National College of Ireland

SDN Based Network Management for Enhancing Network Security

Research Project
Msc Cloud Computing

Ajay Kumar Mishra
Student ID: X01630165

School of Computing
National College of Ireland

Supervisor: Dr Muhammad Iqbal

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Ajay Kumar Mishra
Student ID:	X01630165
Programme:	Msc Cloud Computing
Year:	2020
Module:	Research Project
Supervisor:	Dr Muhammad Iqbal
Submission Due Date:	23/04/2020
Project Title:	SDN Based Network Management for Enhancing Network Security
Word Count:	6786
Page Count:	24

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on TRAP the National College of Ireland's Institutional Repository for consultation.

Signature:	
Date:	26th May 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

SDN Based Network Management for Enhancing Network Security

Ajay Kumar Mishra
X01630165

Abstract

Traditional network management techniques consists many vulnerabilities and are less viable to maintain the privacy of user's data. In this direction, Centralised management system has been emerged as efficient way to manage the network. Software Defined Network (SDN) is the dynamic and easily programmable networking system where the software manages the data flow and maintains the continuity of the networks. SDN separates the data-plane and control-plane in order to manage the network efficiently and enhances the network security. The proposed Smart-Net system where each router maintains a Flow table entry, if the entry existed in flow table the packet will be forwarded to SDN controller. SDN controller checks for the unusual behaviour in the system and takes the preventive actions in order to stop security attacks such as Sniffing attack, re-directional attack, Denial of Service attack. This paper also encompasses a comparative analysis between Traditional networks and SDN based networks with predefined topology.

1 Introduction

The advancement of cloud technology has made a revolution in the digital society. From the last three decades, the researchers have developed sophisticated cloud systems that include the enhancement of networking topologies, data storage, data manipulation and complex computations. Leading from the front, this technology has provided immense opportunities for the small, mediocre and large scale research and business. Through this technology the business enterprises develop the intra-relations with the clients, as well as they use this technology to gain the financial gains. But with the advent of modern cloud systems, the cyber-attacks are also increasing in numbers where the intruders intrude into the system to steal user data. This in-efficacies are mainly due to the network failures that lead to the worst cyber-attacks.

The research of the last three decades has increased the success of cloud networking by the quality transformation from the traditional networks to the software define networks (SDN). The traditional networks make cloud systems apprehensive due to their decentralized nature and reachability problems. For example, in the traditional networking topology, the data is transferred serially i.e. from one node to another. These networks have possible information about the nodes that are adjacent to each other but have no access to the distant or disconnected nodes. This means that these networks are not centralized and may lose the data due to the network mismanagement, cyber-attacks, node failures or unbalance traffic load.

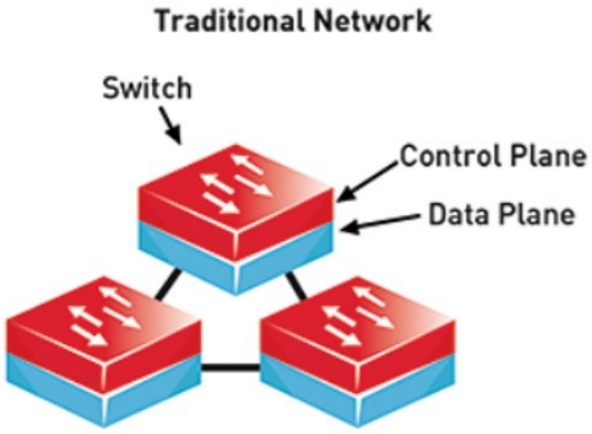

Figure 1: Traditional Network architecture

In general, the network decentralization has caused data reachability and decryption problems. The proposed methodology addresses cyber threats issues which are mainly due to node failures or re-direction attack. The possible reason for the redirection is network decentralization where no information of distant nodes is present so the network manager does not know where the data packet is directed from the one node to other node. This lack of information makes it easy for the hackers to extract the information of the nodes where they redirect the packets at adjacent nodes towards the hacker network. This failure pointed out that the traditional network is not well encrypted and centrally managed thus cause the hijackers to decrypt the data at the multiple nodes that is commonly known as the domain name system(DNS) poisoning and hijacking. The other problem arises which is node failure that is the root cause of traffic congestion and the data redirection. In the system of series connection, one or multiple node failures will cause the data interruption thus redirect the data towards the other adjacent nodes. This makes the nodes loaded with unusual data and produces congestion. In other words, the node failures are also responsible for the data unreachability and help the intruders to decrypt all the data present at the nodes.

Based on these drawbacks the main purpose of this project is to study the software-defined networks that solve the decentralization problem and provide a centralized system for network and its management. SDN is the dynamic and easily programmable networking system where the software manages the data flow and maintains the continuity of the networks. SDN networks are useful because they provide separate data (forwarding layer) and the control layer that helps to prevent the network from physical or cyber failures.

The network security has been considered with implementation of SDN networks. Due to the holistic understanding of the network, the SDN treats the node failures and redirections in an organized way with minimum loss. The advantage of the SDN networks is their autonomous capability to limit the cyber-attacks caused by the ineffective redirection. SDN helps the data to solve the redirection attacks with the centralized capability and its intent-based behaviour where the malicious entities are tracked by the central monitoring tool.

A custom SDN (motivated by Ivey et al. (2016)) which is implemented in python to reduce the redirection attack, damage control and prevent network from DDoS attack. The proposed smart-net primarily provides an effective solution to the redirection attack as well as the techniques to manage the node failures. Our security solution mainly

focuses on two things. First, we redirect the unusual activity from the optimal path to prevent the useful data and the other restricts the inflow to the router as compared to the outflow to prevent the data flooding and TCP back off normally intended by the hackers. Second is to prevent the network from fake packets or heavy traffic, processed by distributed denial-of-service DDoS attack. I have used tree-based network to study the impact and analysis of SDN over a non-SDN network. We implement this system using python classes, that is helpful for network visualization and implementation of the modern techniques. This method successfully limits the traffic congestion, node failure, redirection attacks and DDoS attacks and proves to be efficient as compare traditional networks.

1.1 Research Question

- How efficiently SDN based framework can overcome the challenges of traditional networks.
- How SDN based architecture helps to enhance the security from various network attacks.

2 Related Work

2.1 Role of Software Defined Networks in Network Security

It is obvious that the network is made up of the physical entities like a switch, router. which means that it is impractical to detect the network problems at the hardware level, thus require a software system aligned with the hardware to maintain the network problems and security. It is reported in Rose and Brown (2018) that availability of the resources (like physical and virtual resources), data integrity and confidentiality are the core things addressed by the network security. SDN offers a dynamic data flow policy within the network that is responsible for network abstraction virtually, migration and VM instantiation throughout the data centre. Its dynamic concerns, adjustable programmability and end-end automation boosts the network reliability Vizarreta et al. (2018) and limit the vulnerabilities that harm the network stability.

As compared to the SDN, traditional network architectures (TAN) observes the various protocols, software, and infrastructure threats Khan et al. (2016) that make the TAN unstable. Due to complex intertwined of the control and data layer, it's become difficult for the TAN to restrict the intruder's access and traffic congestion which demands a network that provides the isolated control layer to preserve the network credibility. TAN is also not perfect for the networks that spread around the different geographical locations because the traditional networks are controlled at the hardware level, where dealing with the hardware at the large geographical scale is a tremendous downfall to the network security because it requires huge resources and returns zero to little security and financial gains. It is reported that the TAN's are not capable to control a large number of network devices (fixed devices) responsible for the control inefficiency in the networks Rawat and Reddy (2016) that is also seconded by the lack of suitable geographical control and hence required a software-based network where the instructions are populated by the help of software system autonomously to control the large scale network. As compared to the

traditional network architectures SDN and its role in network security is classified into assessment and challenges that are provided below.

