International Journal of Computer Sciences and Engineering
Vol.**12**, Issue.**2**, pp.**46-52**, February **2024**
ISSN: 2347-2693 (Online)
Available online at: www.ijcseonline.org

ISSN: 2347-2693 (E)

Survey Article

A Survey on Software defined Networking and different types of controllers

R. Karhikeyani[1*], E. Karthikeyan[2]

[1,2]Dept. of Computer Science, Government Arts College, Udumalpet, Tamilnadu, India

**Corresponding Author: keyanirangu@gmail.com*

Received: 27/Dec/2023; **Accepted**: 30/Jan/2024; **Published**: 29/Feb/2024. **DOI**: https://doi.org/10.26438/ijcse/v12i2.4652

Abstract: Distributed Denial of Service (DDOS) attacks represent a tool wielded by hackers, cyber extortionist, and individuals involved in cyber terrorism. These malicious activities result in significant financial losses for the targeted victims. Although a myriad of scientific solutions are available, the frequency and intensity of DDOS attacks continue to escalate. In response to the evolving landscape of security threads, a novel network paradigm has emerged. Software Defined Networking (SDN) has garnered considerable attention from numerous researchers it addresses the specific requirement of modern data centres, drawing inspiration from its capabilities. Our study presents a comprehensive examination of current SDN-based solutions for detection and mitigating DDOS attacks. Building upon our findings, we propose an innovative framework designed for the detection and mitigation of DDOS attacks on a large scale network. The primary contribution of this paper is twofold. Firstly, we furnish an in-depth survey of SDN based DDOS attack detection and mitigation mechanisms. Secondly, we offer a detailed comparison of SDN- based controllers.

Keyword: DDOS, SDN, cyber terrorism, financial losses, Controllers.

1. Introduction

Over the preceding decade, the remarkable progress in information and communication technologies has brought fourth numerous challenges for the traditional internet. The rapid evaluation of wireless technology, for instance, has compelled researchers to delve into novel approaches for mobility management. Simultaneously, the emergence of extensive-scale data centres has given rise to critical concerns regarding energy conservation. Notably, Software-Defined Networking has recently captured the attention of both the academic and industrial realms as a potential solution to these challenges. At the core of SDN lies the concept of dismantling vertical integration, separating the control plane from the data plane, and introducing programmability into the network. Within the framework of SDN, controllers assume the role of the Network operating system [13].

SDN has brought about a significant transformation in network operations, enhancing efficiency through the centralization of control facilitated by the controller. With networks expanding rapidly in both size and complexity, managing hardware switches has become increasingly challenging. SDN represent a paradigm shift in networking, transitioning from a distributed model to a centralized approach. While offering a streamlined method for network control, SDN also introduces new security challenges, such as the susceptibility to DDOS attacks.

A key feature of SDN is the separation of the data plane, encompassing forwarding elements like switches and routers, from the control plane, which includes the controller. In an SDN environment, network engineers or administrators can shape traffic using a centralized control console, eliminating the need to provide network services wherever required, irrespective of specific connections between server devices. The control plane comprises the controller and its three interfaces:

1. **South-bound interface:** Facilitating communication between switches and the controller.
2. **North-bound interface:** Serving as the interface between the controller and application Layer programs.
3. **East-west bound interface:** Providing an interface between controllers [9].

SDN has several advantages that are centralised management, monitoring, Centralized service and security. Various types of controllers available in the market.SDN do not have a security module. It has a main concern of SDN. SDN architecture overcomes some traditional attacks it also give some attack strategy. Figure-1 depicts basic architecture of SDN.

