DEAKIN UNIVERSITY

Statement and Confirmation of Own Work

A signed copy of this form must be submitted with every assignment.
If the statement is missing your work may not be marked.

Student Declaration

I confirm the following details:

Student Name:	Kenisha Corera
Student ID Number:	BSCP|CS|63|134
Qualification:	Bachelor in Cyber Security
Unit:	SIT325 Advanced Network Security
Centre:	CICRA Campus
Word Count:	1843

I have read and understood both *Deakin Academic Misconduct Policy* and the *Referencing and Bibliographies* document. To the best of my knowledge my work has been accurately referenced and all sources cited correctly.

I confirm that I have not exceeded the stipulated word limit by more than 10%.

I confirm that this is my own work and that I have not colluded or plagiarized any part of it.

Candidate Signature:	J. K.
Date:	02/02/2025

Task 4.3HD

Table of Contents

Step 1: Creating the Strategy for Peer Collaboration (Support)…………………....................6

1.1 Comprehending the Peer-Support Approach

1.2 Discussion with Tutor

Step 2: Configuring the Test Bed for the Experiment ………………………………………..6

2.1 Test Configurations and Specifications

2.2 Topology of the Network

2.3 Creating a DDoS Attack Simulation

2.4 Evaluation Metrics

Step 3: Report…………………………………………………………………................15

3.1 Approach

3.2 Results of the Experiment

3.3 The findings

Conclusion..………………………………………………………………………………….19

References...…………………………………………………………………………………22

Table of Figures

Figure 1……………………………………………………………………………………...7
Figure 2……………………………………………………………………………………...7
Figure 3……………………………………………………………………………………...7
Figure 4……………………………………………………………………………………...8
Figure 5……………………………………………………………………………………...9
Figure 6……………………………………………………………………………………...9
Figure 7………………………………………………………………………………………10
Figure 8………………………………………………………………………………………10
Figure 9………………………………………………………………………………………11
Figure 10……………………………………………………………………………………..11
Figure 11……………………………………………………………………………………..12
Figure 12……………………………………………………………………………………..12
Figure 13……………………………………………………………………………………..13
Figure 14……………………………………………………………………………………..13
Figure 15……………………………………………………………………………………..14
Figure 16……………………………………………………………………………………..14

Table of Abbreviations

Abbreviations	Definitions
DDoS	Distributed denial of service
LOIC	Low Orbit Ion Cannon
ONOS	Open Network Operating System
SDN	Software Defined Network
CPU	Central Processing Unit
TCP	Transmission Control Protocol

Table of Tables

Table 1……………………………………………………………………………………6
Table 2…………………………………………………………………………………..12
Table 3…………………………………………………………………………………..16

Step 1: Creating the Strategy for Peer Collaboration (Support)

1.1 Comprehending the Peer-Support Approach

The research paper "Defending against Flow Table Overloading Attack in Software-Defined Networks" by Bin Yuan, Wei Xiao, Chung and Min demonstrates the operation of the peer-support strategy through its evaluation. This research shows how SDN switches facing attack could access idle flow table zones from neighboring switches to decrease the impact of denial of service attacks targeting specific flow tables.

1.2 Discussion with Tutor

Before setting up the peer-support program I would arrange with my tutor a discussion about the implementation process to gain familiarity with the entire system. The talk will establish critical values including the experimental setup and test condition settings as well as measurement protocols.

Step 2: Configuring the Test Bed for the Experiment

2.1 Test Configurations and Specifications

Parameter	Value
Number of switches	10
Controller	POX (version 0.2.0)
Topology	Mesh topology
Flow Table Size (per Switch)	1000 flows
Attack Rate (DDoS)	500 Packets per second
Peer support scheme	Enabled/ Disabled
Test Duration	10 minutes per scenario

Table 1: Test settings and parameters

2.2 Topology of the Network

I selected 10 switches to use as the hardware design but the mesh topology stands as my preferred choice. The research paper technique enables the reproduction of conditions using peer support methods which matches the current experimental setup. The system contains ONOS as its second switch that manages the flow tables for all switches.

The initial step was opening onos through docker followed by running AARNet.py to obtain the topology display inside the onos system.

Figure 1: Running onos in docker

Figure 2: Onos login

Figure 3: There are currently no devices connected.

Now I wrote AARNet.py for running the mininet,

```python
from mininet.net import Mininet
from mininet.node import RemoteController
from mininet.link import TCLink
from mininet.topo import Topo
from mininet.cli import CLI
from mininet.log import setLogLevel

class MeshTopo(Topo):
    def build(self, n=10):
        switches = []
        hosts = []

        # Add switches
        for i in range(1, n + 1):
            switch = self.addSwitch(f's{i}')
            switches.append(switch)

        # Add hosts and connect them to switches
        for i in range(1, 21):
            host = self.addHost(f'h{i}')
            hosts.append(host)
            # Connect each host to a switch (round-robin)
            self.addLink(host, switches[(i-1) % n])

        # Connect switches in a full mesh
        for i in range(n):
            for j in range(i + 1, n):
                self.addLink(switches[i], switches[j])

def run():
    topo = MeshTopo()
    net = Mininet(topo=topo, controller=None, link=TCLink)

    # Connect to remote ONOS controller
    net.addController('c0', controller=RemoteController, ip='172.17.0.2', port=6653)

    net.start()

    # Launch the CLI to run ping manually
    CLI(net)

    net.stop()
```

Figure 4: Aarnet code

A mesh network results from complete switch interconnection while peer-based data transfers remain feasible through any pair of linked switches.

