Survey on DoS Attack Challenges in Software Defined Networking

Prajakta M. Ombase
P.G. Scholar
(Dept. of Computer Science and Technology)
Usha Mittal Institute of Technology,
SNDT Women's University
Mumbai, India

Nayana P. Kulkarni
P.G.Scholar
(Dept. of Computer Science and Technology)
Usha Mittal Institute of Technology,
SNDT Women's University
Mumbai, India

Sudhir T. Bagade
Assistant Professor
(Dept.of Computer Science and Technology)
Usha Mittal Institute of Technology,
SNDT Women's University
Mumbai, India

Amrapali V. Mhaisgawali
Assistant Professor
(Dept. of Computer Science and Technology)
Usha Mittal Institute of Technology,
SNDT Women's University
Mumbai, India

ABSTRACT
Software Defined Networking (SDN) is a new trend in networking. SDN replaces traditional networking by separating control plane and data plane. SDN is managed by centralized controller. SDN has certain challenges such as security, reliability, controller failover, load balancing, traffic engineering. Security in SDN is more challenging than traditional networking. One of the security challenges in networking is DoS (Denial of Service) attack which can be created using various mechanisms. This paper review different DoS attacks which can be possible on control plane and data plane. This paper have surveyed, studied and identified the security challenges and different existing techniques to mitigate Dos attacks in SDN. Future researches on DoS attack mitigation techniques are indicated in this paper.

Keywords
Software Defined Network (SDN), Open Flow, security, DoS, ONF (Open Networking Foundation), TCP-SYN, ICMP, Southbound interface (SBI), Northbound interface (NBI).

1. INTRODUCTION
In traditional networking environment, data plane and control plane runs locally on each device which provides information about forwarding tables. Data plane includes switches and control plane includes different controllers. Forwarding tables are used to decide where to send packets entering in networking device. As the demand on networks is increasing, this is the complex way to work out on the topology of network [1]. Recent trends such as Cloud computing, big data, mobile traffic and internet of things (IoT) are increasing the load on enterprise network [2]. In 2008 OpenFlow standard has defined SDN architecture explaining how data plane and control plane are separated out using open flow protocol and communicate using application programming interface. OpenFlow is the first standard communication interface developed by SDN community. ONF (Open Networking Foundation) manages open flow protocol. ONF is the group that is associated with development and standardization of SDN. According to ONF, SDN architecture is directly programmable, centrally managed, programmatically configured, open standards-based, vender neutral, and cost effective [3]. SDN gives centralized view of network and make easy to manage the network. Mininet [4] is open source network emulator and creates virtual switch, hosts, links and controllers on a single Linux kernel with a single command. It is easy to use. One can create custom topologies using mininet. It is also useful in OpenFlow and software defined networking for development and experiment. It simulate real machine and can create different host. As it uses single Linux kernel to create network, it cannot run on software that depends on BSD, windows etc. Another limitation of mininet is that it doesn't have OpenFlow controller and it runs on slower links 10 or 100 mbps. Floodlight is open source, java based, OpenFlow SDN controller, supported by Big Switch Networks. Floodlight is easy to use and compatible with mixed OpenFlow and non-OpenFlow networks. It works with both physical and virtual switches that use OpenFlow protocol. Floodlight [5] can work with the different number of switches, routers, virtual switches. OpenStack, Cloud orchestration platform also supported by Floodlight. In SDN networks controller is the brain of the network as it maintains all the network rules and gives instruction to the network elements. Open vSwitch [6] is open source virtual switch also referred to as OVS. It is compatible with different standards and protocols in computer network. It can run in different network platforms such as virtualization and cloud computing platform. It can operate as switch running on virtual machine also. Majority of code for OVS is written in C language.

