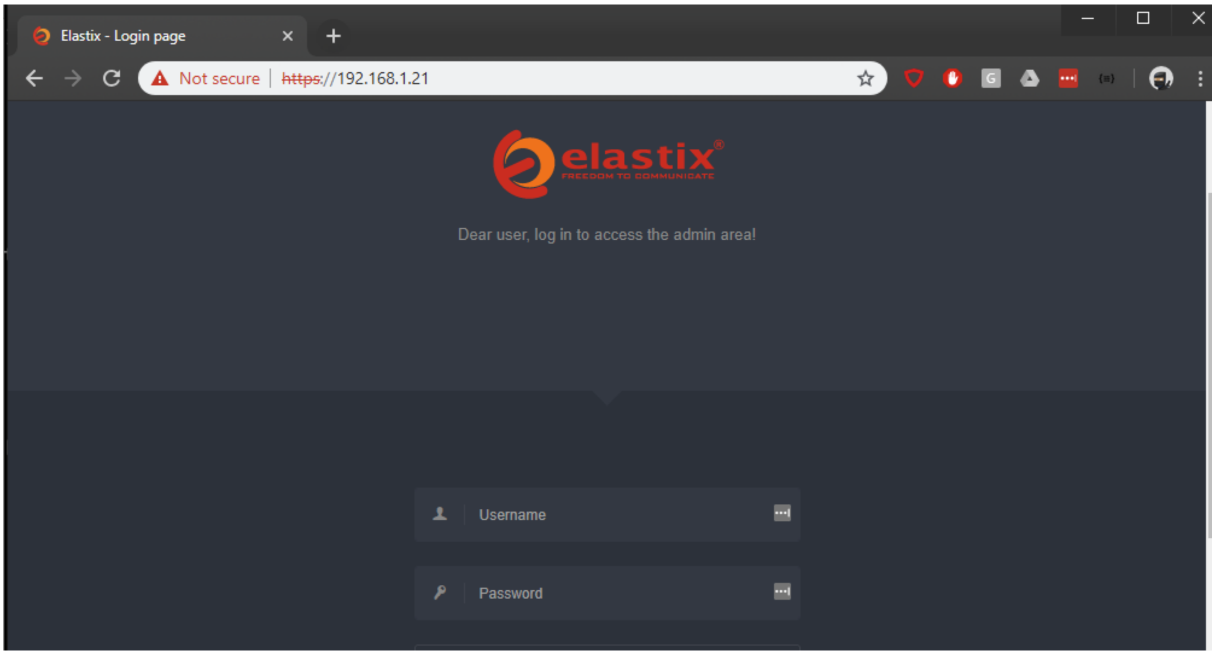
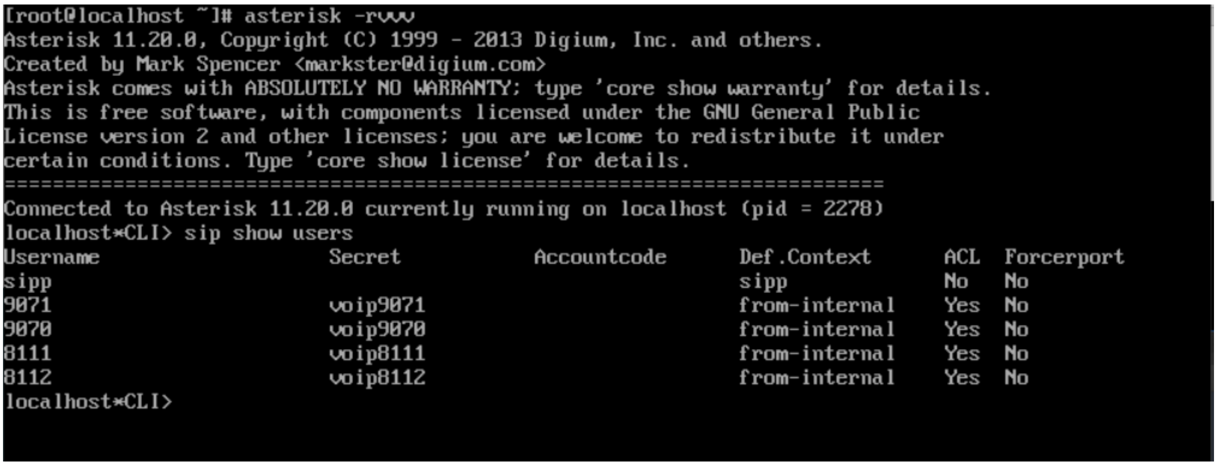


Figure 24: Elastix GUI

Each client should be assigned to a SIP extension, each extension should have at least (User Extension, Display Name, and secret password).

A list of current SIP Users could be shown as the following:



```
[root@localhost ~]# asterisk -rvvv
Asterisk 11.20.0, Copyright (C) 1999 - 2013 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=========================================================================
Connected to Asterisk 11.20.0 currently running on localhost (pid = 2278)
localhost*CLI> sip show users
Username                   Secret           Accountcode      Def.Context      ACL  Forcerport
sipp                                                         sipp             No   No
9071                       voip9071                          from-internal    Yes  No
9070                       voip9070                          from-internal    Yes  No
8111                       voip8111                          from-internal    Yes  No
8112                       voip8112                          from-internal    Yes  No
localhost*CLI>
```

Figure 25: Registered users

1.9 MySIP

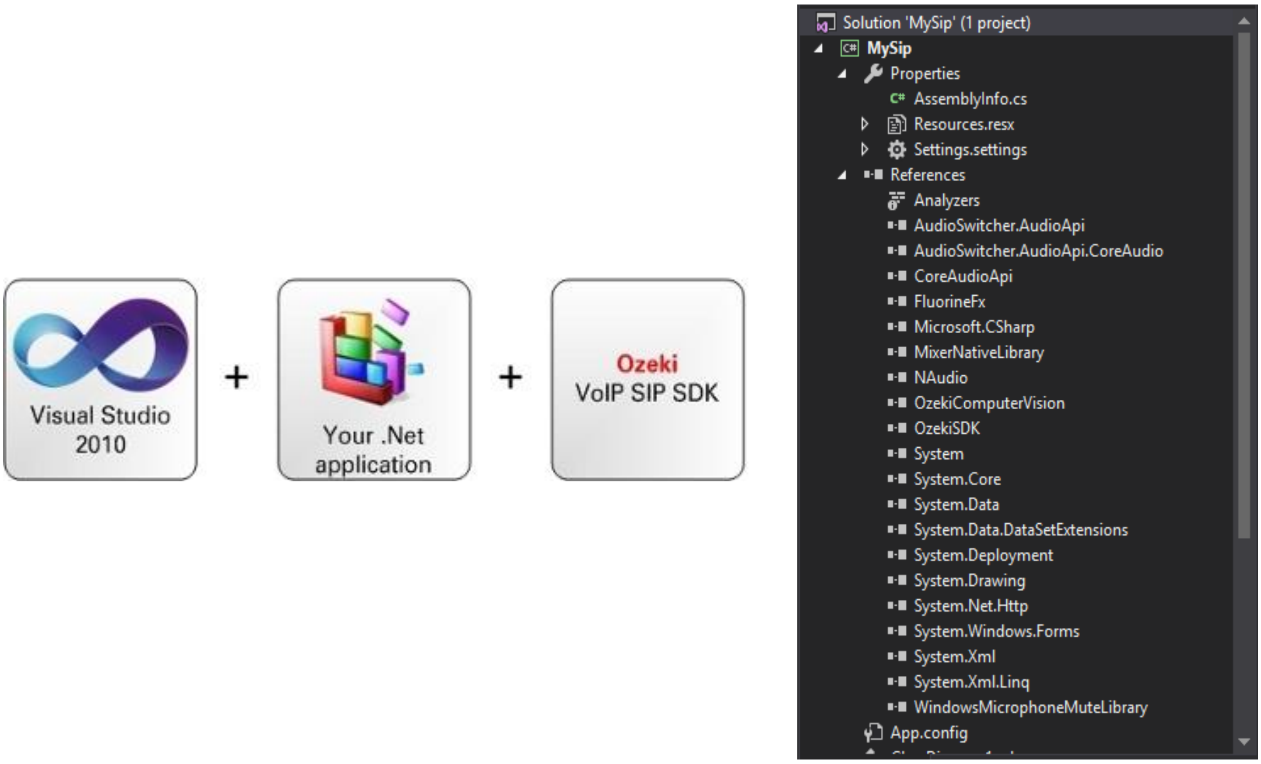
In order to make communication between clients easier and more reliable. MySIP is used, MySIP is a softphone written in C#, this softphone was intended for this project.

It is utilized for VoIP calls. It uses Elastix SIP extension username and password and authenticates with the Elastix server. MySIP can be installed and run on computers operating windows.

The Ozeki VoIP SIP SDK (SDK) is utilized to be able to communicate with VoIP. This library considered one of the best software development kits that allow communication with the VoIP easily and quickly. Some of its advantages (VoIP, n.d.):

- It can be used for any software development under .NET environment.
- Easy to use and user friendly, Demo projects are helpful.
- It is standard compliant and based totally on C# .NET.
- Very optimized for memory and CPU usage.
- Optimized for network resources and bandwidth, also support for example Port-sharing.

It supports Visual Studio starting with Visual Studio 2010



Solution 'MySip' (1 project)
MySip
Properties
AssemblyInfo.cs
Resources.resx
Settings.settings
References
Analyzers
AudioSwitcher.AudioApi
AudioSwitcher.AudioApi.CoreAudio
CoreAudioApi
FluorineFx
Microsoft.CSharp
MixerNativeLibrary
NAudio
OzekiComputerVision
OzekiSDK
System
System.Core
System.Data
System.Data.DataSetExtensions
System.Deployment
System.Drawing
System.Net.Http
System.Windows.Forms
System.Xml
System.Xml.Linq
WindowsMicrophoneMuteLibrary
App.config

Figure 26: .NET Libraries

The SDK in Not free, it works for a trial time, but remove and reinstall the SDK will reactivate the trial time. But DLLs can be used for free with some Commercials.

MySIP uses NAudio Library (Heath) , it supports a variety of APIs and used to record, playback and ready audio. The class relationship diagram showed in figure 27**.**

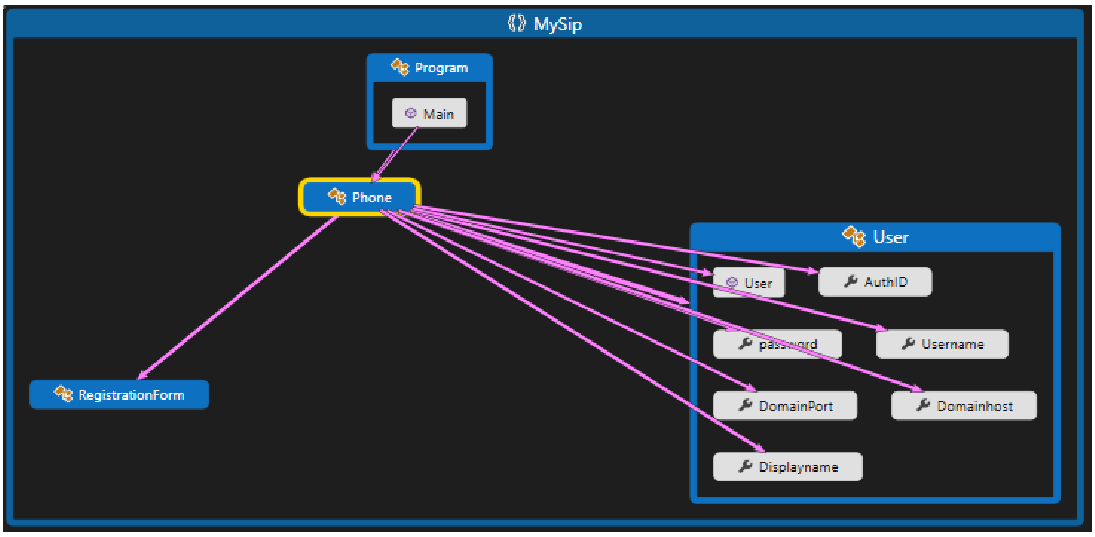


Figure 27: Class relationship diagram

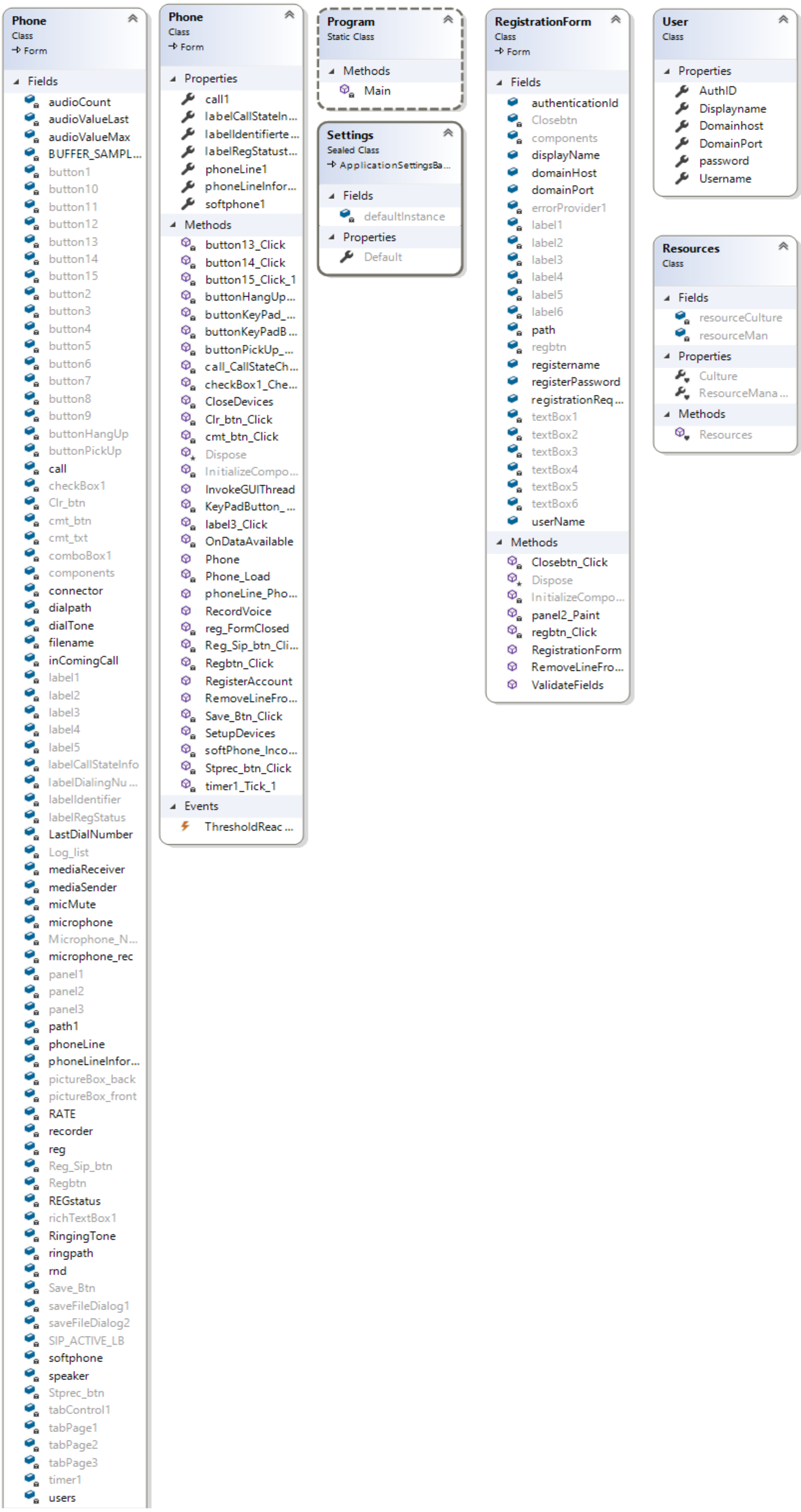


Figure 28: MySIP Class Diagram

There any many other Software in the market, but MySIP is not attached with any VoIP service and it can communicate with any existing SIP Server.

- But why MySIP is better than other produce in the market?

