telecom

MDPI

Article

Studying the Impact of Different TCP DoS Attacks on the Parameters of VoIP Streams

Ivan Nedyalkov

Faculty of Engineering, South-West University "Neofit Rilski", 2700 Blagoevgrad, Bulgaria; i.nedqlkov@swu.bg

Abstract: In today's digital world, no one and nothing is safe from potential cyberattacks. There is also no 100% protection from such attacks. Therefore, it is advisable to carry out various studies related to the effects of the different cyberattacks on the performance of the specific devices under attack. In this work, a study was carried out to determine how individual TCP DoS attacks affect the parameters of VoIP (Voice over IP) voice and video streams. For the purpose of this work, a model of a simple IP network has been created using the GNS3 IP network-modeling platform. The VoIP platform used was Asterisk Free PBX. Tools from Kali Linux were used to implement the individual TCP DoS attacks; IP-network-monitoring tools and round-trip-delay-measurement tools were also used. The proposed study is applicable to multiple VoIP platforms wherein voice and video traffic are passed/processed by the VoIP server. From the obtained results, it was found that Asterisk Free PBX is very well secured against TCP DoS attacks, which do not affect the platform performance or the parameters of the voice and video streams. The values of the observed parameters, such as jitter, packet loss, round-trip delay, etc., are very far from the maximum allowable values. We also observed a low load on the CPU and RAM of the system during the whole study.

Keywords: Asterisk Free PBX; DoS attacks; GNS3; jitter; modeling of IP networks; network monitoring; packet loss; VoIP; voice streams; video streams

Citation: Nedyalkov, I. Studying the Impact of Different TCP DoS Attacks on the Parameters of VoIP Streams. *Telecom* **2024**, *5*, 556–587. https://doi.org/10.3390/telecom5030029

Academic Editor: Maurizio Pizzonia

Received: 25 May 2024
Revised: 30 June 2024
Accepted: 3 July 2024
Published: 8 July 2024

1. Introduction

DoS (Denial of Service) attacks are becoming more frequent. These attacks are probably the easiest type of attack to carry out. Any device that has an embedded web server (runs requests) can be affected by a DoS attack as long as it has network functionality—it can connect to IP networks or the Internet. These devices can be various power electronic devices, IoT (Internet of Things) devices, cyberphysical systems, etc. An attack may cause very serious problems and consequences; take, for example, an attack on a power electronic device that is used by a telecommunications operator to remotely control the power supply (PDU—Power-Distribution Unit) of various communication devices (such as routers or switches). The remote switching on/off (hard restart) of the devices, in case of a detected problem with the operator's communication network, would not be possible if the device were to be affected by a DoS attack. As a result of the attack and the inability to remotely reboot the communication device, the telecom operator will begin to suffer losses and customer dissatisfaction will build due to problems with the network caused by an attacked PDU device. Of course, there are different methods and techniques to protect against this type of attack, such as the following:

- Network segmentation by creating VLANs and using hardware firewalls;
- Load balancing—distributing traffic across multiple servers;
- Blocking traffic from known or suspected IP addresses that have been linked to DoS attacks in the past or present;
- Limiting the speed of the traffic, which can prevent a DoS attack from overloading the server;

- Using Content Delivery Networks (CDNs)—this distributes the content of the website across multiple locations; thus, a DoS attack could not bring down the entire site.

The consideration of these methods and techniques of protection is not the subject of this work. The aim is to observe what happens to an attacked device during a DoS attack and how the attack affects the performance of the device. It is interesting to perform research aimed at monitoring what happens to the attacked device during a DoS attack in order to study the results of these attacks in more detail [1–12].

The question everyone would ask is how to implement such a study. Of course, the best option is to create an experimental IP network and to run various tests with DoS attacks on different devices on the network and observe what happens. An even better option is to use a working network wherein the device under study is connected and attacked. However, here, there would be two main problems: first, there is a danger that the attack could get "out of control", i.e., affect other working devices, and, secondly, no one would allow such a study on a working network. Therefore, the most convenient way to carry out such studies is through modeling or by using IP network-modeling platforms [13–30]. Through the use of IP network-modeling platforms, several problems are solved:

- It solves the problem wherein the expensive physical network devices needed for experimental networks are not available;
- The modelled experimental network is completely closed. Thus, studies of different attacks will be 100% controllable and there is no danger of the attacks going outside the controlled area.

What is the need for such a study? First, it is interesting to observe if and how different TCP DoS attacks change or influence the parameters of VoIP voice and video streams in some way while the VoIP server is being attacked. Additionally, the study will test whether the studied VoIP server provides any protection from DoS attacks. A question arises as to the importance of studying a VoIP server under DoS attack, as voice traffic is exchanged directly between the IP telephones (software or physical) and is not passed/processed by the VoIP server. This is the case, but there are VoIP servers wherein the voice and video traffic are passed/processed by the VoIP server. This study is about those VoIP servers, wherein the voice and video traffic are passed/processed by the server.

In this work, a simple model consisting of an IP network composed of several users, a VoIP server, and an attacker located in an external network has been created. The GNS3 IP network-modeling platform, Asterisk Free PBX, Kali Linux and network-monitoring tools have been used for the purpose of the study. Asterisk Free PBX is one of the VoIP servers described above, wherein voice and video traffic are passed/processed by the server. The research was carried out in two parts. Initially, only voice traffic was exchanged between subscribers in the network, and in the second part, video traffic was exchanged between two subscribers. During all the conversations (voice and video calls), Kali Linux continuously attacked Asterisk with various TCP DoS attacks. Finally, an analysis of the obtained results was carried out.

This article has the following structure: Section 2—Related work; Section 3—Modeling platform, tools and research methodology; Section 4—Results and discussion; and Section 5—Conclusions.

2. Related Work

In [31], the authors propose their mechanism for protecting SIP (Session Initiation Protocol) from attacks, which aims to detect vulnerabilities in lesser-known features of the protocol. For the purpose of the study, the authors used a virtual machine equipped with Trixbox, an Asterisk-based IP-PBX system that served as the attack target, and softphones. Asterisk was affected by SIP-DRDoS (Distributed Reflection Denial-of-Service) attacks; thus, the authors verified the effectiveness of their protection mechanism. These attacks bypassed any defenses such as firewalls, IDS/IPS (Intrusion Detection system/Intrusion Protection System), control lists, etc. As a result of the attack, the CPU was overloaded

multiple times. The proposed protection mechanism reduced the CPU load and the impact of the attack on the system.

In [32], the authors propose a new protocol for SIP security—secure SIP (S-SIP). The proposed protocol was based on a thorough literature review performed by the authors, which examines different ways to secure the SIP protocol. For the purpose of the research and to prove the feasibility of the developed protocol, the authors created an experimental network in which they carried out various experiments. The results show that the proposed protocol offered more security capabilities and imposed a lower computational load.

In [33], the authors propose a method for DDoS-attack (Distributed Denial of Service attack) detection based on the Deep Packet Inspection (DPI) method for packet analysis (a kind of an intrusion-detection system). Through this method, certain information (attack signatures) was extracted from the packets. Using this information, new rules were applied to detect VoIP DDoS attacks. For the purpose of the study, the authors used Asterisk FREE PBX. The experimental studies confirmed that the use of the IDS system prevented DDoS attacks.

In [34], the authors review different methods for detecting DoS and DDoS attacks applied to SIP and classify them based on different factors. Furthermore, the authors explore the strengths and weaknesses of these methods. Finally, the authors provide a discussion of how to improve the reviewed methods and propose future research directions to build more effective solutions for detecting DoS and DDoS attacks.

In [35], the authors propose the use of a deep-learning model based on Recurrent Neural Networks (RNNs) to detect low- and high-intensity DDoS attacks. To prove the feasibility of the proposed method, the authors used real traffic traces (SIP messages) that were "injected" with malicious messages. Based on the obtained results, it was proved that the proposed method had high detection accuracy and low detection times.

In [36], the authors provide an in-depth review of the most-used IDS/IPS systems, with a corresponding analysis for each system considered. From the analysis carried out by the authors, it was found that eDAIT and the e-GAP performed the best out of all the reviewed models.

In [37], the authors developed a Support Vector Machine (SVM) learning algorithm for detection and prevention, which was used to detect DoS attacks. To validate the applicability of the algorithm, the authors used IP-PBX real-time traffic datasets. The obtained results, such as high detection rate, low execution times to classify the attack, low rates of false negatives, and other findings proved the applicability of the algorithm.

In [38], the author again proposes the use of deep-learning and entropy techniques to detect DDoS attacks in SIP-based systems. The proposed algorithm was a combination of deep-learning convolutional neural networks and a stacked bidirectional long short-term memory network. To validate the applicability of the algorithm, the author used a dataset of different types and intensities of DDoS flooding attacks. The proposed algorithm proved its feasibility by achieving high detection rates and low attack-detection times.

In [39], the authors extensively review various technologies and methods for early warning of a DoS attack on a VoIP network intended to protect the users/administrators of the attacked VoIP network from a DoS attack. Additionally, the authors discuss various vulnerabilities in VoIP networks and tests to detect those vulnerabilities.

In [40], the authors consider caller-ID (caller-identification) spoofing attacks because through these attacks, an attacker can very easily access important information transmitted over SIP. Due to the fact that these attacks can take place only in a closed system (VoIP network only), they are not a subject of research and therefore, solutions to deal with these attacks are few. Therefore, the authors propose Blockchain-Based Caller-ID Authentication (BBCA), their blockchain-based defense mechanism, to prevent these attacks.

In [41], the authors propose the use of machine learning as a replacement for rule-based systems to detect DoS attacks. For the purpose of the study, the authors created an experimental setup in which both useful traffic and traffic caused by a DoS attack was exchanged. Through this setup, the authors investigated the effectiveness of certain

machine-learning methods for classifying traffic (useful or dangerous traffic) based on the extracted information.

In [42], the authors propose a new type of SIP-based attack—Distributed Reflection Denial-of-Service (DRDoS). The purpose of this new attack was to expose the little-known capabilities of the SIP protocol. Additionally, the authors developed a simulator for these attacks, called Mr. SIP, which was used to generate the attack. The attack developed by the authors increased the CPU load of the attacked VoIP server dramatically, by up to 100%. Furthermore, due to the structure of the newly created attack, it could not be recognized as an attack, so it could not be filtered by firewalls, IDS/IPS systems, traffic-anomaly-detection systems, etc. The authors propose a defense mechanism for their attack that reduced the increase in CPU load during the attack from up to 100% without the protection mechanism to up to 18%.

In [43], the authors consider Distributed Denial of Service (DDoS) threats, specifically INVITE flooding attacks. To detect these attacks in time, the authors propose the use of a GRU-based Intrusion-Detection System (IDS). This GRU-based Intrusion-Detection System (IDS) was based on recurring neural networks, which process the SIP traffic in real time and efficiently identify attack patterns in the traffic. The ability of the developed IDS to capture temporary dependencies increased the accuracy with which it classified and detected attack behaviors.

In [44], the authors provide a thorough review of methods to secure VoIP platforms. The result of their study was that the existing ways to secure such systems are mainly focused on securing the network layer, while there are not enough measures aimed at securing SIP-based real-time VoIP communication at the application level. The goal set by the authors was to improve the security of the communication in VoIP systems by studying key technologies for detecting network attacks targeting SIP-based calls.

In [45], the authors propose the use of an IDS system they developed that verifies the content of the SIP messages in the processed traffic. The rule definition was applied to intrusion detection, sending fake SIP denial-of-service messages, and flooding with incorrect packets. The data collection to verify the developed IDS system was carried out in two stages; in the first stage, the VoIP system was not attacked, and in the second stage, it was attacked. The tests showed that, when using the k-nearest neighbors, the developed IDS system had the highest detection accuracy, at 99.8%.

In [46], the authors propose the use of a novel framework called "Call Me Maybe", which uses a combination of delay measurement and dynamic protocol switching to prevent DoS attacks in a VoIP system. The authors' proposal is to switch (change) the transport protocol for voice traffic from User Datagram Protocol (UDP) to TCP (Transmission Control Protocol) when the system is subjected to DoS attacks. Validation of the applicability of the proposed framework was accomplished through an established simulation model.

As can be seen from the reviewed works of other authors, as well as from studies not presented in this work, authors mainly propose methods, techniques, or algorithms to detect DoS/DDoS attacks or propose a mechanism to protect against such attacks or improve the protection of the SIP protocol. However, no study has been done on the impact of the DoS attack itself on the quality of the voice and video streams during an attack on a VoIP server through which VoIP streams are processed/exchanged.

3. Modeling Platform, Tools, and Research Methodology Used

3.1. Modeling Platform Used

The IP network-modeling platform used was GNS3 [47]. This platform has many capabilities, such as working with virtual machines; working with disk images of operating systems of real network devices; integration with various tools for monitoring the traffic in the modeled network; the ability to create models of IP networks with arbitrary sizes and numbers of devices, and many other features. The platform is completely free and is used by many of the world's leading network-equipment manufacturers and ICT (Information and Communications Technology) companies. The main requirement when using this

platform is that the computational capabilities of the workstation on which the networks will be modelled should be as high as possible. For this work, a dual-processor workstation with two 18-core processors or a total of 72 logical processors and 192 GB of RAM was used.

3.2. Tools Used

The tools used were as follows:

- Kali Linux (2024.2): this operating system and the multitude of different built-in tools were used for various tests/studies related to determining the level of network security and vulnerability testing [48];
- Wireshark (version 4.0.7): this network protocol analyzer can "capture" all exchanged packets between network devices in an IP network [49]. Due to its integration with GNS3, all nodes in the modeled network can be monitored through Wireshark. This tool "captured" all packets that were exchanged between Asterisk and the users;
- Colasoft Capsa Free (version 11.1): this network analyzer was used to monitor the traffic by displaying information about generated traffic, number of TCP packets, traffic generated by certain protocols, and other traffic-related factors [50];
- Colasoft Ping Tool (version 2.0): a tool that can be used to measure in real time the value of the round-trip delay [51]. The results of the measurement can be used to make graphs to show how the round-trip delay changed over a given period of time.

3.3. Research Methodology

As mentioned above, the study is divided into two parts.

In the first part, only voice traffic is exchanged between the virtual users in the modeled network. The virtual users are implemented using the hypervisor program VMware Workstation Pro 17 [52]. The Windows 10 [53] operating system was installed on these virtual machines. To be able to build VoIP connections between individual Asterisk subscribers, Linphone software for desktop [54] was installed on each of the virtual operating systems. The conversations were grouped into four parts of approximately ten minutes per conversation. During each of the conversations, Kali Linux attacked the Asterisk Free PBX with different TCP DoS attacks: first with TCP SYN flooding; then with TCP ACK flooding; then with TCP RST flooding, and finally with a TCP FIN flooding attack. The size of the packets was set to 900 bytes. This is the maximum packet size that the modeled network devices in the network can process. The inability to process more traffic is due to limitations in the used disk images made by the manufacturers. The packets were sent approximately every 10 µs.

In the second part of the study, only video calls were exchanged between two subscribers (because I have only two webcams). The video calls were grouped into four parts of about 10 min per video call. During each of the calls, Kali Linux attacked the Asterisk Free PBX with different TCP DoS attacks: first with TCP SYN flooding; then with TCP ACK flooding; then with TCP RST flooding, and finally with a TCP FIN flooding attack.

The monitoring was performed using Wireshark, which integrates with GNS3. This functionality makes it possible to observe absolutely all possible links in the modelled network. For the purpose of the study, only the link between Asterisk_Free_PBX and Switch2 was monitored. Only this link was chosen for monitoring because the traffic that is generated by Kali Linux (the various TCP DoS attacks), as well as the voice/video traffic that is exchanged between individual users on the network, passes through this link. As mentioned above, on Asterisk Free PBX, the voice and video traffic pass through the VoIP server, not just between individual subscribers. Because Wireshark was used in both studies, all packets that were exchanged between Asterisk, the virtual users, and the Kali Linux were captured. Once the study was completed, the Wireshark function for studying RTP streams was used to analyze the voice and video streams. Using this function of Wireshark, the observed parameters were analyzed and the impacts of various TCP DoS attacks were evaluated. The parameters observed were jitter values and packet loss.

During all the calls (voice and video), the round-trip delay (RTD) between each of the virtual users and Asterisk was continuously measured. The purpose of this measurement was to observe what happened to the connection between individual subscribers and Asterisk when the VoIP server was subjected to individual TCP DoS attacks—whether the connection broke down or the delay became too huge to be measured, etc.

Using the free Colasoft Capsa software, it was possible to monitor the traffic coming in/out of the Asterisk Free PBX network interface in the modelled network. The parameters monitored were the number of individual types of TCP packet and the amount of traffic processed by Asterisk. Monitoring the values of the individual types of TCP packet made it very easy to detect the occurrence of a TCP DoS attack, the occurrence of a successful TCP session, and other events related to this study. The amount of traffic processed showed the total traffic (voice/video traffic as well as TCP DoS flooding) that the VoIP server had processed. This information was used as a visual representation of how much information Asterisk had to process.

4. Results and Discussion

It should be mentioned that the presented research examines the impact of standard TCP DoS attacks on the performance of Asterisk Free PBX. The attacked device was Asterisk itself, which was considered as a network device, i.e., the VoIP server itself was attacked, not specific elements of it, as in scenarios other authors have examined in their work and research. The VoIP server was subjected to "standard/ordinary" DoS attacks without modification to the contents of the attack packets, as in scenarios some of the authors have examined in the other works mentioned above. Figure 1 presents the topology of the modeled IP network.