1.1 Characteristics of SDN
SDN has logically centralized control as fundamental characteristic. It maintains global view of logically centralized but physically distributed controller. Virtualization in SDN supports multi-tenancy in the infrastructure. AS the data is becoming complex due to increasing communication networking. So managing and deploying the network by network operator is becoming complex in vendor specific configuration. Different virtualization technologies are developed to virtualized network services or functions such as NFV (Network Functions Virtualization). NFV takes function that traditionally runs on network and convert it into the virtual architecture using virtual machine. Without virtualization the use of specific hardware goes on increasing

unnecessarily. Virtualization provides functions virtually without use of hardware. SDN provides integration of the third party network services in the architecture. It allows customization and flexibility of services to adapt new features [7] and reduce the cost of services. Services are managed and controlled centrally by controller. SDN provides scalability by providing any number of switches or routers as required. Third party services can communicate to the controller via internet APIs supported by controller SDN provides protocol which is required for management of the programmable hardware.SDN is vendor neutral. Due to this common network interfaces by hardware and operating system being used by the application doesn't matter. Data centre network needs more number of switches and routers so SDN is preferably used now days.

1.2 Architecture of SDN

SDN architecture has 3 components as shown in Fig.1

1.2.1 SDN Application:

It communicates for behaviour and resources needed with the controller via application program interface (API). Instead of using specialized appliance such as firewall, link load balancer, SDN deploys an application which is used by controller to manage network elements (switch, router etc). SDN applications [8] are the programmes that communicate to the controller through Northbound Interface.

1.2.2 SDN controller:

It resides above the set of OpenFlow switches. It provides necessary flow rule update to networking elements. Each entry in flow table is called as flow rule. Controller uses open flow protocol to communicate and to determine the routing path of the network by adding and modifying the flow rules and deploy it into switch's flow table [9]. First SDN controller was NOX, which was initially developed by Nicira Networks. There are varieties of Open Source Controllers available such as POX, beacon, floodlight, OpenDaylight, etc.

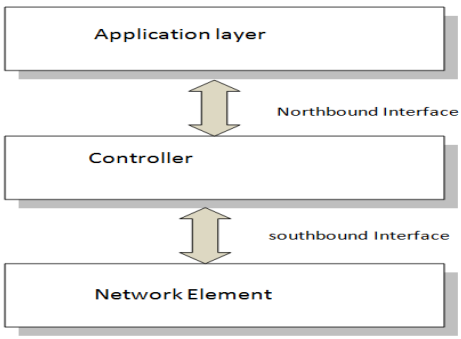

Fig 1: SDN Architecture [10]

Controller can handle prioritizing, de-prioritizing or blocking of packets whenever necessary. SDN controller provides real time feedback and interacts directly with network. Controller implementation offers centralized, hierarchical, fully distributed architecture. Initially SDN control plane was focused on centralized controller architecture, where only single controller has the global view of the network [11]. It has scalability problem. To overcome this limitation hierarchical and fully distributed approach have been proposed. In hierarchical solution, distributed controller operates on the partitioned network view with logically centralized root controller. In fully distributed approach distributed controller operate on their own local network view. Fully distributed approach is more suitable solution.

1.2.3 SDN network element (switch, router):

It controls data forwarding and data processing capability of the network. SDN uses adaptive or dynamic operation mode in which switch requests controller for routing path of a packet that does not have specific route in switch memory. SDN uses South-bound API to allow communication between controller and network element.

1.3 Working of SDN

In open flow switch performs packet forwarding based on flow table. Each networking element such as switch has its own flow table. Flow table represent what actions to be taken for particular SDN switch. It gives match criteria for particular packet. Flow tables consist of different fields like IP source, IP destination, protocol, Series of action. Actions can be drop packet, send packet to port, send packet to controller and modify fields. Working of SDN [12] is illustrated in Fig. 2.

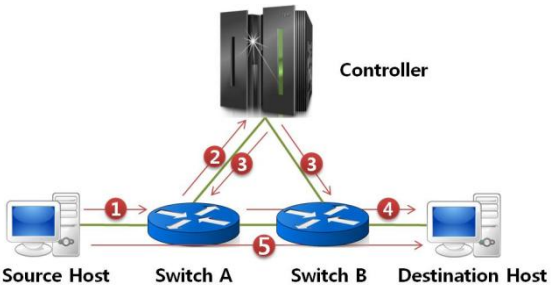

Fig. 2: Working of SDN [12]

(1) When source sends packet to the switch, it matches in the flow table of switch as shown. If match found then the particular action will be taken from flow table of switch. (2) If no match found then the packet will be send to controller. Then controller makes decision based on knowledge of the network and (3) install new flow entry in each network devices. Controller send packet back to the source network device. (4) After that switch will either forward the packet to the destination or particular action will be taken. (5) Finally packet will be send from source host to destination host.