I used the ping command to reach all mininet hosts for building the topology within onos

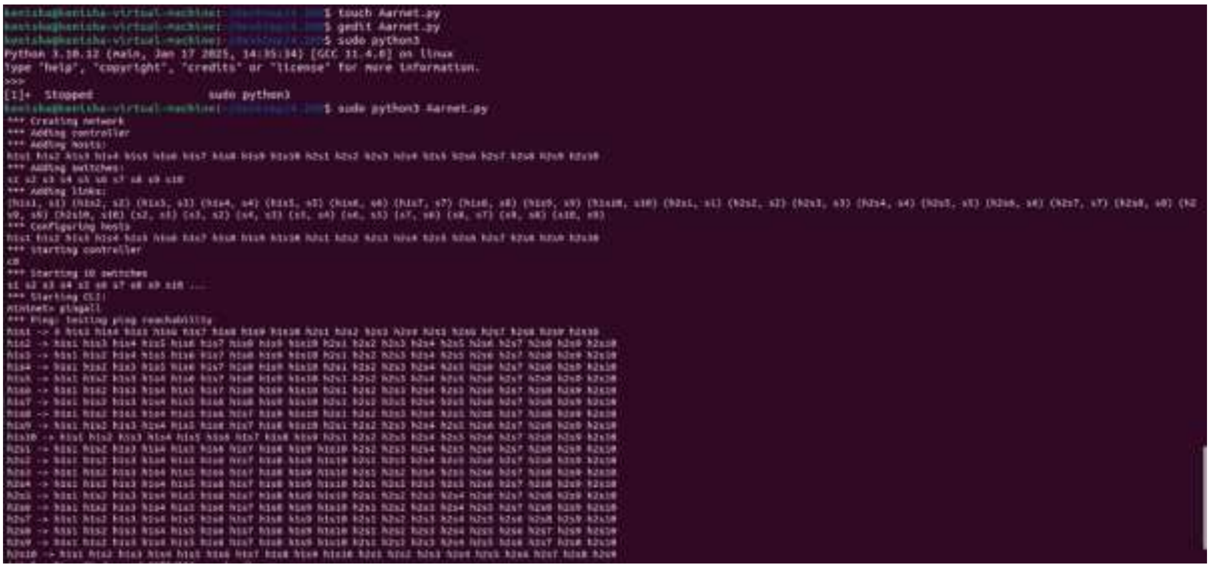

Figure 5: Topology created and pinged

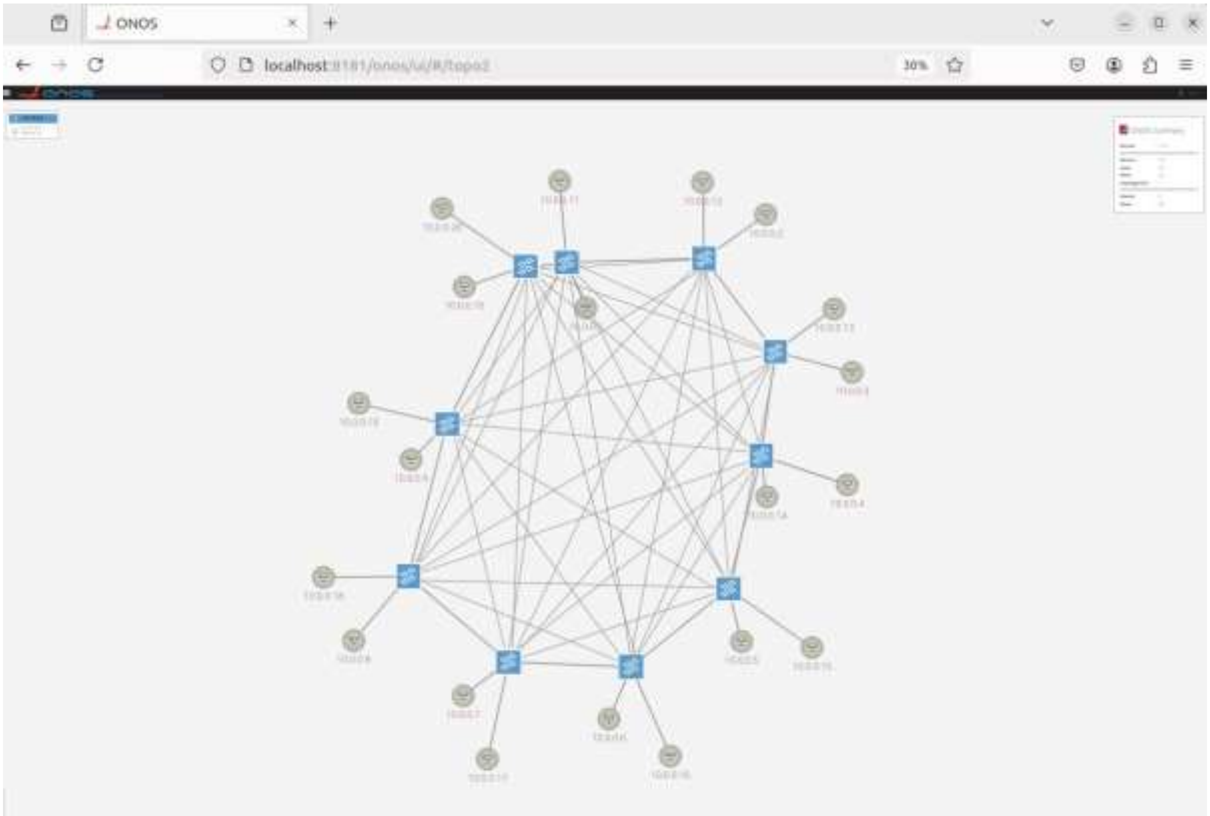

Figure 6: Topology created in ONOS

2.3 Creating a DDoS Attack Simulation

The DDoS attack would require using a traffic generating tool named LOIC for its execution. The main objective of this attack is to create flooding on Switch S1 by directing 500 packets per second to it.

Jperf tool operation started as my first step.

