Evaluation of Denial of Service Attacks in Software Defined-Cognitive Radio Networks

Mampuele Lebepe(✉) and Mthulisi Velempini

University of Limpopo, Private Bag X1106, Sovenga 0727, South Africa
mampuele.lebepe@gmail.com, mvelempini@gmail.com

Abstract. Software defined networks (SDN) offer a novel network resource management framework which addresses network resources management challenges. It addresses the spectrum scarcity problem by employing efficient and dynamic spectrum access. Cognitive radio networks (CRN) enables secondary users to coexist with licensed users in non-interfering manner. Unfortunately, SDN is susceptible to security threats. We integrate a SDN and a CRN and evaluate the denial of service (DoS) in the integrated environment. The DoS attack is a threat to SDN based networks. The DoS attack overloads the controller and floods the switch Content Addressable Memory (CAM tables), which degrades the performance of the network. We evaluate the effectiveness of the SDN-Guard and the Jamming Attack in addressing the effects of the DoS.

SDN-Guard is designed to minimize the overloading of the controller, and the flow tables while managing the flow routes dynamically, timeouts of entry rule and to aggregate flow rule entries given the probability of the threat of the flow which is determined by an intrusion detection system (IDS). IDS is used to detect and control the jamming attack. It is a set of procedures and systems that are able to identify intrusions in a system. This study evaluates the effects of DoS attack on software defined cognitive radio networks. The study observed that the SDN-Guard detects the DoS attack earlier and it reduces the average round trip time and the average processing time compared to the Jamming Attack Defender.

Keywords: Software defined networks · Cognitive radio network · Denial of service · Intrusion detection system

1 Introduction

Software Defined Network (SDN) framework addresses several network resources management challenges. The Cognitive Radio Networks (CRN) on the other hand, is designed to address the spectrum scarcity by employing efficient and dynamic spectrum access (DSA). CRN provides secondary users with the ability to coexist with primary users in non-interfering mode. In this study, we integrate the two networks and evaluate the effects of Denial of Service attack (DoS) in the integrated environment - Software defined Cognitive Radio Network (SD-CRN).

W. Bao et al. (Eds.): ADHOCNETS 2021/TridentCom 2021, LNICST 428, pp. 49–62, 2022.
https://doi.org/10.1007/978-3-030-98005-4_4

A DoS is any type of attack which overwhelms a server and prevents it from servicing its clients. DoS is a challenge in the Internet and other forms of networks. This attack is also a challenge in the SD-CRN. It can overload the controller and overwhelms its processing capacity and floods the switch CAM tables and degrade the performance of the network [1]. This result in loss of revenue for online businesses. Reverse proxy is one effective defense mechanics which counters the DoS [2]. However, it requires complementary schemes to improve its effectiveness [2]. More robust approaches are required to mitigate the effects of the DDoS attacks.

A DoS attack can cause the network to be unstable, unusable by sending data in special patterns or by flooding the network with packets. Remote services can be overwhelmed by a stream of packets from attackers or compromised nodes. However, the effects of DoS can be mitigated in SD-CRN.

The study evaluates the effects of DoS attacks in SD-CRN. The DoS is simulated in SD-CRN environment and two countermeasures are evaluated, the Jamming attack Defender and the SDN-Guard. The effectiveness of these countermeasures is evaluated and comparative results are presented. The following metrics are used for comparison purposes: the controller workload, bandwidth of the control plane, Flow table, bandwidth of the network, Average processing time, Round trip time, and Signal strength.

2 Related Work

Given the significance of 5G enabling technologies such as the SD-CRN, there is need to address the security challenges of such technologies. This Section presents related work and evaluates DoS schemes in SD-CRN. These issues are discussed in [1] and [2]. Nonetheless, we are focus on DoS attacks in SD-CRN in this study.

In [3], a scheme is proposed to mitigate DoS in SDN using a Path Randomization technique. The study focussed on minimizing the effects of DoS on flow tables, which can degrade the network switches. The authors used an algorithm to aggregate flows that produced a positive outcome.

In [4], the effects of DoS on network performance is discussed. The study shows how the attack affects parameters such as the bandwidth of the control plane (controller-switch channel) and latency. The impact on the performance of controller was also analysed. Unfortunately, these issues were not solved.

In addition, a scheme was proposed in [5], the FlowRanger which detects and mitigates the effects of DoS. While the FlowRanger is consist of the following three components:

(1) The trust management element which computes a trust value based on its origin for each packet-in message.
(2) The element of the queuing management which stores the message in the priority queue which corresponds to its trust value and
(3) The message scheduling component which uses a weighted Round Robin strategy to process messages.