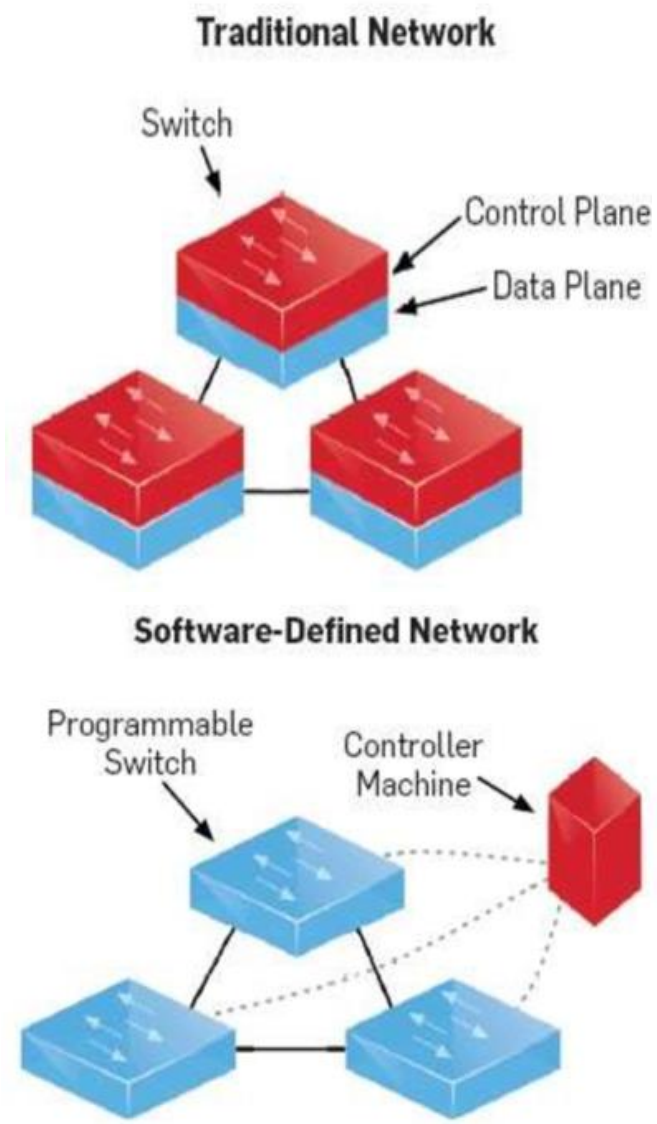

Figure-1 SDN vs. Traditional network architecture

2. Literature Review

Narmeen Zakaria Bawany et. al. [16] proposed paper clearly explains DDOS attacks and its production mechanisms. SDN based DDOS attack detection techniques Machine learning and Entropy based techniques are also clearly explain in this paper and along with traffic pattern analysis and lots of machine learning technique and DDOS attack prevention methods explain in this paper.

Ms.Florance G et. al. [17] proposed paper proposed detailed about the control plane and data plane in SDN. And it explains three types of interfaces in control plane. And also this paper depicts main security issues SDN using mininet and mininet charctristics and its types of layers also proposed in this paper.

Mohammad Alhisnawi et. al. [9] proposed paper explain SDN and type of SDN and general view of controllers. This is clearly explained in Centralised controller and distributed controllers in SDN environment. North bound interface and South bound inter face and east west bound interfaces also explained clearly.

Haeeder Munther Noman et. al[10] proposed paper explain SDN explanation and DDOS attack. This is proposed POX controller detail performance and traffic generation, simulation model and network topology and creating topology and traffic analysis also done using mininet pox controller.

M. Paliwal et. al[19] proposed paper explain distributed controllers and centralised controllers and some examples of centralised and distributed controllers also explain detail in this Paper. Performance also evaluated detail in this paper.

Section I give explanation of SDN and importance of SDN and interfaces of SDN.

Section II gives some literature reviews. Section III explains SDN architecture. Section IV explains advantages of SDN. Section V depicts Different types of controllers. Section VI depicts SDN controller architectures. Section VII gives Comparison of controllers. Section VIII gives conclusion. Section IX gives References.

3. SDN Architecture

SDN architecture is organised into three main layers: the application layer, the control layer, and the infrastructure layer. Each layer is assigned a specific role and engages in interactions with the other layers to effectively oversee and control the network. [9]

1. **Infrastructure Layer:** The infrastructure layer is the bottom most tier of the SDN architecture, also known as the data plane. It combination of physical and virtual network devices including routers, switches and firewalls that are responsible for forwarding network traffic. The instructions guiding this traffic forwarding process are communicated from the control plane, establishing a crucial link between physical and virtual components in the SDN environment.
2. **Control Layer:** The control layer is the middle tier of SDN. It is also known as the control plane. This layer incorporates a centralized controller that communication with devices in the infrastructure layer, device that is responsible for managing and configuring the network. The controller interacts with the devices in the infrastructure layer Utilising protocol like OpenFlow, the controller interacts with devices in the infrastructure layer to program the forwarding behaviour of switches and routers. Decisions concerning the forwarding of traffic are made by the controller, drawing on network policies and rules, and taking into account factors such as network topology, traffic patterns and quality of service requirements.
3. **Application Layer:** The application layer is the top most tier of the SDN and is responsible for providing network services and applications to users. This layer encompasses diverse network applications that engage in

integrations with the control layer to effectively manage the network. Figure-2 explains control layer and infrastructure layer and Application layer architectures.