SDN Security Analysis As mentioned in Rose and Brown (2018), that identification of applications involved in the transactions are important by the system, out of all the quires there should be one identification query for target application involved. It introduces an indexing mechanism for the security of data involved in transaction. SANE was first proposed by Casado et al. (2006) which stands for secure architecture for networked enterprise, which permits users through http server which donot match with the recent topologies. ACL,s stands for Access control list which is restrictive in nature, SANE can be easily clubbed with todays network to address scalability issues. SANE uses DC (domain controller) which works as central component for authentication service, SANE service model involves four steps starting from secure channel authentication for communication to second step which allows server to publish with unique name in NSD(Network service directory) and the third step where permission is granted to access server, the final step involves permssion to establish communication through packets. For instance, the researchers felt Zlomislić et al. (2017) about tangible nature of DDoS attack with the advent of technology the attack mechanism also has evolved with time which requires reliable counter mechanism in todays cyber world, It also emphasis on different vulnerabilities present in the system ie: Disk space, maximum open connection, overloading of communication link may tend to Network attack and last one such as critical node ie: IPS (intrusion prevention system) which cannot be ignored.
The transport layer security (TLS) is equally essential and it is reported in Open Flow vulnerability assessment that lack of TLS increase the denial of service attacks on the network. Thus, the TLS method is reported in that use the mutual authentication of the controllers along with their switches to protect the network from utter vulnerabilities Dayal et al. (2016). However, the research reported in concludes that SDN requires more security layers because the malicious content will be added in the SDN responses to make it unstable Oktian et al. (2017). Also, researchers reported that ProtoGeni Rahouti (2019) is used the study based on airline operations choosing Internet protocol IP based technology due to its complex nature of operation at various levels, it is not possible to cover the entire range of services involved in its operation ie: SATCOM (satellite communication), AERO-MACS (aeronutical mobile airport communication system) and the next generation framework used by U.S aviation industry. It has been observed that the modern avaition industry is more inclined towards COTS (commercial of the shelf) and SOC (system on a chip) range of devices which has replaced the traditional components used in avionics. The flooded malicious attacks in the SDN testbeds to reduce its security. Hence it is concluded that the security of the SDN networks is restricted to generalized cases so in this research we introduced the new redirection technique that reduces the traffic congestion and data flooding for the enhanced security of the SDN. Liang and Znati (2019) has studied the empirical evaluation of machine learning methods for denial of service attack detection. A comparative analysis with ML-based techniques has been performed including the impact of class imbalance problem. The problem of imbalanced classes should not ignored, this highly effects the performance of model, Liang and Znati (2019) has proved by performing various experiments.

Prevention from DDoS attacks Hong et al. (2017) has studies about the slow http DDos attacks, which are difficult to detect in the network. Based on the traffic pattern

they have proposed a defense technique, which uses software defined networks. Another study has been done where author Byun et al. (2019) has introduced an attack scenario in order to avoid the forged packets using SDN. Their risk avoidance methodology is able to prevent from IP forging attack, Openflow protocol was used with the help of SDN to solve this problem. Huang et al. (2020) proposed low cost distributed DDoS architecture, their architecture is based on the Botnet growth model and work discusses about the various DDoS preventive strategies.

SDN Security Challenges The research community emphasizes the concern on the SDN policy conflict resolution. SDN is a dynamic system that provides several settings to policies to be enforced in the system to redirect the correct data flow but the challenge of the policy conflict arises in the advanced software-defined networks. To detect the policy inconsistencies model checking is an important step. Some work related to model checking Vinarskii et al. (2019); Shi et al. (2019) were reported to detect the flow policies but they are single-layered and are not effective in SDN sustainability. On the contrary, multi-layer or slicing methods Wang et al. (2018); Li et al. (2019) were reported in that isolate the network layer from the data layer and is helpful for system integrity confidentiality.

The second challenge in the SDN is its design that is the useful pillar of system stability. According to the literature survey, the design method reported in Ashouri and Setayesh (2018) is a great contribution in SDN stability. This research is useful but still lack in its functionality and is inevitable for future development. The third challenge in the SDN security is scalability. Scalability demands the system virtualization and network verification that is lagged by most of the SDN's. This drawback is somehow addressed by Akiyama et al. (2016) where they worked on the scalability and related issues of the Open Flow system. In addition to this Trakadas et al. (2018), the work reported by also provided the opportunity to the SDN developers to find the root bug in the system by keeping the system scalable and virtual.

Hence it is concluded that the issues of scalability, design, and policy conflict will make the SDN's outdated and not optimized. It is a fact that these factors indirectly reduce the adaptability of the software-defined networks and addresses security challenges by proposing scalable method to optimize the golds of the SDN's.

2.2 Role of SDN in Health, management and Public Network Safety

SDN's are considered vital for the services where the prioritization is required. For example, the software-defined network is helpful to assist the traffic management, health management and other priority services that are required to be executed instantly. The work is reported in Canovas et al. (2020), where they implement the SDN based traffic management system to reduce the traffic delays. This work is of prime importance because the data of the traffic signals and camera surveillance was collected by IoT framework and the routes are optimized by the help of SDN optimization.

Considering health services, SDN is fruitful for the health prioritization tasks. With the advent of wearable robot systems, the data collection is easily accessible. The work is reported in Hu et al. (2015) where the HealthIoT system is developed using the centralized SDN where the patients' record and data are controlled centrally to monitor the health status of the patients. Another work is reported in da Silva et al. (2019), where

the context-aware mobile approach (CAMA) is used with the SDN infrastructure to develop eHealth care in Brazil. This work uses the context-aware approach to handle the eMobility, mobility prediction and other eHealth care management. It is also reported in Andriopoulou et al. (2018) that the traffic congestion and other redirecting attacks make the remote health care and delivery system down thus they have proposed an SDN based network for proper remote media transmission between the doctor and patient that is independent of the Internet service provider connectivity. The interesting work related to health devices communication was reported in Sallabi et al. (2018) where the SDN is deployed to manage the personal device assistant based on the billions of the customer data. The application of this work is to provide useful health assistance to the customers on their request. Overall, it is found that SDN is useful for developing remote and fast health assistance systems for maintaining the balance between online and offline health care services.

Parallel to this, SDN is a great networking tool for home management and maintaining user privacy. The work is reported in Alshnta et al. (2018) that aims to improve user privacy by maintaining the ISP services. This work states that the increasing use of mobile devices at the public or home place has a lot of acquired data thus SDN will help summarize the data for the useful application. Besides, the public safety networks (PSN) are popular among the users as they are designed and dedicated according to the application specification. Software-defined networks are usually famous for the scalability and the flexibility in their software networks. As PSN relies on the application-specific requirements that means it is used for different applications like web streaming, remote localization, location mapping and others that require the flexible network system to be maintained. Thus, the work reported in Wetterwald et al. (2016) has induced the concept of software-defined networks to improve the flexibility of public network security for disaster management services.

Based on the previous research we have found that the SDN scalability and flexibility is vital for the health care, public network security and the management that requires prioritized networking. Here we propose the Smart-Net SDN that will be deployed in health care and disaster management scenarios to remove the potential barrier that non-scalable networking will face during the operation. This thesis proves vital for the application of SDN in health management and emergency networking and leads the new pathways in secured cloud computing.

2.3 Scalability of SDN and its Issues

Traditional networks are directly proportional to user customization services, where they required skilled professional for maintaining virtualization and also required hefty challenges if the physical network manipulation is required Sezer et al. (2013). However, SDN due to its centralized network structure has provided the scalable network solutions to the various networking applications. Scalability in SDN may refer to processing (CPU) optimization that leads the network for parallel processing of different tasks. It also refers to the parallel processing of different inter-connected machines as well as the optimization of SDN algorithms, architectures, etc.

Scalability demands the managed tradeoffs between the different network properties like matureness, reliability, flexibility, resiliency and its performance to cope up with a stable design of the networks Song et al. (2017). Scalability cure of the network may affect the other properties. For example, the proactive rules in the software-defined networks

cause the slow processing speed of the due to the inclusion of the switches and may reduce the dynamic operations management of the SDN. This work yields that scalability has a direct relation with the network synchronization and computational stress on the performance of the controller. The performance of the controller has also a direct impact on the scalable network designs. The work reported in Ahmad and Andras (2018) has claimed that the network controller layer has various metrics to follow i.e. link utilization, path installation time and target dependence. The noble work done in Hu et al. (2015) has studied the control layers of the network by defining the productivity metrics of the parallel distributed system that enhanced the adaptability of the networks.

