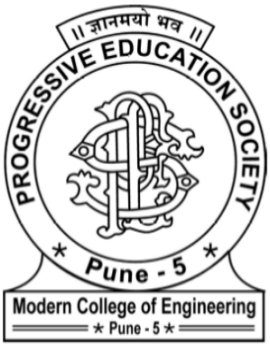
A PRELIMINARY PROJECT REPORT

ON

DETECTION OF DDoS IN SDN ENVIRONMENT USING SVM AND ENTROPY BASED DISCRETIZATION.



BY

ACHYUTH RAO
AKIB SHAIKH
ARUN POTTEKAT
PRANAV TALE

DEPARTMENT OF COMPUTER ENGINEERING
P.E.S MODERN COLLEGE OF ENGINEERING
PUNE - 411005.
[2016 2017]

A PRELIMINARY PROJECT REPORT

ON

"Detection of DDoS in SDN Environment using SVM and Entropy based discretization"

Version: 1.0, April 29, 2017

By

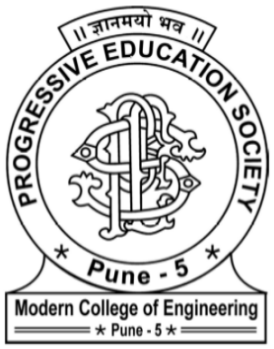
Achyuth Rao
Akib Shaikh
Arun Pottekat
Pranav Tale

Guide:

- **Internal Guide Name:** Ms. Aparna Junnarkar

Presented By:

Date	Version	Title	Authors
6^{th} October, 2016	1.0	Detection of DDoS in SDN Environment using SVM and Entropy based discretization	Achyuth Rao, Akib Shaikh, Arun Pottekat, Pranav Tale



Progressive Education Society's
Modern College of Engineering
Shivajinagar, Pune - 411005.

CERTIFICATE

This is to certify that the following students of Final Year Computer Engineering have successfully completed the preliminary analysis and design of project entitled "Detection of DDoS in SDN environment using SVM and Entropy based mechanism" for the organization "PES Modern College Of Engineering"

The Group Members names are: Sandhyavandanam Achyuth Rao.
Akib Ashraf Shaikh.
Arun Pramod Pottekat.
Pranav Balasaheb Tale.

This is in partial fulfillment of Bachelor of Computer Engineering under Savitribai Phule Pune University.

Date:

Internal Guide	Head Of Dept	External Examiner
(Ms. Aparna Junnarkar)	(Computer Engineering)	
	(Dr. Prof. Mrs. S. A. Itkar)	

Acknowledgement

It gives us pleasure in presenting the preliminary project report on **'Detection of Distributed Denial of service attack in Software Defined Network using Support Vector Machine and Entropy Based Discretization'**.

Firstly, we would like to express our indebtedness appreciation to our internal guide **Ms. Aparna A. Junnarkar**. Her constant guidance and advice played very important role in making the execution of the report. She always gave us her suggestions, that were crucial in making this report as flawless as possible.

We would like to express our gratitude towards **Prof. Dr. Mrs. S. A. Itkar** Head of Computer Engineering Department, PES Modern College of Engineering for her kind co-operation and encouragement which helped us during the completion of this report.

Also, we would like to thank **Mr. Kunal Khadke, Ms. Yogita Narwadkar, Mr. B. D. Phulpagare, Ms. Pallavi Baviskar, Ms. Deipali V. Gore, Ms. Renuka Kajale** and all **Technical assistants** for providing time to time guidance and various resources such as laboratory with all needed software platforms and continuous Internet connection for our Project.

In the end special thanks to all our classmates for helping us out during the entire documentation process.

Achyuth Rao
Akib Shaikh
Arun Pottekat
Pranav Tale

Contents

Abstract

The current networking paradigm involves switches, routers and gateways where these networking devices constitute both logical thinking as well as routing of packets. Traditionally the network administrator is responsible for configuring and managing these devices manually and at all times, which makes it a tedious task.