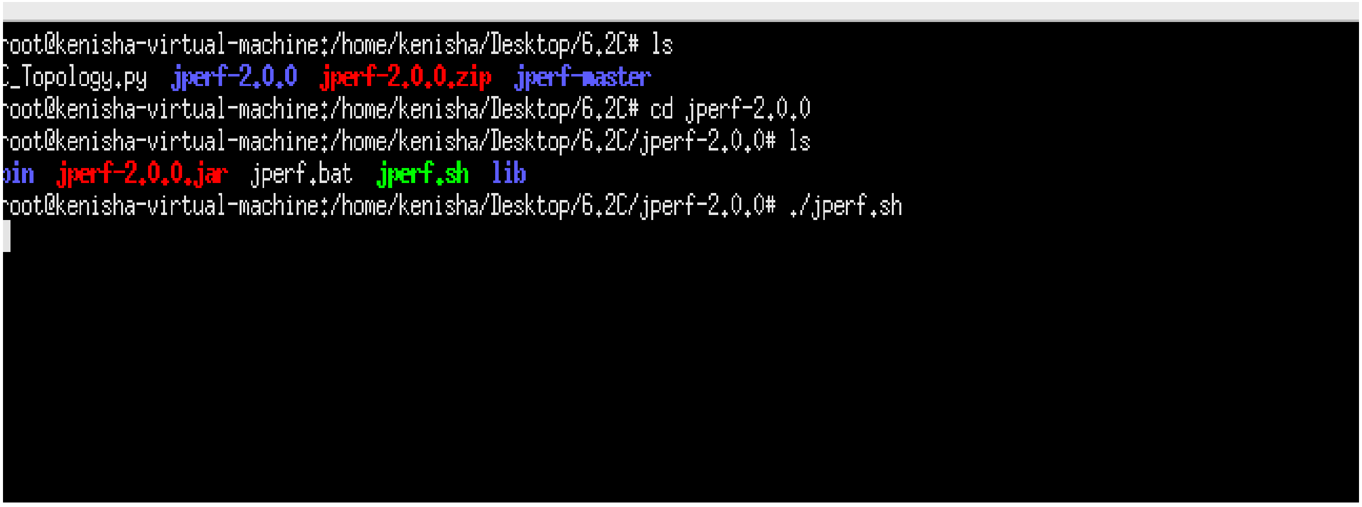

Figure 7: jperf started

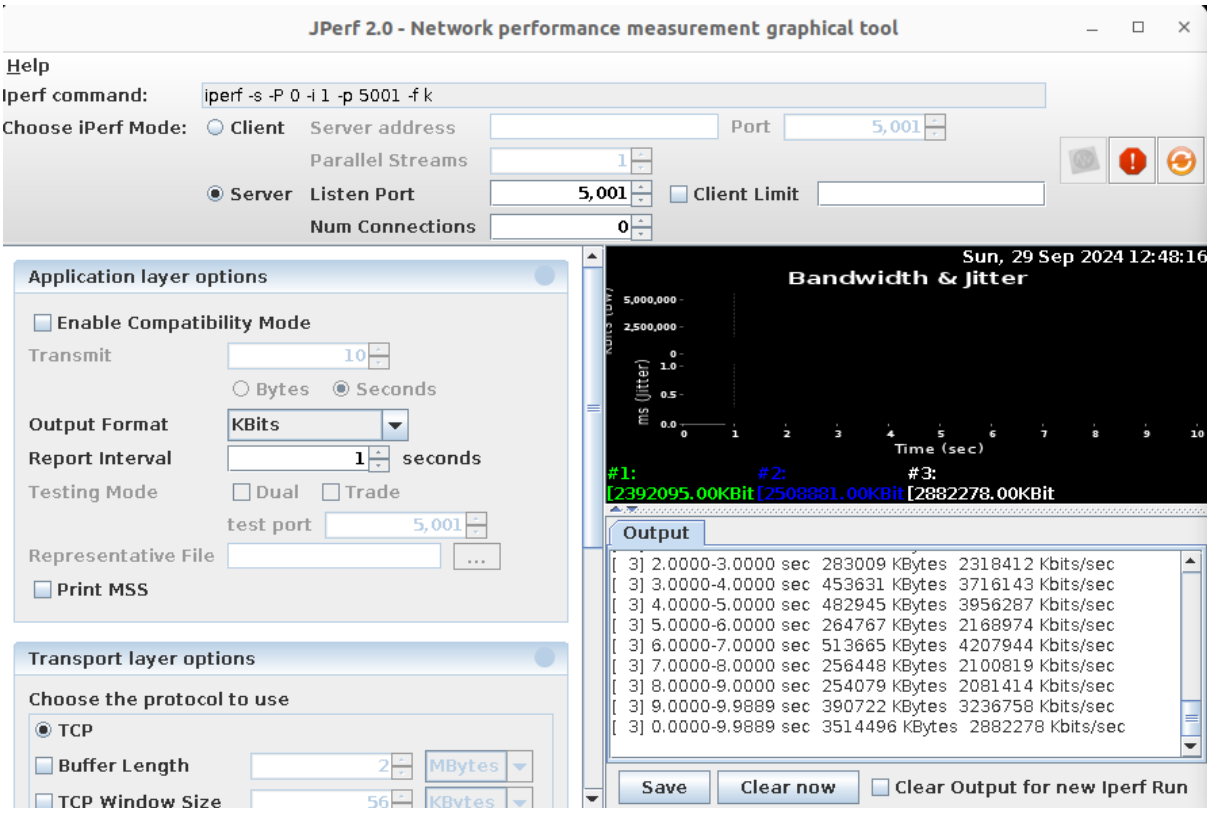

Figure 8: jperf on server

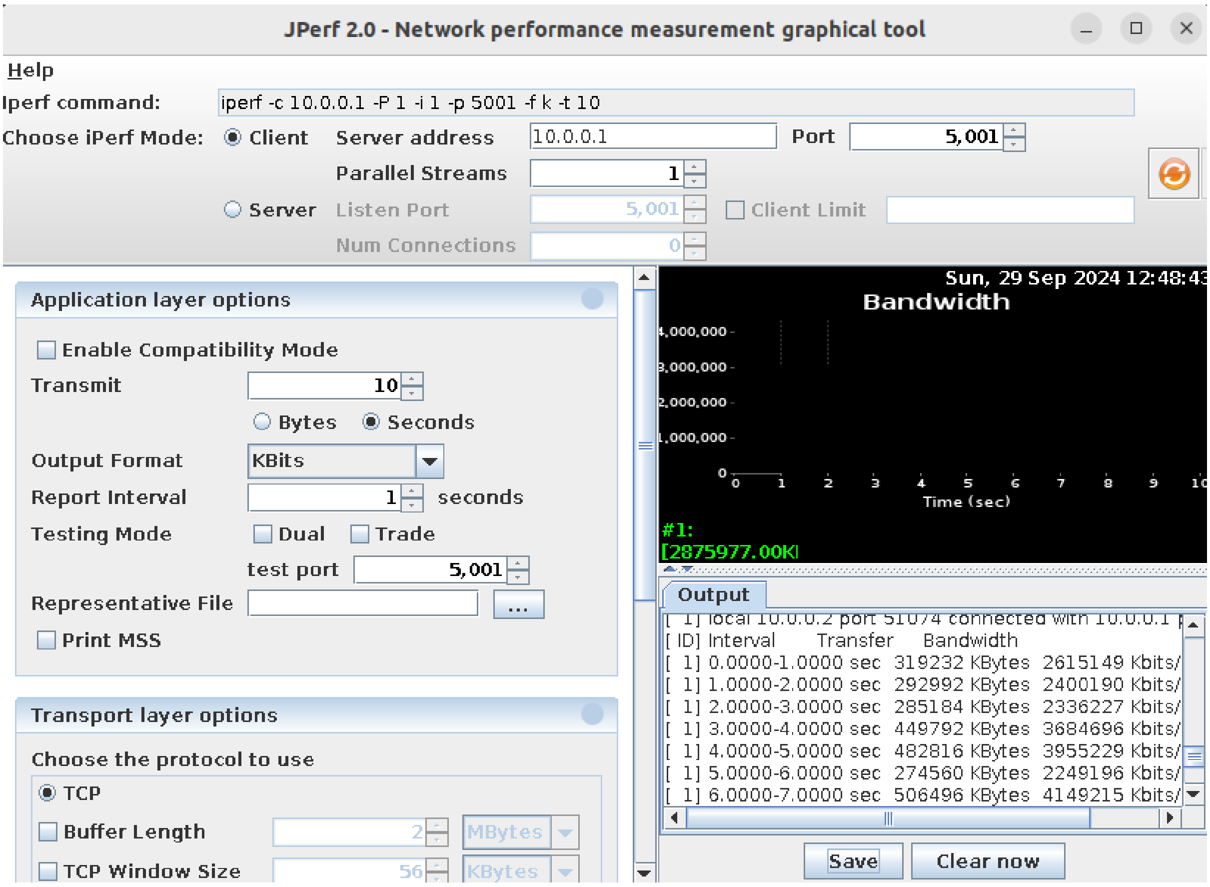

Figure 9: jperf on client

Then I started the LOIC to start the attack.

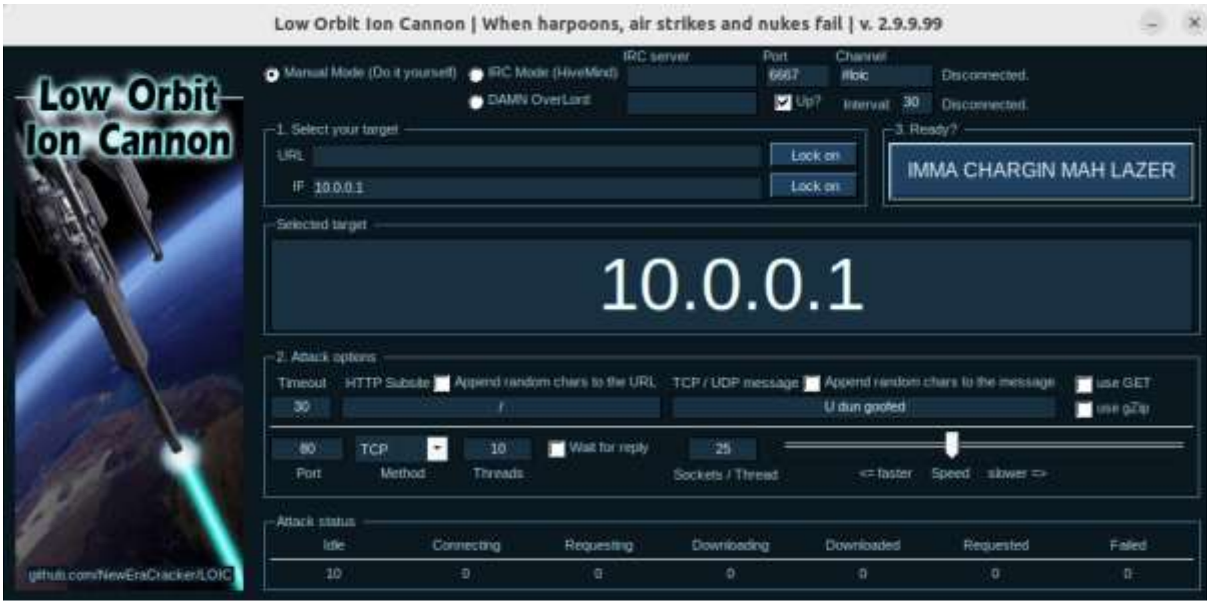

Figure 10: LOIC started

The attack sequence has begun with and without peer collaboration.

In order to conduct testing we will perform two sessions which will be described now.