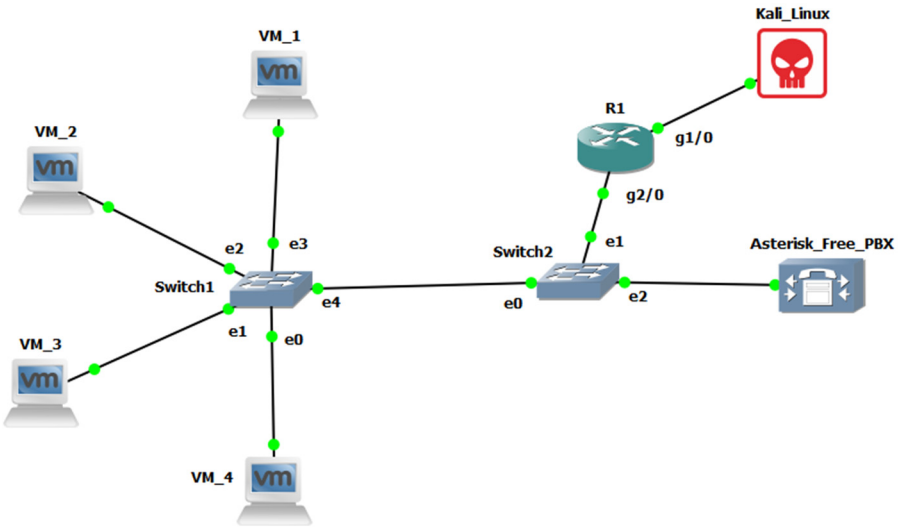


Figure 1. Topology of the modeled network.

The experimental network was composed of two switches (Switch1 and Switch2). E0 to E4 are the ports of the switches to which the VMs, Asterisk and router (R1) are connected. g/1/0 and g2/0 are the router ports to Kali Linux and Switch2 are connected. R1 is an emulated disk image of an operating system of a real router through which a connection to other networks was simulated. Four users (VM_1, VM_2, VM_3, and VM_4) represented four virtual machines with software phones installed on them. The virtual machines were created using hypervisor software. The IP PBX under study was an Asterisk Free PBX, which was also installed on a virtual machine using hypervisor software. The built-in firewall was enabled in the Asterisk settings. Kali Linux was deployed on some other, external network. Kali Linux itself was also installed on a virtual machine.

4.1. Results for Only Voice Streams

Figure 2 represents the number of different TCP packets sent during normal operation. The x-axis represents time, and the y-axis represents the number of TCP packets. As can be seen from the graphs in normal operation mode, when the system was not under attack, there were only a few TCP packets; the presented result is for when the Asterisk settings were accessed through a browser and the page was loaded. There was a successful TCP session. Prove of this statement is the presence of TCP FIN packets, which were sent when the TCP session was closed from one of the two sides.

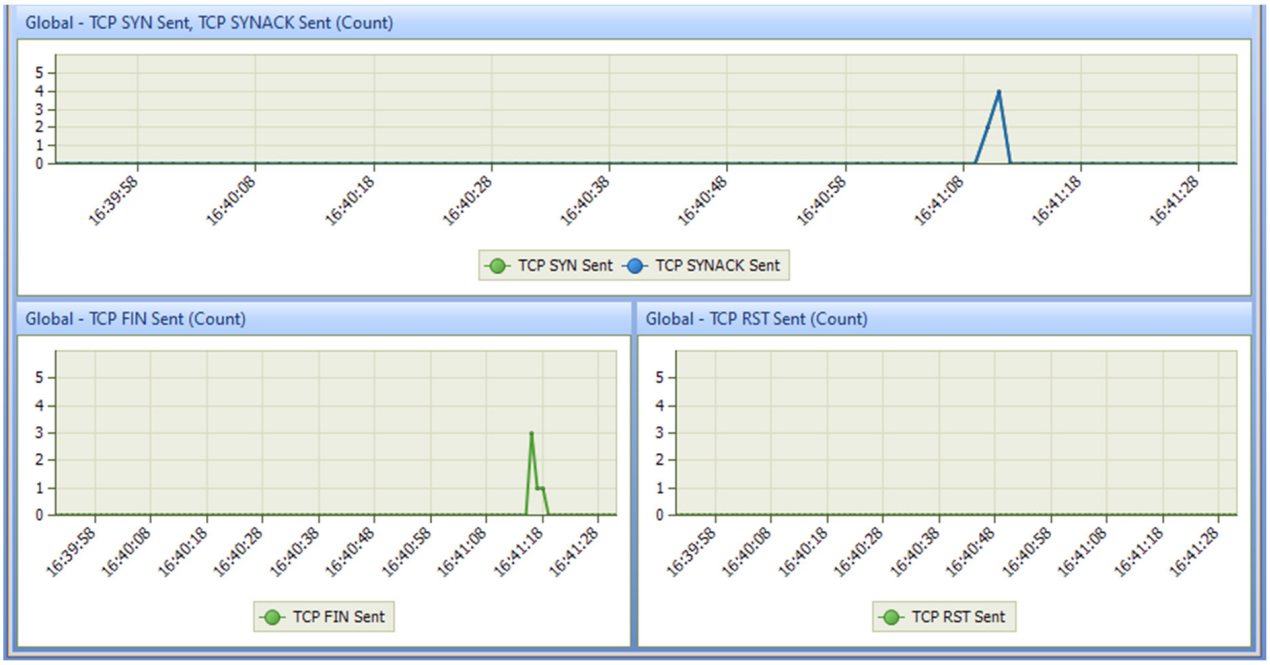


Figure 2. Number of different TCP packets sent during normal operation.

Figure 3 provides the summarized results of the voice traffic exchanged in both directions between VM_4 (10.10.10.2) and VM_3 (10.10.10.6). As can be seen from the results, the traffic was passed through and processed by Asterisk (10.10.10.5). For all subsequent similar figures, the designations are: **SSRC**—indicates the synchronizing source. **Max Delta** is the difference between the arrival of one packet and the arrival of the previous packet; **Max jitter** is the maximum measured jitter for the entire duration of the call; **Mean jitter** is the average value of the jitter for the entire duration of the call; **Max skew** is the maximum distortion; **RTP packets** is the number of RTP packets; **Lost** is the percentage of lost packets out of the total number of packets; **Seq error** indicates errors in TCP/UDP sessions; **Start at** indicates at which moment the corresponding studied voice/video stream starts; **Duration** is the duration of the studied voice/video stream; **Clock drift** shows what the deviation from the clock frequency is; **Freq Drift** shows what the deviation from the sample rate is. In this article, only **max jitter, mean jitter and packet lost** will be monitored. Figure 3a shows the summarized results for the voice stream exchanged between VM_4 and Asterisk, and Figure 3b shows the summarized results for the voice stream exchanged between VM_3 and Asterisk. In terms of the mean jitter, both streams were within the norm, where the maximum allowed value is 30 ms [55,56]. The maximum jitter values were also within the norm. No packet losses were observed.

Forward		Reverse	
10.10.10.2:17078 → 10.10.10.5:18134		10.10.10.5:18134 → 10.10.10.2:17078	
SSRC	0xe0802dcc	**SSRC**	0x5ba1538b
Max Delta	72.37 ms @ 279	**Max Delta**	59.39 ms @ 443
Max Jitter	5.84 ms	**Max Jitter**	6.74 ms
Mean Jitter	1.57 ms	**Mean Jitter**	0.95 ms
Max Skew	-32.13 ms	**Max Skew**	-39.03 ms
RTP Packets	27969	**RTP Packets**	28028
Expected	27969	**Expected**	28028
Lost	0 (0.00 %)	**Lost**	0 (0.00 %)
Seq Errs	0	**Seq Errs**	0
Start at	28.803606 s @ 176	**Start at**	27.629798 s @ 46
Duration	559.34 s	**Duration**	560.54 s
Clock Drift	-9 ms	**Clock Drift**	-0 ms
Freq Drift	8000 Hz (-0.00 %)	**Freq Drift**	8000 Hz (-0.00 %)

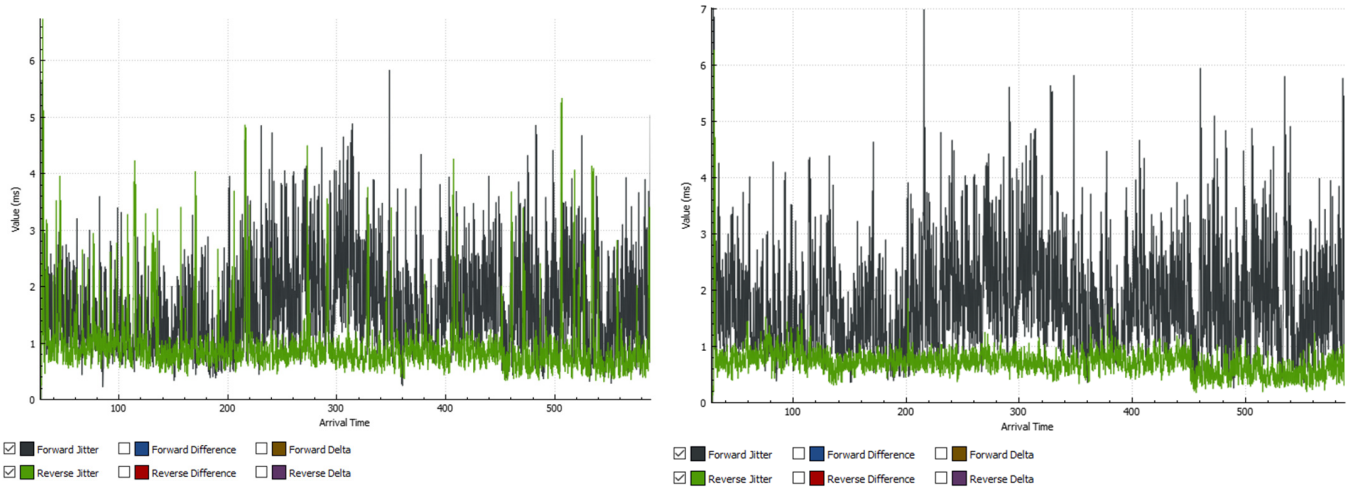
(**a**)

Forward		Reverse	
10.10.10.5:10676 → 10.10.10.6:7078		10.10.10.6:7078 → 10.10.10.5:10676	
SSRC	0x54928994	**SSRC**	0xfd6db222
Max Delta	68.87 ms @ 283	**Max Delta**	58.91 ms @ 441
Max Jitter	7.02 ms	**Max Jitter**	6.27 ms
Mean Jitter	1.74 ms	**Mean Jitter**	0.70 ms
Max Skew	-34.12 ms	**Max Skew**	-39.06 ms
RTP Packets	27969	**RTP Packets**	28029
Expected	27969	**Expected**	28029
Lost	0 (0.00 %)	**Lost**	0 (0.00 %)
Seq Errs	0	**Seq Errs**	0
Start at	28.804110 s @ 178	**Start at**	27.628795 s @ 45
Duration	559.34 s	**Duration**	560.56 s
Clock Drift	-9 ms	**Clock Drift**	-0 ms
Freq Drift	8000 Hz (-0.00 %)	**Freq Drift**	8000 Hz (-0.00 %)

(**b**)

Figure 3. Summarized results for the main parameters of the voice stream between VM_3 and the Asterisk (**a**) and between VM_4 and the Asterisk (**b**) during normal operation mode.

Figure 4a shows the variation in the jitter for the voice stream between VM_4 and Asterisk in both directions during the whole conversation time. Figure 4b shows the variation in the jitter for the voice stream between VM_3 and Asterisk in both directions. The x-axis (Arrival time) represents time, and the y-axis (value ms) represents the value of the jitter in ms. As can be seen from the graph, the values were very far from the maximum allowable levels. Parenthetically, it should be noted that in this type of simulation study, when real-time audio/video streams are studied, the computing capabilities of the workstation used for network modeling are of great importance. Computing capabilities affect the values of the jitter and the delay. If the computing capabilities are low, the jitter and the delay values increase. This finding is a result of many years of working with GNS3 and the use of different kinds of workstations for network modeling.



(**a**) (**b**)

Figure 4. Instantaneous values of the jitter for the voice stream between VM_4 and Asterisk (**a**) and between VM_3 and Asterisk (**b**) during normal operation mode.

4.1.1. Results from the TCP SYN Attack

During this attack, Asterisk was flooded with TCP SYN packets [57]. Figure 5 represents the number of different TCP packets. As can be seen from the graphs, the number of TCP SYN and the TCPSYNACK packets have increased many times, which is normal for this attack. Asterisk responds to the TCP requests with TCP SYNACK, even generating

TCP RST packets to terminate the problematic session. However, terminating the session was impossible. Asterisk is accessed through a browser. The proof for this statement is the presence of TCP FIN packets, which means that there was a successful TCP session that was terminated.

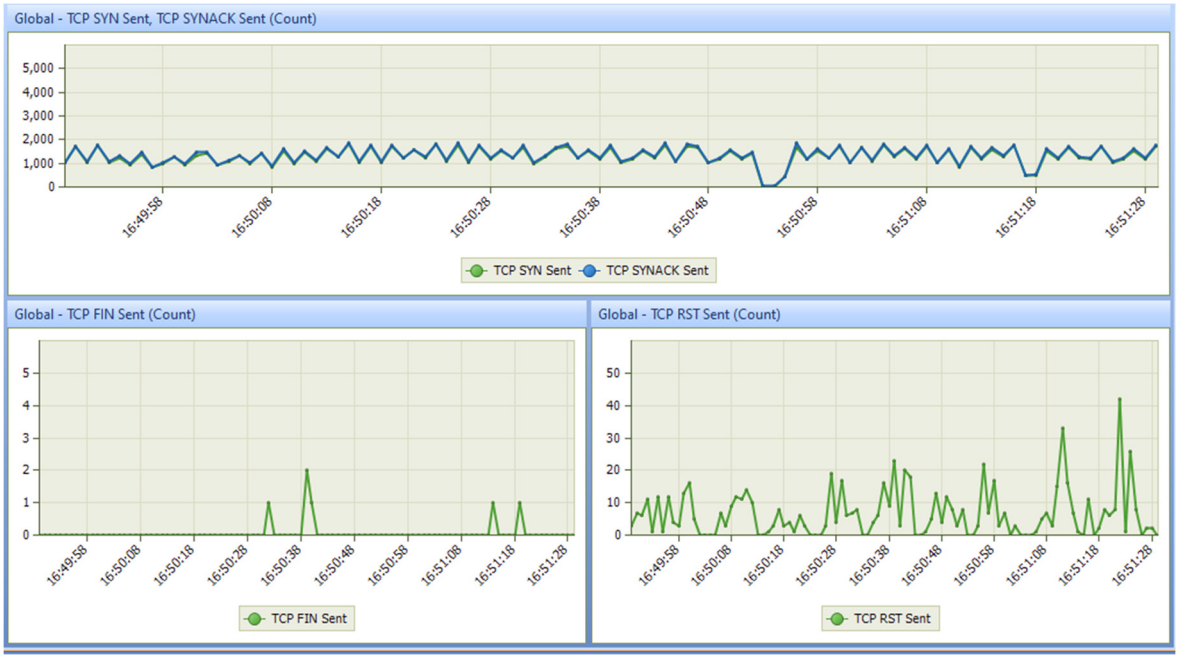


Figure 5. Number of different TCP packets sent during the TCP SYN attack.

Figure 6 presents the summarized results for the voice traffic that was exchanged in both directions between VM_4 and VM_3 during the attack. Figure 6a shows the summarized results for the voice stream that was exchanged between VM_4 and Asterisk, and Figure 6b shows the summarized results for the voice stream that was exchanged between VM_3 and Asterisk. An increase in the maximum jitter value due to the attack was observed. The average jitter values in both streams were elevated but were within the norm. There were no packet losses.

Forward		**Reverse**	
10.10.10.5:17362 → 10.10.10.2:17078		10.10.10.2:17078 → 10.10.10.5:17362	
SSRC	0x103729dc	**SSRC**	0xf3d8e56d
Max Delta	155.71 ms @ 64588	**Max Delta**	1985.74 ms @ 50091
Max Jitter	12.81 ms	**Max Jitter**	49.70 ms
Mean Jitter	1.19 ms	**Mean Jitter**	1.30 ms
Max Skew	-134.58 ms	**Max Skew**	-472.48 ms
RTP Packets	55891	**RTP Packets**	20650
Expected	55891	**Expected**	20650
Lost	0 (0.00 %)	**Lost**	0 (0.00 %)
Seq Errs	0	**Seq Errs**	0
Start at	668.168218 s @ 18486	**Start at**	668.434223 s @ 18525
Duration	1117.80 s	**Duration**	415.33 s
Clock Drift	0 ms	**Clock Drift**	-161 ms
Freq Drift	8000 Hz (0.00 %)	**Freq Drift**	7997 Hz (-0.04 %)

(**a**)

Forward		**Reverse**	
10.10.10.6:7078 → 10.10.10.5:18836		10.10.10.5:18836 → 10.10.10.6:7078	
SSRC	0xd2b9944b	**SSRC**	0x65734348
Max Delta	140.76 ms @ 64570	**Max Delta**	1973.28 ms @ 50092
Max Jitter	12.17 ms	**Max Jitter**	49.81 ms
Mean Jitter	0.87 ms	**Mean Jitter**	1.53 ms
Max Skew	-121.20 ms	**Max Skew**	-472.46 ms
RTP Packets	55892	**RTP Packets**	20650
Expected	55892	**Expected**	20650
Lost	0 (0.00 %)	**Lost**	0 (0.00 %)
Seq Errs	0	**Seq Errs**	0
Start at	668.167224 s @ 18485	**Start at**	668.435222 s @ 18526
Duration	1117.82 s	**Duration**	415.33 s
Clock Drift	0 ms	**Clock Drift**	-160 ms
Freq Drift	8000 Hz (0.00 %)	**Freq Drift**	7997 Hz (-0.04 %)

(**b**)

Figure 6. Summarized results for the main parameters of the voice stream between VM_3 and the Asterisk (**a**) and between VM_4 and the Asterisk (**b**) during the TCP SYN attack.