With the onset of the Software Defined Network, this task of manually managing the devices reduces to some extent, as it seperates the control plane from the data plane i.e. Forwarding of packets is done in the data plane and intelligence of the entire network resides in the control plane.

Data plane constitutes the network devices which comprise switches known as "dumb terminals" and the control plane constitutes a central controller which keeps track of all switches in the network.

Due to the centralized nature of SDN i.e. the controller at the center of SDN taking the logical decisions, there arises a threat of malicious users launching cyber attacks on this central component thereby, dislodging the entire network. Some attacks include Application level attacks, Brute Force attack, man in the middle attacks, DDoS attack etc.

Distributed Denial of Service attack involves a single malicious user controlling different users known as bots to launch an attack against a single entity in the network without even the victim being aware of the attack. DDoS attacks result in direct financial losses along with damage to company reputation and loss of the customer's trust.

As a part of solution to this problem, two algorithms can be used for the detection of DDoS attack i.e. Support Vector Machine, a machine learning algorithm and Entropy based Discretization, a Data Mining algorithm.

Support Vector Machine takes the rate of incoming packets as the input and classifies them as normal traffic or attack traffic. Whereas Entropy based Discretization monitors the entropy of the network i.e. measure of randomness and if it falls below a threshold value then it is classified as attack traffic.

Once the attack is detected by either of the applications, an entry will be made in the log files constantly monitored by network monitoring tools and thus report the incident back to the network adminstrative team.

The network adminstrator would be provided a monitoring application which would display the details of the attack detected such as the source IP and victim IP etc.

As a part of the project, comparison studies will also be done related to both algorithms displaying the rate at which attacks are detected along with their accuracies.

List of Figures

List of Tables

List of Abbreviations

SDN ..... Software Defined Network

SVM ..... Support Vector Machine

DDoS ..... Distributed Denial of Service

ONOS ..... Open Network Operating Switch

OVSDB ..... Open vSwitch Database

EBD ..... Entropy based Discretization

NP ..... Non Polynomial Time

P ..... Polynomial Time

HTTP ..... Hyper Text Transfer Protocol

UDP ..... Uniform Datagram Protocol

TCP ..... Transmission Control Protocol

k-NN ..... K Nearest Neighbour

ICMP ..... Internet Control Message Protocol

FTP ..... File Transfer Protocol

QP ..... Quadratic problem

1.

Software Requirement Specification

1.1 Purpose

The number of Cyber Attacks is increasing day by day and has become a major concern for enterprises. Such attacks include application level attack, man in the middle attack, brute force, DDoS etc. Attackers are waging an asymmetric battle against these enterprises on their networks, assets and data. One major type of attack would be Distributed Denial of Service attack which is launched by flooding the targeted machine with multiple requests to overload the system and prevent the legitimate packets. In the next generation networking, when Software Defined Networks would be deployed in data centers, enterprises or campus networks, security will prove to be a critical issue against such attacks that needs to be taken care of, which is the aim of our project.

In Software Defined Networks, the network intelligence and state are logically centralized which gives programmability, better network control, flexibility and scalability. But in this new architecture the controller which is responsible for controlling the whole network can become a single point of attack. In our proposed method, we will be using Support Vector Machine, Entropy Based Discretization as well as Fuzzy C Means clustering in order to effectively detect any DDoS attack that takes place in a Software Defined Network.

1.2 Project Scope

The scope of the project extends to setting up of a Software Defined Network using OpenFlow configured switches and also use three algorithms i.e. Entropy based discretization, Fuzzy C Means Clustering and Support Vector Machine for the detection of DDoS attacks. As a part of this project we would also be using a network monitoring tool to monitor the software defined network and raise a notification as soon as any of the three algorithms reports a DDoS attack by marking an entry into the log files which could be utilized for future needs.

1.3 Usage Scenario

1.3.1 User profiles

1. **Network Administrative team :** Every software or non-software industry now-a-days involves its own network establishment. A network administrative team is delegated the task of monitoring the network activities and troubleshooting the network in case of any issues.

