The Indonesian Journal of Computer Science
www.ijcs.net
Volume 14, Issue 5, October 2025
https://doi.org/10.33022/ijcs.v14i5.4951

Mitigation Of Denial of Service Attacks in Software-Defined-Cognitive Radio Networks Using Software-Defined-Cognitive Radio Shield

Mampuele Lebepe[1], Mthulisi Velempini[2]
201509073@keyaka.ul.ac.za[1], mthulisi.velempini@ul.ac.za[2]
[1,2] Department of Computer Science, University of Limpopo, Polokwane, 0727, South Africa

Article Information

Received : 8 Jul 2025
Revised : 17 Sep 2025
Accepted : 10 Oct 2025

Keywords

Software-defined network, Cognitive radio networks, intrusion detection system

Abstract

A user-friendly approach to managing network resources which solve a number of management-related concerns is provided by software-defined networks. While on the other hand, a novel paradigm called Cognitive Radio Networks (CRN) was developed to address spectrum scarcity by employing dynamic spectrum access. CRN enables secondary unlicensed users to utilize the licensed spectrum when idle without interfering with authorized users. Unfortunately, the two network technologies are susceptible to security attacks. Network security planning as the first step in network protection is therefore fundamental. The techniques employed by software-defined cognitive radio networks to detect and counteract denial of service (DoS) attacks are presented in this study. We then design an intrusion detection system (IDS) to address the effects of DoS attacks. In this approach, the IDS is connected to the software-defined cognitive radio network's controller. We considered the detection time or the amount of time it takes to detect an attack, the payload, the jitter, and the packet drop rate as evaluation metrics. The round trip time and throughput were also considered. To generate the results and compare them to those of existing schemes we used NetSim, which was installed on the Windows 10 Operating System. The simulation results show that the proposed scheme is efficient.

A. Introduction

Software Defined Networks (SDN) provide a unique approach to managing network resources which addresses a historic network management challenge [1]. While on the other hand, Cognitive Radio Networks (CRN) technology was proposed as a solution to the spectrum scarcity problem. It employs dynamic spectrum access techniques to enable secondary unlicensed users to opportunistically use the licensed spectrum when not in use. The main constraint is to ensure that the secondary users do not interfere with authorized users.

The study aims to design and evaluate the DoS attack mitigation scheme in SD-CRN. It also investigates the impact of IDS placement. To address the security issues in SD-CRN technologies, there is an increased interest in the implementation of SD-CRN. Related work is also reviewed and used to underpin the research approach of this study. The theories and methods which guided the selection of the study's approach are discussed.

We implement a DoS detection and mitigation scheme in SD-CRN. We also investigates the effectiveness of the SD-CRN Shield in reducing the effects of DoS attacks in the software-defined cognitive radio environment. The study enhances and secures emerging technologies and networks.

B. Related Work

In this Section, we review related work which is designed to address the DoS in SD-CRN. It reviews the findings of these schemes and identifies areas for further research. The design techniques and methods used are analyzed.

Availability attacks prevent users from utilizing the services of a given network. One such attack is the DoS attack. It prevents users from accessing the server and the services available in the network [5]. There are two types of DoS attacks, flooding and vulnerability attacks. The flooding attacker overwhelms the victim with streams of packets in an endeavour to exhaust or deplete its resources. The DoS vulnerability attack exploits vulnerabilities in systems for malicious purposes [6]. DoS attacks have increased in complexity. For example, flooding and vulnerability attacks have been integrated given the advancement of technology. It has become complex to the extent that it is difficult to distinguish between malicious and non-malicious traffic [7].

In [8], a technique for minimizing the consequences of DoS attacks in SDN by installing parallel flows was proposed. The scheme does not reject harmful packets but it deletes them and denies malicious users access to the network resources. It reduces significantly the control channel traffic and the utilization of the controller by installing flow rules on a single request from the source to the destination. It improves the SDN efficiency in relation to response time, CPU utilization and bandwidth.

Authors in [9] analysed the STRIDE model and made some recommendations regarding the prevention of DoS attacks in SDN. Rate limitation, flow timeout modifications, and flow aggregation were some of the recommendations.

The other challenging attack is the jamming attack however it can be easily detected in cognitive radio networks [10]. A jamming attack can be launched by both malicious users inside and outside the network. There is therefore a need for

an IDS given that cognitive radio networks and SDN are both susceptible to security attacks.

To detect anomaly behaviour in CRN, the cumulative sum (CUSUM) can be used while in cognitive radio networks, countermeasures can be implemented. For example, an IDS which reduces attacks in cognitive radio networks can be considered.

To counter DoS attacks, the Multi-Layer Fair Queuing (MLFQ) approach was proposed in [11]. The scheme restricts the Packet-In message queues in the controller and minimizes DoS threats. The controller pools requests from various queues using the Weighted Round Robin (WRR) algorithm. A queue is considered as a per-switch queue when the amount of packets in the queue reaches a predetermined threshold.

A queue is changed to a per-port queue if it is still growing. The queues are then combined into one queue. However, if the size of the sub-queue falls below the set threshold, the non-malicious hosts connected to the switch experience an additional delay when under attack due to the increased computation. The queues are combined into one queue if the size of the sub-queue falls below the threshold. The multi-layer queue requires a lot of computation. As a result, nodes experience a lot of delay during an attack.

The Flow-Ranger system, which was proposed in [12] detects and mitigates DoS threats. It executes on the controller and has three parts: trust management which assigns a trust value to each packet-in message based on its source; queuing management which assigns the message priority based on its level of trust and message scheduling which handles messages using a weighted Round Robin algorithm. By ensuring that genuine flows are served first in the controller, Flow-Ranger reduces the effects of DoS attacks and improves network performance.

Distributed Denial of Service (DDoS) attacks target the SDN controller due to its vulnerability [9] [10]. Although the SDN controller is exposed and targeted by DDoS attacks, the authors in [11] argue that the controller presents an opportunity to reduce DDoS attacks within cloud computing for example, in the case of the Flow Ranger.

Flood-Guard was proposed in [12]. It is designed to counter DoS attacks by addressing the "flow request flooding" attacks. When the controller detects a DoS attack, the scheme directs the switch which is under attack to reroute all the new flows to a data location cache. Unfortunately, the Flood Guard is complex as a result it may not be fully executed which may require the placement of additional devices on the data plane.