In [6], a scheme was proposed to protect SDN from the distributed IP filtering DoS attacks. The proposed scheme analyses user behaviour and assigns flow timeouts based

on user behaviour. The flows of malicious users were assigned short timeouts while flows of trusted ones were assigned long timeouts. This approach requires malicious traffic entries to be deleted quickly from CAM table's switches. Nevertheless, if the flow length is greater than the fixed timeout, this may result in new packet-in messages being transmitted to the controller. This approach also eliminates malicious traffic, which can pose problems for false-positive flows.

A scheme in [7] leverages SDN's hierarchical strategy and programmability and proposes a self-management scheme involving an ISP and its clients to address DoS. The ISP collects risk data from users to use it in the implementation of a security approach and to update network flow tables. The ISP controller assigns a high priority value if a flow is assumed to be trustworthy. If the authenticity of the flow is in question, the ISP controller assigns a low priority to the flow and manage it through the path assigned to malicious flow. This reduces the effects of the DoS on the network performance by adjusting the load. Unfortunately, it does not address the overloading of the controller and the flooding of the flow tables within the switches.

The available schemes cannot reduce the load of the controller, the round trip time, the switch-to-controller bandwidth, the average processing time, and the network bandwidth usage while detecting the malicious nodes. Table 1 summarises the research gaps in SD-CRN.

Table 1. Analysis of the gaps in the literature

Approach	**Objective: Minimizing the effects of the following metrics**						
	Controller workload	Control plane bandwidth	Flow table usage	Network bandwidth	Average processing time	Round trip time	Signal strength
SDN-Guard [8]	✓	✓	✓	✓	✓	✓	✓
Flow Ranger Invalid source specified.	✓	×	×	×	×	×	×
IP filtering approach [6]	✓	✓	✓	✓	×	×	✓
Jamming attack defender[1]	✓	✓	✓	✓	✓	✓	✓
Self-management scheme [7]	×	×	×	✓	✓	✓	✓

Regarding security requirements in CRN's, the work in [9] stated that the security requirements such as availability, integrity, identification, authentication, confidentiality are essential in CRNs. Availability refers to the ability of primary users (PU) and secondary users (SU) to access the spectrum timeously.

A number of methods for detecting malicious activity using Openflow have been investigated. These methods vary from local network detection of infected hosts by comparing flows [10] to deterministic sampling using Openflow to inspect certain classes of traffic [11]. With available features in Openflow version 1.0, we explored the possibilities of using Openflow to detect the DoS attacks. An ideal DoS solution may consist of the following: the initial detection, sampling techniques, and blocking behaviour.

In [12], the focus was on mitigating the DoS attack on flow tables which result in the degradation of the network switches. In order to address this issue, a path randomization technique and flow aggregation algorithm was proposed. The system performance was evaluated in a simulation environment that showed some positive results.

3 Methodology

DoS is the most common and unavoidable threats to SDN security and all types of networks. The DoS overloads the network and servers by overwhelming them with streams of traffic while starving legitimate users of valuable service [13].In this study, we investigated the best algorithms designed to detect and address the DoS. This Section, generates statistical data through simulations to meet the objectives of this study. Table 2 depicts the simulation environment and Table 3 presents the parameters used.

Table 2. Simulation environment

Computer	HP L425.SCMSDOM.LOCAL
RAM	8,00 GB
CPU	intel®Pentium(D) CPU 2037 @3.19 GHz
OS (operating system)	Microsoft windows

Table 3. Parameters and tools

1 Parameters	2 Tools
3 Network Simulator	4 Matlab
6 Controller	6 Floodlight 1.2
7 Switch Software	8 OpenFlow V 1.3
9 IDS	10 Snort 2.0.6
11 Simulation Area	12 150 m * 150 m

An SDN-Guard is an SDN application that is plugged into an SDN controller and which uses the network traffic ID to analyse the flow and raise an alarm when a malicious

traffic is detected. Given the alerts and the current state of the network, appropriate decisions designed to minimize the effects of DoS are made for each flow. It consists of the three following modules (Fig. 1):