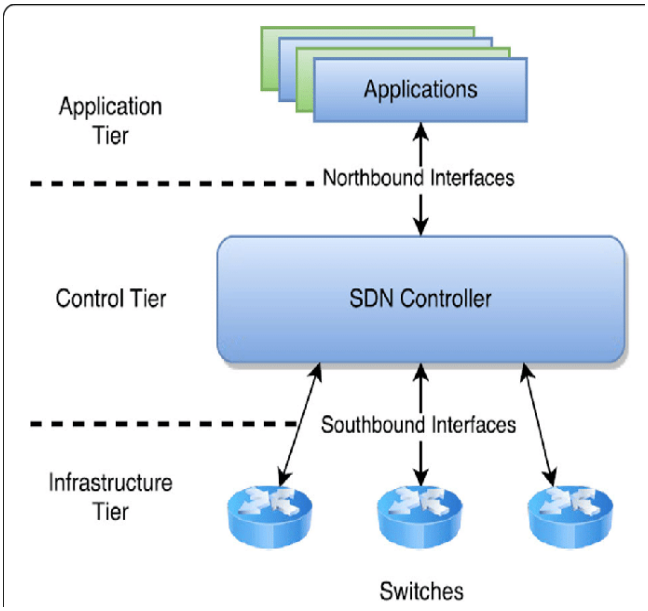

Figure-2 SDN architecture

4. Advantages of SDN

SDN offers several advantages over traditional networking architectures, including:

Centralized Network Control: An essential advantage of SDN is its centralization of network control in a single controller, simplifying the management and configuration of the network. SDN provides network administrators with the capability to define and enforce network policies in a more detailed manner, leading to improved network security, performance, and reliability.

Programmable Network: Within a SDN environment, network devices are programmable and can be dynamically reconfigured to accommodate evolving network administrators to swiftly adapt the network to shifting traffic patterns and demands, ultimately enhancing overall network performance.

Cost Savings: SDN enables network administrators to employ commodity hardware for network construction, thereby lowering the expenses associated with propriety network hardware. Furthermore, the centralization of network control diminishes the necessity for manual network management, resulting in cost savings related to labour and maintenance.

Enhanced Network Security: The centralized control inherits in SDN facilitates more effective detection and response to security threats. The implementation of network policies and rules empowers administrators to establish precise security controls, there by mitigating potential security risks.

Scalability: SDN streamlines the process of scaling the network to adapt to fluctuating traffic demands. Through programmable network control, administrator can swiftly make adjustments to accommodate increased traffic without requiring manual intervention.

Simplified Network Management: SDN contributes to simplified network management by abstracting the underlying network hardware and presenting administrators with a logical view of the network. This abstraction makes network management and troubleshooting more straightforward, ultimately leading to enhanced network uptime and reliability.

5. SDN Controllers

Controller is main application of SDN. It manages flow control for network management and application. Here SDN controllers are typically operated as servers and utilize protocols to instruct switches on the appropriate destinations for packet forwarding. Two well known protocols used in SDN controllers that are open flow and open virtual switch database.

a. Basic functions of controllers

1. Planning, directing and coordinating all operation functions.
2. SDN controllers are mainly used to managing the all financial data necessary for accurate consolidated business result.
3. Creating and preparing internal and external financial list.
4. Running cross platform is allowing multithreading being easy to learn and fast memory access and good memory management are essential programming languages.
5. When choosing particular controller we have to take some factors for considerations. Because that are affect the controller's performance and speed. Python, C++, Java are the most used Programming languages for SDN controller programming generally Java coded controllers have characteristic to run cross platform and presently it is good modularity C coded controllers provide high performance but it is lack for high modularity and good memory management and good GUI and then Python coded lack real multithreading Harding.