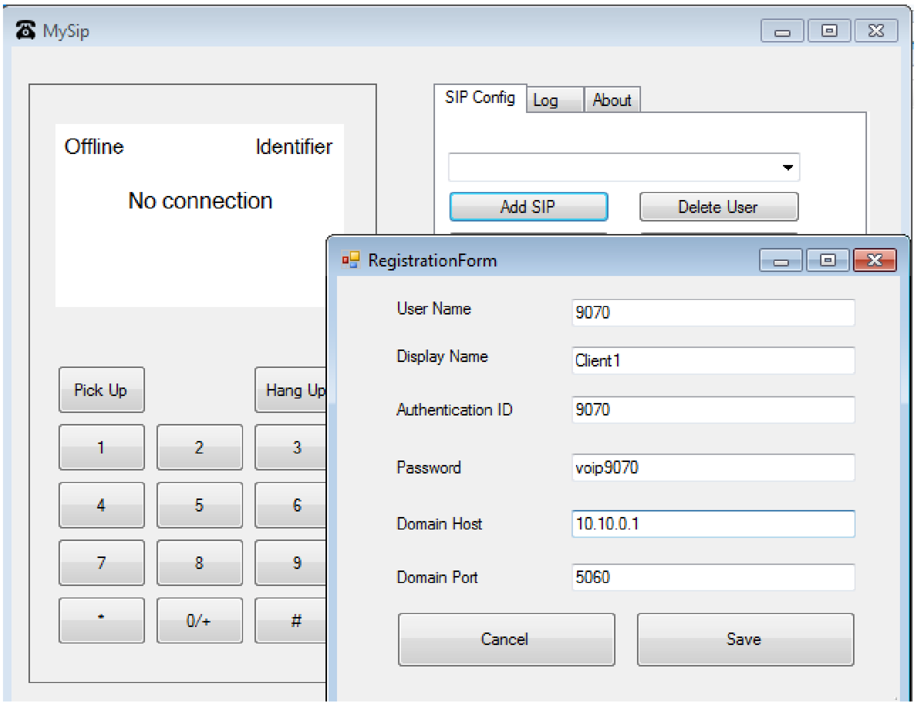
Compared with most famous Commercial Software in the Market X-Lite.

X-Lite is considered as the most popular softphones for VoIP in the market. It is the most basic of the line of VoIP apps that Counter-Path offers, and it is the only free product. Some of the MySIP advantages:

- Free to use, Not attached with any VoIP service
- QoS granted from Ozeki
- LogFile Current SIP Activities (Save + Add Comments)
- Multi SIP Accounts.
- SIP Accounts Manager.
- Record calls.
- Mic Peak-meter.

To add a user, *Add SIP* button will open a new window with the information to fill:

**Domain Host: is the IP Address of the OPENVPN Server (Elastix Server).*



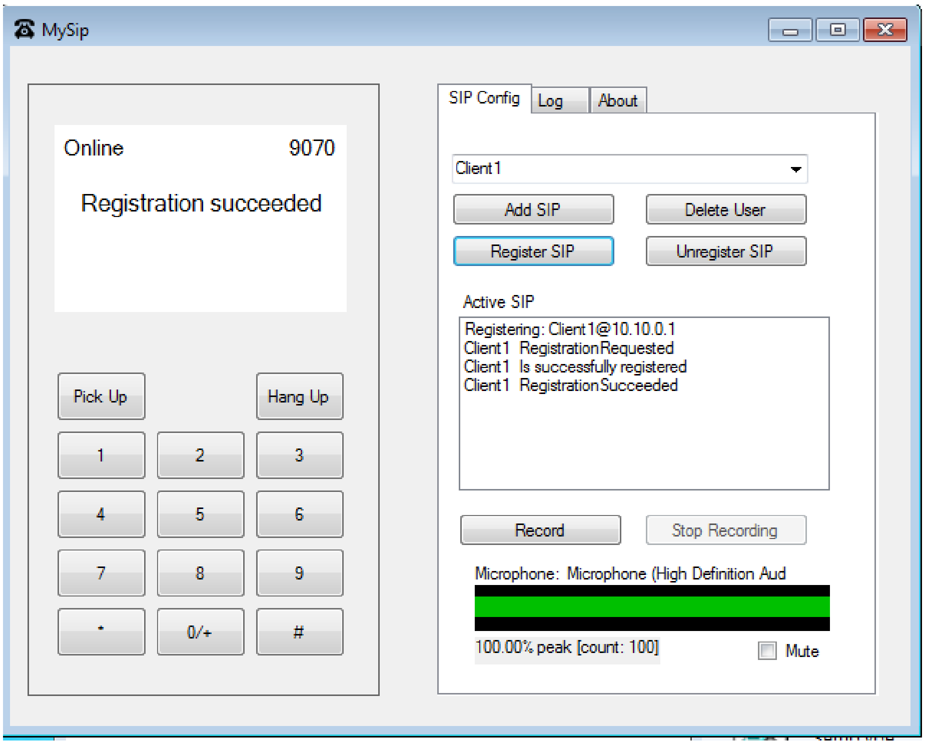
Offline Identifier

No connection

User Name: 9070
Display Name: Client1
Authentication ID: 9070
Password: voip9070
Domain Host: 10.10.0.1
Domain Port: 5060

Figure 29: MySIP Add User

After adding a user *Register SIP* button will register the selected user from the dropdown. A successful message with the current status will be shown in the main display.



Online 9070

Registration succeeded

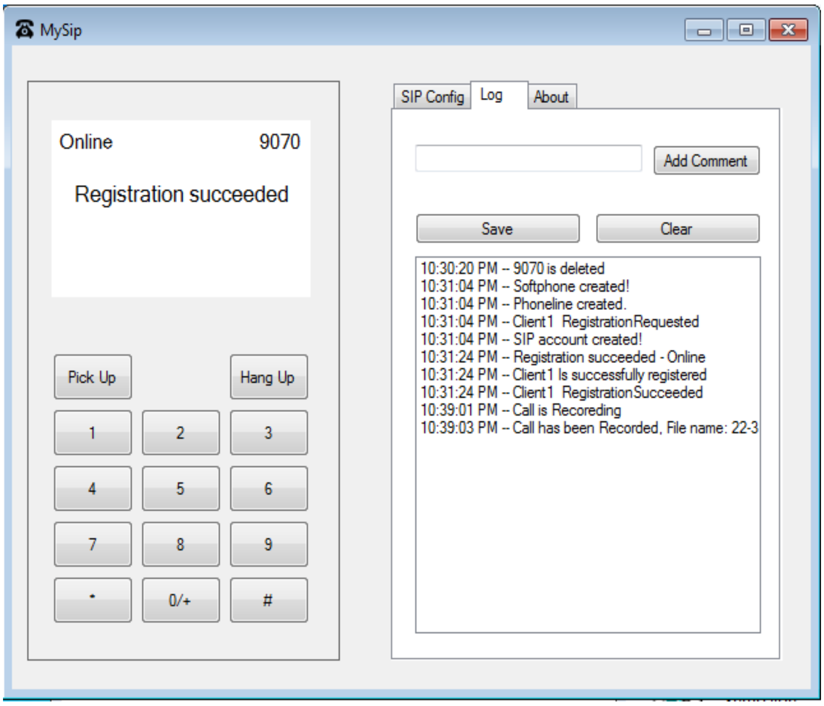
```
Registering: Client1@10.10.0.1
Client1  RegistrationRequested
Client1  Is successfully registered
Client1  RegistrationSucceeded
```

Microphone: Microphone (High Definition Aud

100.00% peak [count: 100]

Figure 30: MySIP Register User

Now making a call to any other client will be possible. Also recording the call by clicking the *Record* button, Recoded File (.wav) will be added to the program directory.



Online 9070

Registration succeeded

```
10:30:20 PM -- 9070 is deleted
10:31:04 PM -- Softphone created!
10:31:04 PM -- Phoneline created.
10:31:04 PM -- Client1  RegistrationRequested
10:31:04 PM -- SIP account created!
10:31:24 PM -- Registration succeeded - Online
10:31:24 PM -- Client1 Is successfully registered
10:31:24 PM -- Client1  RegistrationSucceeded
10:39:01 PM -- Call is Recoreding
10:39:03 PM -- Call has been Recorded, File name: 22-3
```

Figure 31: MySIP Logs

In the Log tab, all activities will be logged. Including adding, deleting, connecting, disconnecting and recording activities. Also adding a Comment to the logfile is possible. This log file can be saved or deleted.

2. Results and analysis

To find the answers to the questions mentioned in section 1, it is necessary to analyze the research data obtained through experiments. This section includes the results of the experiment performed in this research along with a discussion on the prediction of the performance of the Tor network. Lastly, the probability of attackers on the network is also discussed in this section.

2.1 Data Analysis Procedures

The results of the experiment obtained through the above procedure were then analyzed to find out the answers to the research questions. To find out Quality of Service performance in VOIP over the Onion Routing network, the recordings from Wireshark were utilized. Packet loss, latency, and jitter were obtained in this way. Packet loss was calculated from the number of packets which exceeded 400 ms latency. Latency was calculated from the difference of time between the receiving of the call at the receiver and the sender. Moreover, the difference in latency for each voice packet sent gave the average jitter.

Also, during testing, another tool has been used. The StarTrinity SIP Tester (Aleshin, n.d.), it tests the Load and monitors VoIP network. It can also simulate many incoming and outcoming calls with RTP media.

For test there 3 PCs were used, each PC was connected to a different internet provider:

- Server (VPN IP address: 10.10.0.1) connection LIWEST provider Download 200Mbps/Upload 20Mbps.
- PC1 (VPN IP address: 10.10.0.18) connection 4G A1 hotspot (Download 80Mbps/Upload 50Mbps).
- PC2(VPN IP address: 10.10.0.6) connection 4G bob hotspot (Download 100Mbps/Upload 20Mbps).

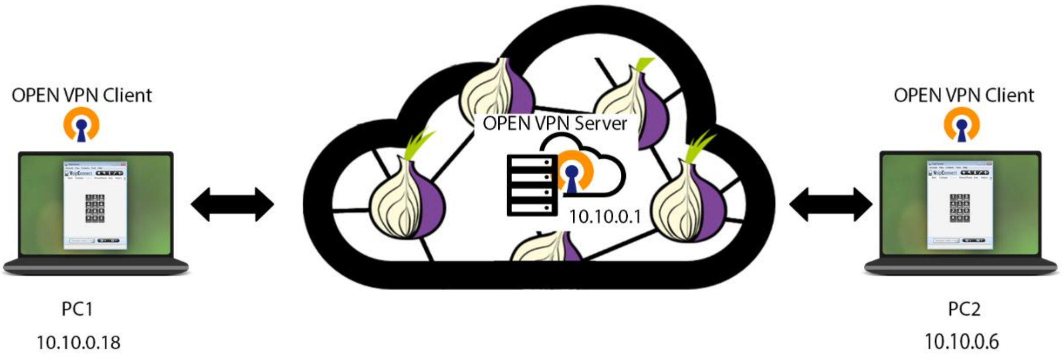


Figure 32: Test structure

2.2 VoIP over Tor

As discussed in the previous section, during the experimentation VoIP calls were made through the Onion Routing network in two different situations. The first situation was the RTP Stream between the callees is redirected through the Elastix server. And the second situation was RTP Stream pass directly between callees, RTP packets will not pass through the Elastix Server. Research data were obtained in many different time intervals. Readings were taken on all different times.

2.3 Non-Direct RTP streams

The aim of the investigation was to calculate Quality of Service performance in the Tor network. By default, Elastix installation will set both endpoints phones to pass their media streams (RTP streams) through the Elastix server itself.SIP packets should pass through the server to initiate the call, the RTP stream would look something like this:

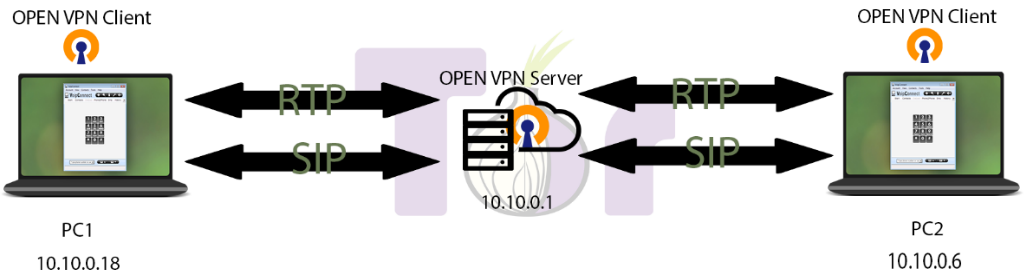
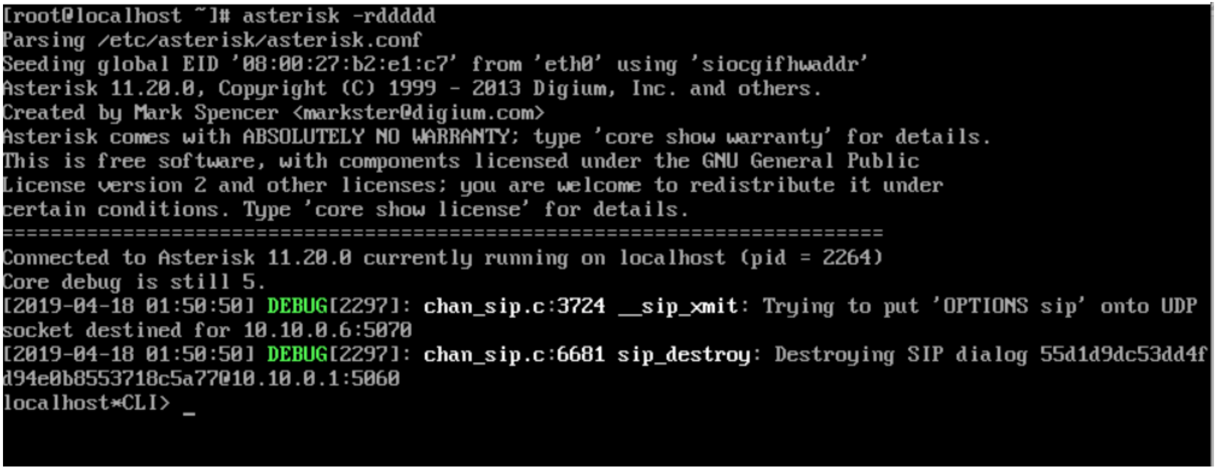


Figure 33:Non-Direct RTP

In the Elastix Server, two clients have been added. Those accounts do not support Direct RTP. It means all traffic should pass through the server. Both clients have connected to the Tor network and the OpenVPN too. Elastix in the debug mode, and it shows all the RTP packets passing through the server.

The Server receives the Packets from Client1 (9070) and IP: 10.10.0.18 and send them to Client2 (9071) and IP: 10.10.0.6.

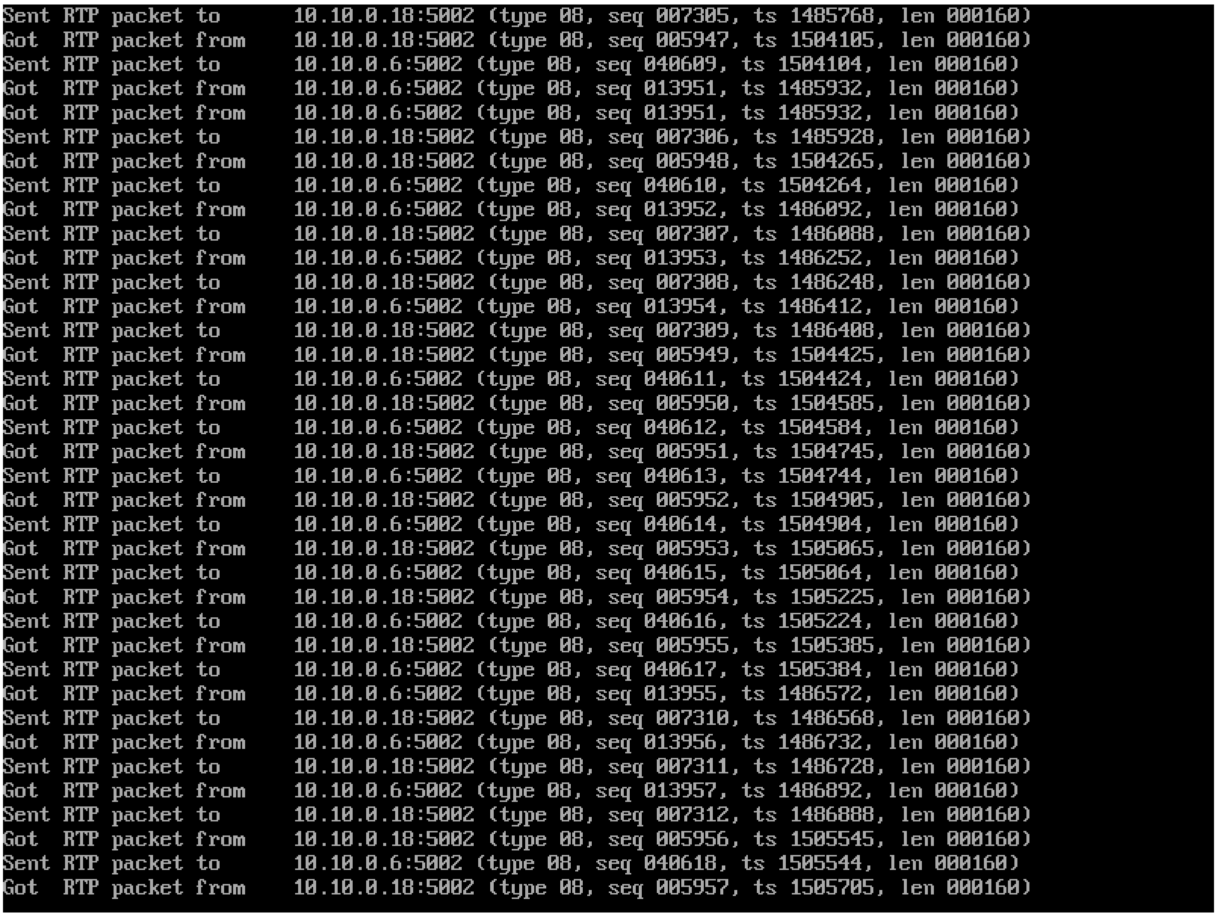
Starting Asterisk in debug mode:



```
[root@localhost ~]# asterisk -rddddd
Parsing /etc/asterisk/asterisk.conf
Seeding global EID '08:00:27:b2:e1:c7' from 'eth0' using 'siocgifhwaddr'
Asterisk 11.20.0, Copyright (C) 1999 - 2013 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=========================================================================
Connected to Asterisk 11.20.0 currently running on localhost (pid = 2264)
Core debug is still 5.
[2019-04-18 01:50:50] DEBUG[2297]: chan_sip.c:3724 __sip_xmit: Trying to put 'OPTIONS sip' onto UDP
socket destined for 10.10.0.6:5070
[2019-04-18 01:50:50] DEBUG[2297]: chan_sip.c:6681 sip_destroy: Destroying SIP dialog 55d1d9dc53dd4f
d94e0b8553718c5a77@10.10.0.1:5060
localhost*CLI> _
```