The major contributors in the scalability of the networks mainly deal with the separation control and the data layer. Here we will explain the prime reasons that contribute to the scalability are:

Separation of the Layers The separation of the control layer with the data layer produces the signalling overhead error that is caused by an additional load on the controller layer. For example, when the two layers are separated the management is carried out by a remote-controlled place. The data layer has no direct access to the data packets thus all the optimization must be processed by the controller that causes the signalling overhead and disturbance between the two layers. This overhead may cause the control plane scalability that indirectly disturbs the network separation.

SDN Controller and its Request Handling Capability Even with the centralized control structure, the single and hybrid control architecture has a direct impact on the SDN controllers and its requests handling capability. It is reported in Karakus and Durresi (2017) that the increasing number of the network devices in the network may cause the disturbance in the single-layer controller besides the multiple layers. The increased flow of the requests and the data from the clients in the network causes the troublesome issues to the scalability of the system, SDN is not capable of removing this bottleneck Benson et al. (2010) and also force the delay of control plane activity by the help of disadvantageous programming.

Communication Delays The flexibility reached with the implementation of controllers which in turn makes network more responsive, It also elaborate link failure as a comprhensive study of control path rerouting which creates the worse case scenario of time delay between switch and controller, It uses PSO (Particle swarm optimization) algorithm to overcome delay issue. Fan et al. (2019). This thesis, introduces smart network techniques to improve communication delays that will utmost reduce the scalability and prioritizing problems in the network.

2.4 Other Applications and Challenges

SDN and IoT The increasing number of the application of the Internet of Things in the digital network has increased its pace over time because it provides the networks to be agile and have multi-purpose communication with the inter-connecting devices. Several IoT systems are highly adaptable in solving communication problems and deals with the other IoT network problems. This adaptability usually personified with the SDN network Chen et al. (2010) where both exhibit the same characteristics of solving the problems in an adaptable way.

SDN is used with the IoT systems to promote cellular communication in the software-defined way that may treat the customizability and automaticity of the network. It is reported in Haque and Abu-Ghazaleh (2016) that centralized structure of the SDN is highly compatible with the wireless communication of the networks which enable the SDN to easily populate the route commands with increased efficiency and decreased communication lag. Tselios et al. (2017) analyzed the common security issues of SDN in order to combinely use the IoT service with block chain technologies.

SDN has unique properties to manage wireless sensor networks. The work is reported in Abdolmaleki et al. (2017) where the SDN-WISE is proposed to support the data aggregation and state full addressing. This work managed to enhance the SDN-IoT policies to retain WSN management during the cellular transmissions. The IoT dependent system is introduced in Uddin et al. (2018) where the architectural solution is proposed that includes the SDN as the base routing or control station with many child stations. This system improves the network agility to maintain network stability. The concept of multiple virtual networks on the single virtual space is proposed in Ma et al. (2017) where the same abstraction is provided to the multiple channels that means the same infrastructure is given to developing the multiple infrastructures that increased the optimization power of the network.

Allocation of resources, quality management and other parameters are the core duties of the IoT systems. SDN has also improved the quality of the IoT services to increase system sustainability. The work is reported in Xu et al. (2019) where the multi-layer network controller is proposed that mimics the SDN layered architecture and is used for the task allocation purpose. The rise of MIOT(mobile internet of things) which outsource the sensing data. It discuss about crowedsensing framework which collects data through cloud which acts as controller from distant edge nodes. SDN is incorporated by cloud and edge nodes which collects data from mobile users. El Kamel et al. (2017). However, the SDN with the IoT system is the best choice for the network operation but there is still a quality lag present. Thus in this work, we have developed the tree-like SDN structure to route the cellular packets that resemble with the IoT structure and is used to minimize the redirection losses.

SDN for Mobile Network Security It is a common problem in the mobile network communication that the drastic increase in the real and virtual mobile networks causes the network working capability Chiesa et al. (2016), service quality, network cost and the speed of transmission. As compared to the traditional IP based mobile networks that are not scalable and reliable, mobile software-defined networks (SDMN) a combination of the SDN and the network function virtualization (NFV) are deployed in the mobile networks to cope up with the mobile congestion and traffic redirection.

With the migration of SDN to SDMN causes security issues as reported in Farshin and Sharifian (2017). The Open Flow network that is next generation of mobile network security is still in challenging phase and left the SDN with the issues of the point of attacks, security metrics, countermeasures, etc. This challenge along with the hybrid networks also increases the security crises that made the single-layer security crisis to the several layers thus increase the network skewness and instability. It is reported in Sinh et al. (2017) that NFV is also responsible for increasing the security threats to the network system. Even with the SDB virtualization, NFV allows the malicious contents in the network that produces negative backhaul thus SDN or SDMN systems must follow the security ethics and prevent the unusual attacks.

Software-Defined Spanning Tree Software-defined networks usually face the problem of forwarding loops that are caused by the flooding of data packets in the network. The spanning-tree solution to SDN is excellent because it decreases the flooding as well as become responsible for the topology awareness in the centralized framework. The main constituents used in Ivey et al. (2016) are also used in this work and these are listed as OpenFlow Protocol, libfluid SDN Library, ns-SDN, SDN Controller, and OpenFlow Switch. One of the advantages of the spanning tree algorithm that it is helpful to offload the forward looping present in the controller. It helps to produce effective communication between the different loops that make it helpful for reducing the hectic redirection attacks and unusual traffic congestions. In short, the spanning tree optimization defines the parallel routing of the data packets and thus optimize the packet processing power, autonomous decision making and SDN dynamic behaviours. The comparative analysis for different work is shown in Table 2.4

Paper title	Method	Advantages	Future Scope/Disadvantages
Detecting DDoS Attack on SDN Due to Vulnerabilities in OpenFlow Ali et al. (2019)	Time and space-efficient solution to detect the DDoS attack on SDN.	Solution consumes less computational resources and space also no need of any additional equipments.	Openflow Protocol has been used, which contains many vulnerabilities. Proposed solution is only for DDos attack, other attacks are not considered.
Detecting DDoS Attack using Software Defined Network (SDN) in Cloud Computing Environment Bhushan and Gupta (2018)	Proposed a customized algorithm to compute information distance to distinguish between attack and non-attack flows.	They claimed their approach can detect the DDoS attacks with very low communicational and computational overhead.	They have not described how they will prevent the system if DDoS attack is detected.
SDN-Based Double Hopping Communication against Sniffer Attack Zhao et al. (2016)	DHC method has been used where, ends in communication packets with routing paths are dynamically changed.	Method is effective for detecting sniffing attacks.	DHC requires both old and new flow entries in flowtable simultaneously , therefore the cost of flow table entries space increases to a great extent.
A Proactive Flow admission and rerouting scheme for load balancing and mitigation of congestion propagation in SDN Data Plane. CN et al. (2019)	Used Bayesian Network for effective port utilization and deciding alternate paths	Effectively utilises the bandwidth and balance the network load to improve QOS and network efficiency	Proposed system only concerned about load balancing of the network, security aspects are not covered.
SDN Based Network Management for Enhancing Network Security (Our Approach)	Uses the concept of SDN and maintaining the Flow table entry also checks for unusual behaviour in the system, All the unknown packets are sent SDN controller for central management.	The proposed method mitigates the DDoS attack with Sniffing attack also maintains smooth functioning of network even after node failures.	In improvisation, more kind of attacks can be integrated.

Table 2.4 : Comparison of Different Works with proposed framework

3 Methodology

The traditional network system are highly prone to network attacks and can lead to data-leak and steal packet information from the network. Also if the router is damaged, it may cause high number of packet drops. Message encryption and network separation are two most prominent solution to prevent the data leakages but if the system is vulnerable intruder can access the private key to decrypt the message and separating the network involves very high budget to implement the model. To prevent the public network from such kinds of attacks we proposes SDN based network mechanism. SDN based network separates the control plane and data plane from a network. A centralised mechanism is established in order to prevent the network from unauthentic source. The proposed method will be explained in Figure 2.