6. SDN controller architecture

There are two approaches in controllers that are
a. Centralized controller and
b.Distributed controller. In Figure-3 explain detail in Centralised on controller and Distributed controllers and its types [12].

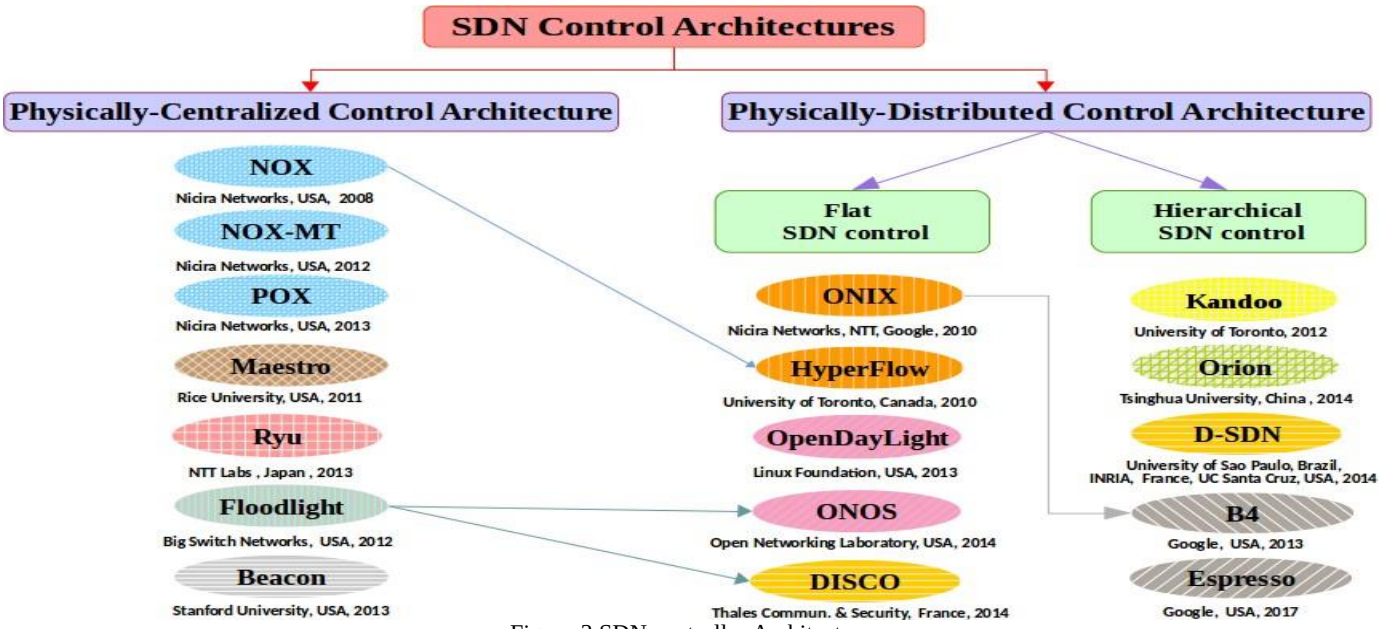

Figure-3 SDN controller Architecture

a. Centralized controller

Centralized control plane is a single controller for entire network. Centralized controllers developed better performance and user experience. But this type of controllers cross point start using controller itself preciously where deploy the controller either enterprise.

Network or data centre network. Since user traffic increase the demand of data centre as well as security, scalability, performance and availability. In Figure-4 explain detail in centralized controller and how it works [12].

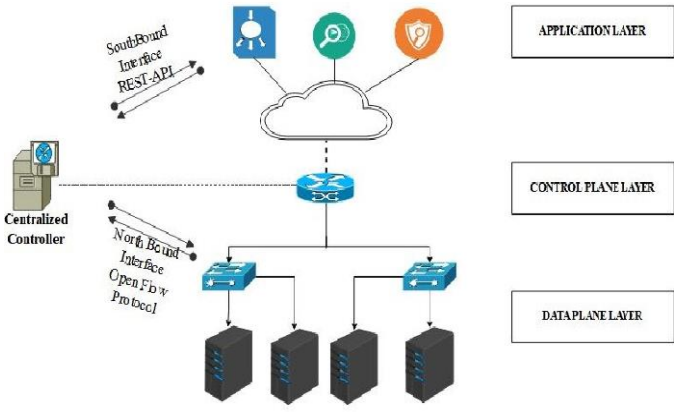

Figure-4: Centralized Controller

Examples of Centralized Controllers

1. NOX

NOX is the first OpenFlow controller written in C++. However, it also provides Python API. This was the idea for a lot of research and improvement work in the early stages of exploring the OpenFlow and SDN space. It serves network management platform that provides an advanced programming interface for managing and developing network management applications.