Figure 34:Asterisk Debug mode

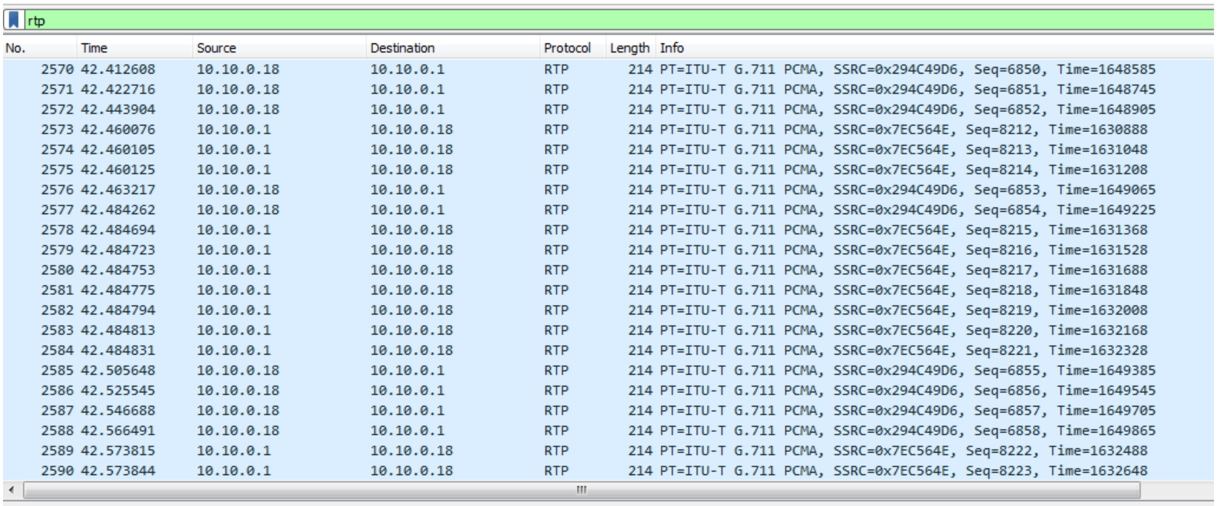
Whenever that call is made the RTP packet will pass all through the Server. In the RTP debug mode on the Server the packets are traceable. The following figure shows the RTP packets sent from both clients.



```
Sent RTP packet to     10.10.0.18:5002 (type 08, seq 007305, ts 1485768, len 000160)
Got  RTP packet from   10.10.0.18:5002 (type 08, seq 005947, ts 1504105, len 000160)
Sent RTP packet to     10.10.0.6:5002 (type 08, seq 040609, ts 1504104, len 000160)
Got  RTP packet from   10.10.0.6:5002 (type 08, seq 013951, ts 1485932, len 000160)
Got  RTP packet from   10.10.0.6:5002 (type 08, seq 013951, ts 1485932, len 000160)
Sent RTP packet to     10.10.0.18:5002 (type 08, seq 007306, ts 1485928, len 000160)
Got  RTP packet from   10.10.0.18:5002 (type 08, seq 005948, ts 1504265, len 000160)
Sent RTP packet to     10.10.0.6:5002 (type 08, seq 040610, ts 1504264, len 000160)
Got  RTP packet from   10.10.0.6:5002 (type 08, seq 013952, ts 1486092, len 000160)
Sent RTP packet to     10.10.0.18:5002 (type 08, seq 007307, ts 1486088, len 000160)
Got  RTP packet from   10.10.0.6:5002 (type 08, seq 013953, ts 1486252, len 000160)
Sent RTP packet to     10.10.0.18:5002 (type 08, seq 007308, ts 1486248, len 000160)
Got  RTP packet from   10.10.0.6:5002 (type 08, seq 013954, ts 1486412, len 000160)
Sent RTP packet to     10.10.0.18:5002 (type 08, seq 007309, ts 1486408, len 000160)
Got  RTP packet from   10.10.0.18:5002 (type 08, seq 005949, ts 1504425, len 000160)
Sent RTP packet to     10.10.0.6:5002 (type 08, seq 040611, ts 1504424, len 000160)
Got  RTP packet from   10.10.0.18:5002 (type 08, seq 005950, ts 1504585, len 000160)
Sent RTP packet to     10.10.0.6:5002 (type 08, seq 040612, ts 1504584, len 000160)
Got  RTP packet from   10.10.0.18:5002 (type 08, seq 005951, ts 1504745, len 000160)
Sent RTP packet to     10.10.0.6:5002 (type 08, seq 040613, ts 1504744, len 000160)
Got  RTP packet from   10.10.0.18:5002 (type 08, seq 005952, ts 1504905, len 000160)
Sent RTP packet to     10.10.0.6:5002 (type 08, seq 040614, ts 1504904, len 000160)
Got  RTP packet from   10.10.0.18:5002 (type 08, seq 005953, ts 1505065, len 000160)
Sent RTP packet to     10.10.0.6:5002 (type 08, seq 040615, ts 1505064, len 000160)
Got  RTP packet from   10.10.0.18:5002 (type 08, seq 005954, ts 1505225, len 000160)
Sent RTP packet to     10.10.0.6:5002 (type 08, seq 040616, ts 1505224, len 000160)
Got  RTP packet from   10.10.0.18:5002 (type 08, seq 005955, ts 1505385, len 000160)
Sent RTP packet to     10.10.0.6:5002 (type 08, seq 040617, ts 1505384, len 000160)
Got  RTP packet from   10.10.0.6:5002 (type 08, seq 013955, ts 1486572, len 000160)
Sent RTP packet to     10.10.0.18:5002 (type 08, seq 007310, ts 1486568, len 000160)
Got  RTP packet from   10.10.0.6:5002 (type 08, seq 013956, ts 1486732, len 000160)
Sent RTP packet to     10.10.0.18:5002 (type 08, seq 007311, ts 1486728, len 000160)
Got  RTP packet from   10.10.0.6:5002 (type 08, seq 013957, ts 1486892, len 000160)
Sent RTP packet to     10.10.0.18:5002 (type 08, seq 007312, ts 1486888, len 000160)
Got  RTP packet from   10.10.0.18:5002 (type 08, seq 005956, ts 1505545, len 000160)
Sent RTP packet to     10.10.0.6:5002 (type 08, seq 040618, ts 1505544, len 000160)
Got  RTP packet from   10.10.0.18:5002 (type 08, seq 005957, ts 1505705, len 000160)
```

Figure 35:RTP packets between callee

Also, in Wireshark on the endpoint, all RTP Packets are sent and receive between the endpoint and server.



rtp

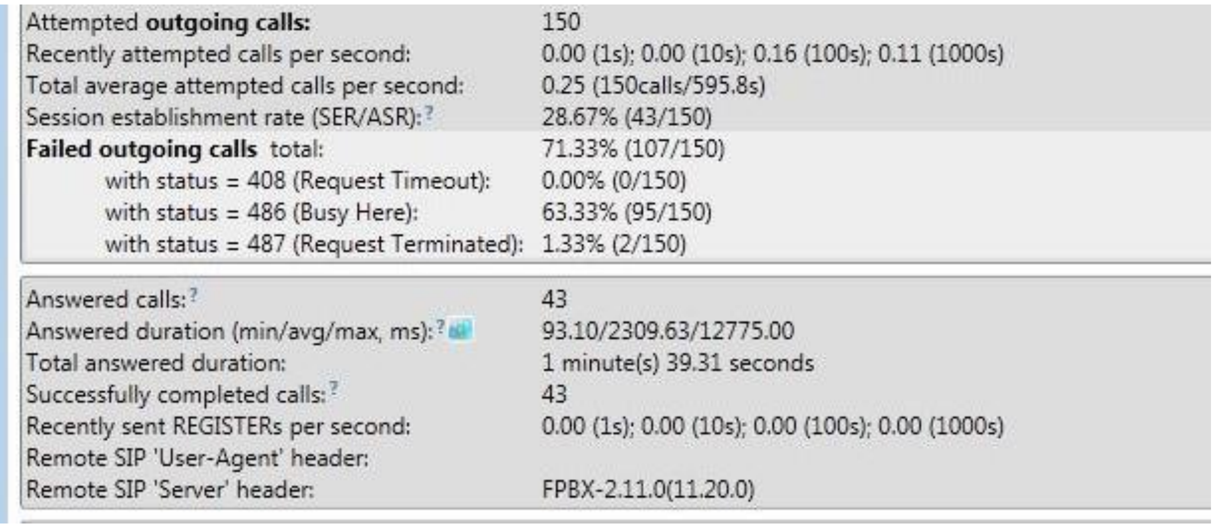
No.	Time	Source	Destination	Protocol	Length	Info
2570	42.412608	10.10.0.18	10.10.0.1	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x294C49D6, Seq=6850, Time=1648585
2571	42.422716	10.10.0.18	10.10.0.1	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x294C49D6, Seq=6851, Time=1648745
2572	42.443904	10.10.0.18	10.10.0.1	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x294C49D6, Seq=6852, Time=1648905
2573	42.460076	10.10.0.1	10.10.0.18	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x7EC564E, Seq=8212, Time=1630888
2574	42.460105	10.10.0.1	10.10.0.18	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x7EC564E, Seq=8213, Time=1631048
2575	42.460125	10.10.0.1	10.10.0.18	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x7EC564E, Seq=8214, Time=1631208
2576	42.463217	10.10.0.18	10.10.0.1	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x294C49D6, Seq=6853, Time=1649065
2577	42.484262	10.10.0.18	10.10.0.1	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x294C49D6, Seq=6854, Time=1649225
2578	42.484694	10.10.0.1	10.10.0.18	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x7EC564E, Seq=8215, Time=1631368
2579	42.484723	10.10.0.1	10.10.0.18	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x7EC564E, Seq=8216, Time=1631528
2580	42.484753	10.10.0.1	10.10.0.18	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x7EC564E, Seq=8217, Time=1631688
2581	42.484775	10.10.0.1	10.10.0.18	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x7EC564E, Seq=8218, Time=1631848
2582	42.484794	10.10.0.1	10.10.0.18	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x7EC564E, Seq=8219, Time=1632008
2583	42.484813	10.10.0.1	10.10.0.18	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x7EC564E, Seq=8220, Time=1632168
2584	42.484831	10.10.0.1	10.10.0.18	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x7EC564E, Seq=8221, Time=1632328
2585	42.505648	10.10.0.18	10.10.0.1	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x294C49D6, Seq=6855, Time=1649385
2586	42.525545	10.10.0.18	10.10.0.1	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x294C49D6, Seq=6856, Time=1649545
2587	42.546688	10.10.0.18	10.10.0.1	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x294C49D6, Seq=6857, Time=1649705
2588	42.566491	10.10.0.18	10.10.0.1	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x294C49D6, Seq=6858, Time=1649865
2589	42.573815	10.10.0.1	10.10.0.18	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x7EC564E, Seq=8222, Time=1632488
2590	42.573844	10.10.0.1	10.10.0.18	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x7EC564E, Seq=8223, Time=1632648

Figure 36: Wireshark Non-direct RTP

For Analysis StarTrinity has been configured. On both sides, StarTrinity gives the possibility to configure a number of call attempts with an interval between those calls, on the other side StarTrinity accepts all the incoming calls and with a call duration after answering option.

In the testcase 150 attempts have been made, each with an interval of 4000 ms and on the receiver side the incoming call duration after the answer was 500 ms.

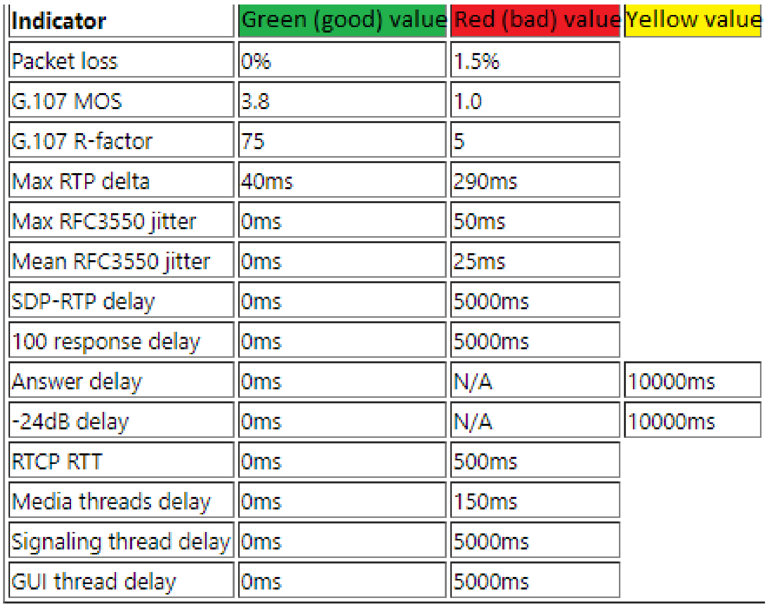
From 150 attempts only 43 have been answered, this caused by the TOR network delay and Elastix Bulletproof VoIP Security, Elastix comes out of the box with built in bulletproof features. Elastix is protected against calls flooding.



Attempted **outgoing calls:**	150
Recently attempted calls per second:	0.00 (1s); 0.00 (10s); 0.16 (100s); 0.11 (1000s)
Total average attempted calls per second:	0.25 (150calls/595.8s)
Session establishment rate (SER/ASR): ?	28.67% (43/150)
Failed outgoing calls total:	71.33% (107/150)
with status = 408 (Request Timeout):	0.00% (0/150)
with status = 486 (Busy Here):	63.33% (95/150)
with status = 487 (Request Terminated):	1.33% (2/150)
Answered calls: ?	43
Answered duration (min/avg/max, ms): ?	93.10/2309.63/12775.00
Total answered duration:	1 minute(s) 39.31 seconds
Successfully completed calls: ?	43
Recently sent REGISTERs per second:	0.00 (1s); 0.00 (10s); 0.00 (100s); 0.00 (1000s)
Remote SIP 'User-Agent' header:	
Remote SIP 'Server' header:	FPBX-2.11.0(11.20.0)

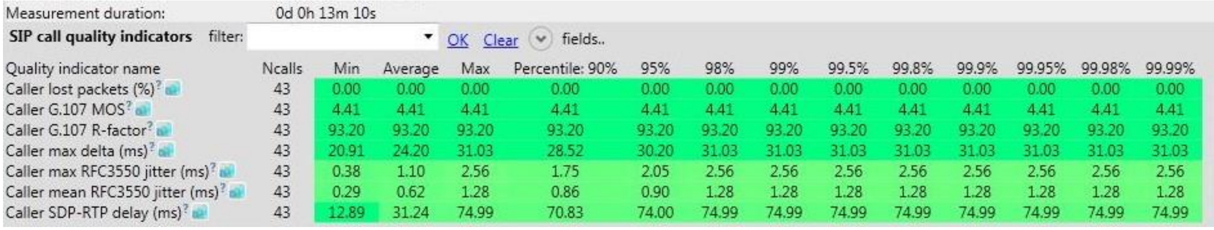
Figure 37: Non-direct RTP StarTrinity

According to Startrinity measured indicators will have different colors in the report. Each color represents the status of the measured value. Green means good and Red not good. The following table shows the good and bad value for each measurement.



Indicator	Green (good) value	Red (bad) value	Yellow value
Packet loss	0%	1.5%	
G.107 MOS	3.8	1.0	
G.107 R-factor	75	5	
Max RTP delta	40ms	290ms	
Max RFC3550 jitter	0ms	50ms	
Mean RFC3550 jitter	0ms	25ms	
SDP-RTP delay	0ms	5000ms	
100 response delay	0ms	5000ms	
Answer delay	0ms	N/A	10000ms
-24dB delay	0ms	N/A	10000ms
RTCP RTT	0ms	500ms	
Media threads delay	0ms	150ms	
Signaling thread delay	0ms	5000ms	
GUI thread delay	0ms	5000ms	

Figure 38: Colors in Startrinity Reports/Statistics *(StarTrinity)*



Measurement duration: 0d 0h 13m 10s

SIP call quality indicators filter: OK Clear fields..