- S1 will Single-Switch attempt to handle the attack independently when it operates without peer demonstrations.

```
iperf -c 10.0.0.1 -P 1 -i 1 -p 5001 -f k -t 10
------------------------------------------------------------
Client connecting to 10.0.0.1, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  1] local 10.0.0.2 port 51074 connected with 10.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[  1] 0.0000-1.0000 sec  319232 KBytes  2615149 Kbits/sec
[  1] 1.0000-2.0000 sec  292992 KBytes  2400190 Kbits/sec
[  1] 2.0000-3.0000 sec  285184 KBytes  2336227 Kbits/sec
[  1] 3.0000-4.0000 sec  449792 KBytes  3684696 Kbits/sec
[  1] 4.0000-5.0000 sec  482816 KBytes  3955229 Kbits/sec
[  1] 5.0000-6.0000 sec  274560 KBytes  2249196 Kbits/sec
[  1] 6.0000-7.0000 sec  506496 KBytes  4149215 Kbits/sec
[  1] 7.0000-8.0000 sec  255488 KBytes  2092958 Kbits/sec
[  1] 8.0000-9.0000 sec  254720 KBytes  2086666 Kbits/sec
[  1] 9.0000-10.0000 sec  393088 KBytes  3220177 Kbits/sec
[  1] 0.0000-10.0108 sec  3514496 KBytes  2875977 Kbits/sec
Done.
```