In [13], the authors proposed an Avant-Guard. When the controller is exposed, TCP SYN flooding can overwhelm its resources, causing the CPU, memory, and bandwidth to be exhausted by flooded requests. In this scheme, Switches can delegate all TCP connections using Avant-Guard. The Avant-Guard submits a flow request after the completion of the handshake for a Transmission Control Protocol (TCP) connection. Unfortunately, this scheme is optimized to prevent only SYN flooding.

The authors in [14] developed a performance and security analysis approach model. This model determines the ideal encryption key length for the best performance and security level of the network. The expected encryption duration

when the network performs at its peak is measured by the model. As seen in Figure 1, the model is equipped with an IDS. However, the model cannot detect DoS attacks in SD-CRN and can only detect attacks which intercept and modify data.

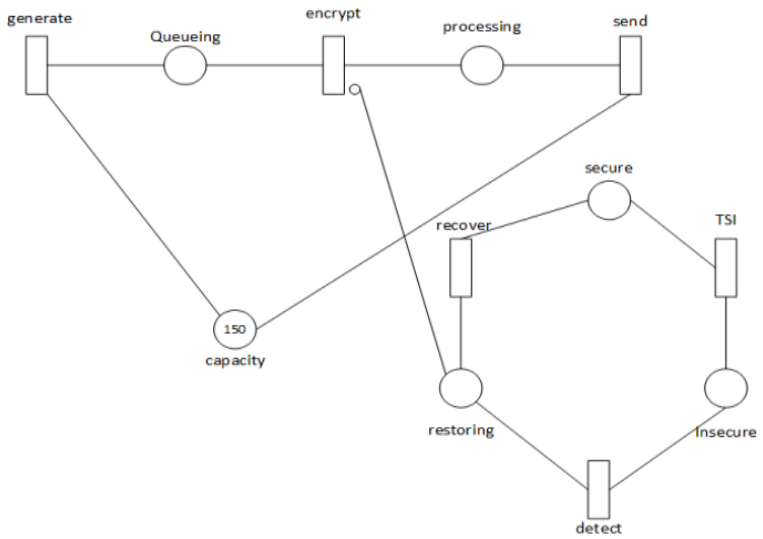

Figure 1. Petri net model for combined performance and security analysis

In SDN, a SRL architecture was proposed to mitigate TCP SYN-flooding in [15]. Two modules from the SRL that are implemented in the controller assist in minimizing the effects of the attack. There are two modules, a flow module and a hashing module. The hash module determines the hash values using the IP address of the SYN flood packets. Then the flow rules are stored in the controller according to their hash value. The ones with low hash values are discarded since they are assumed to be from malicious nodes. Wildcard rules are computed using spoofed IP and MAC addresses by the flow module. Then, all the requests coming from the malicious IP addresses are blocked.

A Flow-IDS system as shown in Figure 2 was proposed in [16]. It is designed to detect and reduce Simple Mail Transfer Protocol (SMTP) DoS Flood attacks in SDN. The scheme is used to detect SMTP flows which are malicious. To determine if the flows are legitimate or not, Deep Learning (DL) and Decision Tree (DT) algorithms are used. The authors demonstrated that SMTP attacks on SDN can be detected and blocked using Flow-IDS.

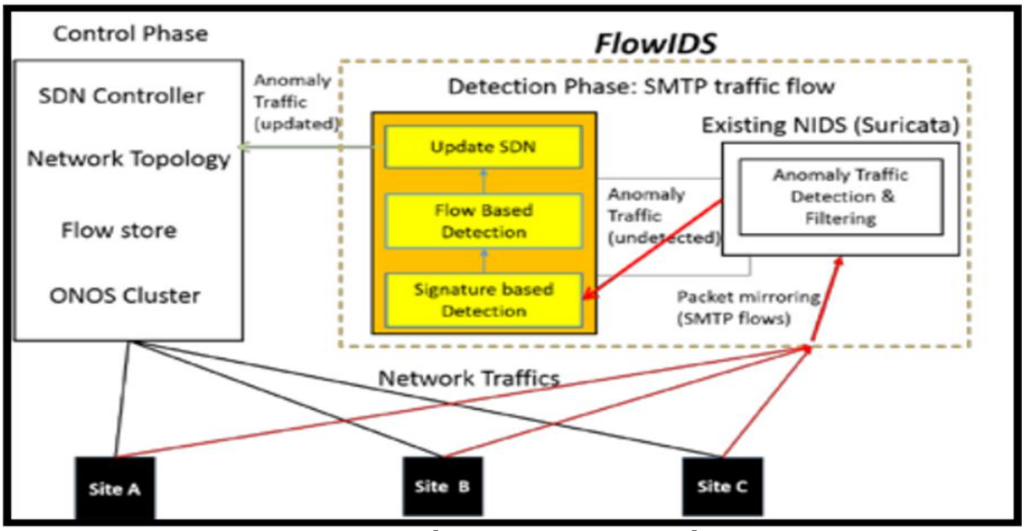

Figure 2. Flow-IDS Framework

In order to minimize the impact of DoS in SDN, Shin in [17] proposed a strategy which implements parallel flows. The approach was able to reduce the DoS attack without impacting on network administration and monitoring. Given a request from the source, flow rules are installed in all the source to destination switches. This reduces the volume of traffic on the control channel and also reduces the utilization of the controller. The CPU utilization, flow requests, control channel bandwidth and response time of the SDN are also improved.

Authors in [18] proposed a SDN Manager. It has up to five parts that cooperate to minimize DoS attacks on SDN. A storage device, an updater, a checker, a monitor and a forecast engine are some of the components of the SDN Manager. The scheme uses a loop to read the statistics of the flows. The network is updated after anticipating changes in the flow based on collected data. Results demonstrated that the approach marginally improves SDN performance.

In cognitive radio networks, it might be challenging to distinguish between networks experiencing DoS attacks and networks that are congested [19]. In order to address this, queue length, packet loss probability, and throughput are calculated in conjunction with the packet send ratio and packet delivery ratio to generate the results which can be used to distinguish between the two.

A scheme known as the SDN-Guard was proposed in [20]. The scheme reduces the workload, average processing time, network bandwidth, controller load, switch-to-controller bandwidth, and the detection time.

DoS attacks continue to pose a threat to the Internet because they improve and increase their complexity with the advancement of technology which makes detection and mitigation a challenging task. There is a need for an approach which could simultaneously implement IDS placement and detect DoS in SD-CRN.