Quality indicator name	Ncalls	Min	Average	Max	Percentile: 90%	95%	98%	99%	99.5%	99.8%	99.9%	99.95%	99.98%	99.99%
Caller lost packets (%)?	43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Caller G.107 MOS?	43	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41
Caller G.107 R-factor?	43	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20
Caller max delta (ms)?	43	20.91	24.20	31.03	28.52	30.20	31.03	31.03	31.03	31.03	31.03	31.03	31.03	31.03
Caller max RFC3550 jitter (ms)?	43	0.38	1.10	2.56	1.75	2.05	2.56	2.56	2.56	2.56	2.56	2.56	2.56	2.56
Caller mean RFC3550 jitter (ms)?	43	0.29	0.62	1.28	0.86	0.90	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28
Caller SDP-RTP delay (ms)?	43	12.89	31.24	74.99	70.83	74.00	74.99	74.99	74.99	74.99	74.99	74.99	74.99	74.99

Figure 39: Non-direct Caller StarTrinity results

In the caller analysis and during the test Packet lost, jitter buffer through caller listening satisfaction (G.107 MOS), max delta, RFC33550 jitter, RTP delay.

1. Packet lost: gives the level of lost RTP packets, larger values more than 3% usually indicate overloads in the IP network. This packet loss value has a direct effect on audio quality. In the 43 calls that have been answered 99.9% percent have very good quality audio and the packet lost was almost zero. This is a very good result.

2. G.107 E-model means opinion score (MOS) used to measure specified jitter buffer settings. According to the G.107 satisfaction level:

- 4.3-5.0: very satisfied
- 4.0-4.3: satisfied
- 3.6-4.0: some users satisfied
- 3.1-3.6: many users dissatisfied
- 2.6-3.1: nearly all users dissatisfied
- 1.0-2.6: not recommended

In the test, more than 99.9% percent of the calls have 4.41 which a very satisfying level.

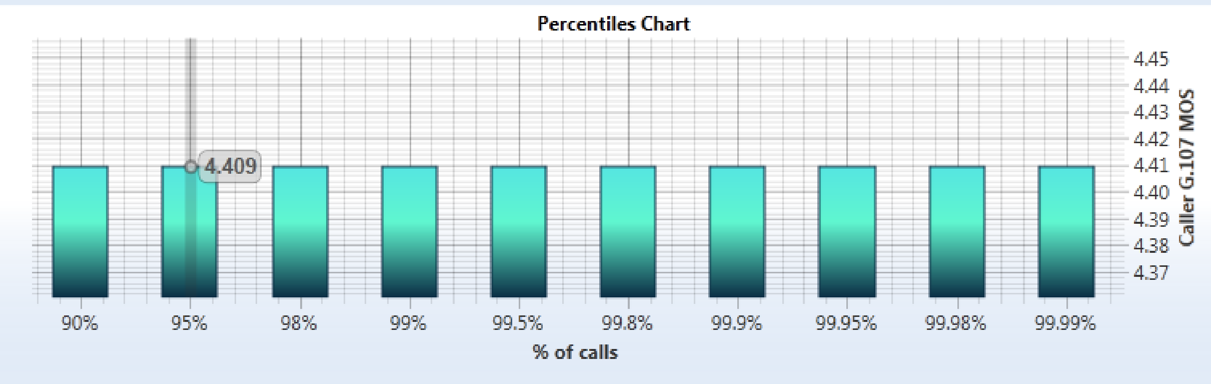
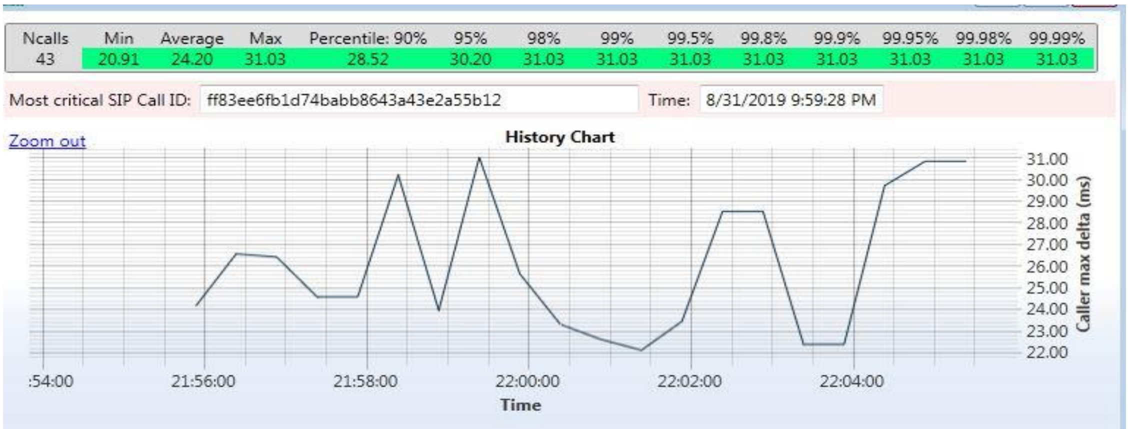


Figure 40: Non-direct Caller G.107 percentiles chart

3. Caller max delta indicates the maximum time between consecutive packets of RTP. which gives information about how unstable the delays in the flow of the media.



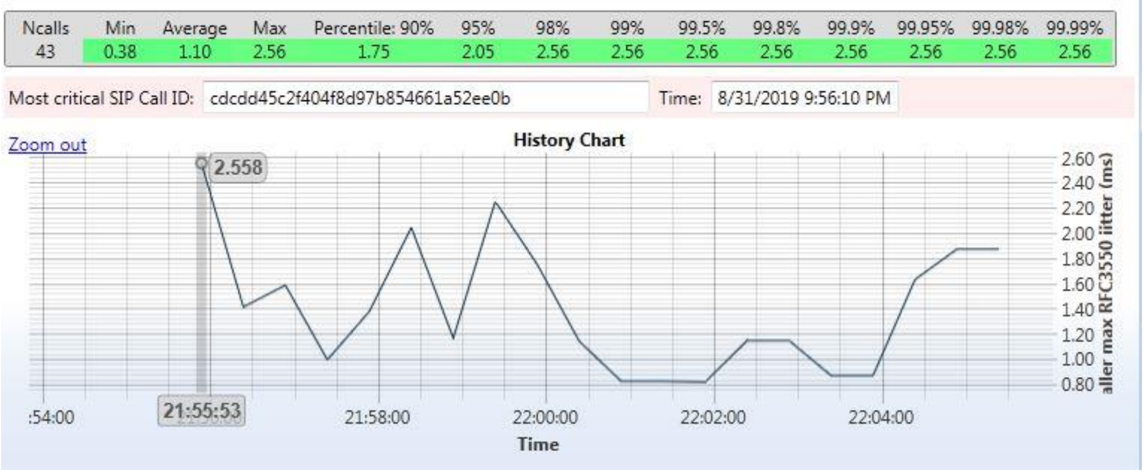
Ncalls	Min	Average	Max	Percentile: 90%	95%	98%	99%	99.5%	99.8%	99.9%	99.95%	99.98%	99.99%
43	20.91	24.20	31.03	28.52	30.20	31.03	31.03	31.03	31.03	31.03	31.03	31.03	31.03

Most critical SIP Call ID: ff83ee6fb1d74babb8643a43e2a55b12 Time: 8/31/2019 9:59:28 PM

Figure 41: Non-direct Caller max delta history chart

Over 150ms means overloads in IP network CPU of server. The RTP delta value effects on audio quality. In the test, the average was around 25ms of calls have achieved.

4. RFC33550 jitter maximum value of RTP stream jitter per call, according to RFC3550 standard over 50ms could mean overloads in the IP network. Also, this value has a direct impact quality of the call and the audio. In the test, max jitter has shown very good results the maximum was 2.5ms which is a very good indicator.



Ncalls	Min	Average	Max	Percentile: 90%	95%	98%	99%	99.5%	99.8%	99.9%	99.95%	99.98%	99.99%
43	0.38	1.10	2.56	1.75	2.05	2.56	2.56	2.56	2.56	2.56	2.56	2.56	2.56

Most critical SIP Call ID: cdcdd45c2f404f8d97b854661a52ee0b Time: 8/31/2019 9:56:10 PM

Figure 42: Non-direct Caller jitter percentiles chart

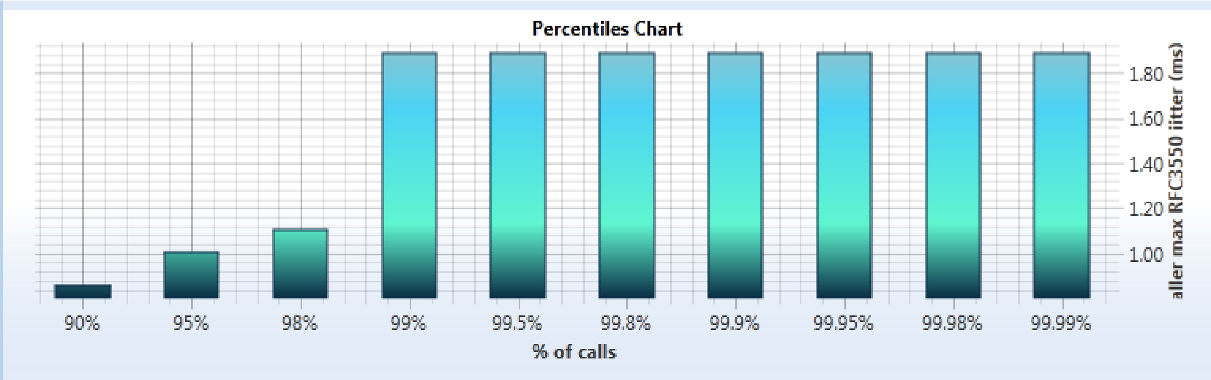
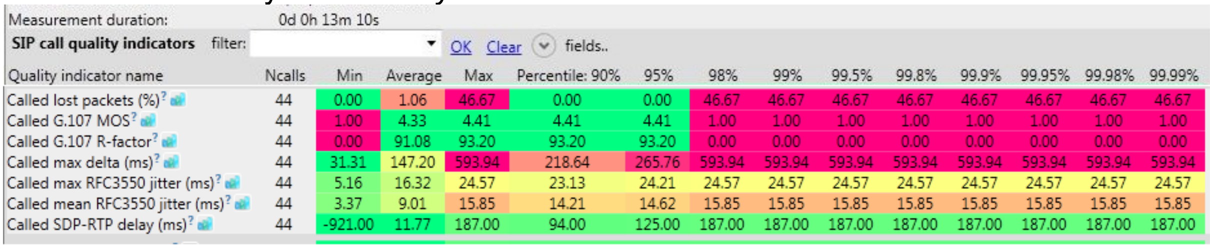


Figure 43: Non-direct Caller jitter percentiles chart

5. Caller SDP-RTP delay: represent the delay between SPD response (183: Session in progress, 200: OK response) and the first RTP packet. The chart shows 12,89 ms, and according the Startrinity measured indicator table the value is acceptable.

On the other hand, the call destination showed acceptable results beside some problems with Packet lost, max delta and jitter. Such problems could be caused by Bulletproof feature on Elastix server and the Tor network delay. Although and during the 43 calls on the caller side, but 44 calls showed on the recipient side. There is extra call was received but only in the recipient side. The ACK for this answer seems to be lost on the way. That’s why the Caller side could not count this call.



Measurement duration: 0d 0h 13m 10s

SIP call quality indicators filter:

Quality indicator name	Ncalls	Min	Average	Max	Percentile: 90%	95%	98%	99%	99.5%	99.8%	99.9%	99.95%	99.98%	99.99%
Called lost packets (%)?	44	0.00	1.06	46.67	0.00	0.00	46.67	46.67	46.67	46.67	46.67	46.67	46.67	46.67
Called G.107 MOS?	44	1.00	4.33	4.41	4.41	4.41	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Called G.107 R-factor?	44	0.00	91.08	93.20	93.20	93.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Called max delta (ms)?	44	31.31	147.20	593.94	218.64	265.76	593.94	593.94	593.94	593.94	593.94	593.94	593.94	593.94
Called max RFC3550 jitter (ms)?	44	5.16	16.32	24.57	23.13	24.21	24.57	24.57	24.57	24.57	24.57	24.57	24.57	24.57
Called mean RFC3550 jitter (ms)?	44	3.37	9.01	15.85	14.21	14.62	15.85	15.85	15.85	15.85	15.85	15.85	15.85	15.85
Called SDP-RTP delay (ms)?	44	-921.00	11.77	187.00	94.00	125.00	187.00	187.00	187.00	187.00	187.00	187.00	187.00	187.00

Figure 44: Called destination StarTrinity results

Hight Packet lost at the max on the recipient, more than 46% in some calls which indicate overloads in the network and affect directly the audio quality. Although the average value of the packet lost is 1.06%, this is still acceptable as long as it is under 3% according to StarTrinity documentation (StarTrinity).

But max RFC3550 jitter and max delta affected only 10% of the calls. In the following charts, both results started after 90% of the calls to happen.

Called SDP-RTP delay showed a negative value -921ms, this is normal in case when RTP packets are detected before SDP negotiation packet, which is 183 session progress or 200 ok.

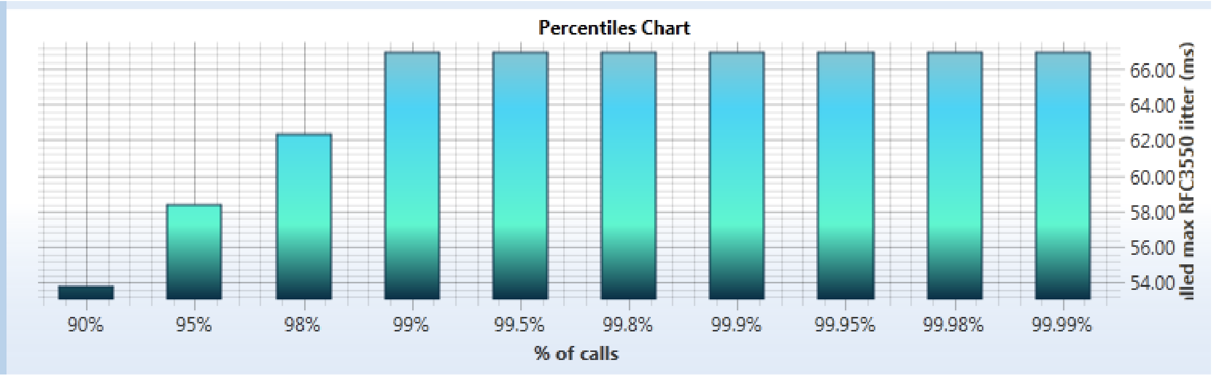


Figure 45:Non-direct destination jitter percentiles chart

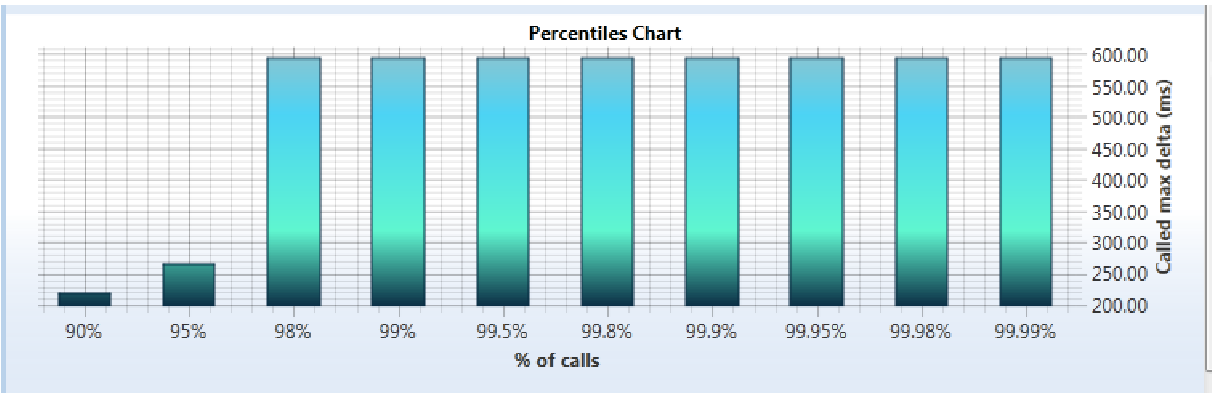


Figure 46: Non-direct destination max delta percentiles chart

According to the results obtained through the experiment, Non-direct connection over Tor networks showed acceptable results which indicate the making VoIP calls over Tor network using OpenVPN is possible but sometimes calls have an audio quality issue due to high Packet lost in some calls.