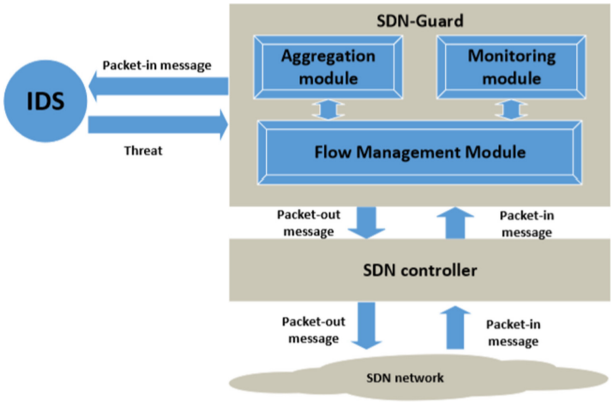

Fig. 1. SDN-Guard layout

The flow management module selects the routing path for each flow and determines the firm timeout of the TCAM entries informed by the risk of flow, to mitigate the effects of the DoS [14].

Rule based aggregation module which aggregates malicious traffic inputs to minimize the number of inputs used in TCAM switches [14].

Monitoring module collects multiple flow, switch and link statistics (e.g. flow, switch TCAM and link bandwidth usages) so that other modules can use them [14].

SDN-Guard communicates with an IDS which analyses packet-in messages and alerts SDN-Guard of the likelihood of flow threats. An IDS can be replaced by a system capable of evaluating accurately the risk of flows like the one used in [15].

To address the DoS on the SD-CRN, the proposed scheme consists of the following:

Threat-Based Routing: To address the effects of DoS on bandwidth usage and queuing delays, SDN-Guard reroutes malicious traffic to the least-used links based on bandwidth requirements and switch ternary content addressable memory (TCAMs). Due to the statistical data collected by the monitoring module, the flow management module has access to the values of these two parameters. A generated path may ensure minimal impact of attack and may not be a shortest path. Malicious traffic must reach the destination where it can theoretically be further analysed by IDS or by a prevention system (which is critical in case of false positive malicious flows). We do not drop malicious traffic to ensure that false positive malicious flows, with higher delays, can reach their destination. The non-malicious flows are routed through the shortest paths to ensure minimum delays [14].

Timeout Management: Depending on the probability of threat, the flow management module set the timeout to each flow. The switch communicates with the controller when

the hard timeout expires. The controller has a shorter hard timeout which increases traffic. This does not only increase the bandwidth usage of the switch-to-controller, but it overloads the controller. If the flow is malicious, the SDN-Guard sets high timeout to the flow. The idea is to ensure that the flows do not trigger higher controller to the switch traffic [14].

Malicious Flow Rule Aggregation: Malicious flows are assigned longer hard timeout. These flows are retained for a longer duration. These may overload entries in the flow tables as the number of entries increase in the flow tables. Flow aggregation is considered as a solution to the challenge where the aggregation module aggregates malicious flows in a given switch on the bases of the same source and destination [14].

When a new flows are received by a switch, which cannot be associated with any rule, control is passed to the controller for an appropriate forwarding rule. The packet-in messages are sent to the IDS to analyse their threats. The threat probability is used for routing decision making and setting timeouts for entries in switches' TCAMs. Two cases can be identified:

Table 4. Flow management decisions

Flow type	Threat probability	Timeout	Path	Rule aggregation
Legitimate	Low	Default	Shortest	Optional
Malicious	High	High	Least-utilized links	Mandatory

The Placement of IDS and Traffic Management.
There are two IDS deployment options:

- Under the first option, multiple IDS can be deployed to one switch. Each IDS then analyses the traffic passing through its associated switch.
- In the second option, a single IDS is deployed which analyse all the traffic.

The following are proposed as possible solutions:

(1) optimal IDS placement and traffic mirroring
(2) switch-to-IDS traffic sampling

The two possible solutions are discussed in detail in the sequel:

Optimal IDS Placement and Traffic Mirroring: An optimal location of IDS determines switches which should mirror the flows to minimize the mirrored traffic and the bandwidth required (by minimizing the number of links used by mirrored traffic). The Integer Linear Program (ILP) can be used to model the placement problem of IDS [4].

Let $G = (N, L)$ represent the network where N is the set of switches and L is the set of links connecting to the switches.

We define $p_{n\overline{n}}$ as the cost of the shortest path from switch $n \in N$, which corrresponds to the number of hops between the two switches.

Let $i \in I$ denote a flow in transit in a network. The throughput of the flow i is denoted by f_i .

Define $r_{in} \in \{0, 1\}$ as a boolean variable which equals to 1 if the flow $i \in I$ passes through the switch $n \in \overline{N}$.