Wireless networks are at risk of DoS attacks. Unfortunately, wireless network security is weak while attacks are getting more sophisticated and complex. However, the Jamming Attack Defender [10] and the SDN-Guard [20] are the best-performing strategies in addressing the effects of DoS in SD-CRN. To improve the performance of IDS, the two strategies may be integrated and enhanced.

C. System Design

SD-CRN Shield was proposed in this work. It is a scheme which is implemented in an SDN controller. In order to inspect the traffic of the network and raise alarms when malicious traffic is detected, an IDS is employed.

The platform used and the parameters are shown in Tables 1 and 2, respectively.

Table 1. Platform for simulation

Computer	**DELL Vostro 3591**
RAM	12,00 GB
CPU	Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz
OS	Microsoft Windows 10 Pro

Table 2. Parameters and tools

Parameters Used	**Tools Used**
Network Simulator	NetSim
Simulation Area	500m*500m

NetSim was used to emulate a given network topology. NetSim builds a network of virtual hosts, connections, controllers, and switches.

The SD-CRN Shield is modelled after the SDN-Guard and the Jamming Attack Defender. The scheme integrates the two schemes into one to produce a hybrid scheme. It consists of a flow management unit, which selects paths to route the flows and sets the hard timeout for the correlating Ternary Content-Addressable Memory (TCAM) entries. This is done based on the likelihood that the flow poses a threat and flows are managed to reduce the impact of DoS attacks. To reduce the number of entries utilized in the switches' Ternary Content-Addressable Memory, a rule aggregation module collects flow entries of malicious traffic.

To alert the SD-CRN Shield of any anomalies, an IDS communicates with the shield as the ongoing behaviour is monitored. The proposed scheme is based on the following three design considerations in mitigating the DoS attacks:

Timeout management: Using the IDS, determine the timeout value for each flow rule based on the likelihood of a threat.

Aggregation of malicious flow rules: Hard timeouts are assigned to malicious flows. These are captured in the TCAM tables and stored for a considerable amount of time. As a result, it may be overloaded with used entries. In order to address this challenge, malicious flow entries are merged if they have the same source-destination pairs and the same outgoing link.

Threat-based routing: This method routes malicious data through the network's least-used channels to conserve bandwidth.

The following metrics are used to evaluate the performance of the proposed scheme:

Packet Drop Rate:

By observing the packet drop rate (PDR) and signal strength (SS), a DoS attack can be detected. The PDR of a user is calculated (using equation (1)) as the ratio of packets transmitted to packets received by the user. If a secondary or unauthorized user is attacked its SS is investigated. If the SS is too high, the packet is dropped. To detect PDR changes of the secondary user, the CUSUM algorithm is used.

One way to determine the PDR is:

For q(q ≤ n)

$$PDR=q!/n!(n-q)!*p^q \llbracket(1-p)\rrbracket^(n-q) \quad (1)$$

where n = a number of trials.
q = number of success
p = probability of success

Each packet contains the SS, which is obtained from the physical layer.

Jitter:

Jitter is used to assess the performance of the network. It is the variable delay in the sending data packets and is measured by computing the average time difference between each packet.

Round trip time

The Round trip time can increase beyond the duration in the presence of the attack.

Throughput:

To examine the behaviour of controllers, we evaluate the source-to-destination achievable throughput as well as the incoming controller throughput.

Payload:

Payload the portion of malware that is added to the size of the packet.

Detection time:

Detection time is the time it takes to detect an attack. This metric is important because the sooner an attack is detected; the sooner it can be mitigated.

In networks that use Dynamic source routing protocol (DSR) at Layer 3, the SD-CRN Shield can avoid malicious nodes and choose less risky paths. The function validates the route reply in the route cache and detect malicious nodes. When the node is malicious a route reply is not processed.

When a malicious node is detected, its route entry is deleted from the cache. The IDS also runs at Layer 2 to detect DoS at lower layers. A watchdog timer starts when a packet is sent if _NETSIM WATCHDOG_ is defined. After a packet has been

forwarded to the next hop node, the current node checks the watchdog timer duration to decide whether to keep sending packets to the target node.

The malicious node does not forward incoming packets if it's watchdog timer times out at the node that transmitted the packet. The scheme has a counter which records the time the watchdog timer expired. These are the packets that are sent but are not forwarded by the next hop node. The current node flags the next hop node as malicious when the value of this counter is more than the failure threshold.

D. Simulation Environment

We created 5 scenarios as depicted in Figures 3-7. The network scenarios consist of two Wired Nodes, one Layer 2 Switch, two routers, one Access Point, and one wireless node in the grid environment which consists of malicious nodes. The traffic is transmitted from the Wired node to the Wireless node. Figure 3 shows a network without malicious nodes. Figure 4 shows the network with one malicious node. Figure 5 shows the network with two malicious nodes while in Figure 6 there are five malicious nodes. Lastly in Figure 7, there are ten malicious nodes.