2.4 Direct RTP Streams

Direct RTP Streams In non-NAT situations, it is desirable over have the RTP streams pass directly between phones. SIP control messages will in any away pass to/from the Asterisk server. However, RTP streams will pass straightforwardly between phones.

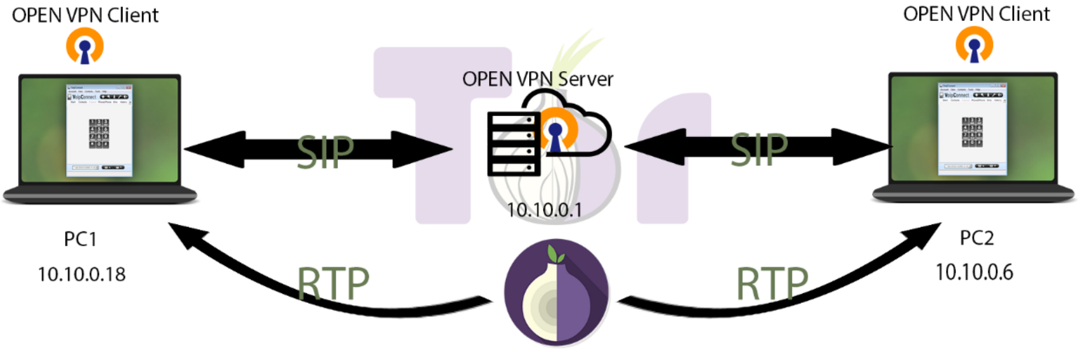
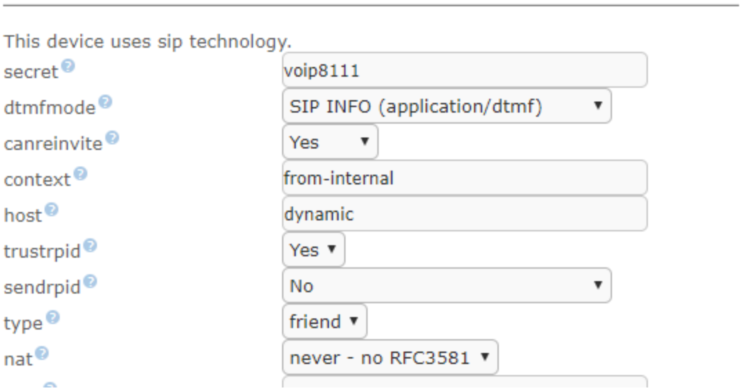


Figure 47:Direct RTP

To achieve a Direct RTP Stream between callees, Elastix caller configuration must be changed. This change should be done in two parts. First, to enable *canreinvite=yes** in sip.confg. And the second part, Disable NAT for each SIP alone and dtmfmode must change: rfc2833 usually for INFO. When "canreinvite=no", everything is sent always via Asterisk (Elastix).



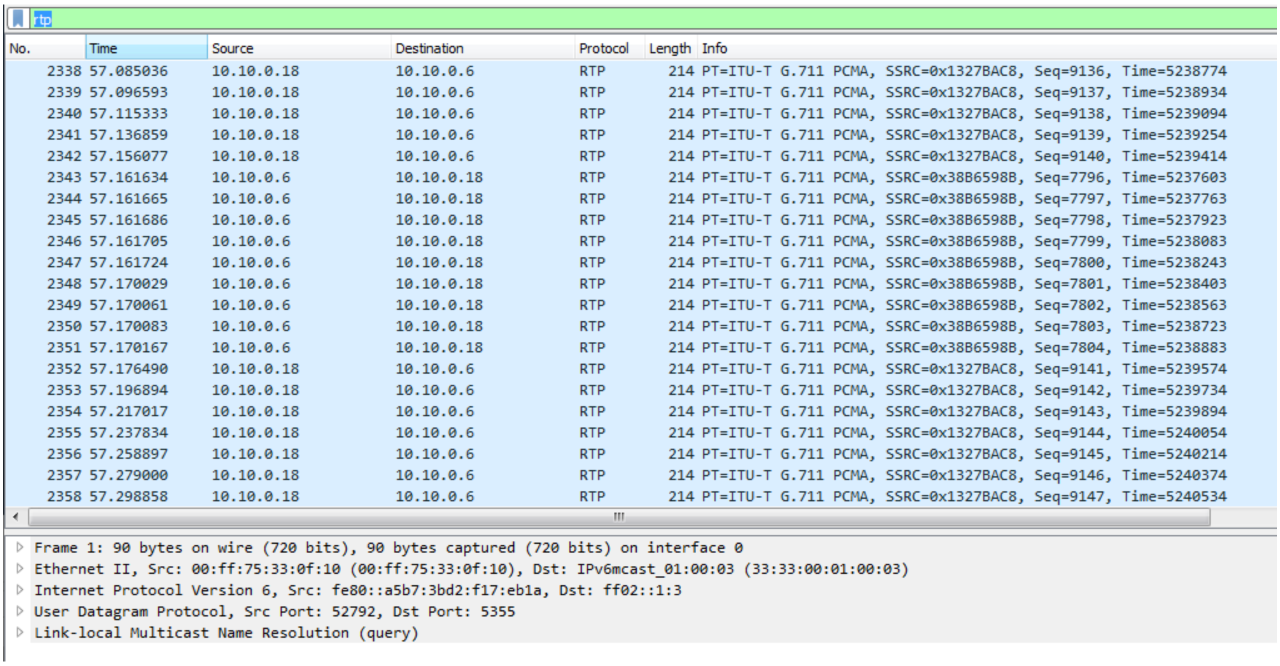
This device uses sip technology.

secret	voip8111
dtmfmode	SIP INFO (application/dtmf)
canreinvite	Yes
context	from-internal
host	dynamic
trustrpid	Yes
sendrpid	No
type	friend
nat	never - no RFC3581

Figure 48: User Direct RTP Server Configuration

**Note: canreinvite= was renamed to directmedia= in Asterisk 1.6.2 to more accurately describe what this setting does. But Elastix 4 runs Asterisk 11.20 older version so canreinvite was used.*

In the Elastix server, two clients have been added. Those accounts support Direct RTP. It means all RTP traffic should pass directly between clients. Both clients have connected to the Tor network and the OpenVPN too. First Client1 (8111) and IP: 10.10.18 and send them to Client2 (8112) and IP: 10.10.0.6.
Wireshark on the endpoint all RTP Packets are sent and receive between the endpoint and another endpoint.

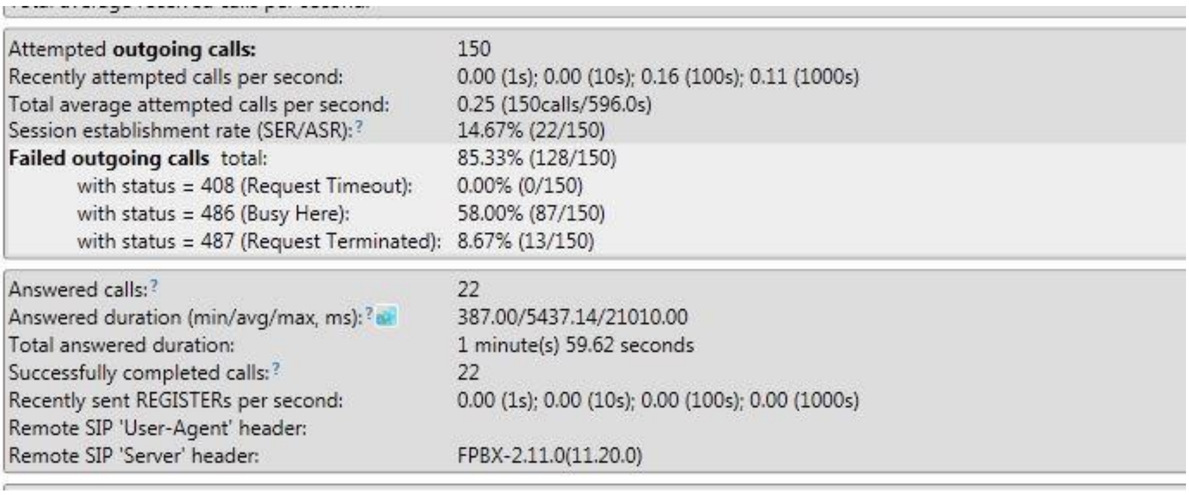


```
No.   Time      Source      Destination  Protocol Length Info
 2338 57.085036 10.10.0.18  10.10.0.6    RTP      214 PT=ITU-T G.711 PCMA, SSRC=0x1327BAC8, Seq=9136, Time=5238774
 2339 57.096593 10.10.0.18  10.10.0.6    RTP      214 PT=ITU-T G.711 PCMA, SSRC=0x1327BAC8, Seq=9137, Time=5238934
 2340 57.115333 10.10.0.18  10.10.0.6    RTP      214 PT=ITU-T G.711 PCMA, SSRC=0x1327BAC8, Seq=9138, Time=5239094
 2341 57.136859 10.10.0.18  10.10.0.6    RTP      214 PT=ITU-T G.711 PCMA, SSRC=0x1327BAC8, Seq=9139, Time=5239254
 2342 57.156077 10.10.0.18  10.10.0.6    RTP      214 PT=ITU-T G.711 PCMA, SSRC=0x1327BAC8, Seq=9140, Time=5239414
 2343 57.161634 10.10.0.6   10.10.0.18   RTP      214 PT=ITU-T G.711 PCMA, SSRC=0x38B6598B, Seq=7796, Time=5237603
 2344 57.161665 10.10.0.6   10.10.0.18   RTP      214 PT=ITU-T G.711 PCMA, SSRC=0x38B6598B, Seq=7797, Time=5237763
 2345 57.161686 10.10.0.6   10.10.0.18   RTP      214 PT=ITU-T G.711 PCMA, SSRC=0x38B6598B, Seq=7798, Time=5237923
 2346 57.161705 10.10.0.6   10.10.0.18   RTP      214 PT=ITU-T G.711 PCMA, SSRC=0x38B6598B, Seq=7799, Time=5238083
 2347 57.161724 10.10.0.6   10.10.0.18   RTP      214 PT=ITU-T G.711 PCMA, SSRC=0x38B6598B, Seq=7800, Time=5238243
 2348 57.170029 10.10.0.6   10.10.0.18   RTP      214 PT=ITU-T G.711 PCMA, SSRC=0x38B6598B, Seq=7801, Time=5238403
 2349 57.170061 10.10.0.6   10.10.0.18   RTP      214 PT=ITU-T G.711 PCMA, SSRC=0x38B6598B, Seq=7802, Time=5238563
 2350 57.170083 10.10.0.6   10.10.0.18   RTP      214 PT=ITU-T G.711 PCMA, SSRC=0x38B6598B, Seq=7803, Time=5238723
 2351 57.170167 10.10.0.6   10.10.0.18   RTP      214 PT=ITU-T G.711 PCMA, SSRC=0x38B6598B, Seq=7804, Time=5238883
 2352 57.176490 10.10.0.18  10.10.0.6    RTP      214 PT=ITU-T G.711 PCMA, SSRC=0x1327BAC8, Seq=9141, Time=5239574
 2353 57.196894 10.10.0.18  10.10.0.6    RTP      214 PT=ITU-T G.711 PCMA, SSRC=0x1327BAC8, Seq=9142, Time=5239734
 2354 57.217017 10.10.0.18  10.10.0.6    RTP      214 PT=ITU-T G.711 PCMA, SSRC=0x1327BAC8, Seq=9143, Time=5239894
 2355 57.237834 10.10.0.18  10.10.0.6    RTP      214 PT=ITU-T G.711 PCMA, SSRC=0x1327BAC8, Seq=9144, Time=5240054
 2356 57.258897 10.10.0.18  10.10.0.6    RTP      214 PT=ITU-T G.711 PCMA, SSRC=0x1327BAC8, Seq=9145, Time=5240214
 2357 57.279000 10.10.0.18  10.10.0.6    RTP      214 PT=ITU-T G.711 PCMA, SSRC=0x1327BAC8, Seq=9146, Time=5240374
 2358 57.298858 10.10.0.18  10.10.0.6    RTP      214 PT=ITU-T G.711 PCMA, SSRC=0x1327BAC8, Seq=9147, Time=5240534

▷ Frame 1: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface 0
▷ Ethernet II, Src: 00:ff:75:33:0f:10 (00:ff:75:33:0f:10), Dst: IPv6mcast_01:00:03 (33:33:00:01:00:03)
▷ Internet Protocol Version 6, Src: fe80::a5b7:3bd2:f17:eb1a, Dst: ff02::1:3
▷ User Datagram Protocol, Src Port: 52792, Dst Port: 5355
▷ Link-local Multicast Name Resolution (query)
```

Figure 49: Wireshark Direct RTP

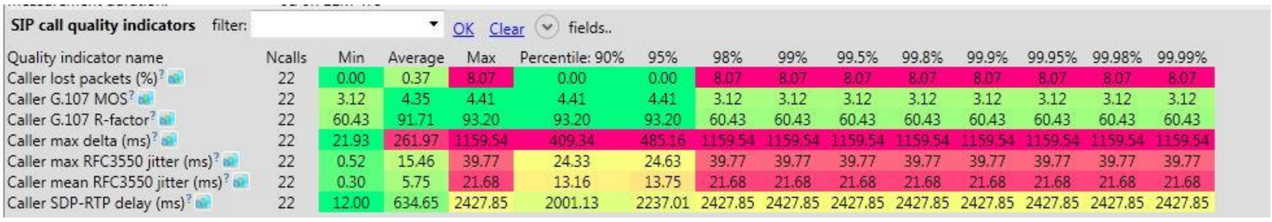
For Analysis StarTrinity also has been used and with the same configurations. The testcase 150 attempts have been made, each with an interval of 4000ms and on the receiver side the incoming call duration after the answer was 500ms.
The answered calls number was identical in both test cases.
From 150 attempts only 22 have been answered. This also can be relayed to the same reasons network delay and Elastix Bulletproof VoIP Security.



Attempted **outgoing calls:**	150
Recently attempted calls per second:	0.00 (1s); 0.00 (10s); 0.16 (100s); 0.11 (1000s)
Total average attempted calls per second:	0.25 (150calls/596.0s)
Session establishment rate (SER/ASR):?	14.67% (22/150)
Failed outgoing calls total:	85.33% (128/150)
with status = 408 (Request Timeout):	0.00% (0/150)
with status = 486 (Busy Here):	58.00% (87/150)
with status = 487 (Request Terminated):	8.67% (13/150)
Answered calls:?	22
Answered duration (min/avg/max, ms):?	387.00/5437.14/21010.00
Total answered duration:	1 minute(s) 59.62 seconds
Successfully completed calls:?	22
Recently sent REGISTERs per second:	0.00 (1s); 0.00 (10s); 0.00 (100s); 0.00 (1000s)
Remote SIP 'User-Agent' header:	
Remote SIP 'Server' header:	FPBX-2.11.0(11.20.0)

Figure 50: Direct RTP StarTrinity

Startrinity gave the following results:

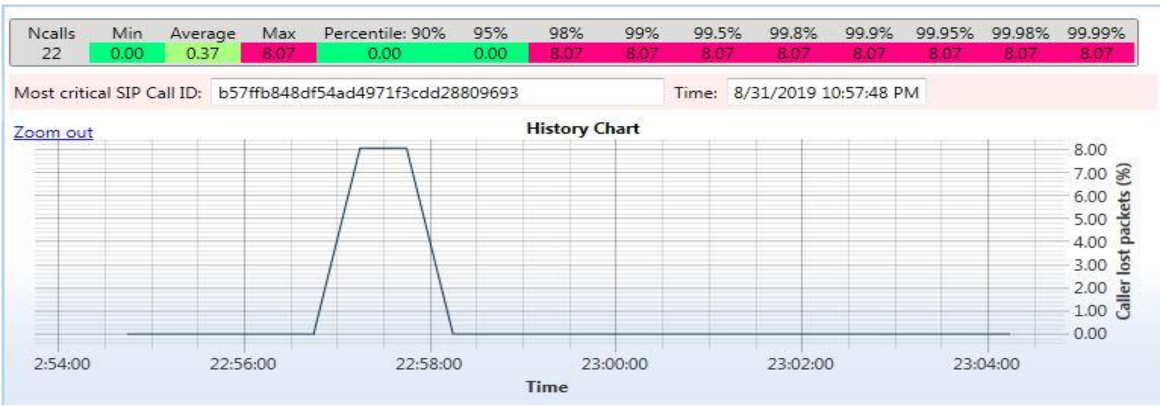


SIP call quality indicators filter: OK Clear fields..