1.4 Challenges in SDN

Though SDN has more advantages over the traditional networking, there exist some challenges [13] also as discussed below.

1.4.1 Quality of service:

In SDN, floodlight controller is used for minimal dependencies and to support number of open flow switches. The source code for interfaces in floodlight is easily available. Existing APIs in SDN needed more functionality to improve quality of service to make better ability of SDN to provide a service. It considers different parameters like minimum packet loss, minimum delay, and bandwidth. In traditional networking quality of service depends on many factors like prioritising a flow by restricting throughput other flow, traffic shaping, traffic policies.

1.4.2 Load balancing:

As SDN has global routing view controller can discover all the paths between source and destination. The SDN controller has capability to observe traffic of each server. It can manage incoming and outgoing traffic from server. Load balancing includes efficient module to reduce the packet latency which improves server and data transmission performance.

1.4.3 Scalability:

Management of large number of controller and switches is needed in SDN. Communication delay can be possible because of the placement of controller and switches. In case of controller failure switches needs the new link for communication. Scalability in SDN considers number of switches that controller can support, flow table entries for each flow and how controller is capable of handling switches spread across the network.

1.4.4 Traffic Engineering:

Traffic engineering include optimized network to provide more services. It allows to travel traffic over the less congested path. To allow the change in network topologies different traffic engineering techniques are needed in SDN. It can reduce service failure and deterioration. Its aim is to improve network performance by providing new path on link failure congestion. Optimization algorithms are useful for calculating new path.

1.4.5 Security:

Security is challenging issue in SDN. It considers protection of controller, DoS attack challenges, Intrusion prevention [1]. In next section security challenges in SDN are discussed.

1.4.6 Controller placement:

Controller placement is a key issue while designing distributed SDN control plane. It is necessary to decide where to position a limited number of resources. Performance of controller, inter controller latency Propagation delay, control path reliability, fault tolerance and application requirements are important parameters to be consider while controller placement.

1.5 Security challenges in SDN

Though the separation of data plane and control plane is fundamental feature of SDN, it also opens new security challenge [14]. Communication channel between the layers can be targeted for attacker. The control plane is more attractive for security attacks specially dos and ddos attacks as it is visible in nature. Security challenges are expected to grow highly as the deployment increases.

1.5.1 Application plane security challenges:

Third party application and multivendor development in different programming environment create security collision. Some of the security challenges are Authentication and authorization [15]. Authentication is major issue in SDN new trends. Most of the functionalities of controller are developed by other parties rather than the controller vendor. They allow using network behaviour without security mechanisms. Hence centralized control architecture security is the major issue.

1.5.2 Control plane security challenges:

Threats from application plane: Different applications developed from application layer can cause security issue in control plane [16]. Different application having different functional requirements needs different security. Before accessing the resources and network information by the applications there is need to provide security. Vendors must have different privileges to access information about network. Control plane security challenges are mentioned as below.

Scalability threats: Controller is able to handle huge amount of data flows. Controller can insert flow rule for new entry. Decisions are taken in centralized manner. The lack of scalability makes attacker to attack on control plane. Controller has to manage number of forwarding devices. As number of flows increases highly, there is possibility that time delay increases which can lead into controller failover. To avoid this multiple controller can be used.

Distributed control plane challenges: Large number of forwarding devices cannot be handled by single controller. The solution is to use number of controller distributed over the network which divides network into sub-networks. But then flow rules in the different sub-networks will be challenging [10]. If application is related to different sub networks then it may cause security problem in authentication and authorization.

1.5.3 Data plane security challenges:

In SDN security in control plane has direct impact on the data plane. If controller is failed then it will affect the whole network as it is a centrally located. If networking element like switch doesn't receive the forwarding rule from controller because of controller failure or any disconnection, the data plane becomes off. So switch to controller link is targeted for attack by attacker [17]. Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) are implemented in OpenFlow. Later on use of TLS kept as optional as the TLS configuration is complicated. After that the network traffic of data plane is passed over the firewall.