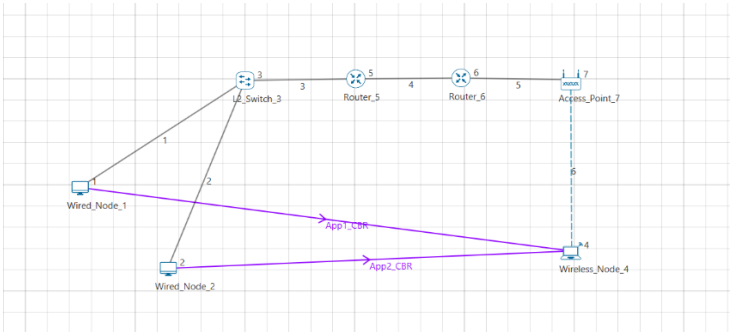

Figure 3. Scenario 1- Analysis of a Network without Malicious Nodes

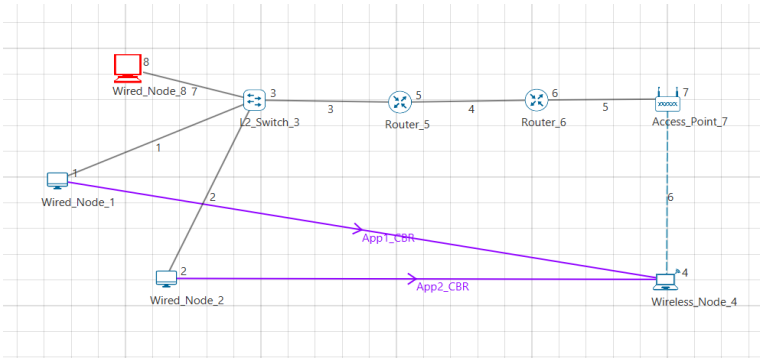

Figure 4. Scenario 2 – A Network with one Malicious Node

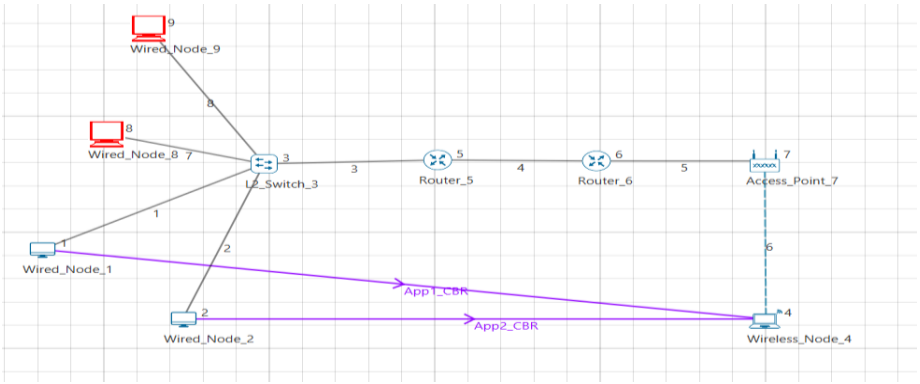

Figure 5. Scenario 3 - A Network with two Malicious Nodes

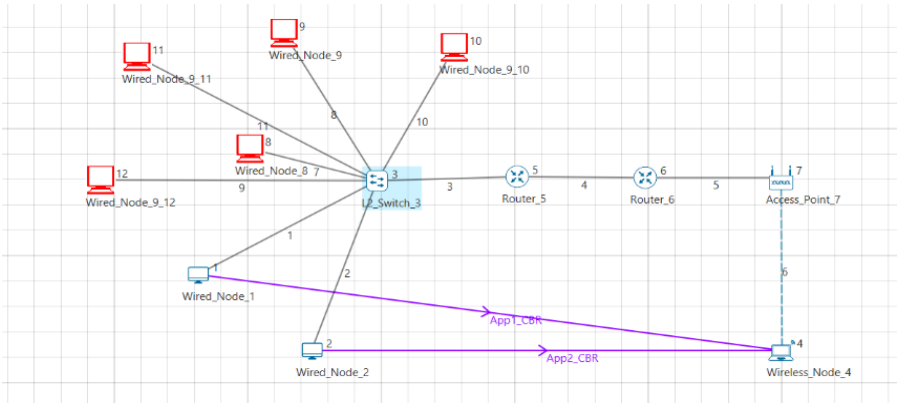

Figure 6. Scenario 4 - A Network with five Malicious Nodes

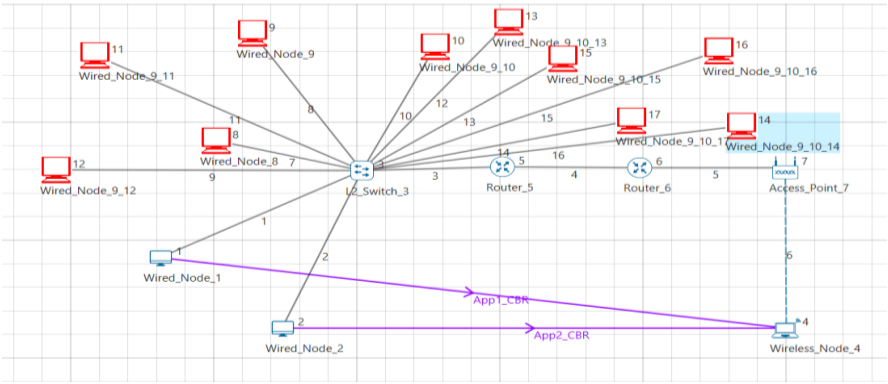

Figure 7. Scenario 4 - A Network with ten Malicious Nodes

E. Results

We propose a scheme called Software Defined-Cognitive Radio Shield which we evaluated using the following five metrics: detection time, PDR, RTT, jitter, payload, and throughput. The results of our experiments are presented in this Section.

IDS Placement

Using sampled traffic, we evaluated the IDS's accuracy in our analysis. We examined how fast packets were processed at various sampling rates. In our

experiment, we generated a TCP-SYN flooding DoS attack and conducted a number of trials with various sampling rates.

The best possible location for the IDS is one which minimizes traffic to the IDS location and minimizes bandwidth consumption by the traffic. We used 8 switches for our research. We first installed the IDS at the controller before placing it on each of the 8 other switches. The results demonstrate that the controller, as depicted in Figure 8, is the ideal location for the IDS.

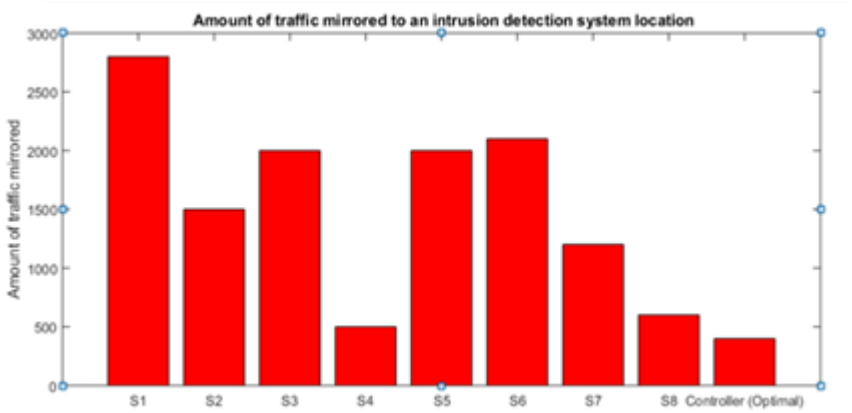

Figure 8. Amount of traffic mirrored to the IDS location

Comparing the controller to switches 1 through 8, the controller has the least amount of mirrored traffic. This shows that it is the ideal location. S4 has the second least amount of mirrored traffic but it is not the lowest, so we take the lowest for the best place to locate the IDS. Three simulations were run for this scenario. A scenario with one malicious node was considered in Figure 4.

Throughput

We evaluated the performance of the two schemes in terms of throughput. Source-to-destination achievable throughput and incoming controller throughput were measured to evaluate the performance of the controllers. The throughput results of the five scenarios are shown in Table 3.