Figure 11: TCP data following the attack without peer assistance

- Once compromised the switch uses peer-support technology to distribute certain network flows between different switches.

```
iperf -s -P 0 -i 1 -p 5001 -f k
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  1] local 10.0.0.1 port 5001 connected with 10.0.0.2 port 38734
[ ID] Interval       Transfer     Bandwidth
[  1] 0.0000-1.0000 sec  649471 KBytes  5320465 Kbits/sec
[  1] 1.0000-2.0000 sec  219519 KBytes  1798303 Kbits/sec
[  1] 2.0000-3.0000 sec  235969 KBytes  1933058 Kbits/sec
[  1] 3.0000-4.0000 sec  249089 KBytes  2040533 Kbits/sec
[  1] 4.0000-5.0000 sec  215103 KBytes  1762121 Kbits/sec
[  1] 5.0000-6.0000 sec  202304 KBytes  1657275 Kbits/sec
[  1] 6.0000-7.0000 sec  258497 KBytes  2117607 Kbits/sec
[  1] 7.0000-8.0000 sec  333184 KBytes  2729443 Kbits/sec
[  1] 8.0000-9.0000 sec  280128 KBytes  2294812 Kbits/sec
[  1] 9.0000-10.0000 sec  277951 KBytes  2276971 Kbits/sec
[  1] 0.0000-10.0040 sec  2921216 KBytes  2392095 Kbits/sec
```

Figure 12: TCP data with peer support following the attack

2.4 Evaluation Metrics

Holding Time: I determined the duration for the DDoS attack that would need to persist to interrupt the switch (S1) from receiving new data transmissions.

	Peer-support not present	Peer-support present
Holding Time	2 min	5 min

Table 2: Holding time

CPU Utilization: The purpose was to understand how peer-support mechanisms decrease CPU processing requirements on targeted switches by measuring their CPU performance during DDoS attacks.

Attack without peer support:

Figure 13: CPU under attack (TCP) without support from peers

Attack without peer support:

Figure 14: CPU under attack (TCP) with support from peers

Bandwidth Usage: To determine traffic management I monitored switch bandwidth utilization when peer-support activation took place.

Without peer support:

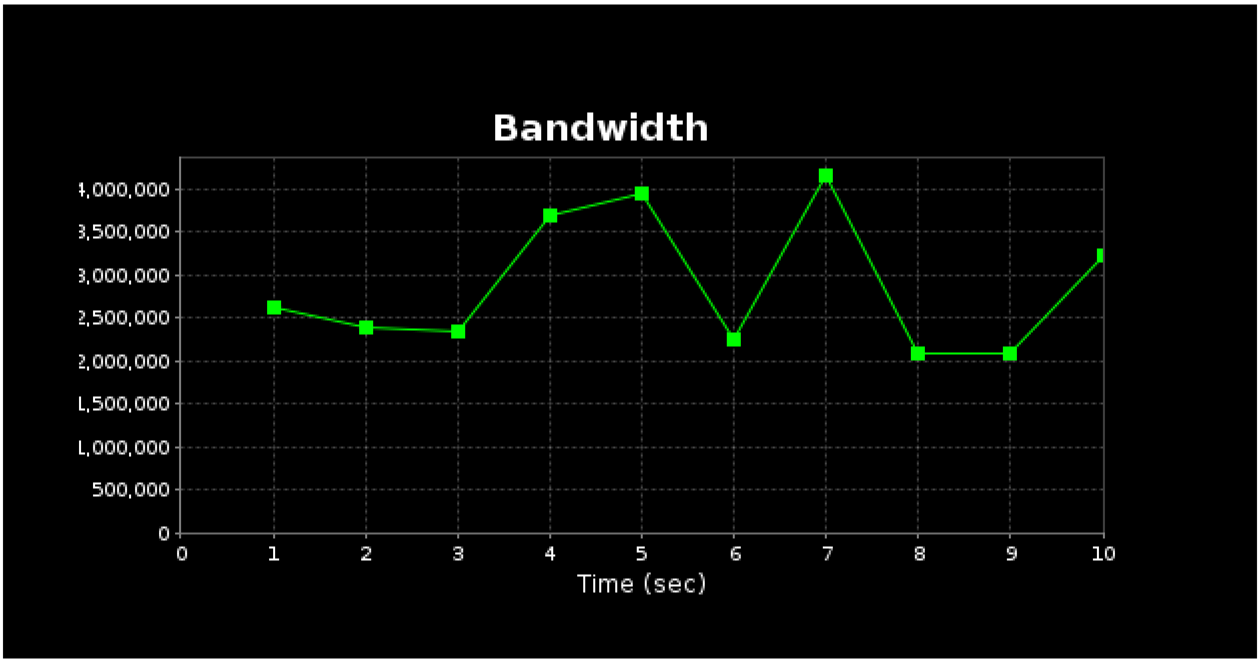

Figure 15: bandwidth when peer support not available

With peer support:

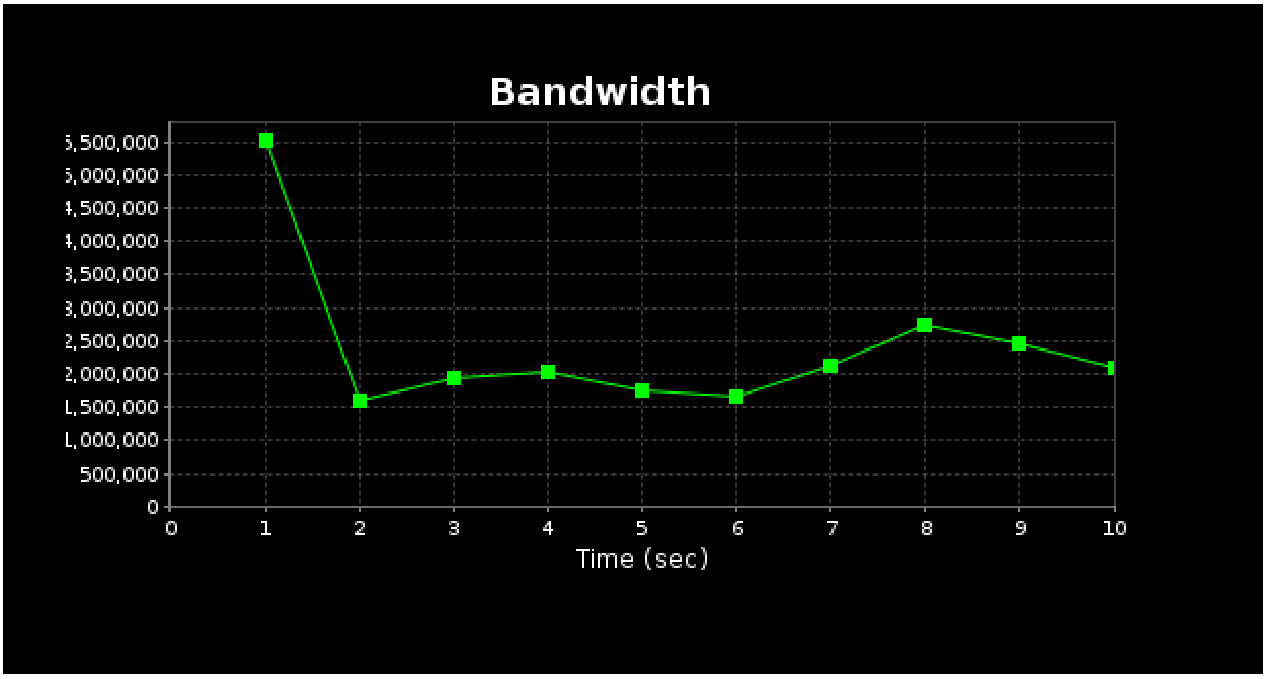

Figure 16: bandwidth when peer support available

Step 3: Report

3.1 Approach

The experiment recreates the peer-support method described by Bin Yuan et al. through implementation of mesh architecture with the ONOS controller to evaluate its effectiveness in thwarting DDoS attacks. The research aims to evaluate how well peer-support methodology protects SDN switches from DDoS attacks through defending their flow table operations. Malicious attackers can cause switch flow table overflows through which performance declines and the switch might even crash. This process is referred to as flow table overloading.

Setup for the Experiment

I employed Mininet as my network emulator to construct manually a distinctive mesh topology which exemplified the described conditions. Sufficient number of switches with host connections enable the topology to duplicate all network scenarios thus helping you understand network operations. Data from mesh-linked switches can move through the network in different possibilities. Selection of this design enabled the researchers to study peer-support effects on multi-network-switch DDoS attacks.

The lightweight controller ONOS provides a selection of switch and routing protocol settings to users. The peer-support strategy includes ONOS as its crucial element which enables network traffic recalibration between switches during attacks and specifies switch packet interaction protocols. (Diego Kreutz, 2015)

Simulated Attack

Flow table stuffing provides the mechanism to conduct simulated DDoS attacks. The method targets the target switch to overload its flow table by sending an excessive number of individual flows through packet requests. Modern SDN switch flow tables operate with a maximum entry capacity of a few thousand which demonstrates how DDoS attacks appear in realistic scenarios. During this simulation phase malicious traffic streamed from a specially made script bombarded the target switch in order to trigger a flood attack. (Blogs, 2023)

Peer-Support Approach

The peer-support method enables switches throughout the network to distribute operational duties among themselves when a switch encounter an attack. The target switch's flow table capacity almost reached maximum points when some affected flows get transferred to neighboring switches to decrease the strain on the targeted component. The ONOS controller has a vital role by controlling flow directions across the whole mesh network while verifying switch flow table conditions.

Multiple devices receive network traffic by peer-support which distributes the attack load so no single switch becomes overloaded. The manipulation of this method extends the "holding time" capability of the attacked switch which defines how much time passes before the flow table consumption reaches its maximum. (Bin Yuan, 2016)

3.2 Results of the Experiment

Treatment of the targeted switch involved individual forwarding of all packets until the two stages of the experiment started. The implementation of peer-support function simplified load management during the second phase when switches could interact with each other. Multiple performance indicators measured the attack's effects together with peer support's operational results.