NOX was originally developed by **Nicira Networks** and now owned by VMware, and was introduced to the community along with OpenFlow in 2009. This was later divided into several development areas.

a. **NOX classic**: This version has been available since 2009.
b. **NOX**: The New NOX supports only C++ and has fewer applications than the old NOX. This is faster and better than codebase.

At the top of we have applications Core, Net, and web. However, the current NOX version only two main applications those are OpenFlow and switch. Middle layer shows in build components of NOX. Figure-5 depicts the architecture of NOX.

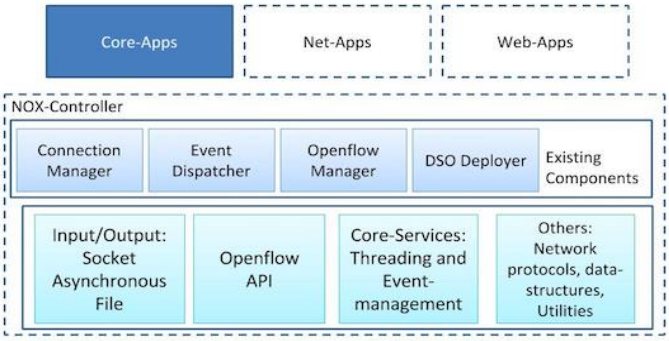

Figure-5: Architecture of NOX

2. POX

POX is called brother of NOX's. POX is a Python based open source Software Defined Controller. POX is used to accelerate the development and design of web applications. POX controllers can be installed on mininet virtual machines. POX controller provides basics of communicating with SDN switches using Open Flow or OVSDB(Open vSwitch) protocols. This is the actual communication protocol between

the controller and the switches. Pox controller simplifies to open flow and SDN experiment.

The general purpose of all SDN controllers (such as pox), is to allow users to write their own applications using the controller as a middleware or application layer, Network applications and the network equipment. POX components are basically python modules. The best way to use with pox is use to a git repository.

Using POX controller you can run different applications such as hub, switch load balancer and firewall. POX allows users to write their own applications and use the controllersto act as intermediary or abstraction layer between network applications and the network devices. It is distributed with some standard applications such as simple layer 2 and layer 3 transmissions. pox is official GUI like NOX.

POX lacks support for distributed controllers and multithreading.Figure-6 depicts detail in Pox Controller and its working flow.

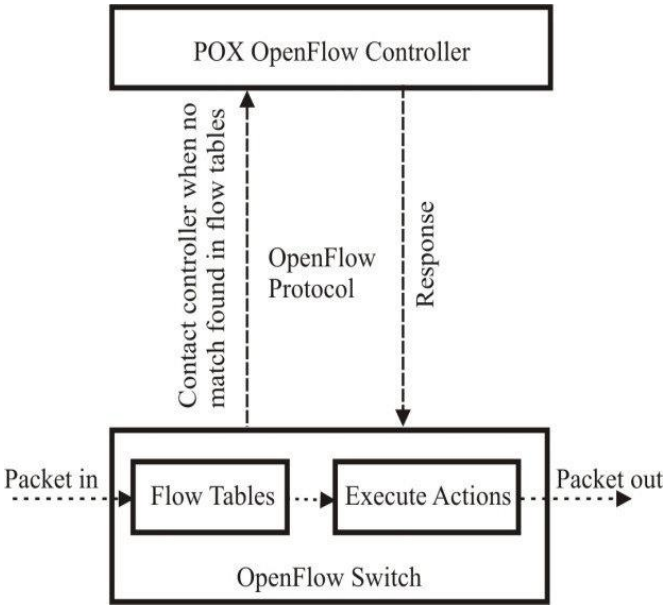

Figure-6: Pox controller

3. RYU

RYU was developed by Nippon Telegraph and Telephone Corporation (NTT) company and follows set of standards to a facilitate modification and expansion to meet new applications. It is one of the SDN controllers, specially designed to managing more traffic rate. RYU includes well defined software frameworks and well-defined APIs that allows developers to easily create.