Quality indicator name	Ncalls	Min	Average	Max	Percentile: 90%	95%	98%	99%	99.5%	99.8%	99.9%	99.95%	99.98%	99.99%
Caller lost packets (%)?	22	0.00	0.37	8.07	0.00	0.00	8.07	8.07	8.07	8.07	8.07	8.07	8.07	8.07
Caller G.107 MOS?	22	3.12	4.35	4.41	4.41	4.41	3.12	3.12	3.12	3.12	3.12	3.12	3.12	3.12
Caller G.107 R-factor?	22	60.43	91.71	93.20	93.20	93.20	60.43	60.43	60.43	60.43	60.43	60.43	60.43	60.43
Caller max delta (ms)?	22	21.93	261.97	1159.54	409.34	485.16	1159.54	1159.54	1159.54	1159.54	1159.54	1159.54	1159.54	1159.54
Caller max RFC3550 jitter (ms)?	22	0.52	15.46	39.77	24.33	24.63	39.77	39.77	39.77	39.77	39.77	39.77	39.77	39.77
Caller mean RFC3550 jitter (ms)?	22	0.30	5.75	21.68	13.16	13.75	21.68	21.68	21.68	21.68	21.68	21.68	21.68	21.68
Caller SDP-RTP delay (ms)?	22	12.00	634.65	2427.85	2001.13	2237.01	2427.85	2427.85	2427.85	2427.85	2427.85	2427.85	2427.85	2427.85

Figure 51: Direct RTP Caller StarTrinity results

6. Packet lost: In the test for the 22 calls 95% percent have good quality audio and the packet lost was almost zero. Although, there was some Packet lost around 8% the overall average of packet lost was 0.37%. This still considered as a good value and under 3%.



Ncalls	Min	Average	Max	Percentile: 90%	95%	98%	99%	99.5%	99.8%	99.9%	99.95%	99.98%	99.99%
22	0.00	0.37	8.07	0.00	0.00	8.07	8.07	8.07	8.07	8.07	8.07	8.07	8.07

Most critical SIP Call ID: b57ffb848df54ad4971f3cdd28809693 Time: 8/31/2019 10:57:48 PM

Figure 52: Direct RTP Caller Packet Lost history chart

7. G.107 In the test more than 98% percent of the calls have 4.41 which a very satisfying level.

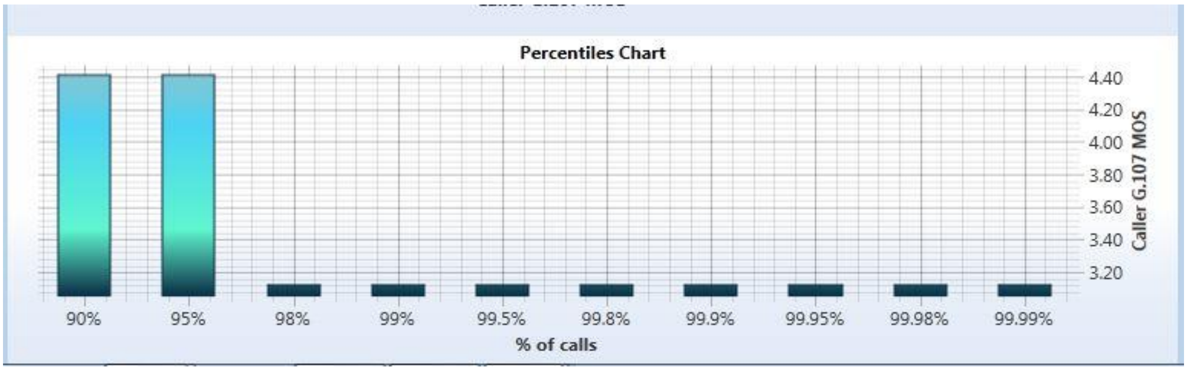
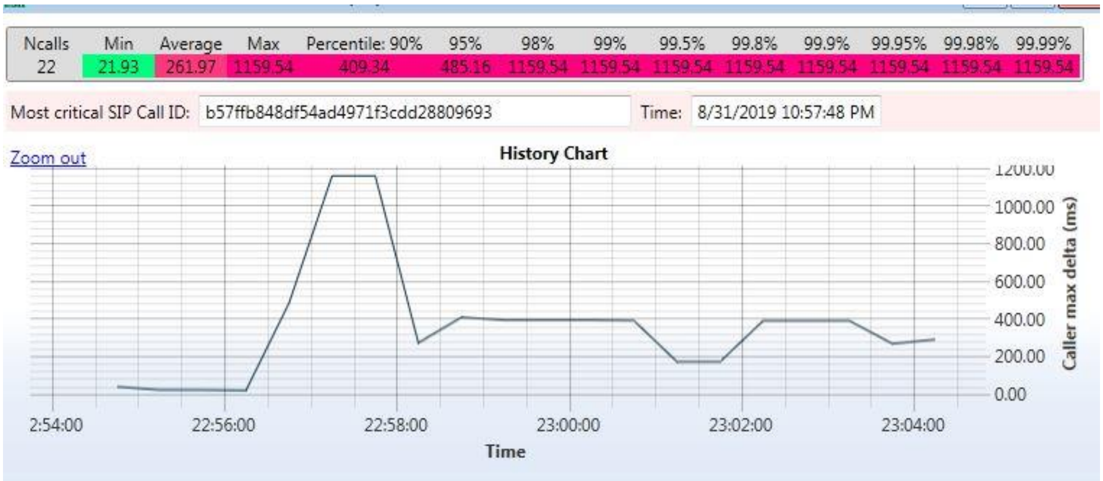


Figure 53: Direct RTP Caller G.107 percentiles chart

8. Caller max delta: Indicates unstable delays in the stream. The average value showed 261ms which is a very high number in the test. This means there are overloads in the network and this will have a direct impact on audio quality.

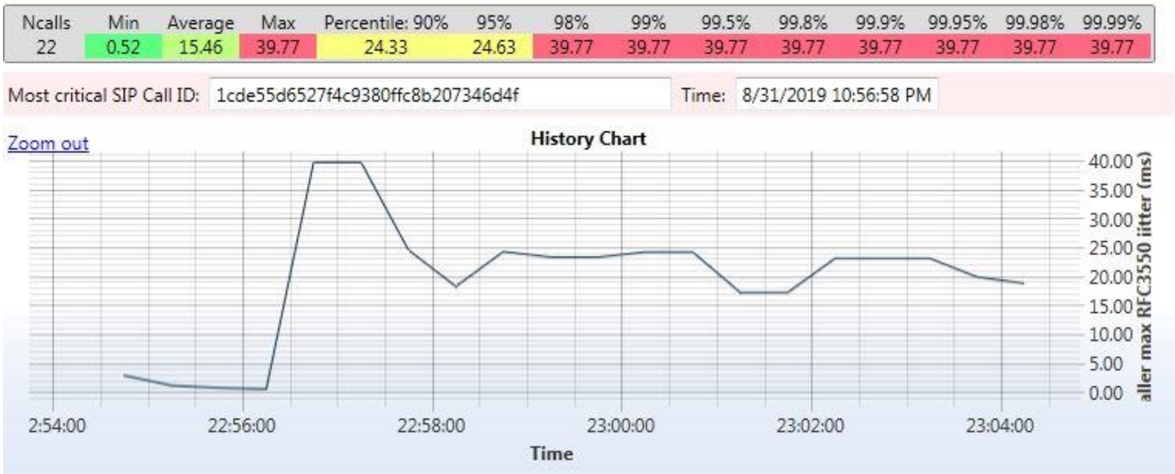


Ncalls	Min	Average	Max	Percentile: 90%	95%	98%	99%	99.5%	99.8%	99.9%	99.95%	99.98%	99.99%
22	21.93	261.97	1159.54	409.34	485.16	1159.54	1159.54	1159.54	1159.54	1159.54	1159.54	1159.54	1159.54

Most critical SIP Call ID: b57ffb848df54ad4971f3cdd28809693 Time: 8/31/2019 10:57:48 PM

Figure 54: Direct RTP Caller max delta history chart

9. RFC33550 jitter maximum: test max jitter showed an acceptable value in the average 15ms.



Ncalls	Min	Average	Max	Percentile: 90%	95%	98%	99%	99.5%	99.8%	99.9%	99.95%	99.98%	99.99%
22	0.52	15.46	39.77	24.33	24.63	39.77	39.77	39.77	39.77	39.77	39.77	39.77	39.77

Most critical SIP Call ID: 1cde55d6527f4c9380ffc8b207346d4f Time: 8/31/2019 10:56:58 PM

Figure 55: Direct RTP Caller jitter history chart

10. Caller SDP-RTP delay: represent the delay between SPD negotiation and the first RTP packet. The chart shows an average of 634 ms. This value still in the green (good) area. The Delay between the SDP and first RTP showed higher delay in the Direct RTP because RTP routed in Tor random routes, on the other hand Non-direct RTP routed to the Server, which decrease the delay since route is already know.

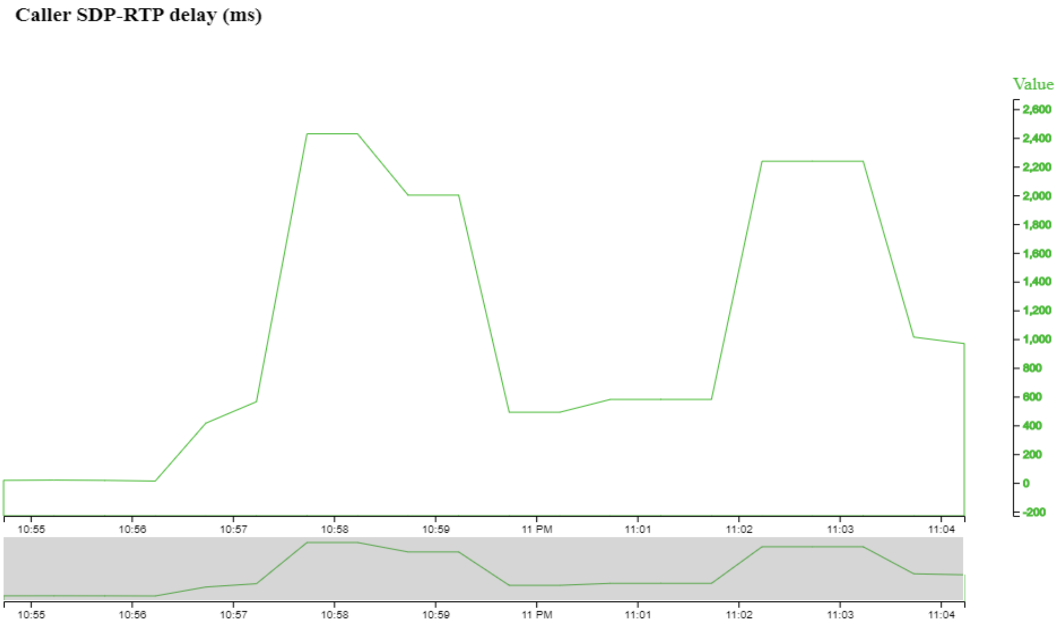
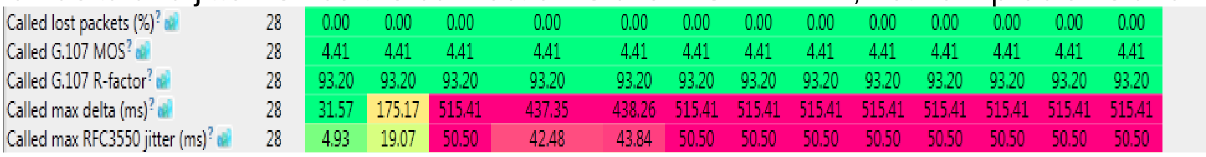


Figure 56: Direct RTP Caller SDP-RTP delay

Call destination showed also some acceptable results except for some problems with max delta and jitter. Since the connection is over TOR + VPN, network problems and



Called lost packets (%)?	28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Called G.107 MOS?	28	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41
Called G.107 R-factor?	28	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20
Called max delta (ms)?	28	31.57	175.17	515.41	437.35	438.26	515.41	515.41	515.41	515.41	515.41	515.41	515.41	515.41
Called max RFC3550 jitter (ms)?	28	4.93	19.07	50.50	42.48	43.84	50.50	50.50	50.50	50.50	50.50	50.50	50.50	50.50

Figure 57: Called destination StarTrinity results

delay can cause high Packet lost and high jitter. TOR is slow by its nature, also traffic in TOR needs to travel through multiple nodes. This makes the network's problems difficult to identify. Also, other problems could be caused by Bulletproof feature on Elastix. On the recipient side 28 calls are received.

The max RFC3550 jitter and max delta affected only 10% of the calls. In the following charts, both results started after 90% of the calls to happen.

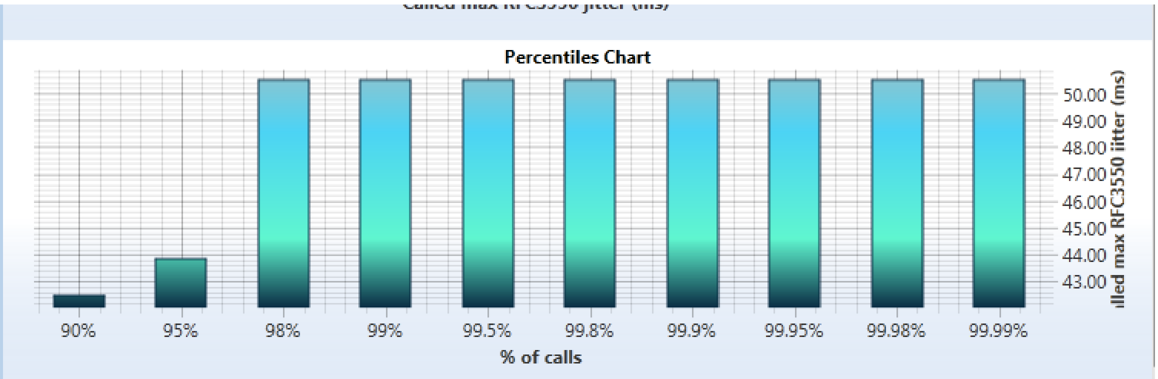


Figure 58: Direct RTP call destination jitter percentiles chart

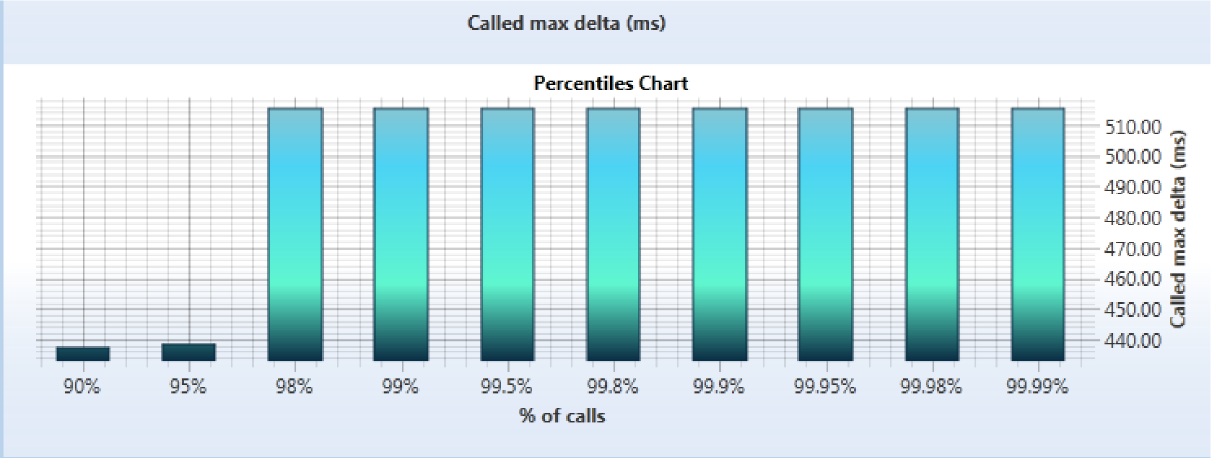
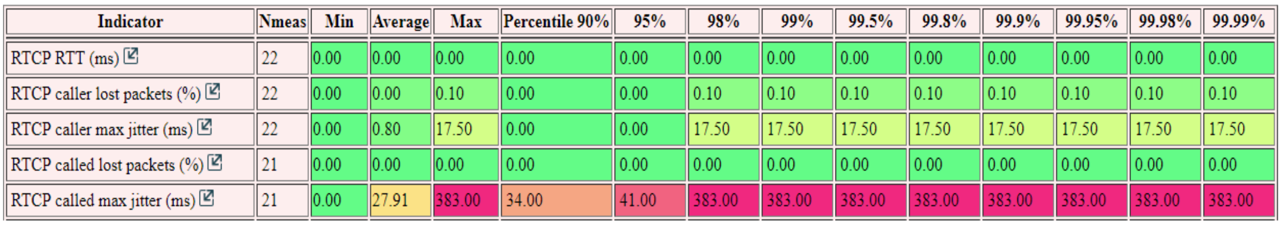


Figure 59: Direct RTP call destination max delta percentiles chart

Using Startrinity RTP analysis for both sides caller and called destination was also possible.