   Currently this team faces a tough task of manually monitoring the network for any attacks at all times. Deployment of our product in an enterprise would reduce the burden of the network team as they would receive a notification whenever DDoS attack occurs thereby allowing them to take precautionary measures.

1.3.2 Use-cases

Sr. No.	Use Case	Description	Actors	Assumptions
1	DDoS Notification raised	Raising DDoS notification as soon as DDoS is detected by any of the three algorithms i.e. Support Vector Machine, Entropy based discretization and Fuzzy C-Means Clustering	Network Administrator	SDN is established within the environment and DDoS attack is launched on the enterprise.
2	Network Monitoring	Monitoring the network statistics for any anomalies using sFlow, ElasticSearch Logstash Kibana (ELK Stack), and Flask based application	Network administrator	The link between switches and the monitoring system is maintained throughout.

Table 1.1: Use cases

1.3.3 Use Case Diagram

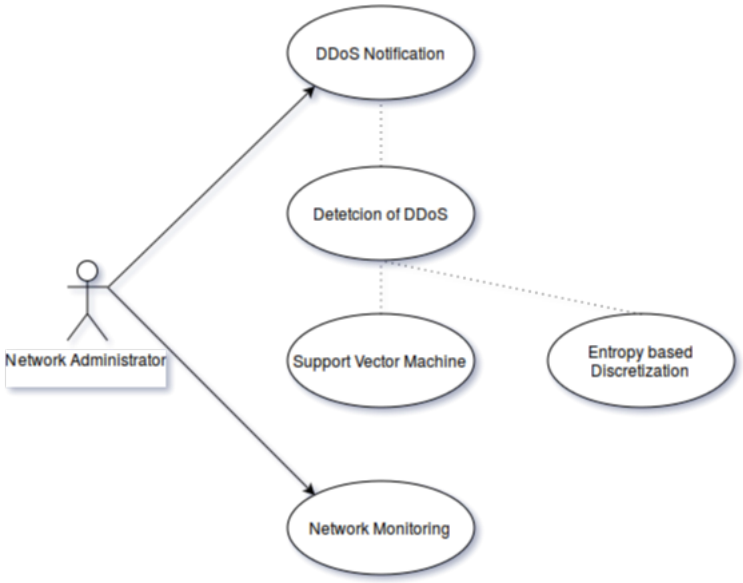


Figure 1.1: Use Case Diagram

1.4 Product Features

1. **Runtime Detection of DDoS attack :-**
   In the current network scenario, there aren't many utilies for efficient DDoS detection. Due to this, the network administrator has to monitor the network continuously. Hence, this product will be step towards the automation of this process wherein the network administrator will have to verify the attack, only when the notification is generated.

2. **Network Environment Specific Product :-**
   The training of the Support Vector Machine algorithm and Fuzzy C Means Clustering is done using an environment specific conditions or depending on the normal network traffic conditions in your network. Thus these algorithms are trained in such a way so as to reduce the false positive rate of detection. Also the threshold for Entropy Based mechanism is decided depending on the normal traffic conditions.

3. **Automated Alert Mechanism :-**
   As soon as the attack is generated the alert can be sent to a specific network monitoring team or can send an email along with the information of where the attack has occurred specifying the IP addresses.

4. **Suitable for all Software Defined Networks :-**
   Independent of what type of Software Defined Network like carrier, service provider, enterprise, data center or campus network, our solution will be effective in all these environments.

5. **Open Source to encourage contribution and research :-**
   As SDN itself is an open standard of next generation networking, our solution will be made open source along with the source code and the method of detection, describing the step-by-step complete procedure. This will help us to encourage the open source community to contribute their ideas and ultimately improve the effectiveness.

6. **SDN Controller Independent :-**
   Since our application will the running on the edge switches, irrespective of which SDN controller is in use our solution can still be used.