Figure 7a,b shows how the jitter values of the voice stream between VM_4/VM_3 and Asterisk changed in both directions. As can be seen from the graphs, excluding

individual spikes, the jitter varied within the acceptable range, but the values were still higher compared to the time when Asterisk was not being attacked.

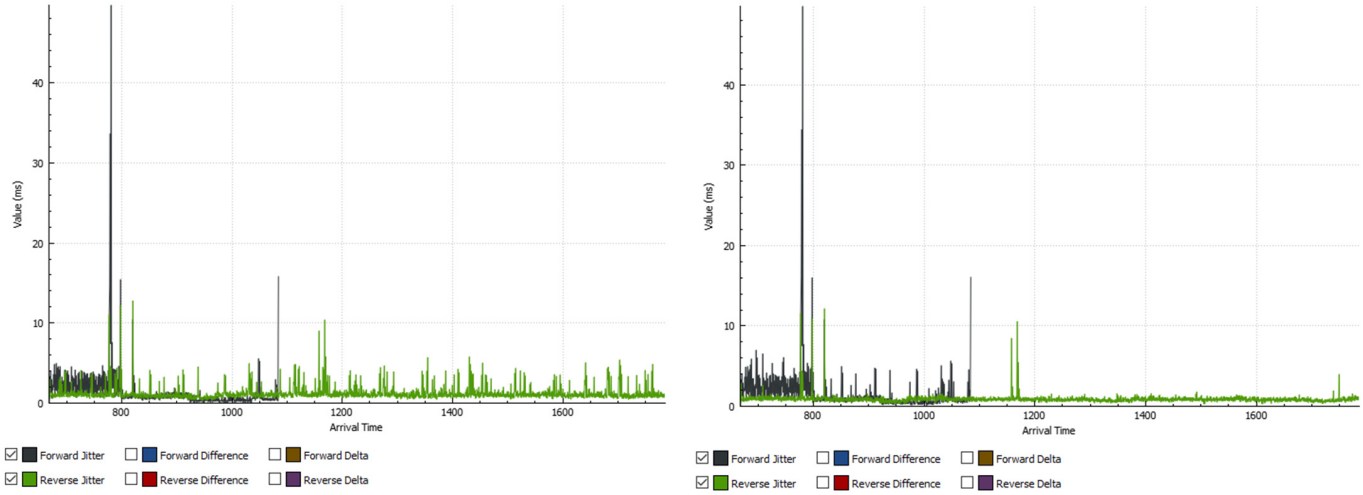


Figure 7. Instantaneous values of the jitter for the voice stream between VM_4 and Asterisk (**a**) and VM_3 and Asterisk (**b**) during the TCP SYN attack.

4.1.2. Results during the TCP ACK Attack

During this attack, Asterisk was flooded with TCP ACK packets [58]. Figure 8 represents the number of different TCP packets. As can be seen from the graphs, the number of TCP SYN and the TCPSYNACK packets is almost zero. There were only a few of these packets and of TCP FIN packets because Asterisk was successfully accessed through a browser and the session was terminated. A huge number of TCP RST packets was noticed. These were generated by the Asterisk web server to terminate the problematic session that was created by the TCP ACK attack. Termination of the session (attack) was impossible.

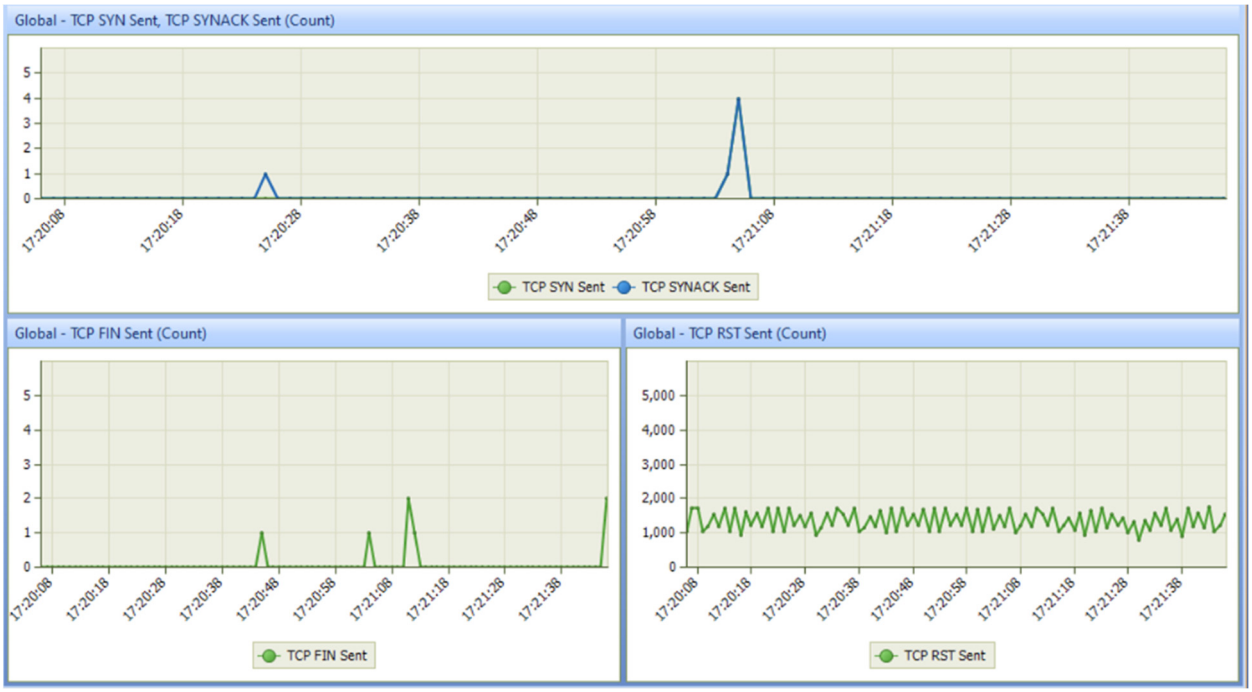
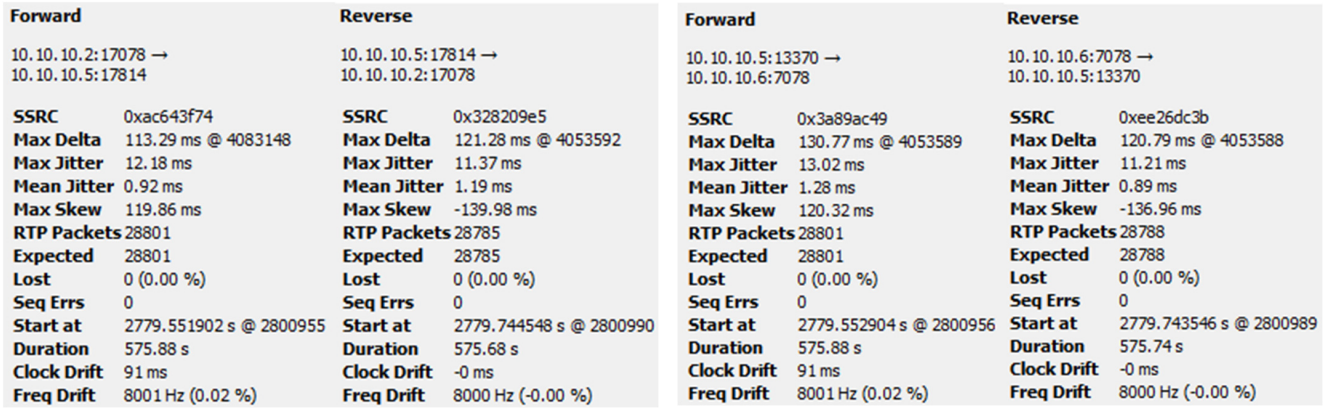


Figure 8. Number of different TCP packets sent during the TCP ACK attack.

Figure 9 presents the summarized results for voice traffic that was exchanged in both directions between VM_4 and VM_3 during the attack. Figure 9a shows the summarized results for the voice stream that was exchanged between VM_4 and Asterisk, and Figure 9b shows the summarized results for the voice stream that was exchanged between VM_3 and Asterisk. There was a decrease in the maximum jitter value and the average jitter value for both streams. Compared to the results from when Asterisk was not being attacked, the maximum jitter values were still higher. The average jitter values were almost identical to those in Figure 4. In spite of the occurrence of the TCP ACK attack, the values were within the norm. Again, there was no packet losses.



Forward		Reverse	
10.10.10.2:17078 → 10.10.10.5:17814		10.10.10.5:17814 → 10.10.10.2:17078	
SSRC	0xac643f74	**SSRC**	0x328209e5
Max Delta	113.29 ms @ 4083148	**Max Delta**	121.28 ms @ 4053592
Max Jitter	12.18 ms	**Max Jitter**	11.37 ms
Mean Jitter	0.92 ms	**Mean Jitter**	1.19 ms
Max Skew	119.86 ms	**Max Skew**	-139.98 ms
RTP Packets	28801	**RTP Packets**	28785
Expected	28801	**Expected**	28785
Lost	0 (0.00 %)	**Lost**	0 (0.00 %)
Seq Errs	0	**Seq Errs**	0
Start at	2779.551902 s @ 2800955	**Start at**	2779.744548 s @ 2800990
Duration	575.88 s	**Duration**	575.68 s
Clock Drift	91 ms	**Clock Drift**	-0 ms
Freq Drift	8001 Hz (0.02 %)	**Freq Drift**	8000 Hz (-0.00 %)

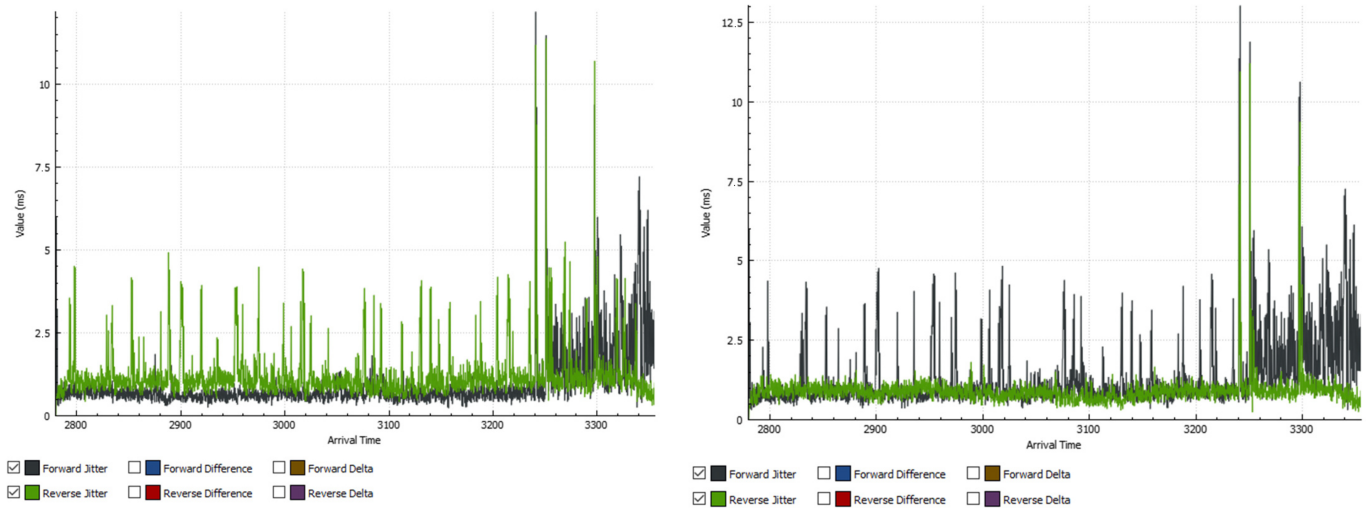
(**a**)

Forward		Reverse	
10.10.10.5:13370 → 10.10.10.6:7078		10.10.10.6:7078 → 10.10.10.5:13370	
SSRC	0x3a89ac49	**SSRC**	0xee26dc3b
Max Delta	130.77 ms @ 4053589	**Max Delta**	120.79 ms @ 4053588
Max Jitter	13.02 ms	**Max Jitter**	11.21 ms
Mean Jitter	1.28 ms	**Mean Jitter**	0.89 ms
Max Skew	120.32 ms	**Max Skew**	-136.96 ms
RTP Packets	28801	**RTP Packets**	28788
Expected	28801	**Expected**	28788
Lost	0 (0.00 %)	**Lost**	0 (0.00 %)
Seq Errs	0	**Seq Errs**	0
Start at	2779.552904 s @ 2800956	**Start at**	2779.743546 s @ 2800989
Duration	575.88 s	**Duration**	575.74 s
Clock Drift	91 ms	**Clock Drift**	-0 ms
Freq Drift	8001 Hz (0.02 %)	**Freq Drift**	8000 Hz (-0.00 %)

(**b**)

Figure 9. Summarized results for the main parameters of the voice stream between VM_3 and the Asterisk (**a**) and between VM_4 and the Asterisk (**b**) during the TCP ACK attack.

Figure 10a shows how the jitter changed for the voice stream between VM_4 and Asterisk in both directions. Figure 10b presents the variation in jitter values for the voice stream between VM_3 and Asterisk in both directions. As can be seen from the two graphs, the jitter values were lower than those obtained during the TCP SYN attack.



(**a**) (**b**)

Figure 10. Instantaneous values of the jitter for the voice stream between VM_4 and Asterisk (**a**) and VM_3 and Asterisk (**b**) during the TCP ACK attack.

4.1.3. Results from the TCP RST Attack

During this attack, Asterisk was flooded with TCP RST packets [59]. Figure 11 represents the number of different TCP packets. As can be seen from the graphs, there were

no TCP SYN and the TCPSYNACK packets because at that time Asterisk was not being accessed. There were a huge number of TCP RST packets because the Asterisk web server was trying to terminate the problematic session that was created by the TCP RST attack. Terminating the session (attack) was impossible. Asterisk was successfully accessed through a browser.

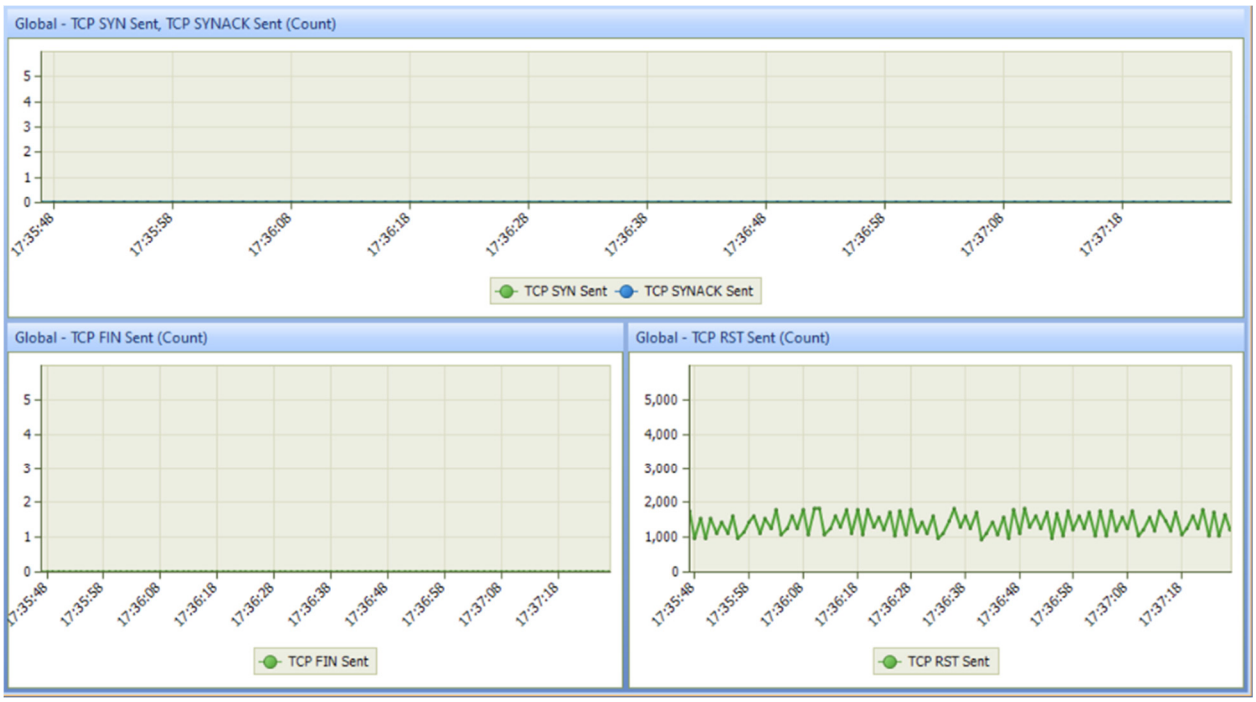


Figure 11. Number of different TCP packets sent during the TCP RST attack.