Table 3. Throughput scenarios

	SDN-GUARD (MBPS)	**SD-CRN SHIELD (MBPS)**
SCENARIO 1: 0 MALICIOUS NODES	0.520928	0.520928
SCENARIO 2: 1 MALICIOUS NODE	0.810496	0.578160
SCENARIO 3: 2 MALICIOUS NODES	0.317328	0.286160
SCENARIO 4: 5 MALICIOUS NODES	0.254457	0.122640
SCENARIO 5: 10 MALICIOUS NODES	0.094385	0.061904

Given the five scenarios we considered, we observed that the throughput of the malicious nodes decreases for both applications as we increased the number of

malicious nodes. This is caused by the SYN flood from the malicious nodes. In scenario 1, there is no malicious node so the throughput remained unchanged. In scenario 2, we had one malicious node and the results show that the SD-CRN shield was able to reduce the throughput from 0.810496 to 0.578160. While in scenario 3 we had two malicious nodes and the achievable throughput further decreased from 0.317328 to 0.286160. In scenario 4, the throughput decreased from 0.254457 to 0.122640. This pattern was observed in scenario 5 in which the throughput decreased from 0.094385 to 0.061904. The throughput results are shown in Figure 9.

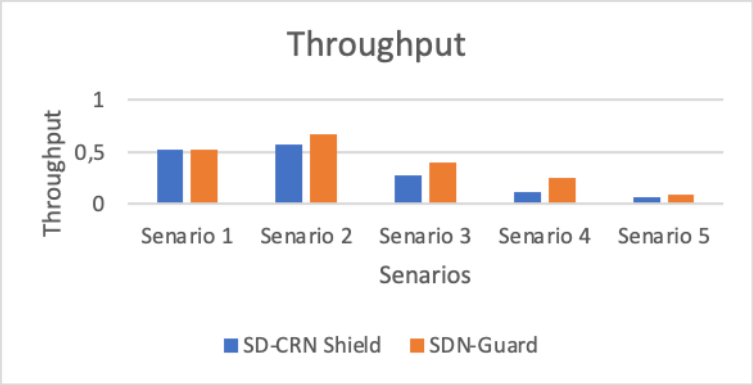

Figure 9. Throughput analysis

The results show that the throughput for scenarios 2 – 5 of the SDN-Guard is higher than that of the SD-CRN Shield. However, it is clear that the SD-CRN Shield significantly lowers the throughput of malicious nodes. The SD-CRN Shield reduces the need for the controller, adds new flow rules by setting high hard timeouts for the malicious traffic. There is also a decrease in the number of packets lost in the SD-CRN Shield because the malicious traffic is load balanced across the least utilized links which minimizes congestion.

The results from the 5 scenarios illustrate that the proposed scheme was able to reduce the throughput of the malicious nodes. Therefore, this means that the SD-CRN Shield is efficient in reducing the throughput of malicious nodes. The scheme also detects the DoS attack and mitigates it by reducing the throughput of the attack.

Packet Drop Rate

We investigated the packet drop rate in the presence of the DoS attack and observed the number of packets our scheme could prevent from being lost. Figure 10 presents the PDR when the DoS attack is launched without the SD-CRN scheme being activated.

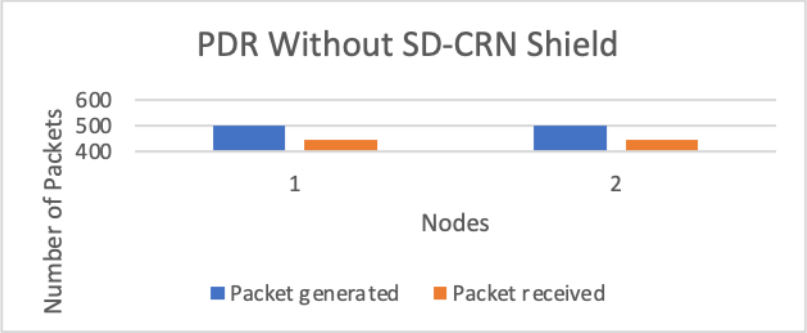

Figure 10. PDR Without SD-CRN Shield

We observed that many packets were dropped when the DoS attack was launched. Node 1 and node 2 generated 500 packets each but received 448 packets and 444 packets respectively. This means that 11.2% of the packets were dropped. Figure 11 shows the results when the SD-CRN Shield is implemented.

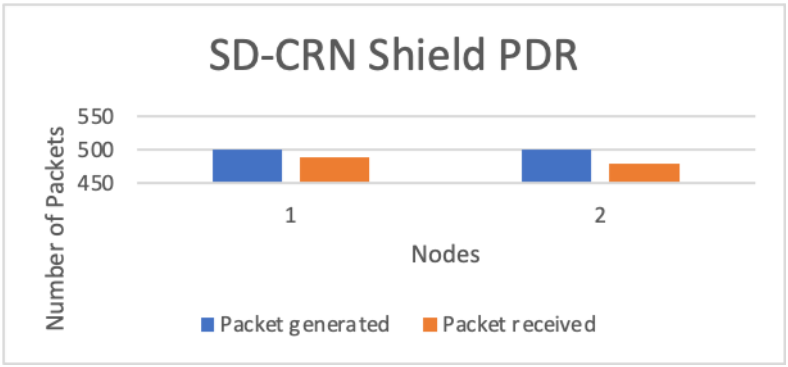

Figure11. PDR with SD-CRN Shield

The results in Figure 11 show that fewer packets were dropped when the SD-CRN Shield was implemented. Node 1 and node 2 generated 500 packets each and it can be observed that Node 1 received 490 while Node 2 received 480 packets. Our scheme was able to reduce the number of dropped packets during an attack. The scheme is therefore efficient in reducing dropped packets during the attack. We then compared the performance of SD-CRN Shield to the SDN-Guard using the PDR. Node 1 sends packets to Node 4 through App1_CBR as shown in figures 12 and then receives packets from the switch. Node 2 sends packets to Node 4 through the App2_CBR as shown in figures 12 and receives packets from the switch.