Indicator	Nmeas	Min	Average	Max	Percentile 90%	95%	98%	99%	99.5%	99.8%	99.9%	99.95%	99.98%	99.99%
RTCP RTT (ms)	22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RTCP caller lost packets (%)	22	0.00	0.00	0.10	0.00	0.00	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
RTCP caller max jitter (ms)	22	0.00	0.80	17.50	0.00	0.00	17.50	17.50	17.50	17.50	17.50	17.50	17.50	17.50
RTCP called lost packets (%)	21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RTCP called max jitter (ms)	21	0.00	27.91	383.00	34.00	41.00	383.00	383.00	383.00	383.00	383.00	383.00	383.00	383.00

Figure 60: Direct RTP results

- RTCP RTT - IP network's round-trip delay by RTCP.
- RTCP caller lost packets (%) – the lost percentage of RTP by caller party. Here no packets were lost from caller side.
- RTCP called lost packets (%) - the lost percentage of RTP by called party. It shows that no packets were lost between the caller and called destination.
- RTCP caller max jitter (ms) - jitter of RTP packets by the caller. No jitter of RTP in the test.
- RTCP called max jitter (ms) - jitter of RTP packets by the callee. RTP jitter at average was around 28ms, an acceptable value but some calls will suffer from jitter.

Above mentioned results of the VoIP call tests on Tor network showed that calls are possible to be done but sometimes those calls will suffer in audio quality due to the packet lost and jitter.
This time interval showed an average of 0.37% on caller side calls have some packet lost. Although 22 calls have made it, this is 14.67% from total calls 150 this represents the Answer Seizure Ratio (ASR) based on ITU Recommendation E.411 (UNION, 1988). This ASR is very low ratio and not recommended, the ASR should be at least over 20%. (Sippy Software, 2018)
On the other hand, the recipient side also has some packet loss. more 10% of calls had jitter.

Direct RTP Steam connection over Tor networks showed not really acceptable results. Jitter and packet loss affected the majority of calls. Although at some points, the average value of jitter and packet lost did not affect all calls, some of them have achieved acceptable call and voice quality. Results indicate the making VoIP calls over the Tor network using OpenVPN is possible but sometimes with jitter and packet loss restriction.

2.5 Anonymity of VoIP

Being anonymous, it means hiding the identity or not showing the real identity (Cambridge). In VoIP being anonymous has many variations based on how to define anonymous:

- The caller or the callee wants to be anonymous for the other side, neither the caller nor the callee knows who is on the other side.
- The caller and callee already know whom they are calling but they want the conversation to remain anonymous to anyone, any 3rd parties on the line or even the central server.

This thesis focused on the second type, which is how to make VoIP communication anonymous to anybody that has access to the communication stream and eavesdropper.

2.5.1 Anonymity in Direct RTP

If the connection uses Direct-RTP, in order to achieve anonymous VoIP with the existence of eavesdropper on the line, the VoIP content should be concealed. This cover will make VoIP content useless for an eavesdropper, by encrypting the End to End connection, using VPN the content will be concealed from an eavesdropper.

However, only encrypting the VoIP calls might not make those calls anonymous. For example, if Alice tries to make an encrypted call to Bob, the eavesdropper using some traffic analysis might be able to relate the VoIP flow between Alice and Bob. Consequently, determining the real IP addresses of both parties will be possible. This will break the anonymity of the VoIP. But how to hide the real IP addresses? This can be achieved by using Tor anonymous network.

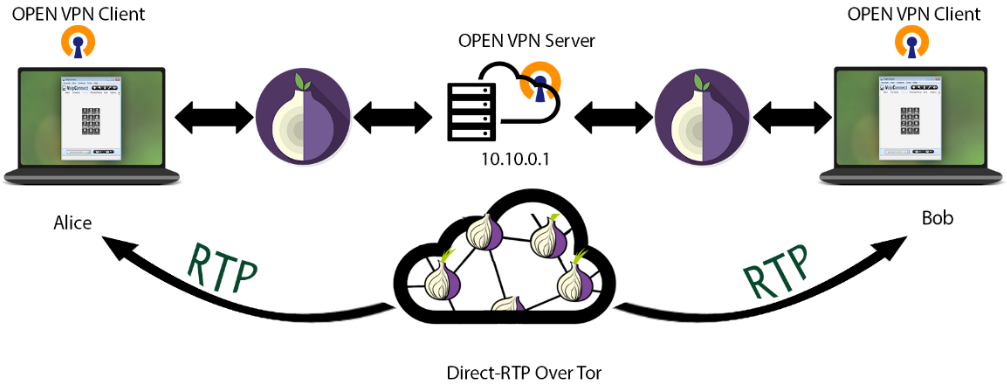


Figure 61: Direct RTP in VPN over Tor network

Using VPN over Tor network will increase the anonymity of end to end connection. First, connection to the Tor network should be established form both sides (Alice and Bob). Then, VPN Client on both sides should establish the connection to the VPN Server, each side will get an VPN IP address, which is not related to the real IP at all. Now, both sides are connected to the same VPN network and IP addresses are related to only to this network. If Alice wants to call Bob, both will register to the Elastix with the Caller ID and VPN IP addresses, Alice will use Bob's VPN IP address as a target IP address.The VoIP traffic will be encrypted using VPN at Tor entry node and will stay encrypted at the exit node. Traffic will be routed over Tor and eavesdropper will not be able to determine the real IP but only the Tor exit node. This is also providing protection against malicious tor exit nodes since the data is already encrypted.

Different possible attacks might cause partial deanonymization of the connection:

- In the project setup, Elastix Server is the VPN Server and has an Onion address. Attacks on the server itself will make the attacker able to decrypt the data. Since the implementation is Direct RTP, only the SIP Invite message will be available. Since the Server is the VoIP and VPN Server the attacker will be able to determine the caller ID and the callee ID beside VPN IP address, also the Tor entry and exit nodes addresses, so real IPs will stay anonymous because parties have registered to Elastix with their VPN IP addresses also the conversation itself because no RTP traffic is going through the Elastix server between parties.

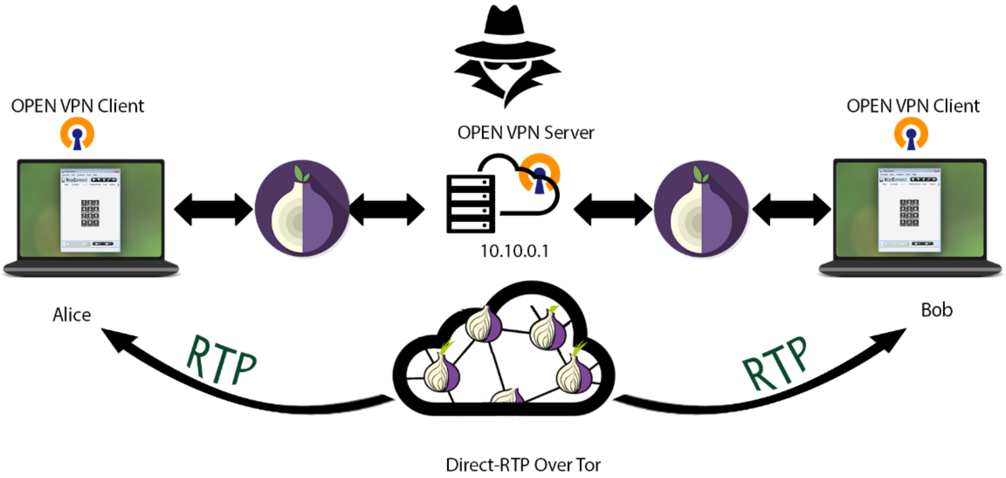


Figure 62: Attacking Elastix Server

- Since the VPN is over the Tor network, attacking the entry node might enable the attacker to determine the caller’s real IP address (Alice), Tor exit node address and the VoIP destination IP (VPN IP address). Attacker might capture the traffic, but the traffic is encrypted using the VPN, so no leak of conversion is possible.

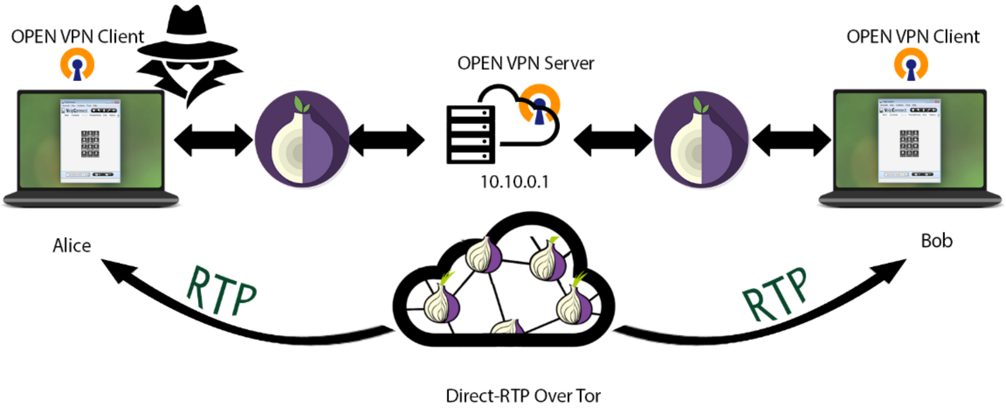


Figure 63: Entry node attack

- Controlling the exit node will make the attacker able to identify the real IP of the callee (Bob). But the caller IP will still be concealed (entry node), also traffic is encrypted (VPN). Which mean attacker cannot analysis the traffic and will not be able to identify the data type for example there is a call happening ,will happened or any other type of data streaming .

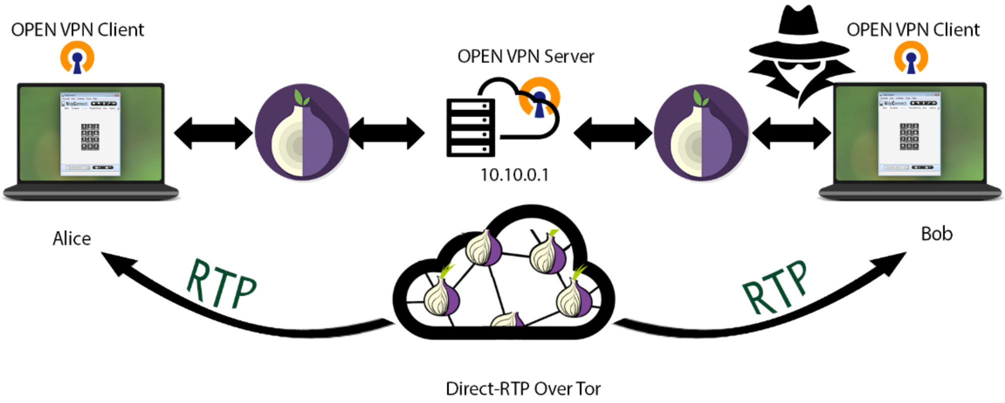


Figure 64: Exit node attack

In the above-mentioned attacks, anonymity was partially broken. Either one of the IPs is revolved or both IDs in case of server attack. But RTP traffic is always encrypted and routed with Tor.

There is only one possible way to deanonymize the connection, identify real IPs and decrypt the conversion:

- To achieve this the attacker should control more than one node. The attacker should control the entry, server and exit node. The combination of all those three will let the attacker identify the caller and callee real IPs and decrypt the traffic using the key from the server.

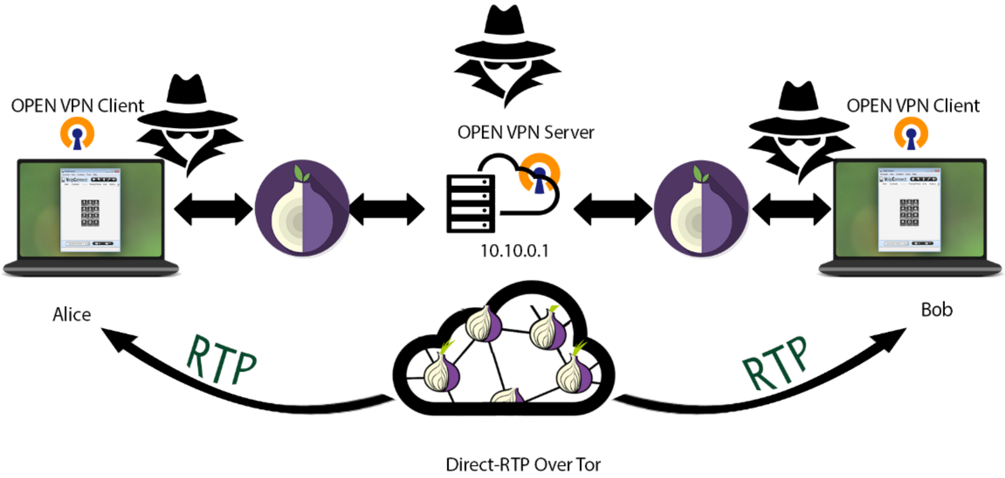


Figure 65: Direct RTP Multiple attacks

Although achieving this combination is very hard it is still possible. There is no 100% secure system yet, and combining Tor with VPN will increase the anonymity for sure but will not provide the ultimate anonymity for VoIP connection.

2.5.2 Anonymity in Non-Direct RTP

In Non-Direct RTP connection, attacks on the entry and exit nodes will remain the same. Controlling the entry or exit node will make the attacker able to identify the real IP of the caller (Alice) in case of entry node and callee (Bob) in case of exit node. But the callee real IP will still be concealed (entry node) and caller real IP will still be concealed (exit node), also traffic is encrypted (VPN). Which mean attacker cannot analysis the traffic and will not be able to identify the data type for example there is a call happening or will happen.

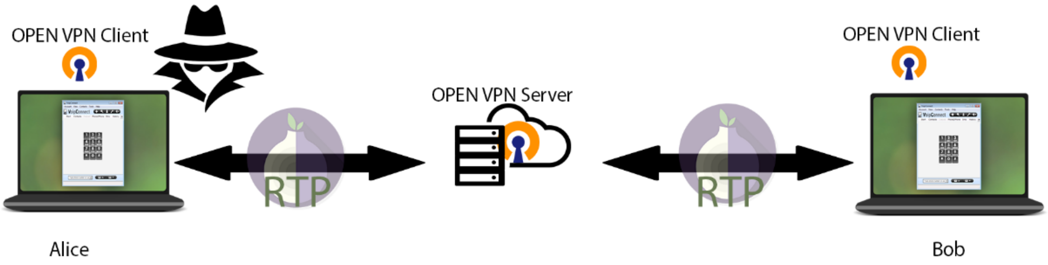


Figure 66:Non-Direct RTP, Entry node attack

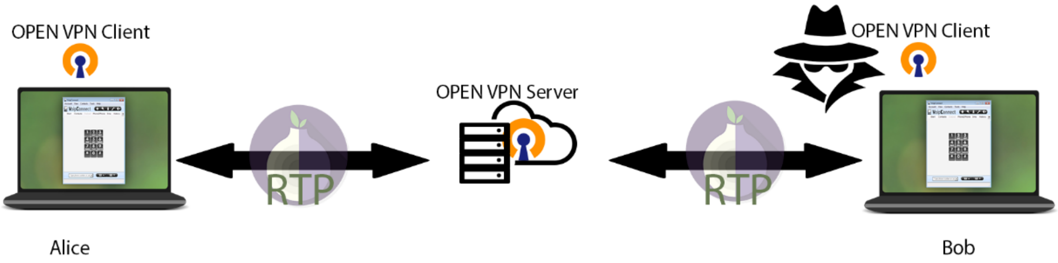


Figure 67:Non-Direct RTP, Exit node attack

One of the problems with Non-Direct RTP is when the attacks on the server itself. Attacker will be able to decrypt the data since the Elastix server is the VoIP and VPN server. Not only the SIP messages will be decrypted but also the RTP packets using the Key from Elastix server. Attacker will be able to listen to the call. Also, he/she will be able to determine the caller ID and the callee ID, VPN IP addresses and the Tor entry and exit nodes addresses. Although, the real IP addresses will stay anonymous because parties have registered to Elastix over Tor.

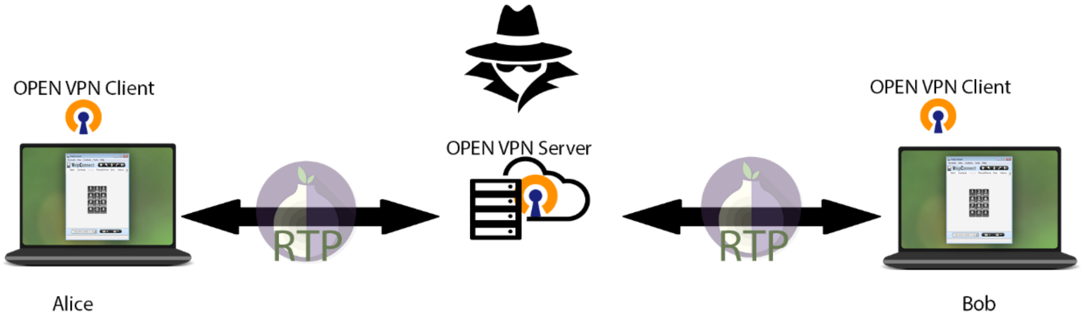


Figure 68:Non-Direct RTP, Server attack

The worst-case scenario is when the attacker/s controlled more than one node. The attacker should control the entry, server and exit node. The combination of all those three will let the attacker identify the caller and callee real IPs and decrypt the traffic from the server.