1.5 Functional Model and Description

1.5.1 Data Flow Diagram

1. **Level 0 Data Flow Diagram :**

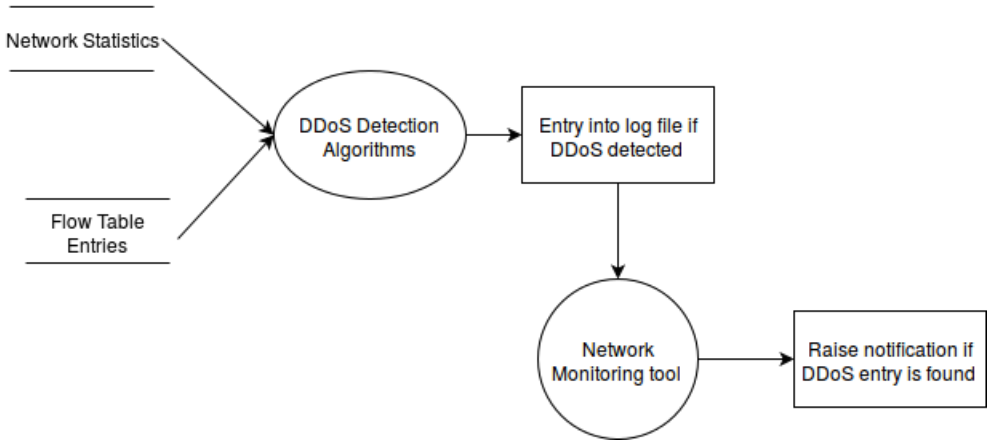


Figure 1.2: Level 0 Data Flow Diagram

2. **Level 1 Data Flow Diagram :**

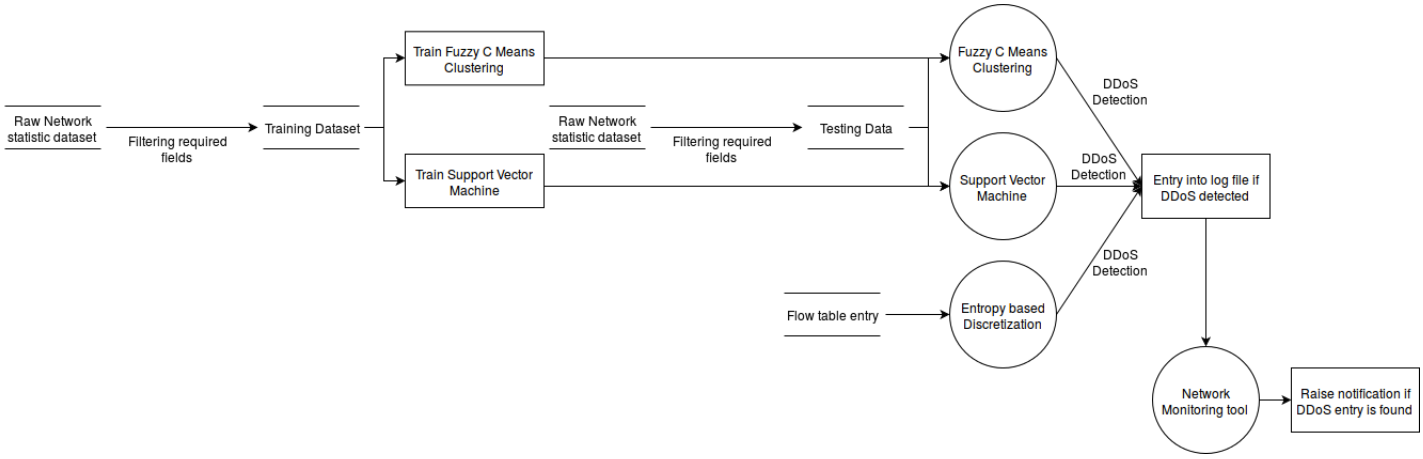


Figure 1.3: Level 1 Data Flow Diagram

1.5.2 Activity Diagram

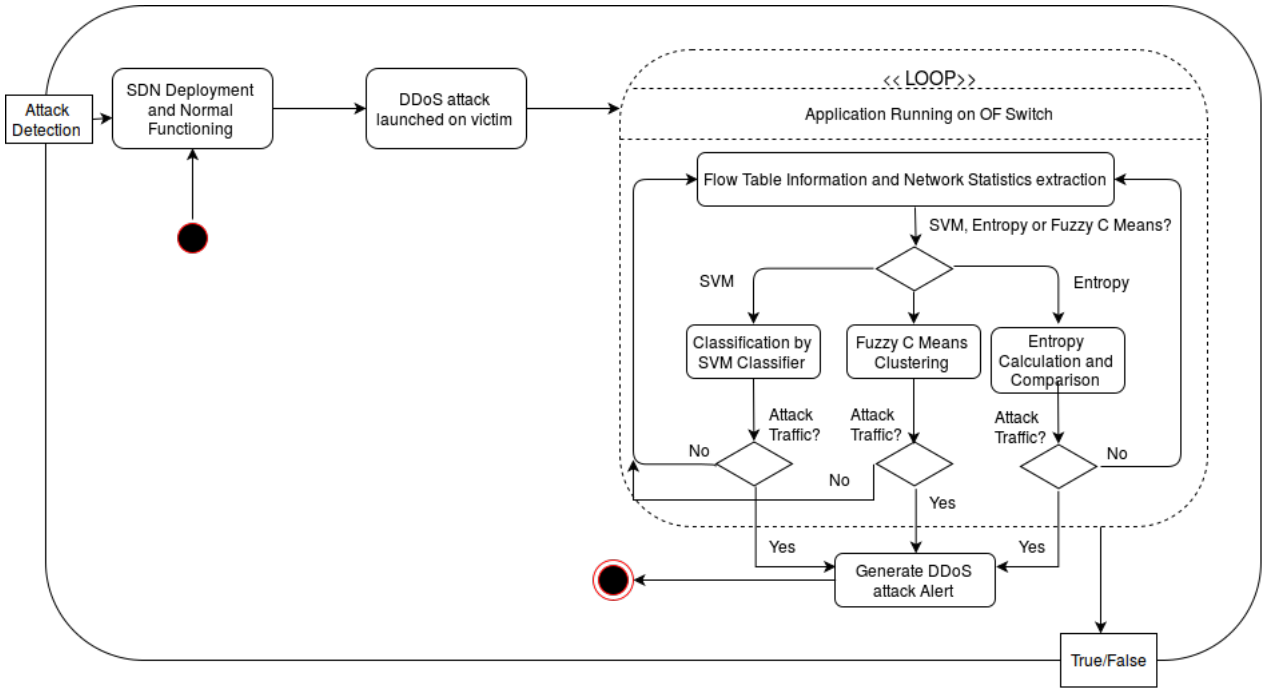


Figure 1.4: Activity Diagram

1.5.3 Non Functional Requirements

1. **Interface Requirements**

   (a) OpenFlow Protocol :
   Provides a medium for the SDN controller to direct traffic along the switches within the network.

   (b) Northbound API's :
   Allow the SDN controller to communicate between services and applications running over the network owing to programmatic nature of SDN.

   (c) Southbound API's :
   Allow the SDN controller to communicate between the network devices within the network.

2. **Software Quality Attributes**

   (a) Correctness:
   The correctness of our product would depend upon the accuracy with which either of the application would detect the attack. For this purpose both applications must be trained in the network which they would monitor to generate the necessary constant values needed for their execution.

   (b) Availability:
   As long as the network is up and running, this service would be also be available continuously monitoring the network for any anomalies.

(c) Usability:
As and when a DDoS attack is detected, a simple notification is sent to the network administrative team. It does not disrupt their normal work flow or involve using complex application for monitoring purposes. Hence no specific training of the user is needed for using this system.

(d) Portability:
Portability is restricted i.e. both applications must be given prior training before being deployed in a network. Thus the system being used in one enterprise cannot be used in another enterpise owing to the fact that the network traffic would vary from enterprise to enterprise such that normal traffic in one could be classified as attack traffic in another enterprise. Hence training in the network is of utmost importance.