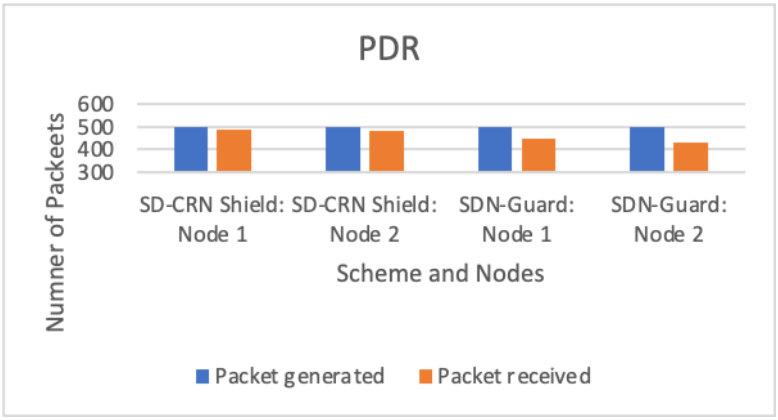

Figure 12. Comparative PDR Results

In the SDN-Guard, fewer packets were received compared to the SD-CRN Shield. The same number of packets were generated for the two schemes. This means that more packets are dropped in the case of the SDN-Guard. We therefore conclude that the SD-CRN is superior in terms of PDR compared to the SDN-Guard.

Round Trip Time

This metric is important in measuring how fast data is transmitted when a DoS attack is launched. Figure 13 shows the performance of RTT when the SD-CRN is not deployed.

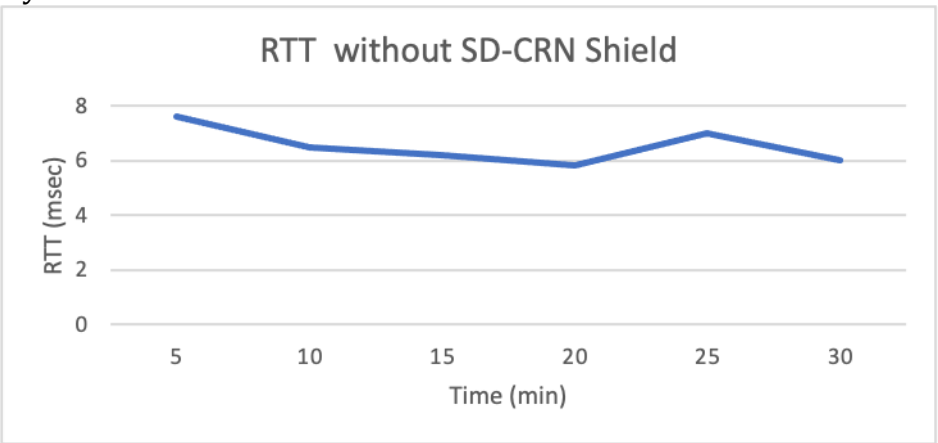

Figure 13. RTT without SD-CRN-Shield

When the simulation is run without the SD-CRN, we observe that the RRT is high. This means that nodes take long time to send and receive data because packets are lost during the DoS attack. The DoS also interfere with the normal transmission of packets. There is no layer of protection to prevent the packets from being dropped. Figure 14 presents the RTT results when the SD-CRN is deployed.

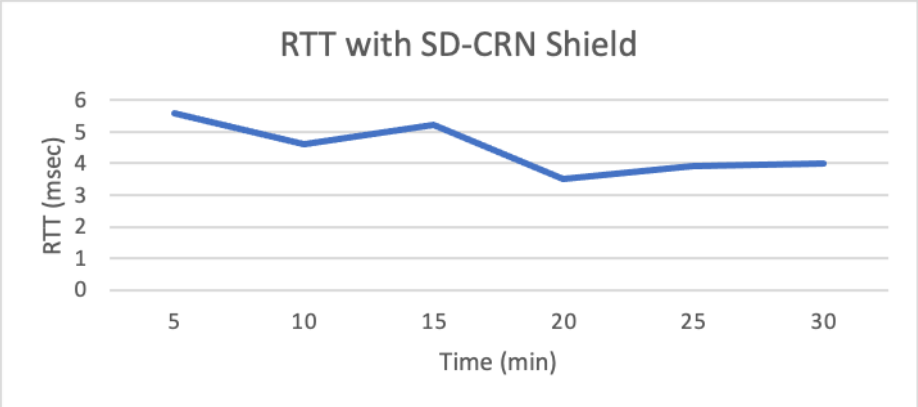

Figure 14. RTT with SD-CRN Shield

In Figure 14 we observed that the RTT has reduced. This means that our scheme is able to reduce the RTT when there is a DoS attack, proving that this scheme is effective. We then compared the performance of SD-CRN Shield to the SDN-Guard based on the RTT metric in Figure 15.

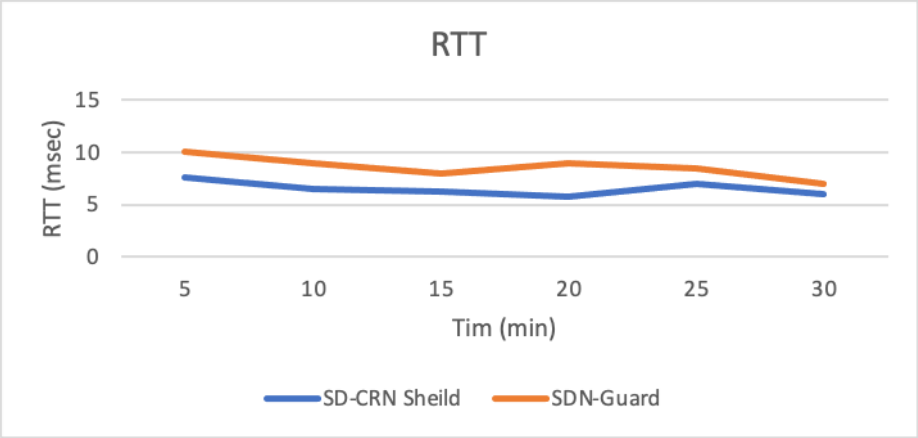

Figure 15. Comparative RTT Results

The figure demonstrates that the SDN-Guard RTT is higher than that of the SD-CRN Shield, meaning that when the attack is launched, the SD-CRN Shield takes less time to send and receive packets. We can therefore conclude that our scheme is superior to the SDN-Guard.

Payload

In the context of a DoS attack, the payload is the portion of malware that the attacker intends to deliver to the victim. We compared the payload of the SD-CRN Shield to the one of the SDN-Guard in Figure 16.