New network management and control applications.
RYU made from Python language. The SDN controller has three layers.. The middle layer consists of network services that are called as Control layer or SDN frame work. Then the bottom of the layer has physical and virtual devices that are called as infrastructure layer. In Figure-6 explain detailed in ryu controller and flow of ryu controller.

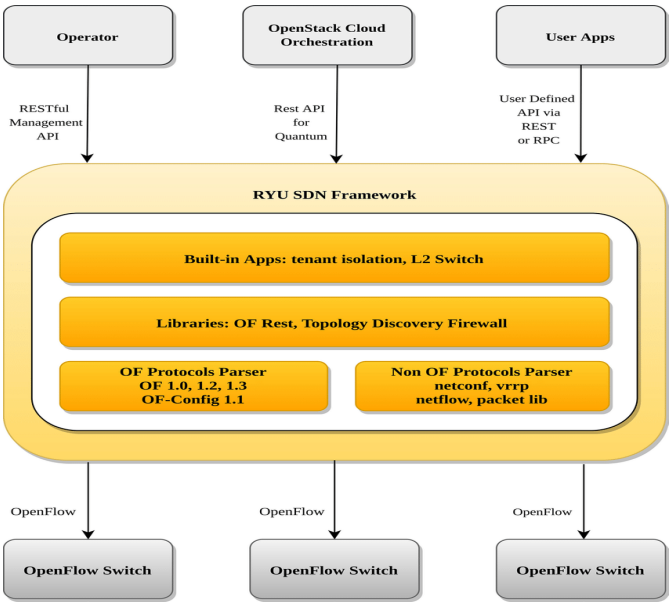

Figure-7: RYU controller

b. Distributed controllers

Distributed control plane means control plane of every network devices reside in the device itself. Each device has its own control plane to data plan. Local controller runs on every compute node and manages and forwarding element directly and locally. Distributed control not has a single point of failure because distributed version control work station creating multiple backup. If source code is corrupted can use any clone as a backup, increasing security since their small risk of damage entire perfect history. In Figure-8 explain distributed controller and its working methods in detail [9].

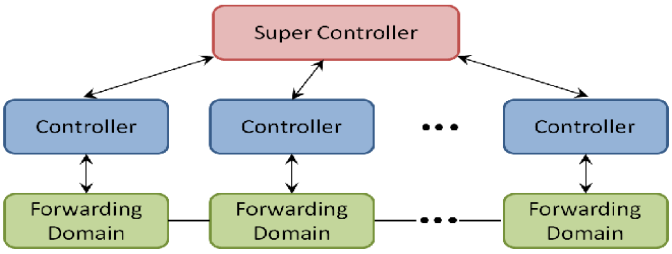

Figure-8: Distributed controllers

Examples of Distributed Controllers

1. OpenDaylight

OpenDaylight controller is Java Virtual Machine software based controller. It can be run from any operating system. Its hardware as long as it supports java. Controller is an implementation of the SDN. It makes use of Maven tool. Maven means popular open source build tool. It is developed by the Apache group to build, publish, and deploy several projects at once for better project management software's. The tool provides developers to build, reporting and documentation from a central piece of information. The OpenDaylight software is built Java programming. The typical OpenDaylight solution consists of five components:

1. Open Daylight APIs
2. Authorization and Accounting,
3. Model Driven Service
4. Abstraction Layer Services and Applications

5. Various south bound plugins.
In Figure-9 explain Open Day Light controller and switch and router working flow.

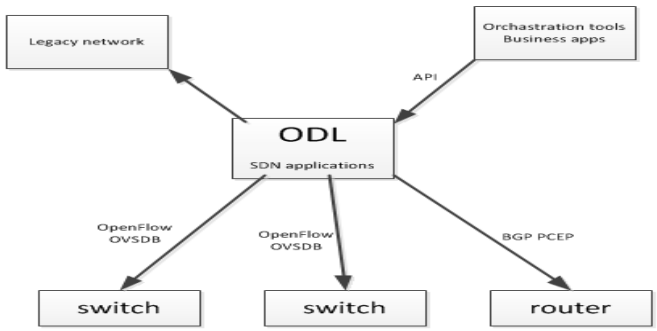

Figure-9: Open-day-light

2. ONOS (Open Network Operating system)
ONOS is the one of the most leading open source SDN controller. ONOS provides a control plane for software-defined network components, such as switches and links, and running software programs or modules to provide communication services to end hosts and neighbouring networks. ONOS was developed by ON.lab in collaboration with a community of leading service providers, vendors, and researchers. Provide visibility into multi-layer network nodes, links, network status, errors, etc. ONOS can run distributed systems on