Different security attacks [18] in SDN are mentioned as follow

IP Spoofing : An attacker access the SDN controller by making fake user identity .An attacker can forward packets to the SDN controller with a source address indicating that the packet is coming from a specific port or system. So the attacker gain unauthorized access to the SDN controller using IP spoofing.

Tampering: Controller software or update packages may be modified by a malicious entity. An attacker can modify the controller's policies to a network element and redirect associated traffic to a specific destination for interception.

Information Disclosure: The risk of information disclosure includes unauthorized access to the sensitive data on the controller, such as backup flow tables, configuration data and topology data, etc.

Repudiation: In this type of attack one party participating in a transaction or communication, and later claiming that the transaction or communication never took place.

Denial of Service: An attacker can modify the flow table entries to perform a DoS attack. DoS attack challenges is discussed in next section

In this paper section 2 defines types of DoS attack in networking. Section 3 explains DoS attack in SDN. Section 4 depicts literature review. Comparative study is mentioned in Section 5. Section 6 represents proposed system. The paper is concluded in section 7.

2. DoS ATTACK IN NETWORKING

DoS attack is most important internet threat. DoS attack have become major threat to the computer network. It disables the service, downgrade the service performance by exhausting the resources. DoS attack becomes successful when the attacker intentionally consume resources that prevent from using the service to the target person. DoS Attack techniques are mentioned below.

2.1 Network based attacks :

TCP SYN Flooding: For client server communication firstly client send SYN message to server. Then server sends acknowledgement by sending SYN-ACK message to the client. Then establishment is completed by responding with ACK message by client [19]. The connection between client and server is open. Client and server can exchange data. The attacker arises at half open connection state that is at the time when server is waiting for client's ACK message after sending SYN-ACK message. ACK message will never send to the victim. It becomes unable to accept any new connection, thus targeted machine cannot provide services.

ICMP Flooding- ICMP ECHO request is send to the computer to check whether computer in internet is responding. After receiving request the computer sends ECHO reply packet. Attack happens when ECHO request overloads the target with so many requests. Target spends all its resources in responding. Thus attacker takes down the victim's computer.

UDP flooding-

This attack is created by sending large number of UDP packets to a random port on remote host. The victim machine becomes unreachable by other clients. Intermediate network can send higher traffic volume than the targeted machine can handle. Flooding can exhausts victim's connection resources.

2.2 Host based attacks:

These attacks are application specific like exploiting particular algorithm, memory structure, and authentication protocol. Attack can be generated from single host or number of hosts. This kind of attack can be happen on E-commerce website such that the site remains available to client but client is unable to purchase the items. The SSL/TLS protocol is used to make sure that the connection between client and server is secure [1].

DDoS is the type of DoS attack where multiple systems are used to target single system causing a DoS attack. In DDoS the incoming flooding traffic is generated from many different sources. So it becomes difficult to stop the attack simply by blocking a single address. It is very difficult to indentify normal traffic and legitimate traffic when it spreads across so many points. Common DDoS attack types are traffic attack, bandwidth attack, application attack. In traffic attack, huge amount of TCP, UDP, ICMP packets are sending. Due to this Legitimate request get lost. Bandwidth attack overloads the target with large amount of junk data, causes loss of network bandwidth.

3. DoS ATTACK IN SDN

Dos attack is intended to flood control plane bandwidth by creating number of new flows which can result into network failure for users. DoS attack on controller can damage the entire network [13] as controller manage the large number of switches/applications. Attackers are intended to flood control plane bandwidth by creating number of new flows which can result into network failure for users [13]. When switch receives new flow it buffers the packet before sending PACKET_IN message to controller. If switch receives number of new packet flows within very short time period its buffer fills up. Then it has to forward complete packet to the controller. This may cause the consumption of control plane bandwidth and it can cause delay in installing new flow rule. At some point switch is unable to forward traffic from new flows [1]. DoS attack on switch is caused by filling up the flow table memory. The networking element or switch has limited memory. This limited memory can be used for DoS attack. If switch's flow table is full, on receiving new flow switch detects that flow table is full. Switch cannot install this rule and send error message to controller and it drops the packet. In this case switch is unable to forward buffered packets until there is free memory space in switch flow table. This attack is local for a particular switch and it does not affect the whole network [20].