Figure 12 presents the summarized results for the voice traffic that was exchanged in both directions between VM_4 and VM_3 during the attack. Figure 12a shows the summarized results for the voice stream that was exchanged between VM_4 and Asterisk, and Figure 12b shows the summarized results for the voice stream that was exchanged between VM_3 and Asterisk. It can be noticed that the jitter values were close to the results that were obtained when Asterisk was not being attacked. Compared to the results of the previous two attacks, a decrease in the jitter values was observed. There were no packet losses.

Forward		Reverse	
10.10.10.2:17078 → 10.10.10.5:18044		10.10.10.5:18044 → 10.10.10.2:17078	
SSRC	0xf6dff38a	**SSRC**	0x2eceebd1
Max Delta	49.91 ms @ 4457335	**Max Delta**	47.40 ms @ 4376427
Max Jitter	5.08 ms	**Max Jitter**	4.95 ms
Mean Jitter	0.98 ms	**Mean Jitter**	1.08 ms
Max Skew	-26.51 ms	**Max Skew**	-27.40 ms
RTP Packets	26782	**RTP Packets**	26797
Expected	26782	**Expected**	26797
Lost	0 (0.00 %)	**Lost**	0 (0.00 %)
Seq Errs	0	**Seq Errs**	0
Start at	3731.810552 s @ 4374413	**Start at**	3731.512619 s @ 4374372
Duration	535.62 s	**Duration**	535.92 s
Clock Drift	0 ms	**Clock Drift**	-0 ms
Freq Drift	8000 Hz (0.00 %)	**Freq Drift**	8000 Hz (-0.00 %)

(**a**)

Forward		Reverse	
10.10.10.6:7078 → 10.10.10.5:19964		10.10.10.5:19964 → 10.10.10.6:7078	
SSRC	0x87448940	**SSRC**	0x49834cb8
Max Delta	28.45 ms @ 4445842	**Max Delta**	49.91 ms @ 4457339
Max Jitter	2.11 ms	**Max Jitter**	5.17 ms
Mean Jitter	0.87 ms	**Mean Jitter**	1.18 ms
Max Skew	-8.52 ms	**Max Skew**	-27.43 ms
RTP Packets	26799	**RTP Packets**	26782
Expected	26799	**Expected**	26782
Lost	0 (0.00 %)	**Lost**	0 (0.00 %)
Seq Errs	0	**Seq Errs**	0
Start at	3731.511605 s @ 4374371	**Start at**	3731.813063 s @ 4374415
Duration	535.96 s	**Duration**	535.62 s
Clock Drift	-0 ms	**Clock Drift**	0 ms
Freq Drift	8000 Hz (-0.00 %)	**Freq Drift**	8000 Hz (0.00 %)

(**b**)

Figure 12. Summarized results for the main parameters of the voice stream between VM_3 and the Asterisk (**a**) and between VM_4 and the Asterisk (**b**) during the TCP RST attack.

Figure 13a shows how the jitter changed for the voice stream between VM_4 and Asterisk in both directions. Figure 13b shows the variation in jitter values for the voice stream between VM_3 and Asterisk in both directions. The result was similar to the result obtained when Asterisk was not being attacked.

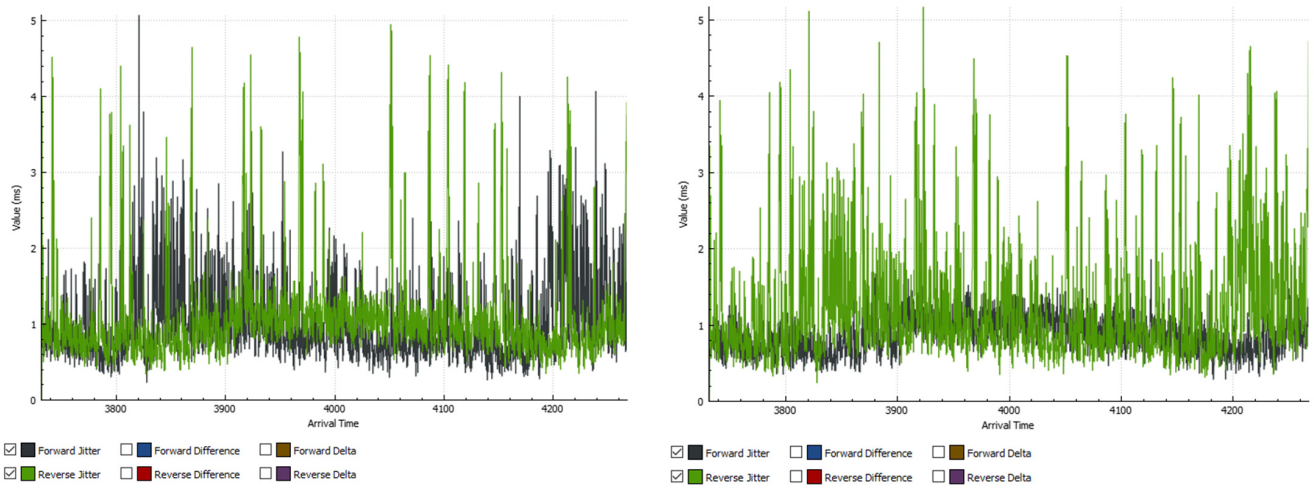


Figure 13. Instantaneous values of the jitter for the voice stream between VM_4 and Asterisk (**a**) and VM_3 and Asterisk (**b**) during the TCP RST attack.

4.1.4. Results during the TCP FIN Attack

During this attack, Asterisk was flooded with TCP FIN packets [60]. Figure 14 represents the number of different TCP packets. As can be seen from the graphs, again, there were no TCP SYN and the TCPSYNACK packets because Asterisk was not being accessed at that moment. There is a huge number of TCP FIN packets because the Asterisk web server generated them as the other side was also sending TCP FIN packets. The Asterisk web server generated these TCP FIN packets because it assumed that the TCP FIN packets received from the other side were generated to terminate the TCP session. Asterisk was successfully accessed through a browser.

Figure 14. Number of different TCP packets sent during the TCP FIN attack.

Figure 15 presents the summarized results for the voice traffic that was exchanged in both directions between VM_4 and VM_3 during the attack. Figure 15a shows the summarized results for the voice stream that was exchanged between VM_4 and Asterisk, and Figure 15b shows the summarized results for the voice stream that was exchanged between VM_3 and Asterisk. As can be seen, the jitter values were close to the results from when Asterisk was not being attacked. There was no packet loss.

Forward		Reverse	
10.10.10.2:17078 → 10.10.10.5:14972		10.10.10.5:14972 → 10.10.10.2:17078	
SSRC	0x0af619ff	**SSRC**	0x3d3cc42c
Max Delta	30.98 ms @ 5260616	**Max Delta**	30.97 ms @ 5268370
Max Jitter	1.56 ms	**Max Jitter**	4.21 ms
Mean Jitter	0.82 ms	**Mean Jitter**	1.12 ms
Max Skew	-11.66 ms	**Max Skew**	-12.07 ms
RTP Packets	1021	**RTP Packets**	1035
Expected	1021	**Expected**	1035
Lost	0 (0.00 %)	**Lost**	0 (0.00 %)
Seq Errs	0	**Seq Errs**	0
Start at	4854.367744 s @ 5260424	**Start at**	4854.068800 s @ 5260321
Duration	20.40 s	**Duration**	20.68 s
Clock Drift	-0 ms	**Clock Drift**	-0 ms
Freq Drift	8000 Hz (-0.00 %)	**Freq Drift**	8000 Hz (-0.00 %)

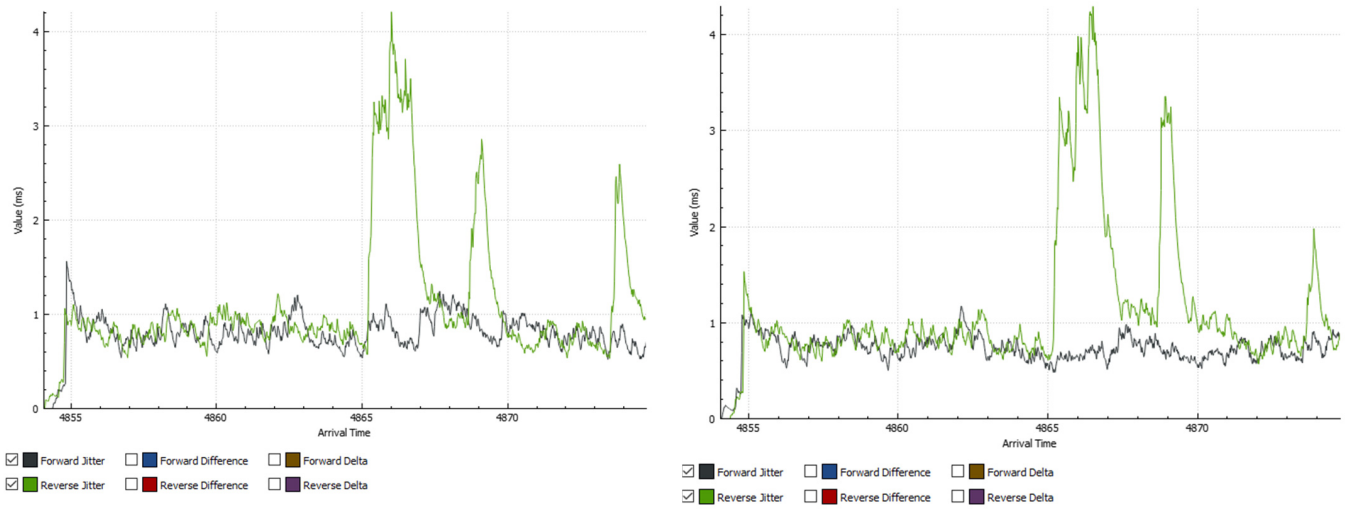
(**a**)

Forward		Reverse	
10.10.10.6:7078 → 10.10.10.5:19676		10.10.10.5:19676 → 10.10.10.6:7078	
SSRC	0x4133e423	**SSRC**	0x073ffdbb
Max Delta	26.94 ms @ 5260589	**Max Delta**	31.95 ms @ 5265384
Max Jitter	1.17 ms	**Max Jitter**	4.29 ms
Mean Jitter	0.74 ms	**Mean Jitter**	1.16 ms
Max Skew	-5.68 ms	**Max Skew**	-13.72 ms
RTP Packets	1036	**RTP Packets**	1021
Expected	1036	**Expected**	1021
Lost	0 (0.00 %)	**Lost**	0 (0.00 %)
Seq Errs	0	**Seq Errs**	0
Start at	4854.068340 s @ 5260320	**Start at**	4854.368773 s @ 5260425
Duration	20.70 s	**Duration**	20.40 s
Clock Drift	0 ms	**Clock Drift**	-1 ms
Freq Drift	8000 Hz (0.00 %)	**Freq Drift**	8000 Hz (-0.00 %)

(**b**)

Figure 15. Summarized results for the main parameters of the voice stream between VM_3 and the Asterisk (**a**) and between VM_4 and the Asterisk (**b**) during the TCP FIN attack.

Figure 16a,b shows the variation in jitter for the voice streams between VM_4/VM_3 and Asterisk in both directions.



(**a**) (**b**)

Figure 16. Instantaneous values of the jitter for the voice stream between VM_4 and Asterisk (**a**) and VM_3 between and Asterisk (**b**) during the TCP FIN attack.

4.1.5. Summarized Results for the Voice-stream study

Figure 17 (green area) represents the total traffic that Asterisk had to process during all attacks. The peaks represent the periods during which Asterisk was being attacked. As can be seen, the amount of traffic processed during each of the attacks was huge.

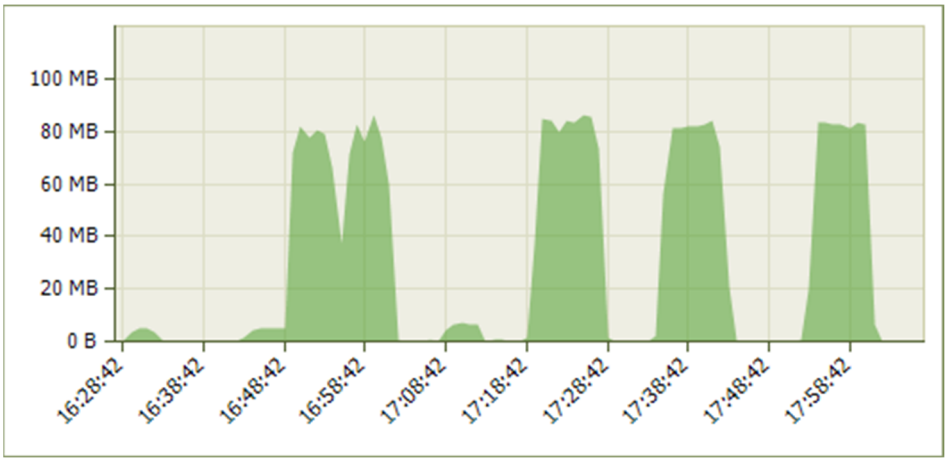


Figure 17. Traffic processed by Asterisk during the whole study period.

Figure 18 presents the number of different TCP packets for the whole study period. As can be seen from the graphs, TCP SYN and the TCPSYNACK packets were present during the TCP SYN attack. TCP FIN was present only during the TCP FIN attack. TCP RST packets were generated during all attacks except the TCP FIN attack.

Figure 18. Number of different TCP packets for the whole study period during the voice-stream study.

The following results show how the round-trip delay (RTD) between the different VMs and Asterisk changed during the whole study period. The results were obtained using the Colasoft Ping Tool (version 2.0). The purpose of these graphs is to show if and how the different TCP DoS attacks affected accessibility to Asterisk. The x-axis represents time, and the y-axis represents the value of the delay in ms. Figure 19 shows how the RTD changed between VM_1 and Asterisk for the whole study period. As can be seen from the graph, the values of the delay were far below the value of 150 ms in one direction.

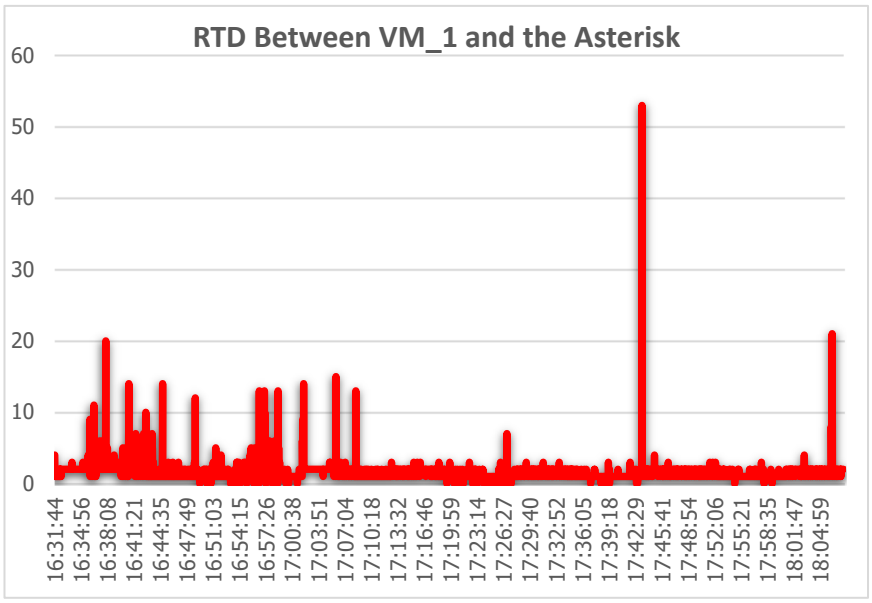


Figure 19. RTD between VM_1 and Asterisk.

Figure 20 presents how the RTD changed between VM_2 and Asterisk for the whole study period.

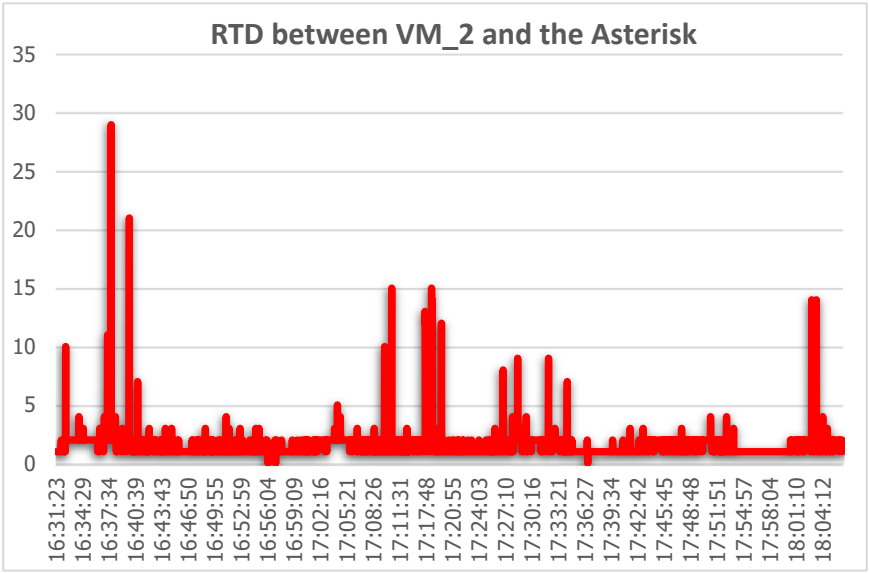


Figure 20. RTD between VM_2 and Asterisk.