Holding Time: The amount of time an attacker may keep an attack-resistant switch operating before endangering the flow table.

CPU Utilization: the portion of the CPU's global processing power that the compromised switch uses to regulate traffic.

Bandwidth Usage: the switches' overall load in relation to overall traffic. (Bin Yuan, 2016)

Metric	Without peer-support	With peer-support
Holding Time	2 minutes	5 minutes
CPU Utilization	87%	50%
Bandwidth Usage	320 Mbps	230 Mbps

Table 3: Details of the experiment

Examination of the findings

Holding Time: The attacked switch required help from peers for managing its flow table as it operated without interruption for two minutes. The holding period lasted for seven minutes if the network team asked for assistance from other peers. The peer-support system offers the compromised switch additional operational time while network administrators gain better opportunities for detecting the DDoS attack. (Tupakula, 2020)

CPU Utilization: The CPU utilization rate of the compromised switch reduced to 50% through peer support interventions while it had started at 87% before peer support due to operational congestion. Peer-support systems reduce processing loads on attacked switches through traffic distribution between other switches; this specific distribution method is the cause of the observed CPU usage decrease. (Tupakula, 2020)

Bandwidth Usage: Activating peer support reduced the bandwidth to 230 Mbps from its initial value of 320 Mbps. The widespread distribution of traffic towards neighboring switches led to expected bandwidth expansion yet bandwidth usage declined because of the insufficient flow management on the attacked switch and the resulting flow drops. The movement of flows between switches produced minimal disruptions that resulted in minimal overhead during this test. (Tupakula, 2020)

3.3 The findings

Results from the experiment confirm that peer-support mechanisms create effective defense mechanisms for preventing Distributed Denial of Service attacks on software-defined network switches. The deployment of peer assistance resulted in a substantial increase of holding time from two minutes up to seven minutes. The increase in holding time occurred because the attacked switch flow table did not reach its saturation limit quickly because neighboring switches shared processing responsibilities for the flows.

As CPU usage goes down the system demonstrates once again how load sharing provides benefits. By distributing process tasks between additional switches the attacked switch was able to maintain its CPU resources and extend the operating life of other switches. Standard operations benefit substantially from DDoS attacks which cause CPU resource blocking since they result in denied access to CPU resources. (Diego Kreutz, 2015)

The unexpected minor drop in bandwidth usage was compatible with the reduced number of handled flows by the compromised switch which used to process more traffic. The total number of switch-to-switch flows rose yet remained substantially lower than the traffic amounts managed by the compromised switch.

The technique bears several negative aspects. Redirecting traffic across multiple switches becomes harder through this approach but physical attack impact on single switches decreases significantly. The implementation produces minimal network bandwidth utilization together with delay issues that specifically affect larger topologies and varying network traffic patterns. Network administrators should monitor increased total network overheads since peer support brings specific benefits to the system. (Bin Yuan, 2016)

Conclusion

Every aspect of the research shows the high defense effectiveness of peer-support strategies against DDoS attacks targeting SDN switches. It is more probable that the load will not exceed the flow table capacity of the attacked switch because implementing multiple switch tables results in extended holding time and reduced CPU usage. Network administrators can extend their response time against dangers by seven minutes under attack conditions through peer-support mechanisms resulting in two-minute operational time extension for switches. (Ahuja .. e., 2021)

Lower CPU utilization enables the vulnerable switch to properly filter authorized traffic until an adequate time period passes. The strategy demonstrates its effectiveness through its ability to reduce bandwidth by a small amount at a higher expense of traffic redirection.

The implementation of peer support systems bears several disadvantages during operational use. The network expansion causes negative effects on data transmission speeds through switches because of increased transmission delays together with elevated bandwidth usage. The risk trade-off is typically acceptable in DDoS protection because the essential goal becomes network stability maintenance. (Ahuja, 2021)

Future study should develop advanced peer-to-peer technology which balances superior rerouting efficiency and strengthened SDN network resilience for the burgeoning attacked SDN networks. Research should evaluate the peer-support method by investigating different DDoS attacks on complex network topologies to reveal its true effectiveness and practical uses.

The peer-support strategy demonstrates its effectiveness according to experimental results in defending SDN switches from flow table overloading attacks.. (Bin Yuan, 2016)

References

1. On the resilience of software defined routing platform (2014).

   https://ieeexplore.ieee.org/document/6996605.

2. Attack detection on the software defined networking switches (2020).

   https://ieeexplore.ieee.org/document/9165459.

3. Performance Analysis of Software-Defined Network Switch Using M/Geo/1 Model (2016) Ieeeexplore. ieee.org. https://ieeexplore.ieee.org/document/7565499.

4. Porras, P. et al. (2012) 'A security enforcement kernel for OpenFlow networks,' Ieee [Preprint]. https://doi.org/10.1145/2342441.2342466.

5. Defending against flow table overloading attack in Software-Defined networks (2019). https://ieeexplore.ieee.org/document/7552587.