The controller has knowledge of the defined variables, a flow i cannot be forwarded from a switch n if it doesn't pass through the switch, hence we have

$$x_{in} \leq r_{in} \quad \forall n \in N \quad \forall i \in I. \tag{1}$$

We also define the decision variable $x_{in} \in \{0, 1\}$ as a boolean variable that indicates whether the flow i is mirrored from the switch n to the IDS. Each flow i is mirrored only once to the IDS. The following constraint may be met:

$$\sum\nolimits_{n\in N} x_{in} = 1 \quad \forall n \in I. \tag{2}$$

The cost of mirroring the flows to an IDS $\overline{n} \in N$ corresponds to the amount of mirrored traffic forwarded from the switches to the IDS. This can be calculated as:

$$C_{\overline{n}} = \sum\nolimits_{i\in I} \sum\nolimits_{n\in N} x_{in} p_{n\overline{n}} f_i \quad \forall n \in N. \tag{3}$$

Finally, the objective is to find the switch $\overline{n} \in N$ that minimizes the mirriring cost:

$$\min_{\overline{n}\in\overline{N}} C_{\overline{n}} \tag{4}$$

The proposed ILP provides the location of the IDS (ie, $\overline{n}$) and switches which forward the traffic to the IDS (using the decision variable x_{in}). Figure 2 depict the experimental environment.

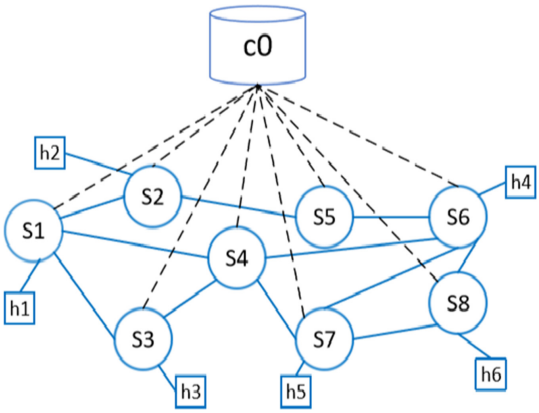

Fig. 2. Experimental setup

Figure 2 shows a topology with eight Openflow switches and six hosts. The controller C0 is connected to switch S5, and data is forwarded through switch S5. The hosts h1, h2, and h3 are malicious nodes in which the server h6 is a DoS target [4].

The experiment begins when normal traffic consisting of TCP flows are sent by all source nodes. In this case, the DoS transmission lasted for 10 min in which the server was flooded with TCP streams. To launch the DoS, TCP traffic was sent using the ping command to generate a streams of traffic designed to flood the target, the server with TCP-SYN, ICMP, and UDP packets from different sources with different IP source addresses. This traffic mimics a DoS originating from different sources [4].

The efficiency of the IDS was evaluated using sampled traffic. The performance was evaluated using packet-processing time when sampling rates were differed. Three types of DoS were generated namely, TCP-SYN, UDP, and ICMP flooding for 30 min. A number of experiments where sampling rates were differed were conducted. A sampling rate of p% depicts that p% of the mirrored traffic is dropped randomly at the switches thereafter it is forwarded to the IDS. The efficiency was determined by the percentage of detected attacks, the number of attacks which were detected successfully in sampled traffic divided by the total number of attacks detected [4].

The second scheme we used is the Jamming Attack Defender.

In jamming attack, the (jammer) attacker maliciously sends or receives data to interfere with genuine users in a session. This situation in turn creates a DoS condition. The jammer may continuously send data packets so that a genuine user may not sense the channel as idle. On the other hand, the legitimate users receive junk packets sent continuously by the jammer. The jammer may overwhelm radio transmission and corrupt the data packets that legitimate users receive. In the worst case, the attacker may jam the dedicated channel used to communicate sensing information among CRs. This attack is called as common control data attack. In addition, if the attacker listens on the control data, the attacker overhears which new channel the CRN is switching to and jams it. These jamming attacks can be done at MAC and physical layers [5].

The attackers have different network attack strategies. The detection of security threats is therefore possible. The attackers may attack both PUs and SUs while in general, SUs are targeted. A number of detection techniques have been introduced in the detection and mitigation of attacks in CRN. The detection technique involves two phases which are the learning phase and the detection phase [5].

In this work, the physical layer attack namely the jamming attack is considered. Jamming attack is detected through the observation of signal strength (SS) and packet delivery ratio (PDR). The collection of information regarding SS and PDR facilitates the detection phase of the IDS to effectively detect the unknown attacks in CRNs. In the learning phase, the normal network behaviour or its performance is observed. In detection phase, the abnormal changes are detected using the non-parametric cumulative sum control chart (cusum) algorithm [5].