Figure 21 presents how the RTD changed between VM_3 and Asterisk for the whole study period. Figure 22 presents how the RTD changed between VM_4 and Asterisk for the whole study period.

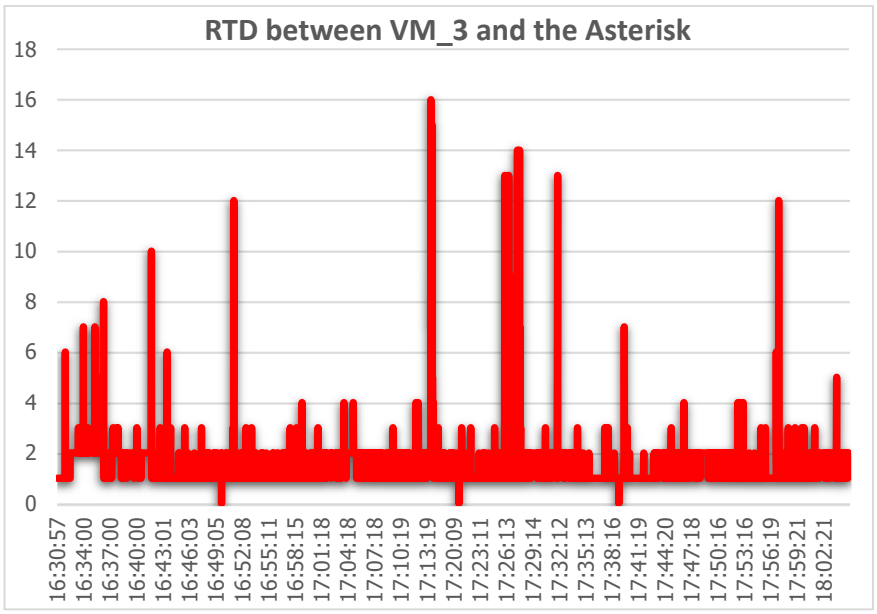


Figure 21. RTD between VM_3 and Asterisk.

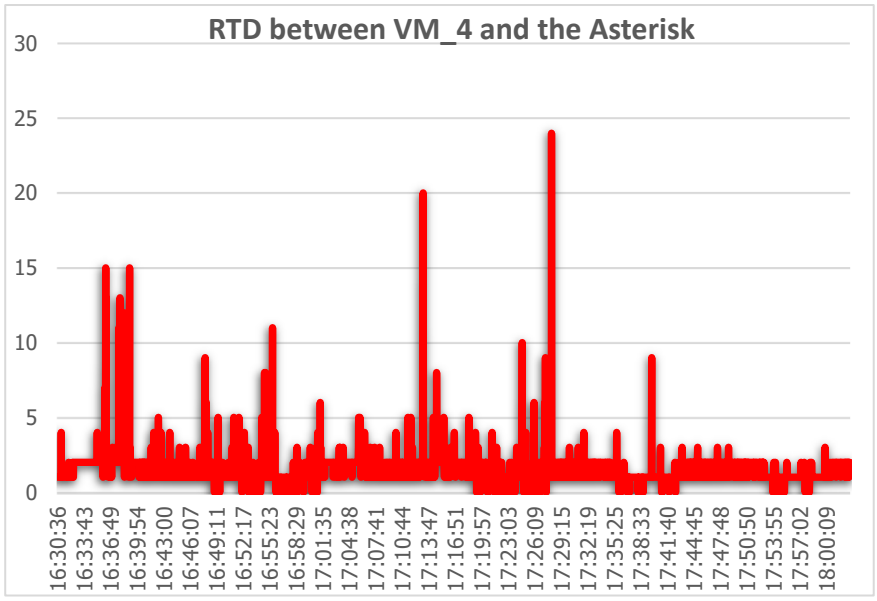


Figure 22. RTD between VM_4 and Asterisk.

As can be seen from the four graphs, the round-trip delay varied but did not reach even half of the maximum allowable value of 150 ms per direction.

4.1.6. Discussion of the Obtained Results for the Voice-Stream study

The obtained results showed something surprising. During all attacks, Asterisk could be accessed through a browser. This was proven by the results for the numbers of different TCP packets. Even during the TCP SYN attack, which is the most burdensome DoS attack, Asterisk was accessible. In terms of the aim of this work, the study showed that the different TCP DoS attacks did not have much impact on either the parameters of the voice flows or on the flows themselves. No call dropping was observed during the study. From the obtained results, it was found that during the TCP SYN attack, the value of the jitter increased, compared to the control result. This increase, however, was very far from the maximum allowed average jitter value of 30 ms. In the other three attacks, the jitter value was close to the jitter value in the control measurement.

Measurement of the round-trip delay between Asterisk and each of the other subscribers showed that the individual TCP DoS attacks had no impact on Asterisk's performance. The expectation was that during the attacks, pings to the VoIP server should have been impossible—the expected results were either "Destination unreachable" or huge delay values. As can be seen from the graphs, there were no lost measurement packets or huge latency values. On the contrary, the RTD values were very far from the allowable delay value of 150 ms per direction.

There were no packets lost because the root causes of packet loss, high jitter levels and high network delays, were not present. As can be seen from the presented graphs, the values of both parameters (jitter and delay) were low.

It should be mentioned that during the measurement, the Asterisk "Responsive Firewall" feature was enabled. Because of this functionality, the effects of the various DoS attacks were neutralized. This functionality is powerful, and the results show that the "Responsive Firewall" protects the system very well from the various TCP DoS attacks, as they do not affect either the parameters of the voice streams or the voice streams themselves

The single spikes with large values above 20 ms that can be observed in the graphs showing the variation in jitter values, as well as the single spikes in the round-trip-delay measurement, are due to an increase in the delay in the modeled network. This increased delay was due to moments of high computational load on the workstation used for modeling the network and all processes in it. Such spikes are always observed in this type of research and real-time traffic modeling; the goal is for there to be as few of these spikes as possible and for them to be of small amplitude.

4.2. Results for Video Streams Only

Figure 23 represents the number of different TCP packets sent during normal operation. As can be seen from the graphs, in normal operation mode, when the system was not under attack, there were only a few TCP packets. The presented result is for a moment when the Asterisk configuration pages were accessed through a browser.

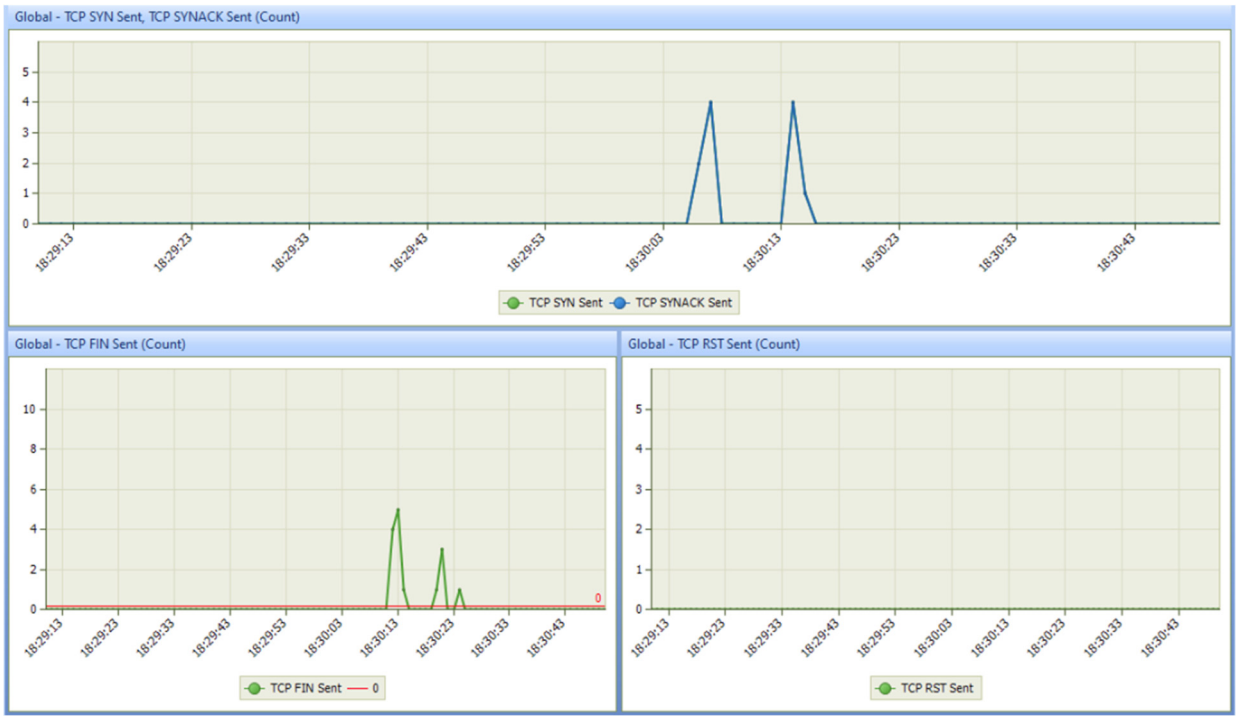


Figure 23. Number of different TCP packets sent during normal operation for a video conversation.

Figure 24 presents the summarized results for the video traffic that was exchanged in both directions between VM_4 and VM_3 during normal operation mode. During the

video calls, the video streams again passed (were processed) through Asterisk (10.10.10.5) rather than directly between the virtual users. Figure 24a shows the summarized results for the video stream that was exchanged between VM_4 and Asterisk, and Figure 24b shows the summarized results for the video stream that was exchanged between VM_3 and Asterisk. As can be seen, the maximum jitter value and its mean value are different compared to the same measurements for the voice-streams study. Both values were far from the maximum allowable levels. There was no packet loss.

Forward		Reverse	
10.10.10.2:19078 → 10.10.10.5:19942		10.10.10.5:19942 → 10.10.10.2:19078	
SSRC	0x88546735	**SSRC**	0x12f255a4
Max Delta	239.02 ms @ 22006	**Max Delta**	458.65 ms @ 81889
Max Jitter	2.15 ms	**Max Jitter**	17.12 ms
Mean Jitter	0.71 ms	**Mean Jitter**	0.98 ms
Max Skew	-16.53 ms	**Max Skew**	-167.45 ms
RTP Packets	21656	**RTP Packets**	22498
Expected	21656	**Expected**	22498
Lost	0 (0.00 %)	**Lost**	0 (0.00 %)
Seq Errs	0	**Seq Errs**	0
Start at	19.574592 s @ 171	**Start at**	19.793680 s @ 261
Duration	322.56 s	**Duration**	322.35 s
Clock Drift	0 ms	**Clock Drift**	-3 ms
Freq Drift	90000 Hz (0.00 %)	**Freq Drift**	89999 Hz (-0.00 %)

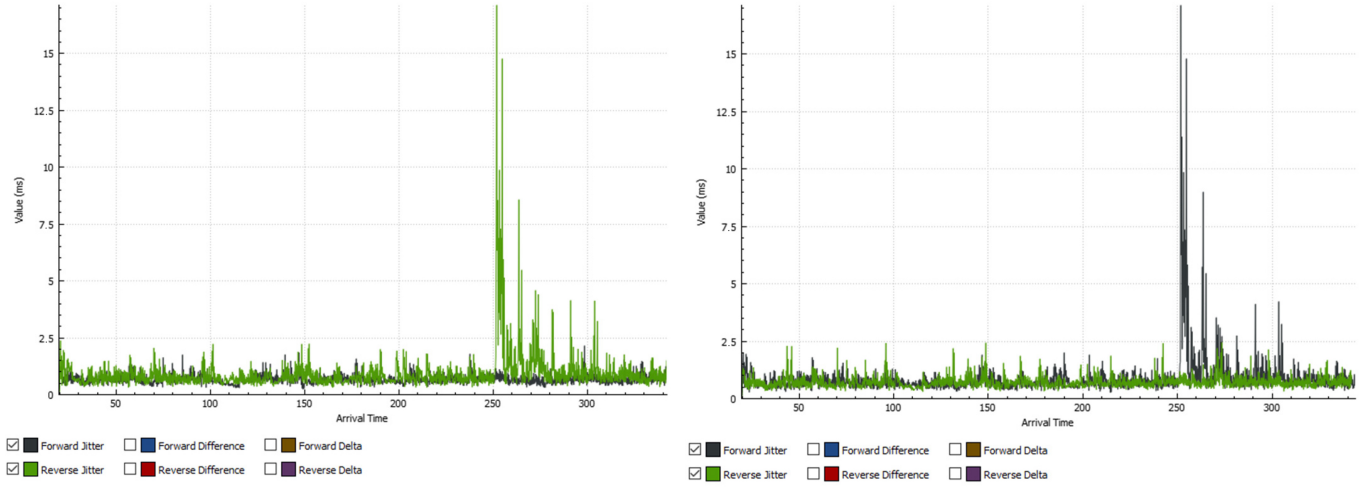
(**a**)

Forward		Reverse	
10.10.10.6:9078 → 10.10.10.5:12426		10.10.10.5:12426 → 10.10.10.6:9078	
SSRC	0x95322440	**SSRC**	0x74352a2d
Max Delta	458.66 ms @ 81886	**Max Delta**	239.06 ms @ 22008
Max Jitter	17.12 ms	**Max Jitter**	2.44 ms
Mean Jitter	0.90 ms	**Mean Jitter**	0.78 ms
Max Skew	-172.44 ms	**Max Skew**	-17.03 ms
RTP Packets	22647	**RTP Packets**	21656
Expected	22647	**Expected**	21656
Lost	0 (0.00 %)	**Lost**	0 (0.00 %)
Seq Errs	0	**Seq Errs**	0
Start at	19.788191 s @ 232	**Start at**	19.575587 s @ 174
Duration	324.35 s	**Duration**	322.56 s
Clock Drift	-3 ms	**Clock Drift**	-0 ms
Freq Drift	89999 Hz (-0.00 %)	**Freq Drift**	90000 Hz (-0.00 %)

(**b**)

Figure 24. Summarized results for the main parameters of the video stream between VM_3 and the Asterisk (**a**) and between VM_4 and the Asterisk (**b**) during normal operation.

Figure 25a,b shows the variation in jitter values for the video stream between VM_4/VM_3 and Asterisk in both directions. The levels of the parameter were low, far from the maximal permissible limits. There were several huge spikes due to computational errors. These were normal jitter graphs.



(**a**) (**b**)

Figure 25. Instantaneous values of the jitter for the video stream between VM_4 and Asterisk (**a**) and VM_3 and Asterisk (**b**) during normal operation mode.

4.2.1. Results during the TCP SYN Attack

Figure 26 represents the number of different TCP packets sent during the attack. As can be seen from the graphs, the numbers of TCP SYN and the TCPSYNACK packets increased, which is normal for this attack. Asterisk responded to the TCP requests with TCP SYNACK; it even generated TCP RST packets to terminate the problematic session

(attack), but these TCP RST packets could not terminate the problematic session. Asterisk was again accessible through a browser.

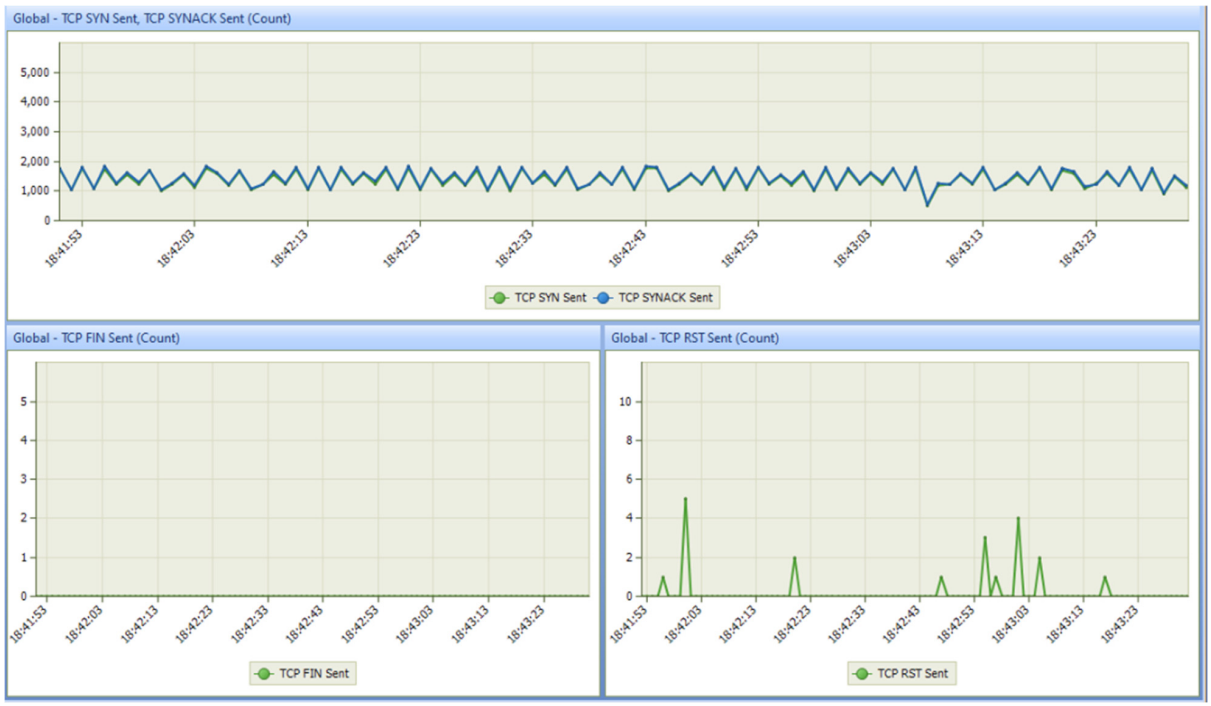


Figure 26. Number of different TCP packets sent during the TCP SYN attack for the video conversation.