4. LITERATURE REVIEW

In literature different mitigation techniques are available for Dos attack in SDN. Recently many techniques have been published for DoS attack detection to separate malicious and legitimate traffic from the network as DoS attack is a real and growing threat.

4.1 Fresco [20]:

It is a security application development framework for OF-enabled network. It exports the scripting API's. It includes security and threat detection logic as modular libraries. Security Researchers can use it for implementation of security detection and mitigation module.

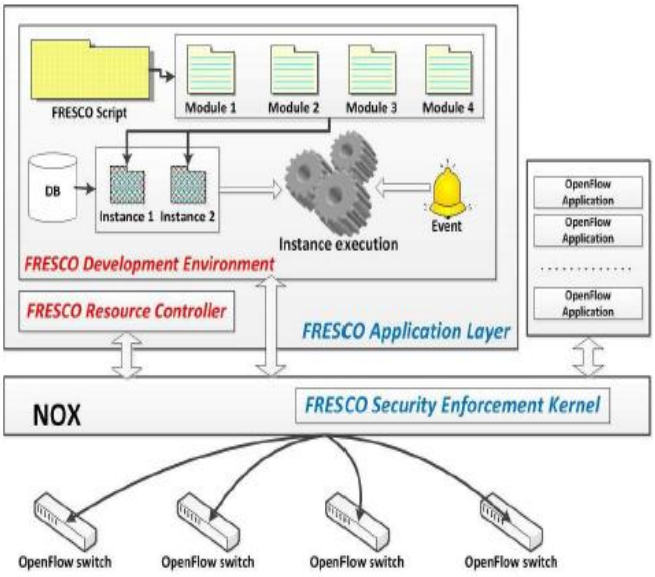

Fig. 3: FRESCO architecture [20].

As shown in Fig.3 FRESCO framework consist of application layer implemented using NOX python module and kernel layer. Application layer provides APIs for application development and kernel layer gives the action from the developed security application.

Developer uses FRESCO script language for interaction between the NOX python modules. Researchers are provided with useful information and tools for security control in FRESCO development environment. Script to module translation automatically translates FRESCO script into modules by creating instance from modules.

4.2 Avant-Guard [21]:

It is use to develop more scalable security features. Architecture of AVANT-GUARD is as shown in fig.4 it is an extension to open flow data plane called connection migration. It responses to handshake pickets if no match found. Connection is established when packet is send to controller. Actuating triggers are introduced for detection and response to the changing flow dynamics in data plane. Data plane proxy the TCP handshake.

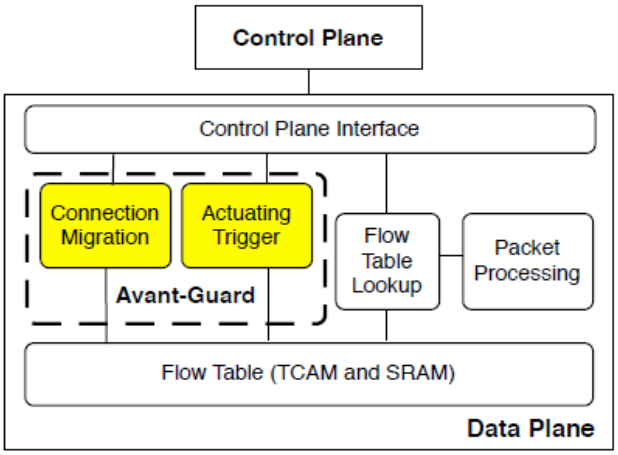

Fig. 4: AVANT-GUARD architecture [21].

It makes those flows visible who completes handshake. Data plane asynchronously reports payload information and network status to control plane. It generates flow rule based on predefined conditions. Control plane defines statistics condition for which notification is needed. Control plane resisters this condition into data plane. Then data plane checks this condition foe currently collected packets. When the condition is satisfied, data plane either give call back event to controller that this condition is met or it insert the entry into particular flow table.