During the detection phase, the IDS detects the point of change in CRN operation. In case of a malicious user, the SU is jammed, the SS is measured at the SU is examined. If the SS is high, then its PDR is dropped. The PDR is the ratio of the number of packets received by user to the number of packets sent [3, 16]. To detect the change in the PDR of SU targeted by jamming attacker, the cusum algorithm based on change point detection

algorithm is employed. It is assumed under normal conditions that the mean value of the random sequence is negative, it becomes positive if any change is detected.

G_n sequence is obtained as:

$$G_n = \beta - F_n \tag{5}$$

Where β is the average of the minimum (negative peak) values of F_n throughout the profiling period. The increase in the mean G_n value can be lower bounded by $h = (2\beta)$ during a jamming attack. Then, the cusum sequence Y_n is expressed as in Eq. (7) where:

$$q^+ = q \; if \; q > 0; \text{otherwise } q^+ = 0 \tag{6}$$

A large value of Y_n implies an anomaly. The detection threshold θ is computed as follows

$$Y_n = (Y_{n-1} + G_n)^+; Y_0 = 0 \tag{7}$$

$$\theta = (m - \beta)t_{des} \tag{8}$$

where t_{des} denotes the desired detection time. It is set to a small value for earliest detection of an anomaly in the CRN. In detection phase, the IDS computes Y_n over a certain period. The value of Yn remains close to zero while the CRN is in normal operation condition. The value of Y_n starts to increase in the presence of a jamming attack. If *Yn* goes above the pre-determined value of θ, and the SS at the SU is high, an alert is generated indicating a possibility of jamming attack [3, 16]. Figure 3 depict the operations of IDS.

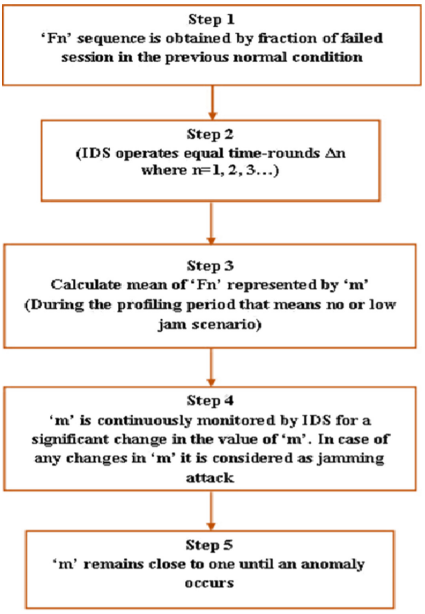

Fig. 3. Flow chart of IDS operation

The IDS is implemented in MATLAB environment. The presence of a licenced users or SU is recognised using the power spectral density in a particular channel. We assume that IDS operates at equal time bounds (Δn where $n = 1, 2, 3\ldots$). Then the operation described in Fig. 3 is performed by the SUs or the cognitive users.

4 Results

In this Section, we present and analyse the generated results of the study which are represented graphically. We considered two schemes in this research, SDN-Guard and the jamming attack defender. The set of results of the two schemes are therefore presented.

Figure 4 depict the simulation area. It shows the 10 nodes which are moving within the 150 m * 150 m grid area. In addition, the network also consists of a base station and four attacking nodes. The malicious nodes launch the DoS attack in the network.