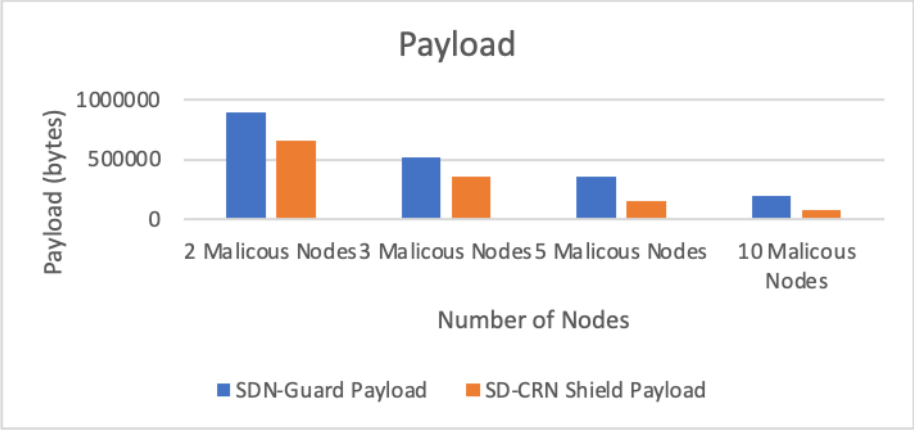

Figure 16. The Analysis of SD-CRN Shield and SDN-Guard Payload Results

The results show that the payload of the SDN-Guard is greater than the one of the SD-CRN Shield in all the cases. This means that the network is more susceptible to DoS attacks in the case of the SDN-Guard. The SD-CRN Shield was able to reduce the payload when we had 2, 3, 5 and 10 malicious nodes which demonstrates that the SD-CRN Shield is superior in terms of reducing DoS attacks in SD-CRNs.

Jitter

Jitter is the variation in the time in delay such as the variability in ping. Jitter is used to assess the performance of the network. It is evident when there is a time delay in sending data packets over a network connection and the delay varies with time. In Figure 17, we evaluated the Jitter of the SD-CRN Shield and compared it to the one of the SDN-Guard.

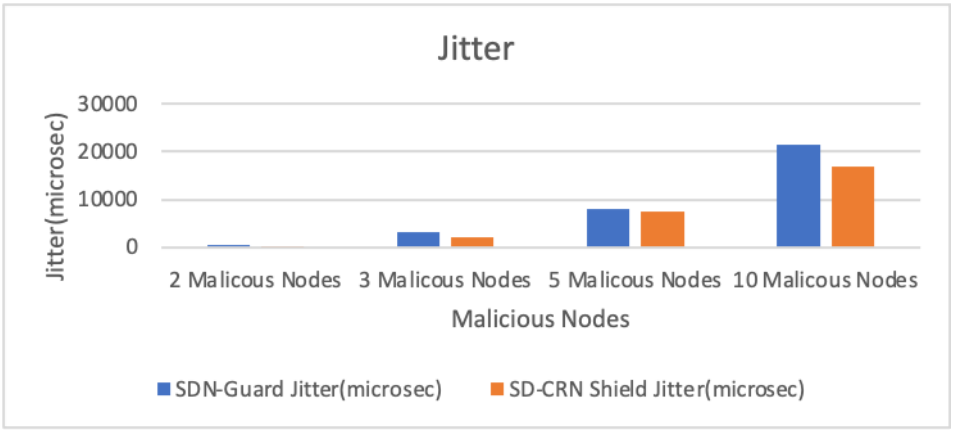

Figure 17. The Comparative Jitter Results

The results in Figure 17 show that the SD-CRN shield has less jitter compared to the SDN-Guard in scenarios with 2, 3, 5, and 10 malicious nodes. For 2 malicious nodes, the jitter of the SDN-Guard is 411,21ms while the one of the SD-CRN Shield is 233,25ms. In 3 malicious nodes, the jitter for the SDN-Guard is 3145,22ms and the jitter of SD-CRN Shield is 2072,66ms. While in the scenarios with 5 malicious nodes, the jitter for the SDN-Guard is 8111,2ms and the for the SD-CRN Shield is 7578ms. Lastly, the one with 10 malicious nodes, the jitter for the SDN-Guard is

21446,54ms and SD-CRN Shield is 17021,15ms. This shows that the SD-CRN Shield is effective in addressing the effects of DoS attacks.

It is evident that the SD-CRN Shield achieved better jitter results, making the SD-CRN Shield a better scheme because a high jitter level is not good for network reliability. The scheme is also effective in addressing the effects of DoS.

Detection Time

We evaluated the detection times of the SD-CRN Shield and the SDN-Guard because early detection of an attack is desirable as a quality-of-service metric. The detection time results are shown in Figure 18.

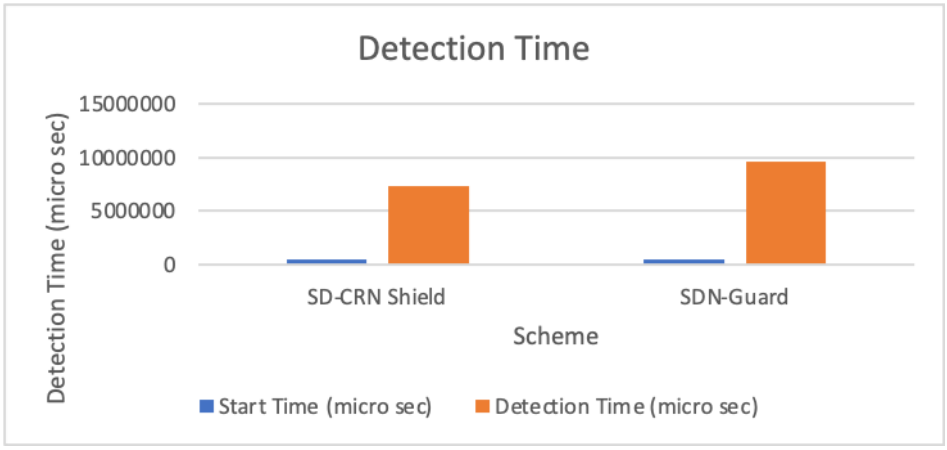

Figure 18. Investigating the Detection Times of the two Schemes

The results show the Start Time and the Detection Time of the DoS attack of the SD-CRN Shield and the SDN-Guard. The start time of the attack is 500000µs for both schemes. The detection time for the SD-CRN Shield is 7386986,02µs and the one for the SDN-Guard is 9595413,82µs. It is evident that the SD-CRN Shield detects the attack earlier compared to the SDN-Guard. We, therefore, conclude that SD-CRN Shield scheme has a higher detection rate coupled with the earlier detection time.