1.5.4 State Diagram

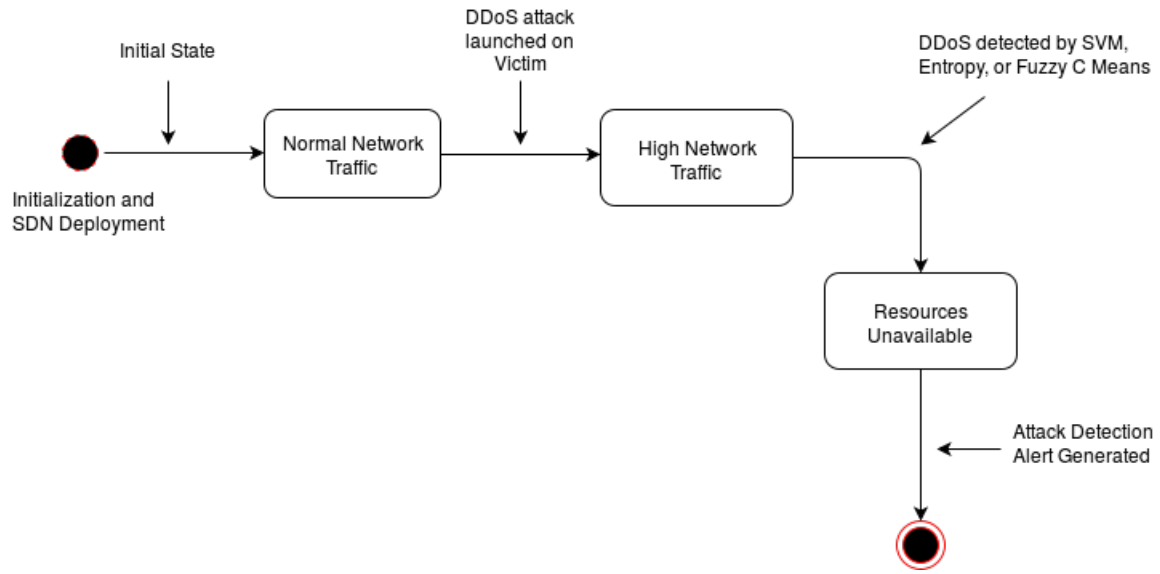


Figure 1.5: State Diagram

2.