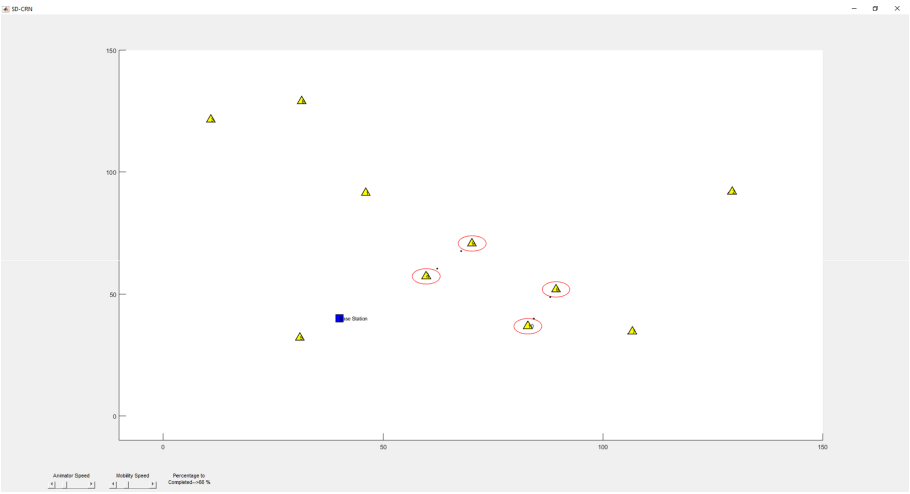

Fig. 4. Simulation area

To evaluate the two schemes and generate comparative results, we considered the following metrics: Average round trip time, Average packet processing time, and Power Spectral density.

Figure 5 presents the average RTT values of the SDN-Guard and the Jamming Attack Defender. The RTT for Jamming Attack Defender is higher than the RTT of the SDN-Guard caused by longer time-outs associated with malicious traffic. This prevent the switches from requesting new flow rules. The requests are also not sent to the controller for flow entry requests. We can see that SDN-Guard is the better scheme because it has lower RTTs.

A Power Spectral Density (PSD) is the measure of signal's power content versus frequency. A PSD characterize broadband random signals. The magnitude of the PSD is normalized by the spectral resolution employed to digitize the signal.

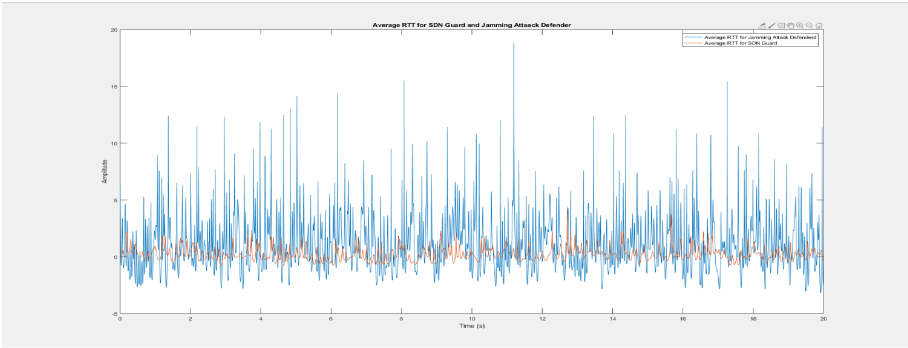

Fig. 5. Average Round-Trip-Time (RTT) for SDN-Guard and Jamming Attack Defender

Figures 6 and 7 present the power density spectrum of one PU available in the slot. The other four user slots are free which means the spectrum is available for SU.

Figure 6 shows the power spectral density of the Jamming Attack Defender. At frequency 0, the magnitude is at 17dB. As the frequency increases to 5 Hz, the magnitude increases to 40dB and it starts decreasing thereafter. The magnitude then remains constant at 15dB as the frequency increases from 25 Hz to 30 Hz.

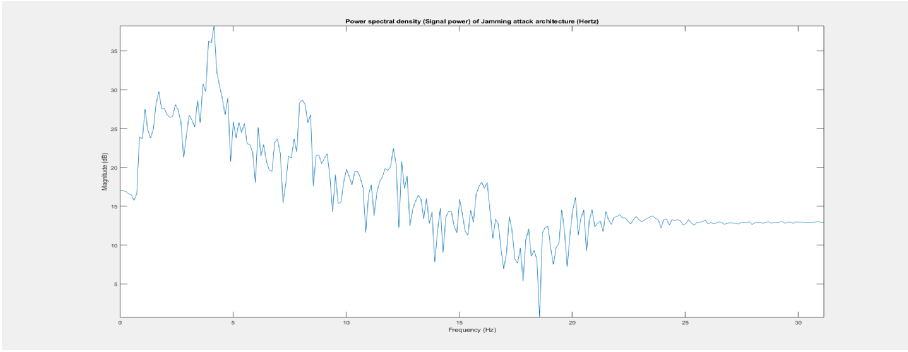

Fig. 6. Power Spectral density of Jamming Attack Architecture

Figure 7 presents the PSD of the SDN-Guard architecture. At frequency 0, the magnitude is at 17 dB. As the frequency increases, the magnitude starts decreasing. This means that the Jamming Attack Defender has a better PSD because it remains constant while the PSD of the SDN-Guard decreases. Therefore, the Jamming Attack Defender has better signal strength than the SDN-Guard.