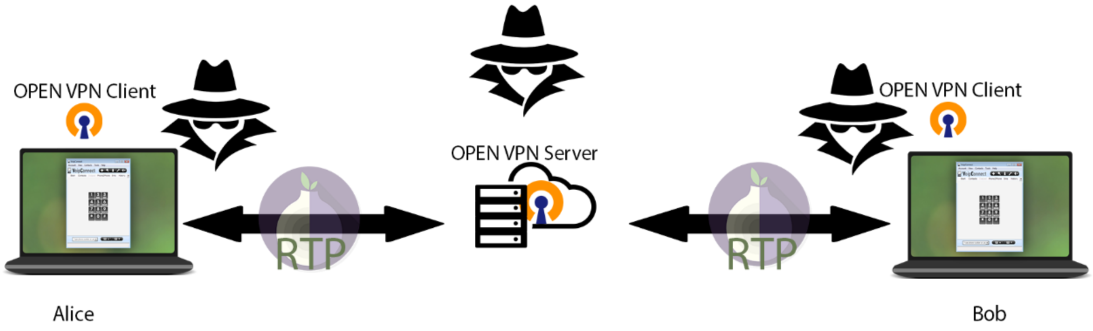


Figure 69:Non-Direct RTP Multiple attacks

3. Conclusion and future work

3.1 Conclusion

This section includes the concluding remarks of the research on the experimental results. The point of this research was to find out the anonymity of VoIP call over Tor network and the Quality of Service execution this VoIP through Tor. The results of the research showed that although voice packets cannot be ideally transmitted via the Tor network it is able to transfer voice packets with lower QoS.

Moreover, the research found out answers to questions of the research. The first question asked for VoIP combination with the Tor network. This study proved that Tor network can be combined with VoIP by using OpenVPN to wrap UDP stream with TCP stream. In OpenVPN, the connection between the sender and the receiver was immediate. The second question asked for the Quality of Service performance of

VoIP through the Tor network. In this regard, this research found out the QoS performance by calculating three QoS metrics namely: latency, jitter and packet loss.

Another question was about the anonymity of the VoIP over Tor. To send anonymous data from a caller to a callee, the Onion Routing network is utilized. This privacy is achieved in Tor by concealing the path of connection between the sender and the receiver. Using VPN over Tor network on Direct-RTP calls showed high anonymity of calls, attacker might need to control entry, exit and server node to be able to deanonymize the call otherwise he might get partial information but still not enough to deanonymize the call. Usually, the Tor network uses three relays to link the caller and the callee. each relay of Tor maintain a level of security and it uses a unique 128 bits key through the Advanced Encryption Standard (AES) encryption.

To transfer data over the Tor network is utilized. As the ITU standard demands the latency of a voice packet to be less than 400 ms. The research showed high jitter and latency in most of the calls, although some of them had an acceptable level of jitter and latency.

At present, the Tor network is being used by many customers to communicate anonymously worldwide. The increase of Tor usage might affect directly the QoS in VoIP by increasing latency and jitter.

To conclude, the research has found out that The Onion Routing network is not ideal for use in VoIP calls, but still possible. Some of the results of the call showed that many VoIP calls sent through the Tor network displayed QoS performance that is acceptable to ITU.

3.2 Future work

The research is based on empirical experiment, and the Tor network has limitations (users, bandwidth and relays) those cannot be adjusted. The focus was to achieve a high level of security and privacy with a good QoS. In the future, research should do more QoS performance in the VoIP and not only on the Tor network, but on other anonymous networks like JAP, P5, and Crowds. Also, another approach will be to define the Tor path which will reduce the jitter and latency in calls.

Also, the possibility of building own anonymous VoIP network.
This research could also find a solution for encapsulation of the VoIP on the client side, like developing VoIP encapsulation plugin. OpenVPN will not be required, the client will use Tor ID to identify himself.

This plugin will be available for MySIP softphone which also will support an extension manager for plugins. This will give the softphone more option to communicate with different VoIP services and support another operating system.

Abbreviations

Abbreviation	Description
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
IP	Internet Protocol
ITU	International Telecommunication Union
IM	instant messaging
JAP	Java Anonymity Proxy
LAN	Local Area Network
MAD	Mean Absolute Deviation
NAT	Network Address Translation
NTP	Network Time Protocol
OR	Onion Routing
PSTN	Public Switched Telephone Networks
QoS	Quality of Services
RTCP	Real Time Control Protocol
RTP	Real Time Protocol

SIP	Session Initiation Protocol
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
VoIP	Voice over Internet Protocol
VPN	Virtual Private Network
WAN	Wide Area Network

References

A. Duric, S. A. (2004). *Real-time Transport Protocol (RTP) Payload Format for internet Low Bit Rate Codec (iLBC) Speech.* The Internet Society.

A. Duric, T. A. (2004). *Real-time Transport Protocol (RTP) Payload Format for internet Low Bit Rate Codec (iLBC) Speech.* The Internet Society.

Adeel Ahmed, H. M. (2010). *VoIP Performance Management and Optimization.* Cisco Press.

Alcantara, M. (2017, November 1). *What is Asterisk® and what are the differences with Elastix?* Retrieved from Elastix: https://www.elastix.org/blog/latestnews/what-is-asterisk-differences-elastix/

Aleshin, S. (n.d.). *StarTrinity SIP Tester™*. Retrieved from http://startrinity.com: http://startrinity.com/voip/siptester/siptester.aspx

Andreas Pfitzmann, M. K. (2009). Anonymity, Unobservability, and Pseudonymity - A Proposal for Terminology. In M. K. Andreas Pfitzmann, *Designing Privacy Enhancing Technologies* (pp. 1-9). Springer Berlin Heidelberg.

Bowei Xi, W. S. (2010). Statistical analysis and modeling of Internet VoIP traffic for network engineering. *Electronic Journal of Statistics*.

Cambridge, U. P. (n.d.). *Meaning of anonymity in English.* Retrieved from Cambridge Dictionary: https://dictionary.cambridge.org/dictionary/english/anonymity

Choi, H., Lee, H., Lee, H., & Kim, H. (2007). Botnet Detection by Monitoring Group Activities in DNS Traffic. *7th IEEE International Conference on Computer and Information Technology.* Aizu-Wakamatsu, Fukushima, Japan: IEEE.

Chu, L., Huo, Z., & Liu, L. (2011). The security research of SIP-based Denial of Service attack. *2011 International Conference on Electrical and Control Engineering.* IEEE.

Claudia Diaz, S. S. (2003). Towards Measuring Anonymity. *The 2nd international conference.* San Francisco: Springer-Verlag Berlin Heidelberg.

D. Mills, U. D. (June 2010). *RFC 5905: Network Time Protocol Version 4: Protocol and Algorithms Specification.* Internet Engineering Task Force.

Dai, W. (n.d.). *PipeNet 1.1*. Retrieved from weidai.com: http://www.weidai.com/pipenet.txt

Dierks Certicom, A. C. (1999). *The TLS Protocol Version 1.0.* The Internet Society(1999).

Gegel, V. (2012). *TORFone - voice add-on for TorChat*. Retrieved from TOR Fone: http://torfone.org/

George Danezis, C. D. (2010). *Systems for anonymous communication.*

Gonia, K. (2004). *Latency and QoS for Voice over IP.* SANS.

H. Sinnreich, A. B. (2006). *Internet Communication Using SIP:.* Wiley Publishing.

Heath, M. (n.d.). *NAudio.* Retrieved from github: https://github.com/naudio/NAudio

ITU. (2003, 05). *ITU-T Recommendation G.114.* Retrieved from https://www.itu.int.

Jiang, W. (n.d.). *A Lightweight Secure SIP Model for End-to-End Communication.* 2005 : Institute of Information Technology, Tsinghua University, Beijing, 100084, P.R.China .

Jie Wu, J. R. (2010). Survey on anonymous communications in computer networks. *Computer Communications archive Volume 33* , 420-431.

Johnston, A. B. (2004). *SIP: Understanding the Session Initiation Protocol.* Artech House.

Jonathan Davidson, J. F. (2006). *Voice Over IP Fundamentals.* Cisco Press.

Keromytis, A. D. (2011). *Voice over IP Security: A Comprehensive Survey of Vulnerabilities and Academic Research.* Springer Science & Business Media.

Keromytis, A. D. (2012). A Comprehensive Survey of Voice over IP Security. pp. 514-536.

Kevin Bauer, M. S. (2012). *ExperimenTor: A Testbed for Safe and Realistic Tor Experimentation.*

Kolesnikov, V. K. (2010). A secure and lightweight scheme for media keying in the session initiation protocol (SIP). Chicago, Illinois, USA: IPTComm.

Landström, S. (2008). *TCP/IP Technology for Modern Network Environments.* Sweden: Luleå University of Technology.

Leif Madsen, J. V. (2011). *Asterisk: The Definitive Guide.* O'Reilly Media, Inc.

Li, C., Li, H., Wang, K., & Nan, K. (2011). Research and Implementation of Unified Communications System Based on Elastix. *7th International Conference on Wireless Communications, Networking and Mobile Computing.* Wuhan, China: IEEE.

Liancheng Shan, N. J. (2009). *Research on Security Mechanisms of SIP-Based VoIP System.* Shenyang, China: IEEE.

LLC, N. (n.d.). *Overview*. Retrieved from Anonymizer.

Lokesh Bhoobalan, P. H. (2011). An Experimental Study and Analysis of Crowds based Anonymity. *International Conference on Internet Computing.*

Lydia Parziale, D. T. (2006). *TCP/IP Tutorial and Technical Overview.* International Business Machines Corporation .

M. Baugher, D. M. (2004). *The Secure Real-time Transport Protocol (SRTP).* The Internet Society.

M.G. Reed, P. S. (1998). Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications* (pp. 482 - 494). IEEE.

Marc Liberatore, B. G. (2011). Empirical tests of anonymous voice over IP. *Journal of Network and Computer Applications*, 341-350 .

Munson, L. (2013, August 29). *Tor usage doubles in August. New privacy-seeking users or botnet.* Retrieved from https://nakedsecurity.sophos.com/2013/08/29/tor-usage-doubles-in-august-new-privacy-seeking-users-or-botnet/

Network, I. V. (2006). *Alvarion's BreezeACCESS.* alvarion.

Ono, A. K., & Tachimoto, S. (2004, March 22). *SIP signaling security for end-to-end communication.* Malaysia: IEEE.

P. Ai-Chun, L. C.-H.-N. (2005). A study on SIP session timer for wireless VoIP. *Wireless Communications and Networking Conference* (pp. 2306-2311). IEEE.

Panchenko, A., Lanze, F., & Engel, T. (2012). Improving performance and anonymity in the Tor network. *IEEE 31st International Performance Computing and Communications Conference (IPCCC).* Austin, TX, USA: IEEE.

Paul Syverson, G. T. (2001). Towards an analysis of onion routing security. *International workshop on Designing privacy enhancing technologies: design issues in anonymity and unobservability* (pp. 96-114). Berkeley, California, USA: Springer-Verlag Berlin, Heidelberg.

Paul, S. (2011). A Peel of Onion. Orlando, Florida: ACSAC.

Pedro Correia, E. R. (2012). *Statistical Characterization of the Botnets C&C Traffic.* Aveiro, Portugal: Instituto de Telecomunicações, University of Aveiro,.

Perkins, C. (2003). *RTP: Audio and Video for the Internet.* Addison-Wesley Professional.

Phillip Kisembe, W. J. (August 2017). Future of Peer-To-Peer Technology with the Rise of Cloud Computing. *International Journal of Peer to Peer Networks (IJP2P) Vol.8, No.2/3*.

Postel, J. (1981). *TRANSMISSION CONTROL PROTOCOL.* Information Sciences Institute University of Southern California.

Prof. Dr. Hannes Federrath, S. K. (n.d.). *Project: AN.ON - Anonymity.Online*. Retrieved from Anonymity and Privacy: https://anon.inf.tu-dresden.de/index_en.html

Project, T. T. (2009, February 18). *One cell is enough to break Tor's anonymity.* Retrieved from blog torproject: https://blog.torproject.org/one-cell-enough-break-tors-anonymity

Project, T. T. (2009–2018). Retrieved from Tor Metrics: https://metrics.torproject.org/

Project, T. T. (2009–2018). *Tor Metrics*. Retrieved from https://metrics.torproject.org/

ProtACT Team, I. T. (2013, September 5). *Large botnet cause of recent Tor network overload.* Retrieved from Fox it: https://blog.fox-it.com/2013/09/05/large-botnet-cause-of-recent-tor-network-overload/

R. Roselinkiruba, R. B. (2013). *Secure steganography in audio using inactive frames of VoIP streams.* Thuckalay, Tamil Nadu, India: IEEE.

Ram Dantua, S. F. (2009). Issues and challenges in securing VoIP. *elsevier*, 2-9.

Ramzi A. Haraty, M. A. (2017). A Systematic Review of Anonymous Communication Systems . *the 19th International Conference on Enterprise Information Systems - Volume 2: ICEIS* (pp. 211-220). Science and Technology Publications, Lda.

Ransome, J. F. (2005). *Voice over Internet Protocol (VoIP) Security.* Elsevier.

Richard Kuhn, T. J. (2005). *Security Considerations for Voice Over IP Systems.* USA: NIST SP 800-58.

Roger Dingledine, N. M. (2004). Tor: The Second-Generation Onion Router. *USENIX Security Symposium .* San Diego, CA.

Roger Dingledine, N. M. (n.d.). *Tor project.* Retrieved from Tor Manual: https://2019.www.torproject.org/docs/tor-manual.html.en

Ronggong Song, L. K. (2002). Anonymous Internet Communication Based on IPSec. *The State of the Art IFIP 17th World Computer Congress — TC6 Stream on Communication Systems: The State of the Art August 25–30, 2002,* (pp. 199-214). Montréal, Québec, Canada: Springer.

Rubin, M. K. (1997). *Crowds: Anonymity for Web Transactions.* Center for Discrete Mathematics & Theoretical Computer Science.

S. Andersen, A. D. (2004). *Internet Low Bit Rate Codec (iLBC).* The Internet Society. Retrieved from iLBC Freeware.

S. Yoon, H. J.-S. (2009). A Study on the Interworking for SIP-Based Secure VoIP Communication with Security Protocols in the Heterogeneous Network. pp. 165-175.

Schulzrinne, H. (2003, July). *RTP: A Transport Protocol for Real-Time Applications.* The Internet Society.

SDK, O. V. (n.d.). *Ozeki VoIP SIP SDK.* Retrieved from Ozeki VoIP SIP SDK: http://www.ozeki.hu/index.php?owpn=1017&download_product_id=2

Shannon, C. E. (1948). A Mathematical Theory of Communication. In *A Mathematical Theory of Communication* (pp. 379-423, 623-656). The Bell System Technical Journal.

Shiping Chen, X. W. (2006). On the anonymity and traceability of peer-to-peer VoIP calls. *IEEE Network (Volume: 20)* (pp. 32 - 37). IEEE.

Sippy Software, I. (2018, 2 27). *Understanding Answer Seizure Ratio (ASR)*. Retrieved from sippysoft.com/: https://support.sippysoft.com/support/solutions/articles/3000080552-understanding-answer-seizure-ratio-asr-

StarTrinity. (n.d.). *SIP Tester Tutorial.* Retrieved from Startrinity: https://startrinity.com/VoIP/SipTester/SipTesterTutorial.aspx

Syverson, P. (2013). *Why I'm not an Entropist.* Springer.

Thomas Porter, C. C. (2011). *How to Cheat at VoIP Security.* Syngress.

Tim Szigeti, C. H. (1994). *End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs* . Cisco Press.

UNION, I. T. (1988, 11). INTERNATIONAL NETWORK MANAGEMENT – OPERATIONAL GUIDANCE. *SERIES E: OVERALL NETWORK OPERATION, TELEPHONE SERVICE, SERVICE OPERATION AND HUMAN FACTORS*, p. 18.

VoIP, O. (n.d.). *Ozeki VOIP SIP SDK*. Retrieved from Ozeki VOIP SIP SDK: http://www.voip-sip-sdk.com/p_11-differences-between-ozeki-voip-sip-sdk-and-other-sip-sdks-voip.html

Volker Fusenig, D. S. (2008). Anonymous Communication in Multi Hop Wireless Networks. *Journal of Research and Practice in Information Technology*.

Wikipedia. (n.d.). *Diffie–Hellman key exchange.* Retrieved from https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange

Xu Jing, W. Z. (2010). Recipient Anonymity: An Improved Crowds Protocol Based on Key Sharing. *2010 WASE International Conference on Information Engineering* (pp. 60-64). Beidaihe, Hebei, China: IEEE.

Yong Guan, X. F. (2002). An optimal strategy for anonymous communication protocols. *Proceedings 22nd International Conference on Distributed Computing Systems.* Vienna, Austria: IEEE.