This research presented the results of the simulations and the different scenarios considered in this study. The results generated were also interpreted and discussed. We evaluated the results of our scheme, the SD-CRN Shield and compared its performance to the SDN-Guard. We evaluated the throughput, PDR, RTT, payload, jitter and detection time results of the two schemes. The comparative results show that our scheme mitigates the effects of the DoS attack in SD-CRN under different scenarios more effectively than the SDN-Guard. This shows that the SD-CRN Shield is superior.

F. Conclusion

The adoption of SD-CRN software is on the rise, therefore addressing its security challenges is more urgent than before. In this work, we focused on DoS attacks. DoS attacks overwhelm servers or networks making it inaccessible. We simulated SD-CRN and a DoS attack. We then designed the SD-CRN Shield to detect

and mitigate the DoS attack. To evaluate the efficiency of the SD-CRN Shield, we compared its performance to the SDN-Guard.

The proposed SD-CRN Shield detects and mitigates DoS attacks in SD-CRNs. We considered 5 scenarios to evaluate the effectiveness of SD-CRN Shield and its performance. The comparative results of SD-CRN Shields and SDN-Guard are presented and analyzed. The first scenario has no malicious nodes, the second has 1 malicious node, the third has 2 malicious nodes, the fourth scenario has 5 malicious nodes and the fifth scenario has 10 malicious nodes. The analysis of the simulation results showed that our scheme, the SD-CRN Shield, performed well given the different scenarios in terms of throughput, PDR, jitter, payload, detection time, and RTT. Our scheme achieved a faster detection time and lower jitter. The payload of our scheme was less compared to the SDN-Guard. Our results therefore demonstrate that the SD-CRN Shield is an effective scheme in detecting and mitigating DoS attacks in SD-CRN.

The SD-CRN Shield could be improved in terms of detection latency and detection capability. In addition, it could be compared to the other schemes. Other metrics such as CPU Usage and Threat Detection Time could be used to evaluate the performance of improved schemes.

G. Acknowledgment

This work is based on the research supported wholly / in part by the National Research Foundation of South Africa (Grant Numbers: 141918)

H. References

[1] K. Y. Jararweh, "SD-CRN:Software Defined Cognitive Radio Network Framework," in 2014 IEEE International Conference on Cloud Engineering, 2014.

[2] S. Kumar, M. Sachdeva and . D. K. Kumar, "Flooding Based DDoS Attacks and Their Influence on Web Services," International Journal of Computer Science and Information Technologies, vol. 2, pp. 1131-1136, 2011.

[3] C. Douligeris and A. Mitrokotsa, "DDoS attacks and defense mechanisms: classification and state-of-the-art," Computer Networks, vol. 44, no. 5, pp. 643-666, 2004.

[4] A. Kuzmanovic and E. W. Knightly, "Low-Rate TCP-Targeted Denial of Service Attacks and Counter Strategies," IEEE/ACM TRANSACTIONS ON NETWORKING, vol. 14, no. 4, 2006.

[5] M. Imran, M. D. Hanif, F. A. Khan and A. Derhab, "Reducing the effects of DoS attacks in software-defined networks using parallel flow installation," Saudi Arabia, 2019.

[6] R. Kloti, V. Kotronis and P. Smith, "OpenFlow: A Security Analysis," in 21st IEEE international conference on network, New York, 2013.

[7] E. J. Leavline, M. Dinesh and D. A. A. G. Singh, "Jamming Attack Detection Technique in Cognitive Radio," International Journal of Applied Engineering Research, vol. 10, no. 55, pp. 2347 - 2812, 2015.

[8] P. Zhang, H. Wang, C. Hu and C. Lin, "On Denial of Service Attacks in Software Defined Networks," IEEE Network, vol. 30, no. 6, pp. 28-33, 2016.

[9] L. Wei and C. Fung, ""FlowRanger: A request prioritizing algorithm for controller DoS attacks in software-defined networks," in 2015 IEEE International Conference on Communications (ICC), London, UK, 2015.
[10] S. S. Hayward, S. Natarajan and S. Sezer, "A survey of security in software defined networks," IEEE Communications Surveys & Tutorials, vol. 18, pp. 623-654, 2016.
[11] I. Ahmad, S. Namal and M. Ylianttila, "Security in sofware defned networks: a survey," IEEE Communications Surveys& IEEE Communications Surveys&, vol. 17, pp. 2317-2346, 2015.
[12] Q. Yan and F. R. Yu, "Distributed denial of service attacks in software-defined networking with cloud computing," IEEE Communications Magazine, vol. 53, pp. 52-59, 2015.
[13] H. Wang, L. Xu and G. Gu, "FloodGuard: A DoS Attack Prevention Extension in Software-Defined Networks," in Proc IEEE/IFIP Int'l. Conf. Dependable Systems and Networks, 2015.
[14] S. Shin, "AVANT-GUARD: Scalable and Vigilant Switch Flow Management in Software-Defined Networks," in Proc. ACM CCS, 2013.
[15] K. Wolter and Reineck, "Performance and security tradeoff.," In International School on Formal Methods for the Design of Computer, Communication and Software System, Springer Berlin Heidelberg, 2010.
[16] T. Ubale and J. A. Kumar, "SRL: An TCP SYNFLOOD DDoS Mitigation Approach in Software-Defined Networks," in Proceedings of the 2nd International Conference on Electronics, Communication and Aerospace Technology, 2018.
[17] M. Z. A. Aziz and K. Okamura, "Leveraging SDN for Detection and Mitigation SMTP Flood Attack through Deep Learning Analysis Techniques," International Journal of Computer Science and Network Security, vol. 17, no. 10, pp. 166-172, 2017.
[18] M. Imran, M. H. Durad, F. A. Khan and A. Derhab, "Reducing the efects of DoS attacks in software defned networks using parallel fow," in Hum. Cent. Comput. Inf. Sci, 2019.
[19] T. Wang, H. Chen, G. Cheng and Y. Lu, "SDNManager: A Safeguard Architecture for SDN DoS Attacks Based on Bandwidth Prediction," Security and Communication Networks, p. 16, 24 November 2017.
[20] C. Ejike and D. Kouvatsos, "Detection of Network Congestion and Denial of Service (DoS) Attacks in Cognitive Radio Networks," in 2019 7th International Conference on Future Internet of Things and Cloud, 2019.
[21] L. Dridi and M. F. Zhani, "SDN-Guard: DoS Attacks Mitigation in SDN," in Ecole de Technologie Superieure(ETS), Canada, 2016.