4.3 Flood Guard [22]:

It controls data to control plane saturation attacks using proactive flow rule analyser extension. FLOODGUARD introduces two modules: 1) a proactive flow rule analyzer module, and 2) a packet migration module. The proactive flow rule analyzer can be activated at any time after detection of saturation attack. The attack is informed by migration module. Once activated it generates flow rules and install it into the switches as shown in fig.5.

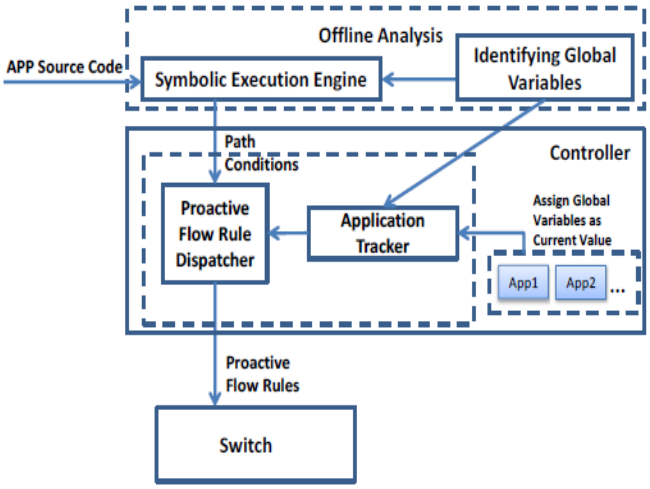

Fig. 5: FLOOD GUARD architecture [22].

4.4 Of-Guard [23]:

It prevents data to control plane saturation attack by packet migration and data plane cache. It filters attacking packets before sending them to controller. Architecture of OF-GUARD is shown in fig.6. This requires all switches to be equipped with OF-Guard extension. Packet migration computes rate of packet in data plane and get the percentage capacity of SDN network. Data plane cache stores flow rules, table-miss packets and it differentiate the fake packets.

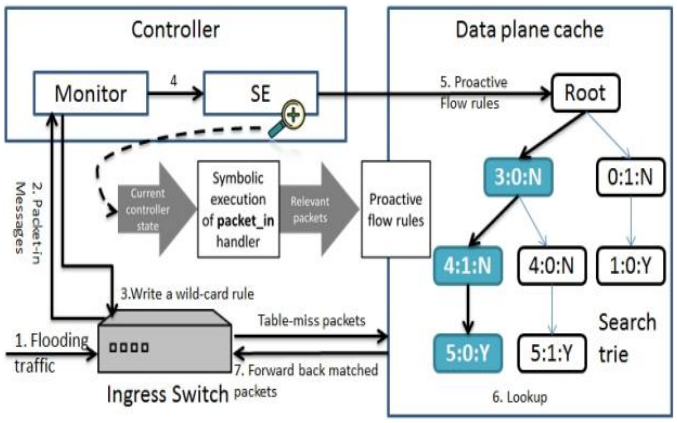

Fig. 6: OF-GUARD architecture [23].

4.5 Flow Ranger [17]:

It is a buffer prioritising solution for controller to handle the flow request. Routing requests are buffered into multiple queues. Higher priorities are given for legitimate users and lower priorities are given for new users. So attacking requests are served from lower priority instead of normal regular requests. It can reduce possibility of attack and maintain normal operation in the network Trust management tracks the trust values of the request users in SDN. If the user uses SDN when there is no sign of attack then trust value increases otherwise value decrease .In queuing management module each request is labelled with trust value of sender. If request is from highly trusted user then it will be served in higher priority, controller maintain buffer queues with different priority levels. Request scheduling modules computes weight of each buffer queue based on length and priority level of the queue.

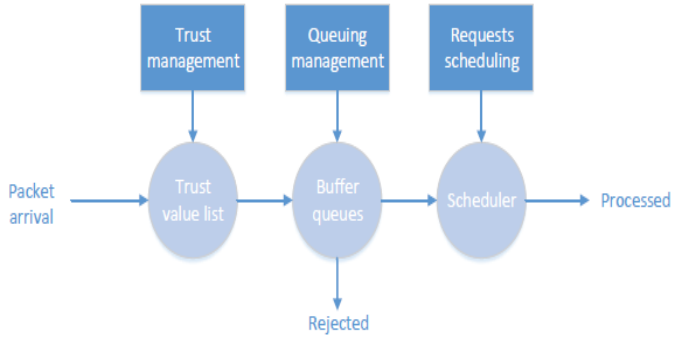

Fig. 7: FLOW RANGER architecture [17].