Figure 27 presents the summarized results for the video traffic that was exchanged in both directions between VM_4 and VM_3 during the attack. Figure 27a shows the summarized results for the video stream that was exchanged between VM_4 and Asterisk, and Figure 27b shows the summarized results for the video stream that was exchanged between VM_3 and Asterisk. As can be seen, the maximum jitter value for both streams increased significantly. Due to this increase in jitter, there were packet losses in both streams, but just a few, only 0.01% of the total traffic. The maximum allowed packet loss for this kind of traffic is 1%.

Forward		Reverse	
10.10.10.2:19078 → 10.10.10.5:11280		10.10.10.5:11280 → 10.10.10.2:19078	
SSRC	0x5f2e39b7	**SSRC**	0x640e028d
Max Delta	386.79 ms @ 673566	**Max Delta**	418.73 ms @ 465947
Max Jitter	27.57 ms	**Max Jitter**	18.53 ms
Mean Jitter	0.58 ms	**Mean Jitter**	0.83 ms
Max Skew	-257.99 ms	**Max Skew**	-147.21 ms
RTP Packets	38569	**RTP Packets**	35059
Expected	38571	**Expected**	35062
Lost	2 (0.01 %)	**Lost**	3 (0.01 %)
Seq Errs	2	**Seq Errs**	1
Start at	580.420448 s @ 127248	**Start at**	580.674471 s @ 127354
Duration	519.93 s	**Duration**	519.67 s
Clock Drift	1 ms	**Clock Drift**	-1 ms
Freq Drift	90000 Hz (0.00 %)	**Freq Drift**	90000 Hz (-0.00 %)

(**a**)

Forward		Reverse	
10.10.10.5:15194 → 10.10.10.6:9078		10.10.10.6:9078 → 10.10.10.5:15194	
SSRC	0x14780871	**SSRC**	0x44fff0d6
Max Delta	386.28 ms @ 673571	**Max Delta**	420.73 ms @ 465936
Max Jitter	27.60 ms	**Max Jitter**	18.45 ms
Mean Jitter	0.68 ms	**Mean Jitter**	0.68 ms
Max Skew	-260.51 ms	**Max Skew**	-147.19 ms
RTP Packets	38569	**RTP Packets**	35207
Expected	38571	**Expected**	35210
Lost	2 (0.01 %)	**Lost**	3 (0.01 %)
Seq Errs	2	**Seq Errs**	1
Start at	580.421420 s @ 127251	**Start at**	580.673491 s @ 127350
Duration	519.93 s	**Duration**	521.76 s
Clock Drift	1 ms	**Clock Drift**	-0 ms
Freq Drift	90000 Hz (0.00 %)	**Freq Drift**	90000 Hz (-0.00 %)

(**b**)

Figure 27. Summarized results for the main parameters of the video stream between VM_3 and the Asterisk (**a**) and between VM_4 and the Asterisk (**b**) during the TCP SYN attack.

Figure 28a represents the variation in jitter values for the video stream between VM_4 and Asterisk in both directions. Figure 28b shows the variation in jitter values for the video

stream between VM_3 and Asterisk in both directions. As can be seen from the graph, there were many spikes because of computational errors, but nevertheless, the values were very far from the maximum acceptable levels.

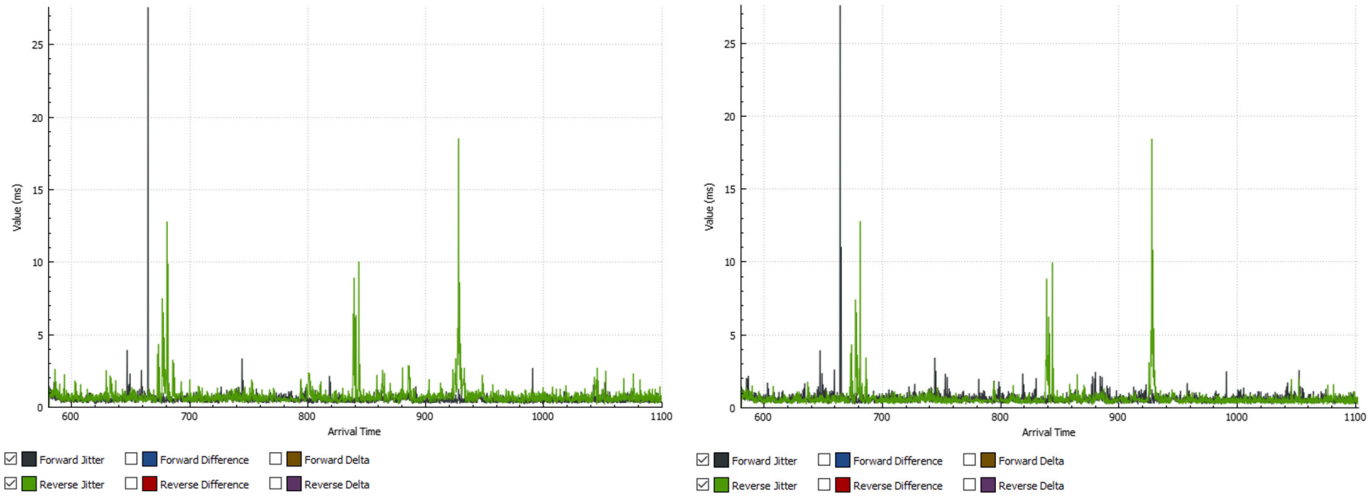


Figure 28. Instantaneous values of the jitter for the video stream between VM_4 and Asterisk (**a**) and VM_3 and Asterisk (**b**) during the TCP SYN attack.

4.2.2. Results during the TCP ACK Attack

Figure 29 represents the number of different TCP packets. As can be seen from the graphs, there were no TCP SYN or TCPSYNACK packets because the VoIP server was not being accessed at this moment. Only TCP RST packets were generated because the web server of Asterisk had detected a problematic TCP session and, through these packets, it tried to terminate this session. Asterisk was accessible through a browser.

Figure 29. Number of different TCP packets sent during the TCP ACK attack for the video conversation.

Figure 30 presents the summarized results for the video stream that was exchanged in both directions between VM_4 and VM_3 during the attack. Figure 30a shows the summarized results for the video stream that was exchanged between VM_4 and Asterisk, and Figure 30b shows the summarized results for the video stream that was exchanged between VM_3 and Asterisk. During this attack, the jilter values decreased because this was a TCP ACK attack, which is not highly burdensome, unlike a TCP SYN attack. As a result, there were no packet losses.

Forward		Reverse	
10.10.10.2:19078 → 10.10.10.5:10014		10.10.10.5:10014 → 10.10.10.2:19078	
SSRC	0x7044d626	**SSRC**	0x7784f0ed
Max Delta	331.38 ms @ 1926681	**Max Delta**	1023.60 ms @ 1386343
Max Jitter	2.71 ms	**Max Jitter**	2.83 ms
Mean Jitter	0.54 ms	**Mean Jitter**	0.79 ms
Max Skew	16.44 ms	**Max Skew**	-21.54 ms
RTP Packets	44000	**RTP Packets**	39266
Expected	44000	**Expected**	39266
Lost	0 (0.00 %)	**Lost**	0 (0.00 %)
Seq Errs	0	**Seq Errs**	0
Start at	1471.218199 s @ 1386251	**Start at**	1470.476079 s @ 1386062
Duration	587.40 s	**Duration**	588.24 s
Clock Drift	-1 ms	**Clock Drift**	2 ms
Freq Drift	90000 Hz (-0.00 %)	**Freq Drift**	90000 Hz (0.00 %)

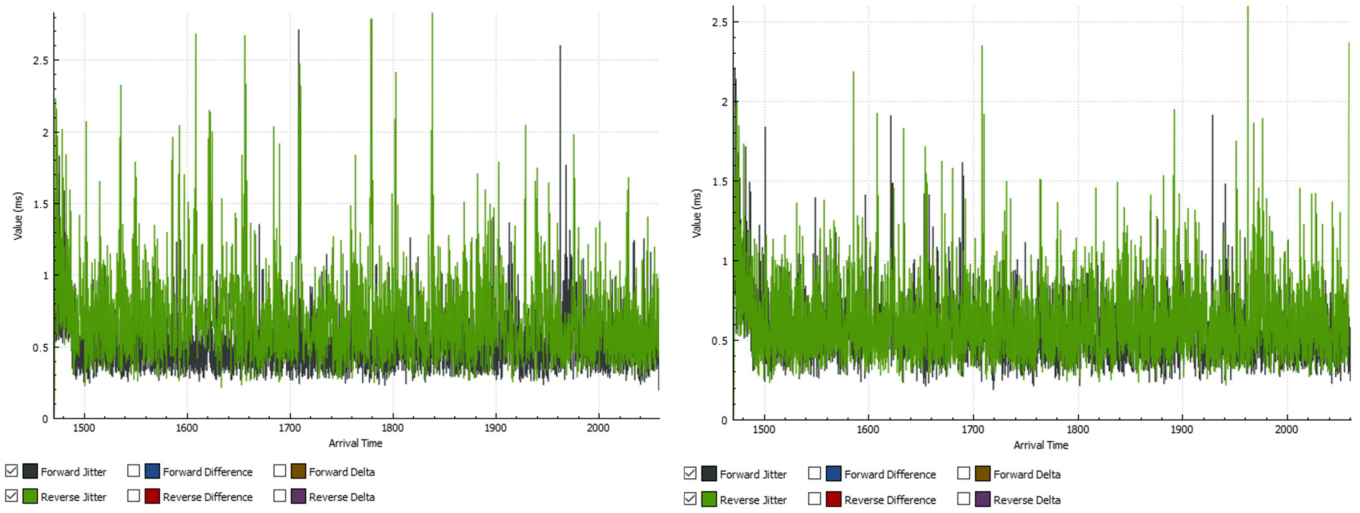
(**a**)

Forward		Reverse	
10.10.10.6:9078 → 10.10.10.5:10620		10.10.10.5:10620 → 10.10.10.6:9078	
SSRC	0x9e58648d	**SSRC**	0x6c024ff6
Max Delta	1023.63 ms @ 1386339	**Max Delta**	328.89 ms @ 1926686
Max Jitter	2.21 ms	**Max Jitter**	2.60 ms
Mean Jitter	0.63 ms	**Mean Jitter**	0.66 ms
Max Skew	-16.55 ms	**Max Skew**	16.44 ms
RTP Packets	39421	**RTP Packets**	44000
Expected	39421	**Expected**	44000
Lost	0 (0.00 %)	**Lost**	0 (0.00 %)
Seq Errs	0	**Seq Errs**	0
Start at	1470.475080 s @ 1386059	**Start at**	1471.218699 s @ 1386254
Duration	590.29 s	**Duration**	587.40 s
Clock Drift	2 ms	**Clock Drift**	-1 ms
Freq Drift	90000 Hz (0.00 %)	**Freq Drift**	90000 Hz (-0.00 %)

(**b**)

Figure 30. Summarized results for the main parameters of the video stream between VM_3 and the Asterisk (**a**) and between VM_4 and the Asterisk (**b**) during the TCP ACK attack.

Figure 31a,b shows the variation in jitter values for the video stream between VM_4/VM_3 and Asterisk in both directions. There was an improvement in the values because a TCP ACK attack is not highly burdensome, unlike a TCP SYN attack.



(**a**) (**b**)

Figure 31. Instantaneous values of the jitter for the video stream between VM_4 and Asterisk (**a**) and VM_3 and Asterisk (**b**) during the TCP ACK attack.

4.2.3. Results Obtained during the TCP RST Attack

Figure 32 represents the number of different TCP packets. The results are similar to the results obtained for the TCP ACK attack.

Figure 32. Number of different TCP packets sent during the TCP RST attack for the video conversation.

Figure 33 presents the summarized results for the video stream that was exchanged in both directions between VM_4 and VM_3 during the attack. The results are similar to the results obtained during the TCP ACK attack.

Forward		Reverse	
10.10.10.2:19078 → 10.10.10.5:12560		10.10.10.5:12560 → 10.10.10.2:19078	
SSRC	0xe3527a46	**SSRC**	0x2c76bb8c
Max Delta	396.26 ms @ 3288501	**Max Delta**	678.74 ms @ 3199748
Max Jitter	6.20 ms	**Max Jitter**	2.73 ms
Mean Jitter	0.51 ms	**Mean Jitter**	0.60 ms
Max Skew	-35.58 ms	**Max Skew**	22.43 ms
RTP Packets	40671	**RTP Packets**	43360
Expected	40671	**Expected**	43360
Lost	0 (0.00 %)	**Lost**	0 (0.00 %)
Seq Errs	0	**Seq Errs**	0
Start at	2431.705314 s @ 3199691	**Start at**	2431.040051 s @ 3199528
Duration	539.36 s	**Duration**	540.02 s
Clock Drift	-0 ms	**Clock Drift**	-0 ms
Freq Drift	90000 Hz (-0.00 %)	**Freq Drift**	90000 Hz (-0.00 %)

(**a**)

Forward		Reverse	
10.10.10.6:9078 → 10.10.10.5:18454		10.10.10.5:18454 → 10.10.10.6:9078	
SSRC	0xf7847b0d	**SSRC**	0x1fa41af7
Max Delta	679.23 ms @ 3199745	**Max Delta**	390.77 ms @ 3288507
Max Jitter	2.61 ms	**Max Jitter**	6.25 ms
Mean Jitter	0.52 ms	**Mean Jitter**	0.58 ms
Max Skew	21.95 ms	**Max Skew**	-36.09 ms
RTP Packets	43464	**RTP Packets**	40671
Expected	43464	**Expected**	40671
Lost	0 (0.00 %)	**Lost**	0 (0.00 %)
Seq Errs	0	**Seq Errs**	0
Start at	2431.039054 s @ 3199525	**Start at**	2431.705813 s @ 3199694
Duration	541.28 s	**Duration**	539.36 s
Clock Drift	-0 ms	**Clock Drift**	-0 ms
Freq Drift	90000 Hz (-0.00 %)	**Freq Drift**	90000 Hz (-0.00 %)

(**b**)

Figure 33. Summarized results for the main parameters of the video stream between VM_3 and the Asterisk (**a**) and between VM_4 and the Asterisk (**b**) during the TCP RST attack.

Figure 34a,b represents the variation in jitter values for the video stream between VM_4/VM_3 and Asterisk in both directions. Excluding the single spike, which may be due to computational errors, the obtained results are very close, almost identical to the results obtained during the TCP ACK attack.

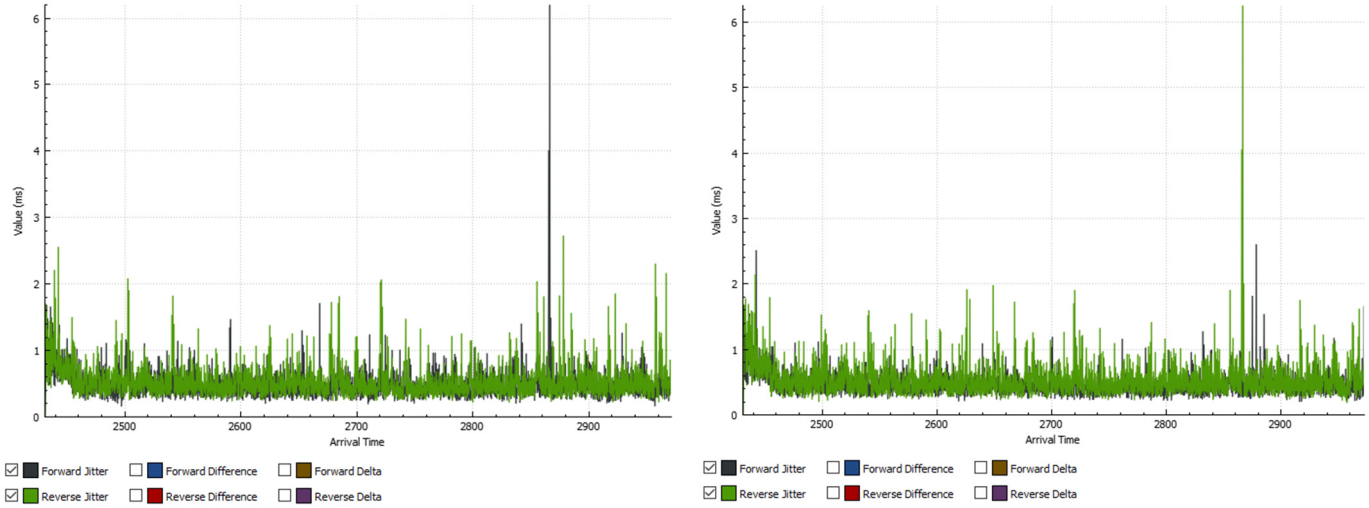


Figure 34. Instantaneous values of the jitter for the video stream between VM_4 and Asterisk (**a**) and VM_3 and Asterisk (**b**) during the TCP RST attack.