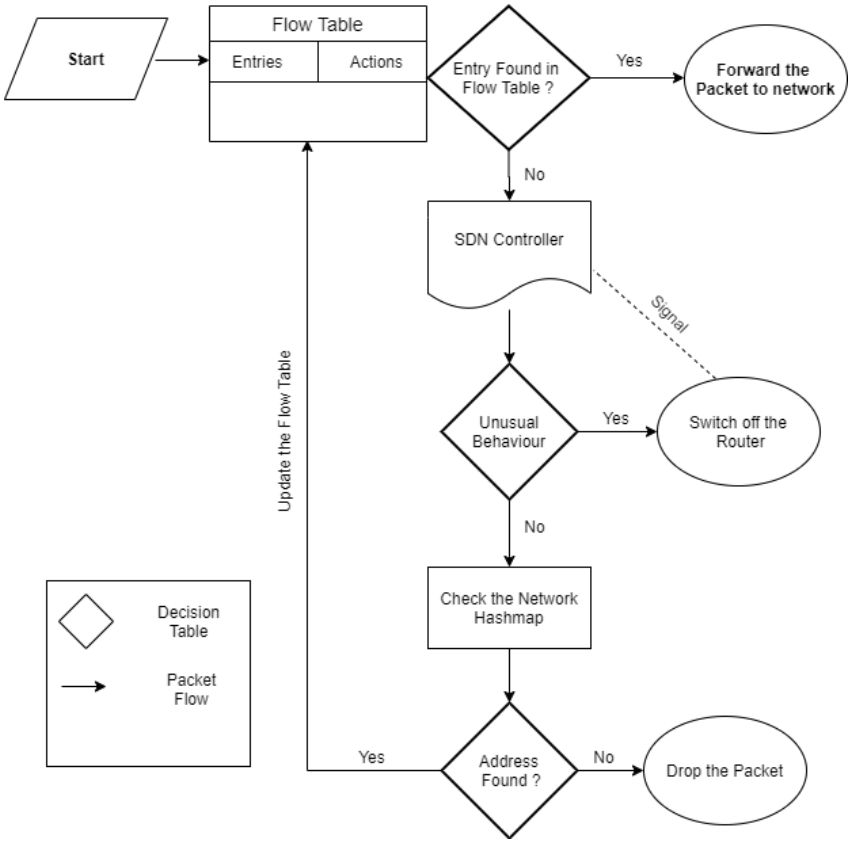

Figure 2: Flow Diagram of Proposed work

In the proposed architecture each router maintains its own flow table. So when the packet arrives at the Router. It check its flow entry from its local flow table. If the packet entry matches then it process its flow table entry (FTE) and forwards the packet to other router or towards the destination. If the Flow Table entry (FTE) is not found then packet is forwarded to the SDN controller. To prevent the network from the Distributed denial of service (DDoS) attack, SDN detects for unusual behaviour in the network component.

If the high amount of network traffic is found and number of packets in the router are more than the average then SDN controller will send the signal to disable the router. If the unusual behaviour is not found in the system SDN controller will check its network hashmap for finding its address. If the address is not found then SDN controller will drop the packet and if the address is found then SDN will send the instructions to router for updating their flow table. The proposed architecture is able to detect the redirection and DDos attacks from the network and is also able to take necessary and preventive action in order for smooth functioning of network.

4 Design & Implementation

To implement the proposed architecture we have simulated various scenarios in order to represent the effectiveness for software defined networks. We have designed a network topology which will help to understand the concepts in detail. Proposed network topology consists of total 13 nodes. Nodes can be a client, server or a router. All network component will communicate with each other via IP address. For identification, Static IP is assigned to each of the component in the network. There will be 4 clients which will generate the packets that will be used to feed into the network and 3 servers will be used to receive the packets. The topology also consists of 5 routers to carry out the packet forwarding among the 13 nodes, 5 nodes will be used as the routers for packet forwarding. To demonstrate the attacks on the network we will simulate the hacker node, which will act as a server and try to steal the packet from the network. Here, the hacker node tries to capture the forwarding table of routers. Now we will discuss about the multiple scenarios in subsections. To explain these multiple scenarios we have considered two network topologies one with SDN and another is without SDN (Traditional Network toplogy) shown in Figure 3 and Figure 4

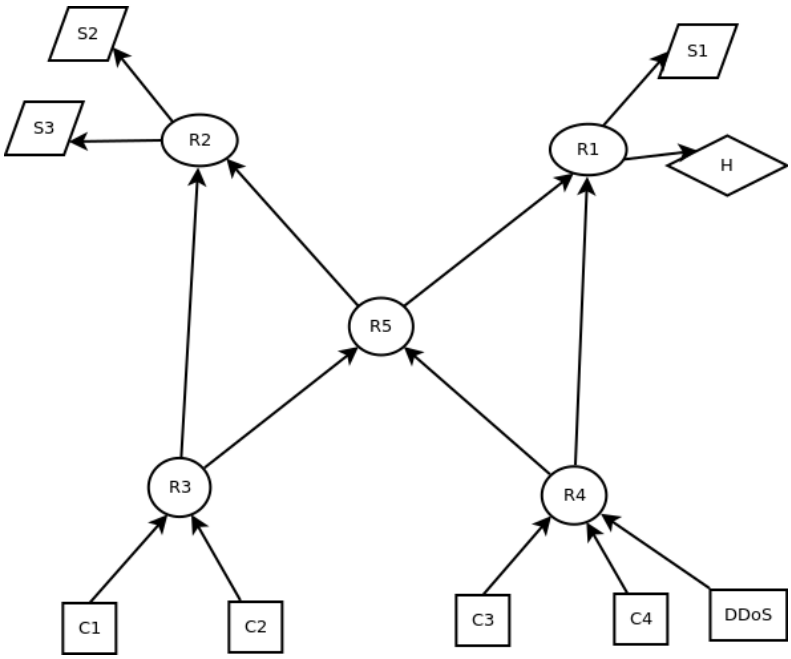

Figure 3: Network Toplogy (Without SDN)

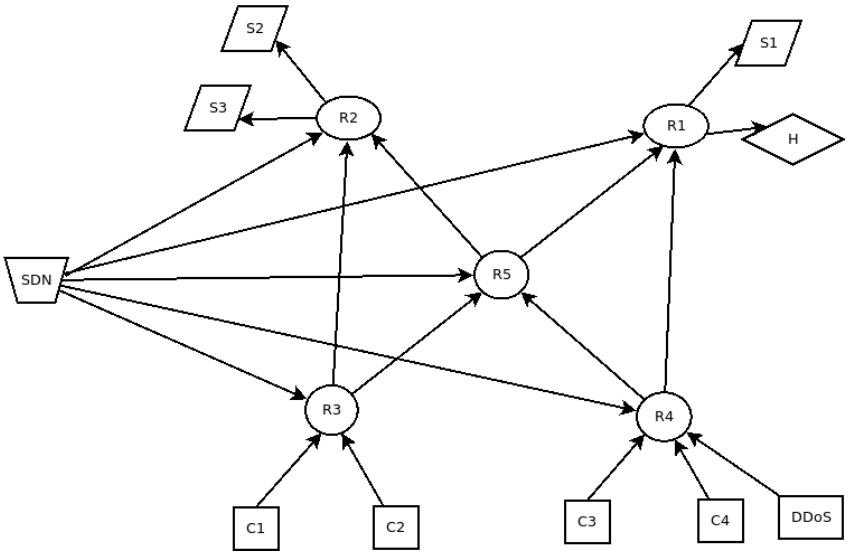

Figure 4: Network Toplogy (With SDN)

4.1 Network Topology (Without SDN)

Traditional network mainly relies on the physical infrastructure. The physical infrastructure includes router, switches, servers etc. The target of network is to securely transfer the information ie:message from one end to another. Here we have implemented 2 different attacks which includes Sniffing attack and DDoS attack. We will intentionally attack on our traditional network.