5. COMPARATIVE STUDY

This section compares the existing security techniques as mentioned in section 4. It represents solution and limitations on respective techniques. Different Security techniques for DoS attack are considered in table 1. They provide security in either data plane or control plane. All security techniques in table 1 provide southbound security in SDN.

Table I: Comparison of Different Security Techniques

Techniques	SDN Layer	Working mechanism	Limitation
FRESCO [20], 2013	Application layer	Application development framework to help to develop new security service for SDN in Application layer. Provides module linking using event triggering and data sharing	Limited security modules
AVANT-GUARD [21], 2013	Data plane	Secures Attack using TCP 3way handshake in data plane. Extension to data plane as connection migration and actuating trigger	Not effective for DoS attacks on controller
FLOOD-GUARD [22], 2015	Data to control plane	Secure Generic saturation attack for different protocols in data to control plane using proactive flow rule analyzer and packet migration	Difficult to handle table miss packets
OF-GUARD [23],2014	Data plane	Uses intermediate server that is data plane cache to filter attacks	Limited to known attacks and not effective on controller
FLOW RANGER[17], 2015	Control plane	Secure dos attack in control plane using buffer prioritizing algorithm	Rule cloning may be needed

6. PROPOSED SECURITY FRAMEWORK

Traditional security functions such as IDS, IPS, and Firewall can be implemented in SDN environment. As shown in Fig.8 IDS can be placed between controller and network element.

When host A sends a data flow to C the first packet of the flow will be sent to the controller because flow is unknown to switch. Then the packet is sent to NIDS. NIDS decides that whether the flow should go through packet inspection process. When it is required then it is notified to controller that the flow should be passed to both NIDS and destination. New flow rule is inserted in the switch. Switch duplicates the flow to its destination and NIDS. After that packet inspection is done and alert is generated if malicious traffic occurred.

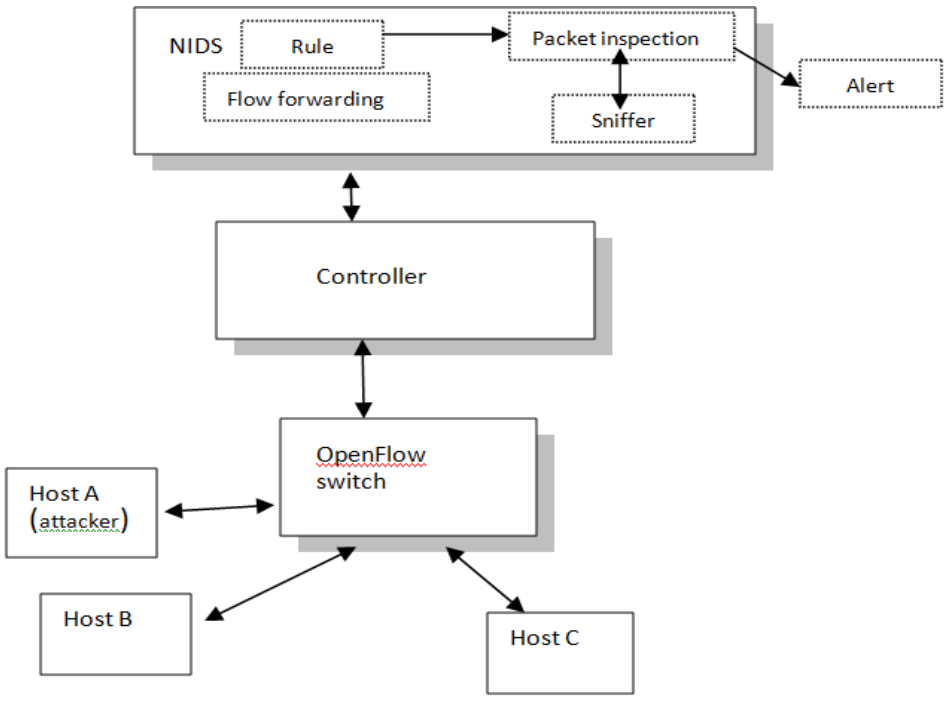

Fig. 8: Proposed security framework.