Sampling reduces the IDS workload which reduces the packet-processing time of the IDS. It relates to amount of time an IDS takes to analyse a packet. It consists of a number of security rules and the IDS workload. Figure 8 depicts average packet processing time for the SDN-Guard in which sampling rates were differed. When sampling is not considered, the average packet-processing time is about 14 s. The sampling rate later

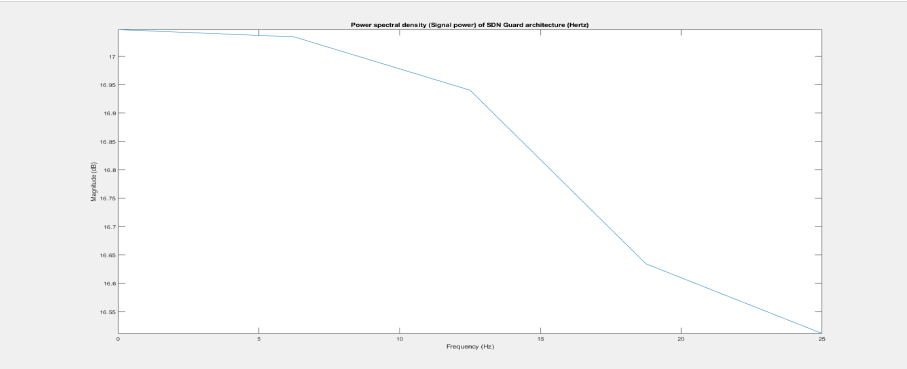

Fig. 7. Power Spectral density of SDN-Guard Architecture

decreased to 10.5 s and thereafter gradually as the sampling rate increases to 80%. Which means that as we increased the size of the sample, the processing time decreased which shows that the SDN-Guard can process large number of packets at a faster rate.

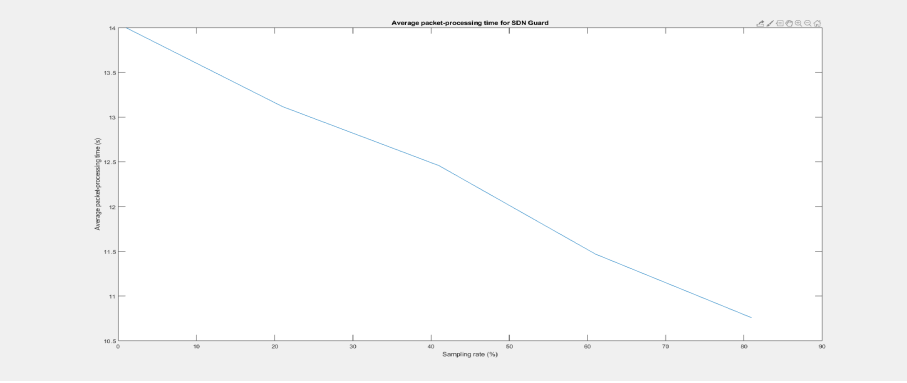

Fig. 8. Average packet processing time for SDN-Guard

Figure 9 shows the average packet processing time of the Jamming Attack Defender where sampling rates were differed. When sampling is not considered, the average packet-processing time is about 23 s. The sampling rate later decreased to 19.5 s as the sampling rate was increased to 80%. We observed that as we increased the number of packets the processing time decreased.

Given the results in Figs. 8 and 9, we can conclude that the sampling rate of 80% reduces the mirrored traffic while the packet-processing time reduces to 50% with IDS accuracy remaining at 100%. We can also conclude that the SDN-Guard reduces the packet processing time efficiently as compared to the Jamming Attack Defender. Which means that the SDN-Guard is superior to the Jamming Attack Defender.