4.1.1 Sniffing Attack

For sniffing attack, hacker will try to capture and analyze network communication information. That will be done in following way. Client C1 will generate the data packets and send to Router R3. When packet reaches to router, it can extract the source IP and destination IP from the packet. Then the router searches for its local forwarding table. If forwarding table contains the destination address, it forwards the packet to destination IP. If forwarding table contains the address of other router rather than destination address router forwards the data packet to that router or address is not found it will ping the servers to ask if that packets belongs to them. In our case, suppose R3 will forwards the packet to R5 and R5 will forward the packet to Router R1. Now, Hacker node will impersonate itself as a Server S2. All the data packets will be transmitted to hacker node by R1. This activity can not be stopped because no other component of network knows that network is compromised, all the packets of S2 can be received by hacker node.

4.1.2 DDoS Attack

DDos attack directs extensive amount of traffic to web servers, router or switch in order to render the website or services inoperable. The high amount of traffic will be generated in the network by sending the fake packets requests. In this experiment, the DDos node will keep generating the fake packet request and send to router R4 and Router 4 either will forward these packets to other routers or will not be able to handle the request and

will be crashed. C1 and C2 will not be able to transfer any packets. This may affect the complete network and can disable the services.

4.2 Network Topology (With SDN)

Using the Software defined network all the network elements will be managed with the help of software. SDN separates the control plane and data plane from a network. Our proposed model using SDN can centrally manage the Flow table entry of routers and can prevent the system from various network attacks. The network topology in the Figure 4 will be followed and we will show how the SDN prevents the network from being attacked.

4.2.1 Sniffing Attack

The same scenario shown in subsection 4.1 will be considered here. Client C1 will generate the data packets and send to Router R3. When packet reaches to router, it can extract the source IP and destination IP from the packet. Each router maintains its own flow table. If the packet entry matches then it forwards the packet to destination IP or to the next available router. In our case, suppose R3 will forwards the packet to R5 and R5 may forward the packet to Router R1. Now, When hacker node attacks on router it may redirect the packets to hacker node instead of server S2, As all the tables are managed centrally by SDN controller there is a no chance that packet will be forwarded to R1. In this way, SDN prevents the network from sniffing attack.

4.2.2 DDos Attack

As we know that, DDoS attack directs extensive amount of traffic to web servers, router or switch in order to render the website or services inoperable. The high amount of traffic will be generated in the network by sending the fake packets requests. In this experiment, the DDoS node will keep generating the fake packet request and send to router R4. If the router will have the Flow table entry about the packet it will forward the packet else it will send the information about it to SDN controller. SDN will detect unusual behaviour in the network if unusual behavior is found, it will disable the resepective router. If unusual behaviour is not found SDN will check the network hashmap if address is found it will update the local flow table of all the routers. If address is not found it will drop the packets. In this scenerio, SDN prevents the network from DDoS attack.

4.2.3 Damage Control

SDN is efficient to manage the damage control in the network. If any node is physically damaged and the other routers are still sending the packets in tradtional network the packets will be dropped. As SDN centrally manages the information about network equipment it will update the flow table of every router and prevent the packet drops. All the results of this implementation will be shown in result section.

In order to implement the proposed architecture we have used the python language. To simulate the network environment we have used concept of threading. Each of the network and server will have capacity that can be defined by number of packets or number of tasks. Client, server and router are the python classes which simulate the virtual networking environment. The message queue where the packets are stored uses the concept

of queue data structure. Similarly, every node in the network are connected via local IP address. The monitor class of proposed architecture is used to provide graphical representation for analysis activities. The system with following configuration is used in order to implement the proposed architecture shown in Table 4.2.3.

System/Software	Specifications
Operating system	Ubuntu18.04
RAM	4GB
CPU	4 Cores
Programming Language	Python3.8
Data Structure	Queue
Libraries	Threads, matplotlib

Table 4.2.3 : System Specification

5 Results and Discussion

In order to have a comparative analysis between traditional network and SDN, we have performed different experiments for different attacks on the network. All the details of experiments and their results will be explained in the subsections.

5.1 Experiment 1 - Sniffing attack (Without SDN)

In this experiment, hacker node try to capture the forwarding table of router. In the traditional network as there is no central administration to control the network. So even R1 node is manipulated by hacker node, Still the other nodes/routers will not update their forwarding table and send the packet to r1 only. In such scenario sniffing of packets become very easy and hacker node will continuously receives the information/message which should be actually received by S1. The flow of packet per unit time in traditional network (Without SDN) is shown in Figure 5

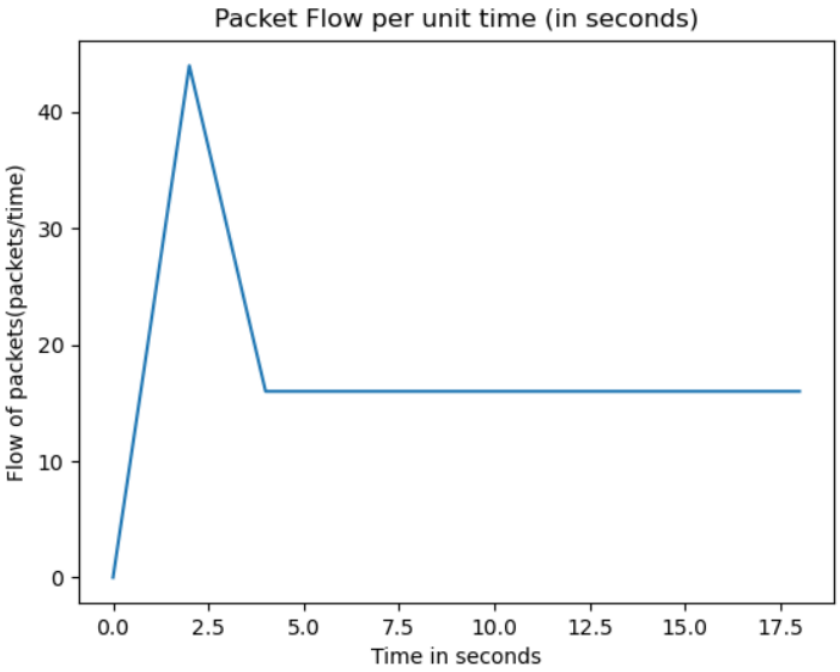

Figure 5: Packet flow per Unit time for Sniffing attack (Without SDN)

In this experiment, hacker node is able to capture the packets of S2 by capturing the forwarding table of R1, this information can be clearly seen in the log file of traditional network. The screenshot of log is attached in the Figure 6 . The graph shown in Figure 5 also represents that number of packet per unit time is very high due to unnecessary forwarding of packets in the network. The sniffing attack is implemented with damage control activity, which controls the network, after the failure of router. The number of packet flow has been reduced as many packets can be dropped because R5 is crashed and is not available for the operation.

```
message->8 added to the que
packet flow 6
router 10.0.0.4 got 10.0.0.4 in flow-table
message->8 removed from the que
message->8 added to the que
packet flow 7
router 10.0.0.5 got 10.0.0.1 in flow-table
message->8 removed from the que
message->8 removed from the que
hacker received packet for 10.0.0.1 processed the message from 10.0.0.9
message->9 added to the quemessage->9 added to the que
```

Figure 6: Log for for Sniffing attack (Without SDN)

5.2 Experiment 2 - Sniffing attack (With SDN)

Now the same experiment will be performed using SDN, where It hacks the router R1, hacker node try to represent itself as server S2. As SDN can centrally manage the flow entry table of network. SDN will update the flow entry of every router and completely isolate the R1 from the network. This step, can prevent the sniffing of upcoming packets.