4.2.4. Results Obtained during the TCP FIN Attack

Figure 35 represents the number of different TCP packets. As can be seen from the graph, there were only TCP FIN packets because Asterisk was receiving multiple TCP FIN packets due to the attack. The embedded web server of Asterisk generated TCP FIN packets because it assumed that the other side sent the TCP FIN packets to terminate the TCP session.

Figure 35. Number of different TCP packets sent during the TCP FIN attack for the video conversation.

Figure 36 presents the summarized results for the video stream that was exchanged in both directions between VM_4 and VM_3 during the attack. Figure 36a shows the summarized results for the video stream that was exchanged between VM_4 and Asterisk, and Figure 36b shows summarized results for the video stream that was exchanged between

VM_3 and Asterisk. As can be seen, the maximum jitter value was very high, but the plots of variation in jitter values should also be examined to understand what caused this high value.

Forward		Reverse	
10.10.10.2:19078 → 10.10.10.5:10124		10.10.10.5:10124 → 10.10.10.2:19078	
SSRC	0x60ca107f	**SSRC**	0x52567e24
Max Delta	823.97 ms @ 1087201	**Max Delta**	389.28 ms @ 1203549
Max Jitter	71.64 ms	**Max Jitter**	3.40 ms
Mean Jitter	0.66 ms	**Mean Jitter**	0.76 ms
Max Skew	-785.23 ms	**Max Skew**	-25.46 ms
RTP Packets	53717	**RTP Packets**	46709
Expected	53717	**Expected**	46709
Lost	0 (0.00 %)	**Lost**	0 (0.00 %)
Seq Errs	0	**Seq Errs**	0
Start at	506.051228 s @ 1172	**Start at**	505.878544 s @ 1093
Duration	738.73 s	**Duration**	738.86 s
Clock Drift	-1 ms	**Clock Drift**	2 ms
Freq Drift	90000 Hz (-0.00 %)	**Freq Drift**	90000 Hz (0.00 %)

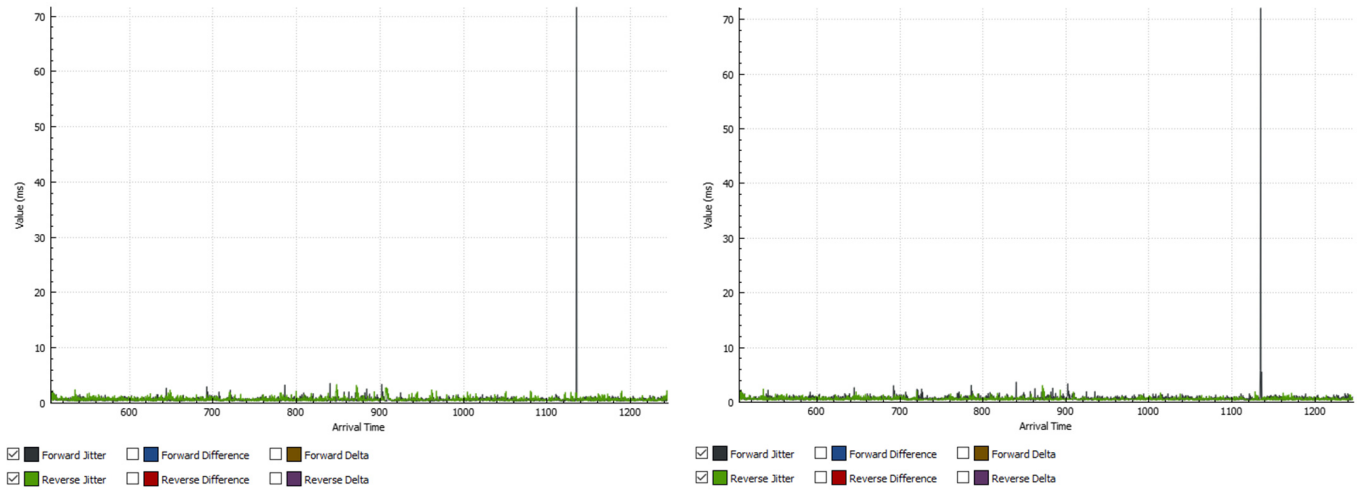
(**a**)

Forward		Reverse	
10.10.10.5:12250 → 10.10.10.6:9078		10.10.10.6:9078 → 10.10.10.5:12250	
SSRC	0x6772edbf	**SSRC**	0x1473da1b
Max Delta	826.46 ms @ 1087235	**Max Delta**	389.80 ms @ 1203545
Max Jitter	72.09 ms	**Max Jitter**	3.10 ms
Mean Jitter	0.72 ms	**Mean Jitter**	0.68 ms
Max Skew	-788.73 ms	**Max Skew**	-24.44 ms
RTP Packets	53717	**RTP Packets**	46765
Expected	53717	**Expected**	46765
Lost	0 (0.00 %)	**Lost**	0 (0.00 %)
Seq Errs	0	**Seq Errs**	0
Start at	506.052226 s @ 1175	**Start at**	505.878065 s @ 1091
Duration	738.73 s	**Duration**	740.95 s
Clock Drift	-1 ms	**Clock Drift**	2 ms
Freq Drift	90000 Hz (-0.00 %)	**Freq Drift**	90000 Hz (0.00 %)

(**b**)

Figure 36. Summarized results for the main parameters of the video stream between VM_3 and the Asterisk (**a**) and between VM_4 and the Asterisk (**b**) during the TCP FIN attack.

Figure 37a presents the variation in jitter values for the video stream between VM_4 and Asterisk in both directions. No degradation of the parameters was observed. As can be seen from the graph, the measured high jitter value is due to a single peak. In the rest of the video stream, the jitter levels varied between 1 and 2 ms. This single peak was due to a computational error. Figure 37b shows the variation in jitter values for the video stream between VM_3 and Asterisk in both directions. The results are identical.



(**a**) (**b**)

Figure 37. Instantaneous values of the jitter for the video stream between VM_4 and Asterisk (**a**) and between VM_3 and Asterisk (**b**) during the TCP FIN attack.

4.2.5. Summarized Results for the Video-Stream Study

Figure 38 (green area) presents the traffic that Asterisk processed during the entire study when it was exchanging video calls. As can be seen from the graph, the traffic handled by Asterisk increased many times, much more than the traffic handled during the voice-streams exchange. The peaks represent the duration of the periods of the individual attacks. As can be seen from the graph, Asterisk processed a large amount of traffic during each of the attacks.

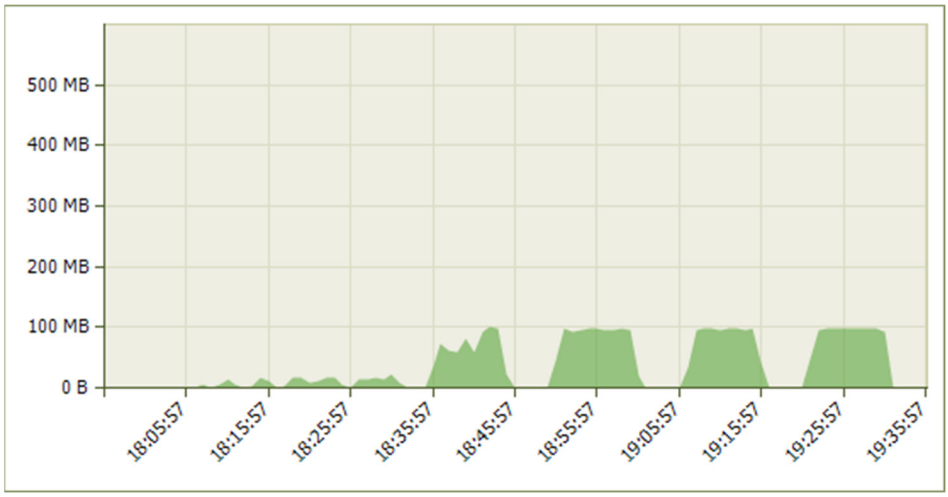


Figure 38. Proceeded video traffic from Asterisk during the whole study period.

Figure 39 presents the number of different TCP packets for the entire study period. As can be seen from the graphs, the results are identical to the results from the voice-flow study.

Figure 39. Number of different TCP packets for the whole study period during the video-stream study.

Figures 40 and 41 presents how the RTD changed between VM_3/VM_4 and Asterisk for the whole study period. As can be seen from the graphs, the values of the delay were far below the limit of 150 ms in one direction.

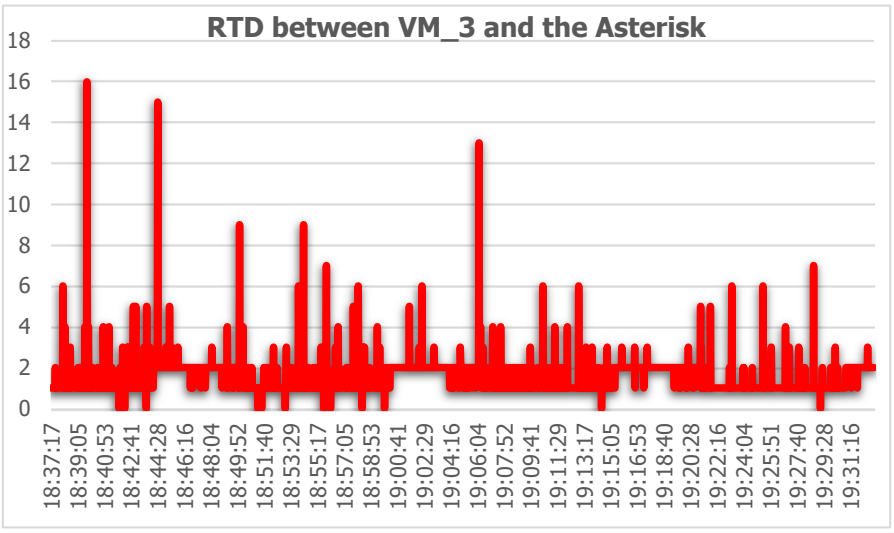


Figure 40. RTD between VM_3 and Asterisk for the video-stream study.

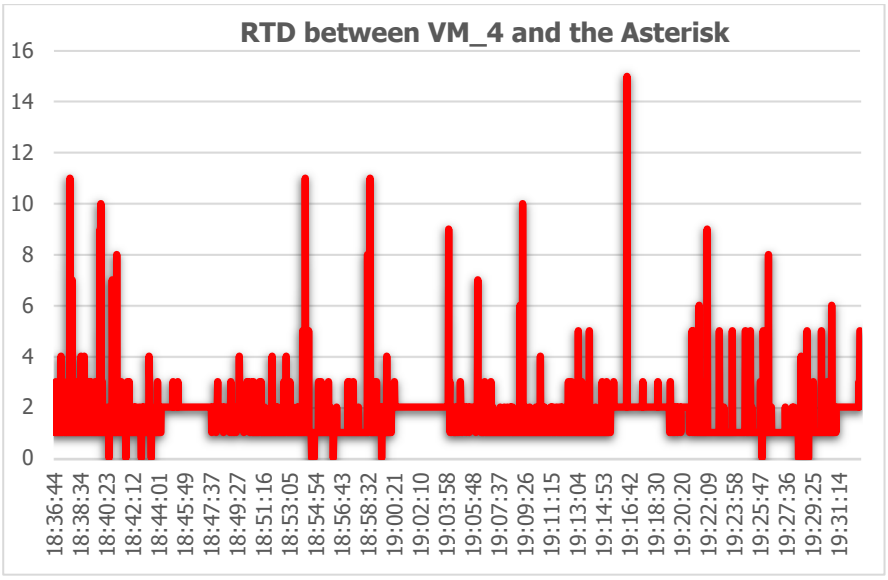


Figure 41. RTD between VM_4 and Asterisk for the video-stream study.

4.2.6. Discussion of the Results Obtained for the Video-Stream Study

The study of the impact of the different TCP DoS attacks on the parameters of the video streams showed the same results as the study of the voice streams. More "serious" distortions were observed in the results obtained during the TCP SYN attack because this is the most burdensome DoS attack compared to the others. Due to this attack, there was an increase in the value of the jitter. This increase led to a packet loss of 0.01%. These obtained values were very far from the maximum allowed ones of 1%. The results obtained during the other three attacks were close to the control results.

The results for the RTD measurement between Asterisk and two subscribers were also within the normal limits. No measurement interruptions of the type "Destination unreachable" were observed. The measured values of the RTD were close, almost identical, to the results obtained during the voice-flow study.

Again, it is noticeable that the "Responsive Firewall" was able to neutralize the effect of the individual attacks on the video streams.

Again, single spikes of large magnitude were observed in the jitter and round-trip-delay graphs during this study. These spikes were due to moments of computational errors.

In terms of traffic processed, Asterisk was able to handle this task in both studies. Despite being subjected to DoS attacks, due to which a large amount of traffic is sent to the VoIP server in the form of various TCP packets, voice and video streams, Asterisk was able to process this traffic. Processing this huge amount of traffic had no impact on the system performance. This can be seen very well in the CPU-load statistics for the day of the two tests (Figure 42). It is important to clarify that the VoIP server was only in use at the time of both studies. From the CPU-utilization graph, it can be seen that the CPU was only 0.33–0.34% utilized, which, for a device under DoS attack, is a very good indicator. In the studies by other authors that were discussed in Section 2, similar values were obtained after applying the methods, techniques, and algorithms developed by the authors to detect DoS attacks. The DoS attacks that were applied during the other studies were specific and included modified packet contents.

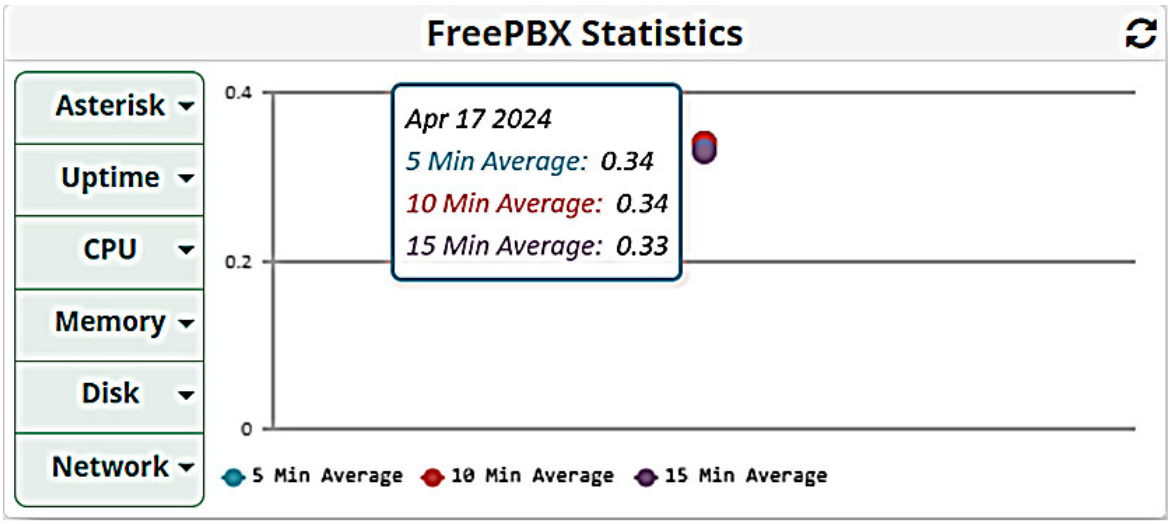


Figure 42. CPU load of the Asterisk Free PBX during the two studies.

Asterisk's memory utilization (Figure 43) during both studies was high, at 69%. It should be noted that this is an average value for the day of the two studies. Therefore, it can be assumed that for the individual studies, the memory load was not very high.