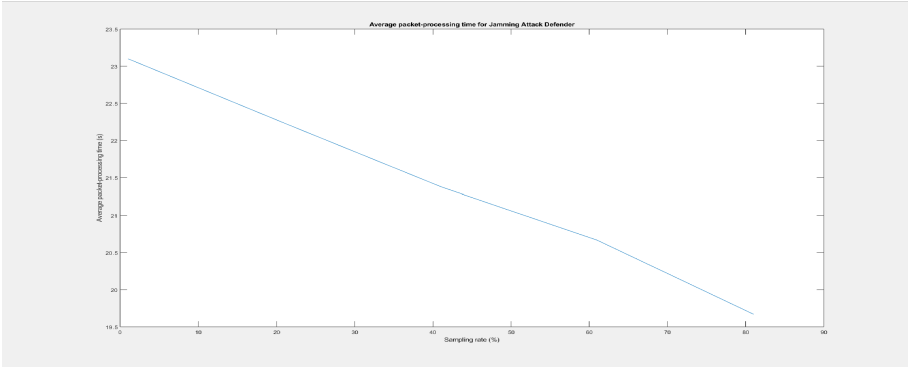

Fig. 9. Average packet processing time for Jamming Attack Defender

5 Conclusion

The study compared SDN-Guard to the Jamming Attack Defender. The objective was to evaluate the performance of the two schemes in order to have an in-depth understating of the two schemes with a view of designing a new scheme best on their best performing attributes of the two schemes. SDN-Guard and Jamming Attack Defender rely on IDS alarms in analysing the network traffic to efficiently protect the SD-CRN.

We also investigated the use of sampling to reduce mirrored traffic. We observed that the SDN-Guard is efficient in reducing the amount of mirrored traffic compared to the Jamming Attack Defender. We also observed that in terms of the source-to-destination RTT, the SDN-Guard takes less time compared to the Jamming Attack.

Lastly, we observed that the Jamming Attack Defender outperforms the Jamming Attack Defender in terms of PSD.

The main objective of the study was to compare the SDN-Guard and the Jamming Attack Defender to find out which scheme detects and mitigates the DoS in SD-CRN efficiently with a view of improving the two schemes. We considered the average round trip time, average packet processing time, and the Power Spectral Density. The results show that the SND-Guard outperforms the Jamming Attack Defender.

References

1. Weiss, A.: A Denial of Service attack can disrupt your organization's web site and network services. Here's how to defend yourself (2012)
2. Manogna, C., Naik, K.: Detection of jamming attack in cognitive radio networks. Int. J. Recent Adv. Eng. Technol. **2014**(6, 7), 2347–2812 (2014)
3. W, Xu, Trappe, W., Zhang, Y., Wood, T.: The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks. In: Proceedings ACM international symposium on Mobile Ad Hoc (2005)
4. Zhani, M.F., Dridi, L.: A holistic approach to mitigating DoS attacks in SDN networks. Int. J. Network Mgmt. **28**(1), e1996 (2017) (Montreal, QuebecH3C1K3,Canada)
5. Leavline, E.J., Dinesh, M.: Jamming attack detection technique in cognitive radio networks. In: Jamming Attack Detection Technique in Cognitive Radio Networks. India (2015)

6. Park, J., Cho, Z.Z.S.: A feasible method to combat against DDOD attack in SDN. In: International Conference on Information Networking (ICON) (2015)
7. Sahay, R., Blanc, G.: Towards autonomic DDoS mitigation using SDN. In: Network and Distributed System Security (NDSS) Symposium (201)5
8. Dridi, L., Zhani, M.F.: SDN-guard: DoS attacks mitigation in SDN. In: Ecole de Technologie Superieure(ETS). Canada (2008)
9. Hanen, I., Kevin, D., Mustafa, S.: Security challenges in cognitive radio networks. In: Proceedings of the World Cognress on Engineering. London, U.K (2014)
10. Sahay, R., Blanc, G., Zhang, Z., Debar, H.: Towards autonomic DDoS mitigation using software defined networking. In: Networks (2015)
11. Shirali, S.S., Ganjali, Y.: Flexible Sampling Extension for Monitoring and Security Applications in OpenFlow (2011)
12. Bharathi, N.A., Vetriselvi, V., Parthasarathi, R.: Mitigation of DoS in SDN Using Path Randomization. In: Smys, S., Bestak, R., Chen, J.I.-Z., Kotuliak, I. (eds.) International Conference on Computer Networks and Communication Technologies: ICCNCT 2018, pp. 229–239. Springer Singapore, Singapore (2019). https://doi.org/10.1007/978-981-10-8681-6_22
13. Jararweh, A.K.Y.: SD-CRN: software defined cognitive radio network framework. In: IEEE International Conference on Cloud Engineering. Boston, MA, USA (2014)
14. Lobna, D., Mohamed, F.Z.: SDN-Guard: DoS Attacks Mitigation in SDN Networks. Canada (2016)
15. Sahay, R., Blanc, G., Zhang, Z., Debar, H.: Towards autonomic DDoS mitigation using software defined networking. In: Network and Distributed System Security (NDSS) Symposium (2015)
16. Axelsson, S.: Intrusion detection systems: a survey and taxonomy. In: Technical report, Department of Computer Engineering, Chalmers University of Technology. Sweden (2000)