Also SDN provides the optimal solution for the flow of packets in the network. The flow of packet per unit time for Software defined network is shown in Figure 7

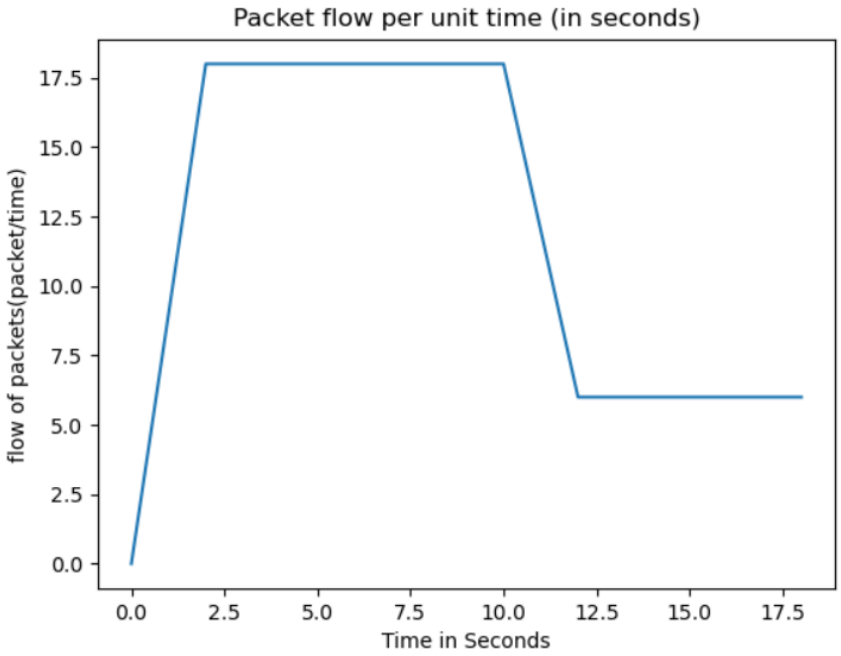

Figure 7: Packet flow per Unit time for Sniffing attack (With SDN)

In this experiment, hacker node is able to capture the packets of S2 by representing the itself as S2 to R1. But as the SDN has its own flow table entry, no other packet can be sent to R1 unnecessarily, this information can be clearly seen in the log file of SDN network. where no hacker node is found. The graph shown in Figure 5 also represents that number of packet per unit time is very less as compared to traditional network because it prevents the flow of unnecessary packets forwarding.

5.3 Experiment 3 - DDoS attack (Without SDN)

In this experiment we will try to send unnecessary packets to network at high rate. The routers will forward the unnecessary packets to other routers in the network and this will create unnecessary traffic in the network and will interrupt the actual service needs to be performed. The flow of packet per unit time is shown with the help of graph in Figure 8

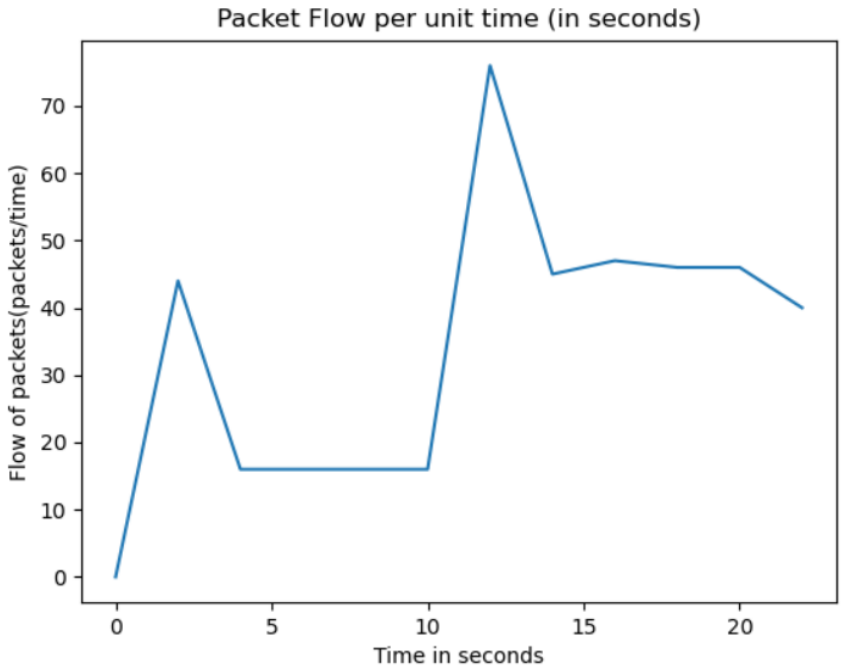

Figure 8: Packet flow per Unit time for DDoS attack (Without SDN)

From the graph shown in Figure 8 it is clear that. After the DDoS attack there is increase in the number of packets. All the network activities can be stored in the form of log, where we can check many junk packets are sent through different IP. A screenshot of DDoS attack log can be shown in Figure 9

```
packet flow 11
router 10.0.0.4 got 10.0.0.30 in flow-table
message->346 added to the que
DDOS junk message to 10.0.0.7
message->347 added to the que
DDOS junk message to 10.0.0.7
message->348 added to the quemessage->79 removed from the que

DDOS junk message to 10.0.0.7message->79 added to the que

packet flow 12message->74 removed from the que
```

Figure 9: Screenshot of log for DDoS attack (Without SDN)

5.4 Experiment 4 - DDoS attack (With SDN)

When the router will be not able to find the address of packet, it will transmit to SDN controller. SDN controller will check for unusual behaviour in the system and then check its hashmap. If the address is not available in its hashmap it will drop the packet. In this way, our proposed method prevents the network from DoS attack. The flow of packet per unit time is shown will be shown for SDN in Figure 10

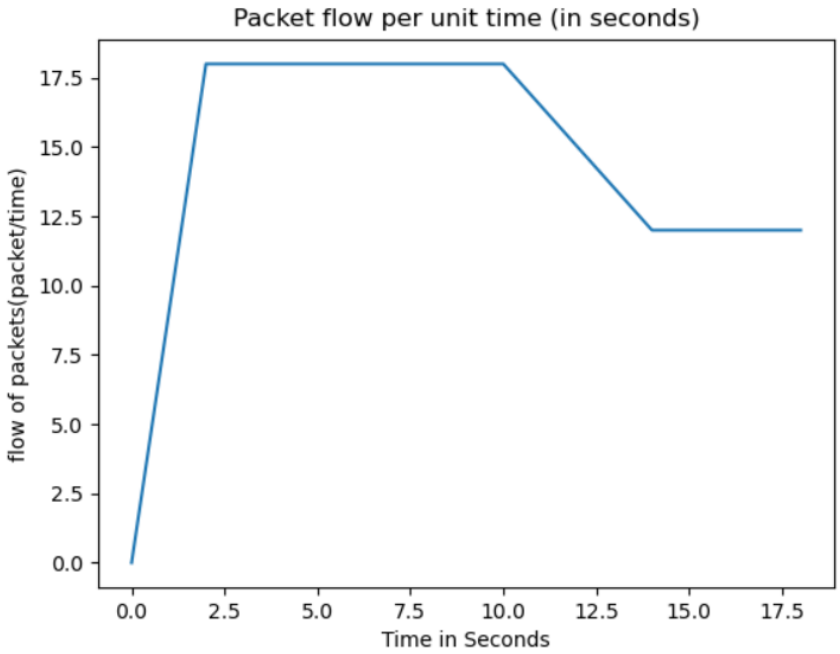

Figure 10: Packet flow per Unit time for DDoS attack (With SDN)

The graph in Figure 10 shows that even after the DOS attack there is not much change in the flow of network packets. Unknown packets are directly discarded by SDN. Even though DDoS attack can be applied to network but it does not effect the network performance with the help of SDN. All the network activities can be stored in the form of log, where we can check many junk packets are sent through different IP. We can also check that SDN is dropping the fake packets. A screenshot of DDoS attack log can be shown in Figure 11

```
DDOS junk message to 10.0.0.7message 96 dropped by 10.0.0.7

message->428 added to the que
DDOS junk message to 10.0.0.7
message->429 added to the que
DDOS junk message to 10.0.0.7
message->430 added to the que
DDOS junk message to 10.0.0.7
message->97 removed from the quemessage->431 added to the que

message 97 dropped by 10.0.0.7DDOS junk message to 10.0.0.7
```