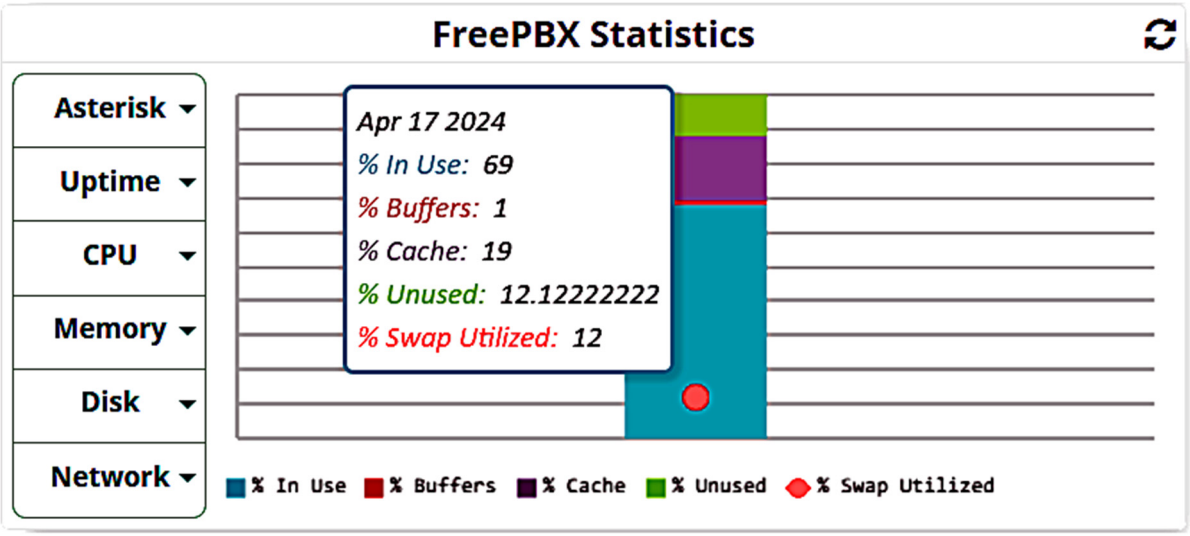


Figure 43. Memory load of the Asterisk Free PBX during the two studies.

An idea for the future development of this research is to replicate these attacks, but instead of them being TCP DoS attacks, they will be UDP DoS attacks because the UDP protocol is used to transmit both the voice and video streams, as well as the signaling (SIP) in Asterisk. Additionally, these UDP attacks will also be applied to specific ports or a range of ports.

5. Conclusions

A model of a working IP network has been created in which both voice and video streams were successfully exchanged.

From this study, it was found that in a VoIP network in which the VoIP server is an Asterisk Free PBX with the "Responsive Firewall" feature enabled, TCP DoS attacks do not affect the voice and video streams or the parameters of these streams. The measured average jitter values (at an average of about 1.1 ms for voice traffic and at an average of about 0.7 ms for video traffic) and packet losses (both measured to be about 0%) are far from the maximum allowable levels.

The results for the round-trip delay also prove that the different attacks do not affect Asterisk's performance. As can be seen from the graphs, during the entire period of both studies, there were no loss of measurement packets, although it was expected that ping would not be possible during the attacks.

The study shows that the Responsive Firewall feature successfully protects the system from the various TCP DoS attacks. Furthermore, it neutralized the impact of these attacks on Asterisk's performance. This could be seen very well in the CPU-utilization statistic, which is 0.34%.

This study could be applied to VoIP platforms where the voice and the video traffic are passed/processed through the VoIP server rather than directly between the IP phones. In this way, it will be possible to determine if, how, and in what way the different TCP DoS attacks affect the performance of the systems.

Funding: The APC was funded by South–West University "Neofit Rilski".

Data Availability Statement: Data are contained within the article.

Acknowledgments: The researcher would like to thank the South–West University "Neofit Rilski" for the APC.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Liu, C.; Du, D.; Zhang, C.; Peng, C.; Fei, M. Observability Analysis of Networked Control Systems Under DoS Attacks. In Proceedings of the IECON 2023 49th Annual Conference of the IEEE Industrial Electronics Society, Singapore, Singapore, 16–19 October 2023.
2. Sinha, S. Network layer DoS Attack on IoT System and location identification of the attacker. In Proceedings of the Third International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2–4 September 2021.
3. Gogoi, B.; Ahmed, T. HTTP Low and Slow DoS Attack Detection using LSTM based deep learning. In Proceedings of the 19th India Council International Conference (INDICON), Kochi, India, 24–26 November 2022.
4. Li, J.; Zhang, Y. Resilient DoS Attack Detector Design for Cyber-Physical Systems. In Proceedings of the 12th International Conference on Renewable Energy Research and Applications (ICRERA), Oshawa, ON, Canada, 29 August–1 September 2023.
5. Ramadhan, U.F.; Prastiantono, A.; Park, J.; Kim, D.; Yoon, M. Impact Analysis of DoS Attack at Vulnerable Point with the Exchange of Frequency Containment Reserves Control in MIDC System. In Proceedings of the International Conference on Technology and Policy in Energy and Electric Power (ICT-PEP), Jakarta, Indonesia, 18–20 October 2022.
6. Sarkunavathi, A.; Srinivasan, V. A Scrutinized study on DoS attacks in Wireless Sensor Networks and need of SDN in Mitigating DoS attacks. In Proceedings of the International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 27–29 January 2021.
7. Mladenov, B.; Iliev, G. Studying the effect of internal DOS attacks over SDN controller during switch registration process. In Proceedings of the 2022 International Symposium on Networks, Computers and Communications (ISNCC), Shenzhen, China, 19–22 July 2022.

8. Jekov, B.; Dimitrov, W.; Panayotova, G.S.; Kovatcheva, E. Intelligent protection of Internet of things systems. In Proceedings of the 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), Maldives, Maldives, 16–18 November 2022.
9. Dimitrov, W.; Spasov, K.; Trenchev, I.; Syarova, S. Complexity Assessment of Research Space for Smart City Cybersecurity. *IFAC-Pap.* **2022**, *55*, 1–6. [CrossRef]
10. Lacerda, M.J.; Oliveira, P.M.; Palma, J.M. Control design for cyber-physical systems under DoS attacks. In Proceedings of the 2022 IEEE International Conference on Automation/XXV Congress of the Chilean Association of Automatic Control (ICA-ACCA), Curicó, Chile, 24–28 October 2022.
11. Sriharipriya, K.C.; Mary, G.I.; Abishek, R.; Panja, A. Manipulation and Detection of DOS attacks on IEEE802. 11 Protocol. In Proceedings of the 2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN), Vellore, India, 5–6 May 2023.
12. Taher, M.A.; Iqbal, H.; Tariq, M.; Sarwat, A.I. Disruptive Effects of Denial-of-Service (DoS)Attacks on Microgrid Distributed Control: Altered Communication Topology, Voltage Stability, and Accurate Power Allocation. In Proceedings of the 2023 IEEE International Conference on Energy Technologies for Future Grids (ETFG), Wollongong, Australia, 3–6 December 2023.
13. Gore, S.; Nagalakshmi, Y.; Knowles, P.; Gupta, K.G.; Jagtap, N.S.; Sali, R.P. Improvised Ensemble Model for Fast Prediction of DoS/DDoS Attacks in Various Networks. In Proceedings of the 2023 1st International Conference on Cognitive Computing and Engineering Education (ICCCEE), Pune, India, 27–29 April 2023.
14. Srivastava, A.; Sharma, H.S.; Rawat, R.; Garg, N. Detection of Cyber Attack in IoT Based Model Using ANN Model with Genetic Algorithm. In Proceedings of the 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT), Greater Noida, India, 9–10 February 2024.
15. Siriyapuraju, S.J.; Gowri, V.S.; Balla, S.; Vanika, M.K.; Gandhi, A. DoS and DDoS attack detection using Mathematical and Entropy Methods. In Proceedings of the 2nd International Conference on Paradigm Shifts in Communications Embedded Systems, Machine Learning and Signal Processing (PCEMS), Nagpur, India, 5–6 April 2023.
16. Zhou, L.; Peng, C.; Cao, Z. Communication and Control Co-design for Networked Control Systems under DoS Attacks and Time-varying Delays. In Proceedings of the 4th International Conference on Control and Robotics (ICCR), Guangzhou, China, 2–4 December 2022.
17. Lazarova, M.; Sapundzhi, F. Stochastic Modeling with Applications in Supply Chain Management and ICT Systems. *Computation* **2023**, *11*, 21. [CrossRef]
18. Zoraida, B.S.E.; Indumathi, G. Comparison of software defined networking with traditional networking using NS2 simulator. *Int. J. Inf. Technol. Secur.* **2023**, *15*, 3–14. [CrossRef]
19. Zelmanov, S.S.; Krylov, V.V. Computer simulation of strength testing of an object based on signal shaped resources. *Int. J. Inf. Technol. Secur.* **2023**, *15*, 59–68. [CrossRef]
20. Wang, Y.; Zheng, H.; Ye, Y.; Li, L.; Hu, H.; Zhang, J. Modeling and Analysis of SYN Flooding Attack in Power SCADA System Based on Queuing Theory. In Proceedings of the International Conference on Wireless Communications and Applications (ICWCAPP), Haikou, China, 20–21 August 2022.
21. Tasho, D.T.; Marin, B.M.; Radostina, P.T.; Alexander, K.A. Generalized nets model of the LPF-algorithm of the crossbar switch node for determining LPF-execution time complexity. In Proceedings of the AIP Conference 2333, 090039 (2021), Sofia, Bulgaria, 7–13 June 2020.
22. Hensel, S.; Marinov, M.B.; Koch, M.; Arnaudov, D. Evaluation of Deep Learning-Based Neural Network Methods for Cloud Detection and Segmentation. *Energies* **2021**, *14*, 6156. [CrossRef]
23. Tashev, T.D.; Marinov, M.B.; Arnaudov, D.D.; Monov, V.V. Computer simulations for determining of the upper bound of throughput of LPF-algorithm for crossbar switch. In Proceedings of the AIP Conference Proceedings, Técnica, Manabí, 11 January 2022; Volume 2505, p. 080030.
24. Tashev, T.D.; Alexandrov, A.K.; Arnaudov, D.D.; Tasheva, R.P. Large-Scale Computer Simulation of the Performance of the Generalized Nets Model of the LPF-algorithm. In *Large-Scale Scientific Computing*; LSSC 2021. Lecture Notes in Computer Science; Lirkov, I., Margenov, S., Eds.; Springer: Cham, Switzerland, 2021; Volume 13127.
25. Sapundzhi, F.I.; Popstoilov, M.S. Maximum-Flow Problem in Networking. *Bulg. Chem. Commun.* **2020**, *52*, 192–196.
26. Qaid, A.; Ertuğ, Ö. Transition from IPv4 to IPv6 Mechanisms by GNS3 Emulation: YPTC as a Case Study. In Proceedings of the 2021 International Symposium on Networks, Computers and Communications (ISNCC), Dubai, United Arab Emirates, 31 October–2 November 2021.
27. Biradar, A.G. A Comparative Study on Routing Protocols: RIP, OSPF and EIGRP and Their Analysis Using GNS-3. In Proceedings of the 2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Jaipur, India, 1–3 December 2020.
28. Parwani, R.; Al-Amoudi, H.M.S.; Jhummarwala, A. Modeling and Simulating large scale Cyber Effects for Cybersecurity Using Riverbed Modeler. In Proceedings of the 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 29–31 January 2020.

29. Li, F.; Gao, W.; Chen, L.; Liu, W. Modeling and Simulation of Network-on-Chip Routing Algorithm Based on OPNET. In Proceedings of the 2020 International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI), Sanya, China, 4–6 December 2020.
30. Konshin, S.; Yakubova, M.Z.; Nishanbayev, T.N.; Manankova, O.A. Research and Development of an IP network model based on PBX Asterisk on the Opnet Modeler simulation package. In Proceedings of the 2020 International Conference on Information Science and Communications Technologies (ICISCT), Karachi, Pakistan, 8–9 February 2020.
31. Tas, I.M.; Baktir, S. A Novel Approach for Efficient Mitigation against the SIP-Based DRDoS Attack. *Appl. Sci.* **2023**, *13*, 1864. [CrossRef]
32. Younes, O.; Albalawi, U. Securing Session Initiation Protocol. *Sensors* **2022**, *22*, 9103. [CrossRef] [PubMed]
33. Amalou, W.; Mehdi, M. An Approach to Mitigate DDoS Attacks on SIP Based VoIP. *Eng. Proc.* **2022**, *14*, 6. [CrossRef]
34. Nazih, W.; Elkilani, W.S.; Dhahri, H.; Abdelkader, T. Survey of Countering DoS/DDoS Attacks on SIP Based VoIP Networks. *Electronics* **2020**, *9*, 1827. [CrossRef]
35. Nazih, W.; Hifny, Y.; Elkilani, W.S.; Dhahri, H.; Abdelkader, T. Countering DDoS Attacks in SIP Based VoIP Networks Using Recurrent Neural Networks. *Sensors* **2020**, *20*, 5875. [CrossRef] [PubMed]
36. Armoogum, S.; Mohamudally, N. A Comprehensive Review of Intrusion Detection and Prevention Systems against Single Flood Attacks in SIP-Based Systems. *Int. J. Comput. Netw. Inf. Secur.* **2021**, *13*, 13–25. [CrossRef]
37. Jama, A.M.; Khalifa, O.O.; Subramaniam, N.K.; Kumar, N. Novel Approach for IP–PBX Denial of Service Intrusion Detection Using Support Vector Machine Algorithm. *Int. J. Commun. Netw. Inf. Secur.* **2021**, *13*, 249–257. [CrossRef]
38. Younes, O.S. A hybrid deep learning model for detecting DDoS flooding attacks in SIP–based systems. *Comput. Netw.* **2024**, *240*, 110146. [CrossRef]
39. Khan, H.M.A.; Inayat, U.; Zia, M.F.; Ali, F.; Jabeen, T.; Ali, S.M. Voice Over Internet Protocol: Vulnerabilities and Assessments. In Proceedings of the 2021 International Conference on Innovative Computing (ICIC), Lahore, Pakistan, 9–10 November 2021; pp. 1–6.
40. Tas, I.M.; Baktir, S. Blockchain-Based Caller-ID Authentication (BBCA): A Novel Solution to Prevent Spoofing Attacks in VoIP/SIP Networks. *IEEE Access* **2024**, *12*, 60123–60137. [CrossRef]
41. Çakır, S.; Sertbaş, A.; Aydın, M.A. Machine Learning-Based Security Test Model and Evaluation for SIP-Based DoS Attacks. In Proceedings of the 2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), Biarritz, France, 8–12 August 2022; pp. 1–5.
42. Tas, I.M.; Unsalver, B.G.; Baktir, S. A Novel SIP Based Distributed Reflection Denial-of-Service Attack and an Effective Defense Mechanism. *IEEE Access* **2020**, *8*, 112574–112584. [CrossRef]
43. Sbai, O.; Allaert, B.; Sondi, P.; Meddahi, A. SIP-DDoS: SIP Framework for DDoS Intrusion Detection Based on Recurrent Neural Networks. In *Machine Learning for Networking*; MLN 2023. Lecture Notes in Computer Science; Renault, É., Boumerdassi, S., Mühlethaler, P., Eds.; Springer: Cham, Switzerland, 2024; Volume 14525. [CrossRef]
44. Wang, S.; Li, H.; Song, P.; Xu, L. A SIP-Based Flooding Attack Detection Method in VoIP Environment. In Proceedings of the 6th International Conference on Information Technologies and Electrical Engineering (ICITEE '23). Association for Computing Machinery, New York, NY, USA, 26 March 2024; pp. 680–684. [CrossRef]
45. Choti, C.; Hnoohom, N.; Tritilanunt, S.; Yuenyong, S. Prediction of Intrusion Detection in Voice over Internet Protocol System using Machine Learning. In Proceedings of the 9th International Conference on Computer and Communications Management (ICCCM '21). Association for Computing Machinery, New York, NY, USA, 28 October 2021; pp. 149–155. [CrossRef]
46. Kafke, J.; Viana, T. Call Me Maybe: Using Dynamic Protocol Switching to Mitigate Denial-of-Service Attacks on VoIP Systems. *Network* **2022**, *2*, 545–567. [CrossRef]
47. Getting Started with GNS3. Available online: https://docs.gns3.com/docs/ (accessed on 25 May 2024).
48. Kali Docs, Official Documentation. Available online: https://www.kali.org/docs/ (accessed on 25 May 2024).
49. Wireshark. Available online: https://www.wireshark.org/docs/wsug_html_chunked/ (accessed on 25 May 2024).
50. Capsa Free Network Analyzer. Available online: https://www.colasoft.com/capsa-free/ (accessed on 25 May 2024).
51. Colasoft Ping Tool. Available online: https://www.colasoft.com/ping_tool/ (accessed on 25 May 2024).
52. VMware Workstation Pro. Available online: https://www.vmware.com/products/workstation-pro/html.html (accessed on 24 June 2024).
53. Windows Technical Documentation for Developers and IT pros. Available online: https://learn.microsoft.com/en-us/windows/ (accessed on 24 June 2024).
54. Linphone for Desktop. Available online: https://www.linphone.org/ (accessed on 24 June 2024).
55. Tim, S.; Christina, H. End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs. In *Part of the Networking Technology Series*; Cisco Press: Indianapolis, Indiana, 2004; ISBN-10: 1-58705-176-1.
56. Cisco-Understanding Delay in Packet Voice Networks, White Paper. Available online: https://www.cisco.com/c/en/us/support/docs/voice/voice-quality/5125-delay-details.html (accessed on 25 May 2024).
57. TCP SYN Flood Attack. Available online: https://www.imperva.com/learn/ddos/syn-flood/ (accessed on 25 May 2024).
58. What Is an ACK Flood DDoS Attack? Available online: https://www.cloudflare.com/learning/ddos/what-is-an-ack-flood/ (accessed on 25 May 2024).

59. RST Flood Attack. Available online: https://kb.mazebolt.com/knowledgebase/rst-flood/ (accessed on 25 May 2024).
60. FIN Flood Attack. Available online: https://kb.mazebolt.com/knowledgebase/fin-flood/ (accessed on 25 May 2024).