Detailed Design Document

2.1 Architectural Design

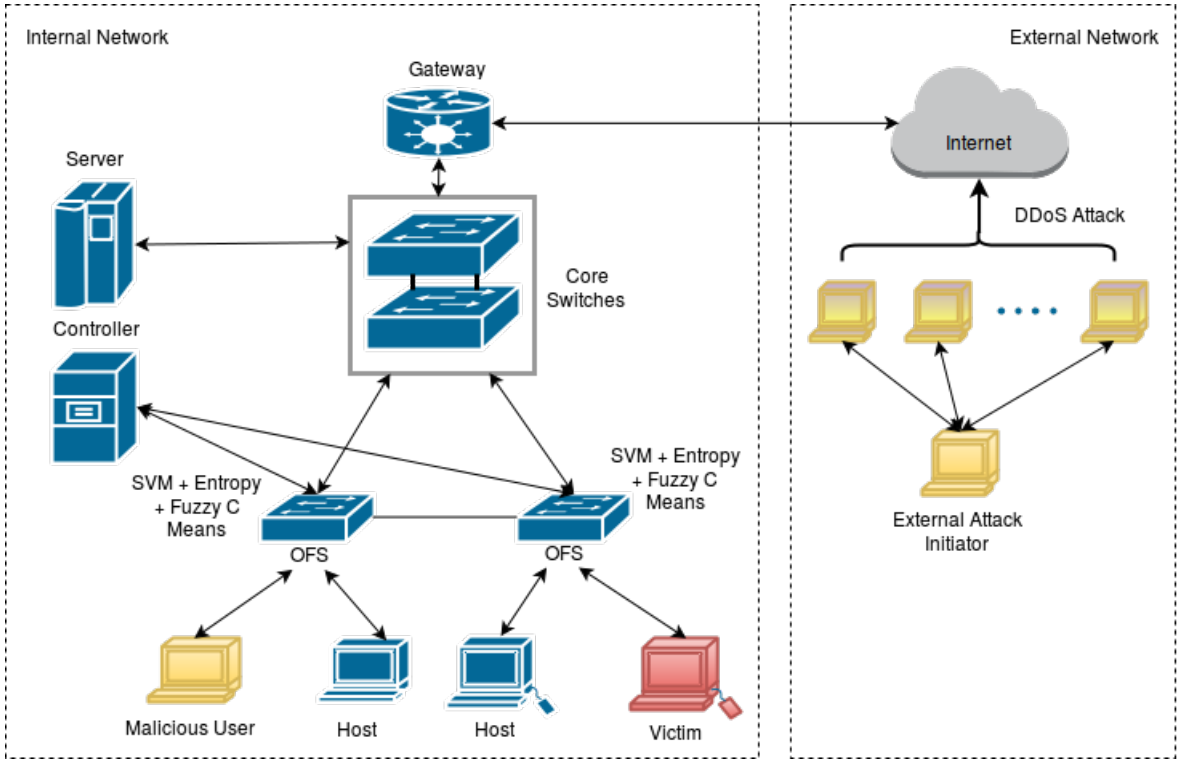


Figure 2.1: System Architecture

During normal state of communication between two users in the network, the first packet is transmitted from the host to it's nearby switch. Initially the switch does not have any entry in its flow table as to where it should forward the incoming packet, hence the default route is to forward the packet to the controller. The controller then dictates the switch to forward the packet to the respective destination port. Along with this the controller also makes an entry in the flow table of the switch for the particular source and destination such that the future packets with the same source and destination are forwarded directly without any intervention of the controller.

This helps in reducing the time taken for communication between source and destination systems. After a particular interval of time the flow table entries are flushed and the process is repeated all over again.

The system is designed to detect two types of DDoS attacks namely :-

1. Internal DDoS attack
2. External DDoS attack

In the first case, i.e. internal DDoS attack, the malicious user and the victim are located within the software defined network. whereas in external DDoS attack the host machine is from a different network. In both cases losses generated are immense.

The malicious machine can now launch attacks at two different points, one being the controller which would result in downfall of entire enterprise network incase there aren't any secondary controllers available, or the second point of attack could be one of host machines.

In case of the attack point being the controller, the attack is generated by spoofing the source and destination ip address so that it does not match any flow table entry and hence must make a trip to the controller, this would burden the controller and thereby render its services unavailable. To overcome this issue, SVM monitors the rate of incoming packets and classifies it as normal traffic or attack traffic, similarly Entropy based mechanism would find a drop in the entropy in the network owing to the large

number of same class of packets hence would be able to detect the DDoS attack, and Fuzzy C Means clustering would assign a membership value closer to ”1” near the core switch since all packets are being directed towards the controller and thus would be able to detect a DDoS attack.

Another method of generating an attack would be by generating a multitude of requests at the host thereby rendering the system void of performing any other operations. This type of attack is also solved along similar guidelines such that SVM would monitor the rate of incoming packets and Entropy based mechanism would monitor the entropy of the network which would drop due to large number of same class of packets, similarly Fuzzy C Means, running on the edge switches would be able to place the incoming packets in the attack cluster thereby detecting a DDoS attack.

Hence, a comparison can be deduced as to which method would generate a result instantaneously and with higher accuracy.

Once the attack is detected by any of the applications an entry is made into the log file of the respective switch, which is constantly monitored by the network monitoring tool which on finding the entry raises a notification to the network admnistrative team. Further it is the responsibility of the network administrative team to take preventive measures to eliminate any damage that could be caused by an impending full fledged DDoS attack.

2.2 Data Design

2.2.1 Dataset Description

For the purpose of detection, all three algorithms use some form of data on which computation is performed and results are predicted. In case of Support Vector Machine the dataset consists of the network statistics, which have been converted to a ”csv” file with limited fields using a terminal based tool ”tshark”.