Figure 11: Screenshot of Log for DDoS attack (With SDN)

5.5 Evaluation

In the result section we have performed experiment with different scenario ie: with SDN and without SDN. Having comparative analysis of the graph under each method we have found that sniffing attack and DDoS attack are difficult to prevent in the traditional network it is also evident that packet flow per unit time in traditional network is very high as it travels through the various routers. In any case, if router is not available or physically damaged the packets will be dropped. Damage control can be easily managed by the

SDN which prevents from various network attacks as well as prevents the packet drop. Shin et al. (2016) shared the opportunities of SDN to network security, which has been implemented practically using our proposed approach. According to Shin et al. (2016) Dynamic flow control, network-wide visibility, network programmablity and simplified data plane are the important features which keeps the network secured with the help of SDN. The statistical analysis for both the methods can be shown with the help of following graphs shown in Figure 12 and Figure 13. The following graph clearly indicates that Proposed architecture with SDN completely avoids the unecessary flow of packets overflooding from 70 to a controled state of 18 under DDoS attack in Fig-12 and Sniffing attack where packet is reduced to 18 from 45 during attack phase Fig-13. Thus it prevents the network from DDoS and Sniffing attacks.

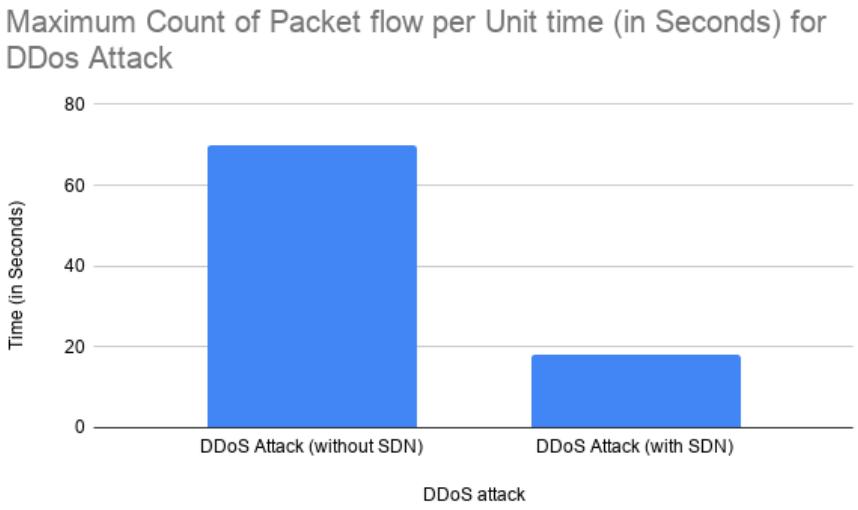

Figure 12: Statistical analysis for DDoS attack

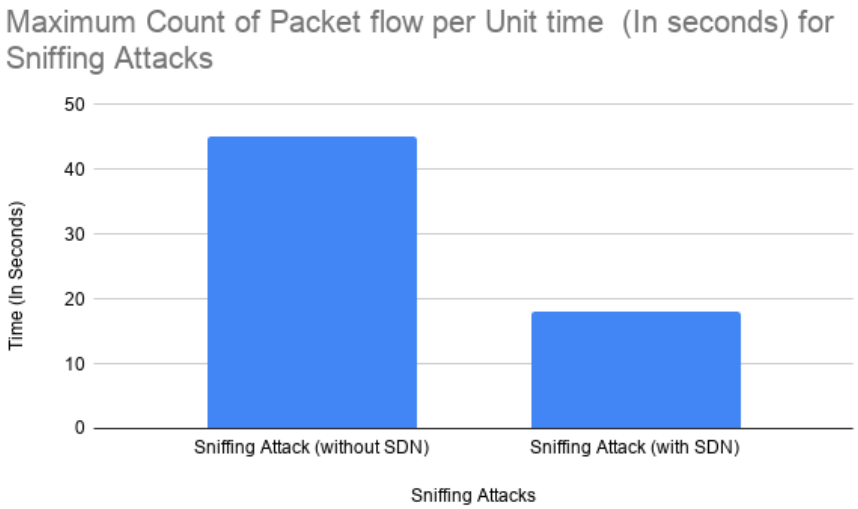

Figure 13: Statistical analysis for Sniffing attack

6 Conclusion and Future Work

Various experiments have been performed in order to prevent the network from the attacks. A comparative analysis between SDN and without SDN (Tradition networks) has been carried out to demonstrate effectivness of SDN network over traditional network. Where we have have found that the traditional network sometimes unecessarily forwards the packets from one router to another router. Anyhow, the information is transmitted but it can be delayed. Traditional network also contains many vulnerabilities due to which it can be easily attacked. SDN enables to administrate the network centrally, through which we can prevent the network from various attacks such as redirectional attack, DDoS attack and SDN is also useful for Damage control. In future work,we can use machine learning, deep learning technology in order to enable better decision making ability for SDN. Also the list of attack is endless, we will consider kinds of attacks and prevent the network system from intruders using SDN.

References

Abdolmaleki, N., Ahmadi, M., Malazi, H. T. and Milardo, S. (2017). Fuzzy topology discovery protocol for sdn-based wireless sensor networks, *Simulation Modelling Practice and Theory* **79**: 54–68.

Ahmad, A. A.-S. and Andras, P. (2018). Measuring the scalability of cloud-based software services, *2018 IEEE World Congress on Services (SERVICES)*, IEEE, pp. 5–6.

Akiyama, T., Teranishi, Y., Banno, R., Iida, K. and Kawai, Y. (2016). Scalable pub/sub system using openflow control, *Journal of Information Processing* **24**(4): 635–646.

Ali, S., Alvi, M. K., Faizullah, S., Khan, M. A., Alshanqiti, A. and Khan, I. (2019). Detecting ddos attack on sdn due to vulnerabilities in openflow.

Alshnta, A. M., Abdollah, M. F. and Al-Haiqi, A. (2018). SDN in the home: A survey of home network solutions using Software Defined Networking.

Andriopoulou, F., Birkos, K., Mantas, G. and Lymberopoulos, D. (2018). Software-defined networking for ubiquitous healthcare service delivery, *International Conference on Broadband Communications, Networks and Systems*, Springer, pp. 95–104.

Ashouri, M. and Setayesh, S. (2018). Enhancing the performance and stability of sdn architecture with a fat-tree based algorithm.

Benson, T., Akella, A. and Maltz, D. A. (2010). Network traffic characteristics of data centers in the wild, *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pp. 267–280.

Bhushan, K. and Gupta, B. B. (2018). Detecting ddos attack using software defined network (sdn) in cloud computing environment, *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 872–877.

Byun, M., Lee, Y. and Choi, J.-Y. (2019). Risk and avoidance strategy for blocking mechanism of sdn-based security service, *2019 21st International Conference on Advanced Communication Technology (ICACT)*, IEEE, pp. 187–190.

Canovas, A., Rego, A., Romero, O. and Lloret, J. (2020). A robust multimedia traffic sdn-based management system using patterns and models of qoe estimation with brnn, *Journal of Network and Computer Applications* **150**: 102498.

Casado, M., Garfinkel, T., Akella, A., Freedman, M. J., Boneh, D., McKeown, N. and Shenker, S. (2006). Sane: A protection architecture for enterprise networks., *USENIX Security Symposium*, Vol. 49, p. 50.

Chen, T.-L., Chung, Y.-F. and Lin, F. Y. (2010). An efficient date-constraint hierarchical key management scheme for mobile agents, *Expert Systems with Applications* **37**(12): 7721–7728.

Chiesa, M., Dietzel, C., Antichi, G., Bruyere, M., Castro, I., Gusat, M., King, T., Moore, A. W., Nguyen, T. D., Owezarski, P. et al. (2016). Inter-domain networking innovation on steroids: empowering ixps with sdn capabilities, *IEEE Communications Magazine* **54**(10): 102–108.