7. CONCLUSION

The proposed system works on dos attack detection and mitigation using intrusion detection system. This system is accomplished by flooding the targeted machines by attacker. IDS is used to detect and mitigate the dos attacks. We can extend this work to mitigate distributed Daniel of service attack that is ddos attack, where the traffic flooding originates from different sources.

This paper categorized security challenges in SDN. Details of DoS attack in SDN are described in the section 3. Different tools available to prevent DoS attack are given in section 4. The proposed solution over the DoS attack is mentioned in section 5. By resolving current security issues, implementing more securing techniques and further exploring characteristics, Software Defined Network may be well more secure.

8. REFERENCES

[1] Rajat Kandoi, Markku Antikainen "Denial-of-Service Attacks in OpenFlow SDN", IFIP 2015.

[2] Software-Defined Networking (SDN) Definition https://www.opennetworking.org/sdn-resources/sdn-definition

[3] The Open Networking Foundation (ONF) https://www.opennetworking.org/about/onf-overview

[4] Mininet Wiki https://github.com/mininet/mininet/wiki/

[5] Floodlight Is an Open SDN Controller http://www.projectfloodlight.org/floodlight/

[6] Production Quality, Multilayer Open Virtual Switch http://openvswitch.org/

[7] Software Defined Network https://en.wikipedia.org/wiki/Software Defined Network

[8] Understanding the SDN Architecture https://www.sdxcentral.com/sdn/definitions/inside-sdn-architecture

[9] Mehiar Dabbagh, Bechir Hamdaoui, Mohsen Guizani, and Ammar Rayes, "software-defined networking security: pros and cons", IEEE Communications Magazine — Communications Standards Supplement, June 2015.

[10] Hiep T. Nguyen Tri, Kyungbaek Kim, "Assessing the Impact of Resource Attack in Software Defined Network" 2nd ed., IEEE2015.

[11] Kannan Govindarajan , Kong Chee Meng , Hong Ong,"A Literature Review on Software-Defined Networking (SDN) Research Topics, Challenges and Solutions",2013

[12] Adnan Akhunzada, Ejaz Ahmed, Abdullah Gani, Muhammad Khurram Khan, Muhammad Imran, and Sghaier Guizani,"Securing Software Defined Networks:Taxonomy, Requirements, and Open Issues", IEEE 2015.

[13] Ijaz Ahmad, Suneth Namal, Mika Ylianttila, and Andrei Gurtov, "Security in Software Defined Networks: A Survey",2015

[14] Sakir Sezer, Sandra Scott-Hayward, and Pushpinder Kaur Chouhan, CSIT, Queen's University Belfast Barbara Fraser and David Lake, Cisco Systems Jim Finnegan and Niel Viljoen, Netronome Marc Miller and Navneet Rao, Tabula, "Are We Ready for SDN? Implementation Challenges for Software-Defined networks", IEEE Communications Magazine, July 2013.

[15] Lei Wei. , Carol Fung," FLowranger: A Request prioritizing algorithm for controller DoS attack in Software Defined Networking", IEEE ICC 2015.

[16] Threat Analysis for the SDN Architecture. Open networking foundation , July 2016.

[17] "Denial of Service attacks and mitigation techniques: Real time implementation with detailed analysis", The SANS Institute, 2011.

[18] S. Shin, V. Yegneswaran, P. Porras, and M. Tyson, "Fresco: Modular composable security services for software defined network In NDSS",2013

[19] Seungwon Shiny Vinod Yegneswaranz Phillip Porrasz Guofei Guy, "AVANT-GUARD: Scalable and Vigilant Switch Flow Management in Software-Defined Networks",2013.

[20] Haopei Wang, Lei Xu, Guofei Gu,"FloodGuard: A DoS Attack Prevention Extension in Software-Defined Networks", IEEE/IFIP 2015.

[21] H. Wang, L. Xu, and G. Gu,"Of-guard: A dos attack prevention extension in Software defined network, In Open Networking", Summit 2014, Poster Session. USENIX.