The fields present in the dataset have been shown in table 2.1

Packet Interval	Source IP	Destination IP	Protocol
0.000300000	192.168.1.14	192.168.1.11	1

Table 2.1: Dataset Fields - Support Vector Machine

1. **Packet Interval :**
   Time interval between two incoming packets
2. **Source IP :**
   The source ip address of the sender machine.
3. **Destination IP :**
   The destination ip address of the receiver machine.
4. **Protocol**
   Protocol of the incoming packet.

This dataset is again categorized into attack traffic dataset and normal traffic dataset, which can be shown in table 2.2 and 2.3, wherein the most standout variation is the packet interval field between normal traffic and attack traffic.

Packet Interval	Source IP	Destination IP	Protocol
0.000300000	192.168.1.14	192.168.1.11	1
0.000100000	192.168.1.12	192.168.1.11	1
0.000200000	192.168.1.13	192.168.1.11	6

Table 2.2: Attack Dataset Fields - Support Vector Machine

Packet Interval	Source IP	Destination IP	Protocol
0.02000000	192.168.1.14	192.168.1.11	1
0.01000000	192.168.1.14	192.168.1.11	1
0.01000000	192.168.1.12	192.168.1.11	1

Table 2.3: Normal Dataset Fields - Support Vector Machine

Similar datasets are created during runtime using the packet capturing tool ”tshark” and the classification is carried out on the same.

In case of Fuzzy C Means clustering, the nature of datasets used are the same. The only difference is the number of fields used. For Fuzzy C Means only the packet interval field is used to generate clusters and assign membership values to each packet. Also like SVM, Fuzzy C Means uses two datasets i.e. attack traffic and normal traffic for training the model, using the packet capturing tool ”tshark”, again which is stored in a csv file format. The training dataset appears to be like the one showed in table 2.4 and 2.5

Packet Interval
0.000300000
0.000100000
0.000200000

Table 2.4: Attack Dataset Fields - Fuzzy C Means clustering

Packet Interval
0.02000000
0.01000000
0.01000000

Table 2.5: Normal Dataset Fields -Fuzzy C Means clustering

However, unlike SVM and Fuzzy C Means that use packet level statistics for classification and clustering, Entropy based discretization uses the flow table entries residing in the switch to perform its computation. The flow table entries are dumped into a file during each time interval, which is then looked up by the entropy based application to detect what is the entropy of the network and accordingly detect whether a DDoS attack has been launched or not.

Sample flow entry :
$cookie = 0x0, duration = 240.478s, table = 0, n\_packets = 21975265, n\_bytes = 2153575970, idle\_timeout = 60, idle\_age = 0, priority = 65535, icmp, in\_port = 2, vlan\_tci = 0x0000, dl\_src = 4a:06:64:e9:ef:f0, dl_dst = f2:57:bd:4c:19:28, nw\_src = 10.0.0.2, nw\_dst = 10.0.0.1, nw\_tos = 0, icmp\_type = 0, icmp\_code = 0actions = output : 1$

wherein the fields used during processing are:

1. *n_packets* **:**
   The total number of packets transitted by the switch having the same flow details.

2. *protocol* **:**
   The protocol of the incoming packets.

3. *nw_src* **:**
   Source IP address

4. *nw_dst* **:**
   Destination IP address

3.

PICT

3.1 PICT

At the end of Term 1 assessment i.e. after completing the design phase of our project the reviews received were :

1. To make sure high accuracy will be achieved.

Taking into consideration this review, instead of using libraries for developing the algorithms, the algorithms have been developed right from scratch to mould them into models very specific to the use case of our project i.e. DDoS Detection.

Along with the feedback report of Term 1 assessment, we had also participated in the annual project competition PICT-Impetus and Concepts 2017, gaining valuable insights and reviews for our project. Industrial personnel belonging to companies like VmWare, Calsoft, Synerzip, ParallelMinds, Veritas, Cybage and many more. Having vast expertise in the domains of our project and were well versed with the latest technologies like SDN, ELK Stack, sFlow. Some of the criticism/feedbacks we received were :

1. What kind of DDoS attacks are being detected?
2. Why mitigation is not being included in the scope of the project?
3. Change the controller used, from pox to OpenDayLight.
4. Include adaptive learning module to adapt to the changing network scenarios.