CN, S. et al. (2019). A proactive flow admission and re-routing scheme for load balancing and mitigation of congestion propagation in sdn data plane, *International Journal of Computer Networks & Communications (IJCNC) Vol* **10**.

da Silva, M. P., Gonçalves, A. L. and Dantas, M. A. R. (2019). A conceptual model for quality of experience management to provide context-aware ehealth services, *Future Generation Computer Systems* **101**: 1041–1061.

Dayal, N., Maity, P., Srivastava, S. and Khondoker, R. (2016). Research trends in security and ddos in sdn, *Security and Communication Networks* **9**(18): 6386–6411.

El Kamel, A., Majdoub, M. and Youssef, H. (2017). A fast bit-level mpls-based source routing scheme in software defined networks: Sd-{W, L} an, *International Conference on Mobile, Secure, and Programmable Networking*, Springer, pp. 109–121.

Fan, Z., Yao, J., Yang, X., Wang, Z. and Wan, X. (2019). A multi-controller placement strategy based on delay and reliability optimization in sdn, *2019 28th Wireless and Optical Communications Conference (WOCC)*, IEEE, pp. 1–5.

Farshin, A. and Sharifian, S. (2017). A chaotic grey wolf controller allocator for software defined mobile network (sdmn) for 5th generation of cloud-based cellular systems (5g), *Computer Communications* **108**: 94–109.

Haque, I. T. and Abu-Ghazaleh, N. (2016). Wireless software defined networking: A survey and taxonomy, *IEEE Communications Surveys & Tutorials* **18**(4): 2713–2737.

Hong, K., Kim, Y., Choi, H. and Park, J. (2017). Sdn-assisted slow http ddos attack defense method, *IEEE Communications Letters* **22**(4): 688–691.

Hu, L., Qiu, M., Song, J., Hossain, M. S. and Ghoneim, A. (2015). Software defined healthcare networks, *IEEE Wireless Communications* .

Huang, K., Yang, L.-X., Yang, X., Xiang, Y. and Tang, Y. Y. (2020). A low-cost distributed denial-of-service attack architecture, *IEEE Access* **8**: 42111–42119.

Ivey, J., Yang, H., Zhang, C. and Riley, G. (2016). Comparing a scalable sdn simulation framework built on ns-3 and dce with existing sdn simulators and emulators, *Proceedings of the 2016 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, SIGSIM-PADS '16, Association for Computing Machinery, New York, NY, USA, p. 153–164.
URL: *https://doi.org/10.1145/2901378.2901391*

Karakus, M. and Durresi, A. (2017). A survey: Control plane scalability issues and approaches in software-defined networking (sdn), *Computer Networks* **112**: 279–293.

Khan, S., Gani, A., Wahab, A. W. A., Guizani, M. and Khan, M. K. (2016). Topology discovery in software defined networks: Threats, taxonomy, and state-of-the-art, *IEEE Communications Surveys & Tutorials* **19**(1): 303–324.

Li, X., Vrzic, S. and Rao, J. (2019). Systems and methods for sdt to interwork with nfv and sdn. US Patent 10,298,466.

Liang, X. and Znati, T. (2019). An empirical study of intelligent approaches to ddos detection in large scale networks, *2019 International Conference on Computing, Networking and Communications (ICNC)*, IEEE, pp. 821–827.

Ma, Y.-W., Chen, Y.-C. and Chen, J.-L. (2017). Sdn-enabled network virtualization for industry 4.0 based on iots and cloud computing, *2017 19Th international conference on advanced communication technology (ICACT)*, IEEE, pp. 199–202.

Oktian, Y. E., Lee, S., Lee, H. and Lam, J. (2017). Distributed sdn controller system: A survey on design choice, *computer networks* **121**: 100–111.

Rahouti, M. (2019). Board 128: Understanding global environment for network innovations (geni) and software-defined networking (sdn) for computer networking and security education, *2019 ASEE Annual Conference & Exposition*.

Rawat, D. B. and Reddy, S. R. (2016). Software defined networking architecture, security and energy efficiency: A survey, *IEEE Communications Surveys & Tutorials* **19**(1): 325–346.

Rose, N. and Brown, K. (2018). Systems and methods for use in indexing applications based on security standards. US Patent 9,860,250.

Sallabi, F., Naeem, F., Awad, M. and Shuaib, K. (2018). Managing iot-based smart healthcare systems traffic with software defined networks, *2018 International Symposium on Networks, Computers and Communications (ISNCC)*, IEEE, pp. 1–6.

Sezer, S., Scott-Hayward, S., Chouhan, P. K., Fraser, B., Lake, D., Finnegan, J., Viljoen, N., Miller, M. and Rao, N. (2013). Are we ready for sdn? implementation challenges for software-defined networks, *IEEE Communications Magazine* **51**(7): 36–43.

Shi, H., Zhang, W. and Zhang, X. (2019). Model checking the openflow protocol using spin, *IOP Conference Series: Earth and Environmental Science*, Vol. 234, IOP Publishing, p. 012071.

Shin, S., Xu, L., Hong, S. and Gu, G. (2016). Enhancing network security through software defined networking (sdn), *2016 25th international conference on computer communication and networks (ICCCN)*, IEEE, pp. 1–9.

Sinh, D., Le, L., Tung, L. and Lin, B. (2017). The challenges of applying sdn/nfv for 5g & iot, *14th IEEE-VTS Asia Pacific Wirel. Commun. Symp.(APWCS), Incheon, Korea*.

Song, S., Park, H., Choi, B. Y., Choi, T. and Zhu, H. (2017). Control Path Management Framework for Enhancing Software-Defined Network (SDN) Reliability, *IEEE Transactions on Network and Service Management* **14**(2): 302–316.

Trakadas, P., Karkazis, P., Leligou, H.-C., Zahariadis, T., Tavernier, W., Soenen, T., Van Rossem, S. and Miguel Contreras Murillo, L. (2018). Scalable monitoring for multiple virtualized infrastructures for 5g services, *SoftNetworking 2018, The International Symposium on Advances in Software Defined Networking and Network Functions Virtualization*, pp. 1–4.

Tselios, C., Politis, I. and Kotsopoulos, S. (2017). Enhancing sdn security for iot-related deployments through blockchain, *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, IEEE, pp. 303–308.

Uddin, M., Mukherjee, S., Chang, H. and Lakshman, T. (2018). Sdn-based multi-protocol edge switching for iot service automation, *IEEE Journal on Selected Areas in Communications* **36**(12): 2775–2786.

Vinarskii, E., López, J., Kushik, N., Yevtushenko, N. and Zeghlache, D. (2019). A model checking based approach for detecting sdn races, *IFIP International Conference on Testing Software and Systems*, Springer, pp. 194–211.

Vizarreta, P., Trivedi, K., Helvik, B., Heegaard, P., Blenk, A., Kellerer, W. and Machuca, C. M. (2018). Assessing the maturity of sdn controllers with software reliability growth models, *IEEE Transactions on Network and Service Management* **15**(3): 1090–1104.

Wang, S., Liu, B. and Feng, Y. (2018). Design of multi-service network slicing scheme based on sdn/nfv, *2018 International Conference on Sensor Networks and Signal Processing (SNSP)*, IEEE, pp. 344–351.

Wetterwald, M., Saucez, D., Nguyen, X.-n. and Turletti, T. (2016). SDN for Public Safety Networks To cite this version : HAL Id : hal-01400746 SDN for Public Safety Networks.

Xu, Q., Su, Z., Dai, M. and Yu, S. (2019). Apis: Privacy-preserving incentive for sensing task allocation in cloud and edge-cooperation mobile internet of things with sdn, *IEEE Internet of Things Journal* .

Zhao, Z., Gong, D., Lu, B., Liu, F. and Zhang, C. (2016). Sdn-based double hopping communication against sniffer attack, *Mathematical Problems in Engineering* **2016**.

Zlomislić, V., Fertalj, K. and Sruk, V. (2017). Denial of service attacks, defences and research challenges, *Cluster Computing* **20**(1): 661–671.