Multiple servers, sharing CPU and memory resources of multiple servers while providing resilience in the event of server failure. Errors may occur and potentially supporting live upgrades of hardware and software without interrupting of network traffic. ONOS applications and uses often consist of custom communication routing management or monitoring services for SDN.
Figure-10 explain ONOS controller working flow and its packet forwarding mechanism

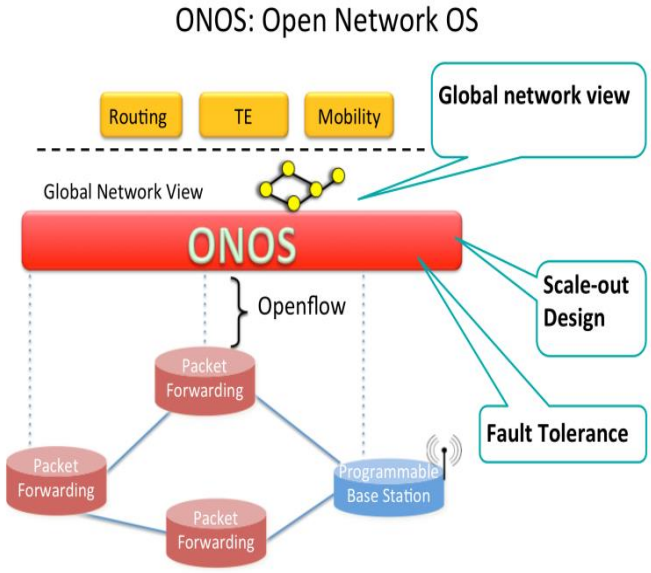

Figure-10 ONOS controller

7. Comparison of controllers

The controllers have been compared based on availability, interfaces, GUI, programming languages and complexities, OpenFlow, platform support, distributed controller, backup controller, first release.
The result of the comparison shown in Bellow table.

Table-1: Comparison among the most popular open source controllers

	NOX	POX	RYU	ONOS	OPENDAYLIGHT
Programming language	C++	Python	Python	Java	Java
Complexity	Moderate	Simple	High	High	High
GUI	Python +QT4	Python +QT4	Yes	Web based	Web based
OpenFlow support	OF v1.0	OF v1.0	OF v1.0,v2.0, v3.0	Of v1.0	OF v1.0
Platform support	Most supported on linux	Linux, MAC OS, and windows	Most supported on linux	Linux, MAC OS, and windows	Linux, MAC OS,and windows
Open source	yes	yes	yes	yes	yes
Distributed controllers	No	No	No	Yes	Yes
Backup Controllers	No	No	No	Yes	Yes
First release	2008	2011	2012	2014	2013
OpenStack neutron support	No	No	Yes	Yes	Yes

8. Conclusion

The rapid growth of internet services and applications increases the number stream present. Many practical solutions developed in past provide secure network infrastructure. SDN is one of the most powerful solutions in to address the network-wide challenges of flexibility control and adaptability as the internet and services grow. SDN simplifies of network into three layers revolutionizes the field of by dividing network functions into three layers. However this paradigm is also vulnerable to DDOS an attack making deployment very difficult. This article provides comprehensive overview of layered SDN architectures and SDN controllers and as well as examples of controllers. The Controller is the most important of component of SDN network and is therefore a prime target for attackers. There is also a list of centralized controller and distributed controllers. Future work should capture all SDN controllers and use some techniques and algorithms to predict and mitigate DDOS attack in software define networking.

References

[1]. Bawany, N.Z., Shamsi. J.A & Salah. K, "DDoS Attack Detection and Mitigation Using SDN: Methods, Practices, and Solutions**".** Arab J Sci Eng 42, pp.**425–441,2017**.

[2]. YooJin Kwon, "A Survey on Defence Mechanism against Distributed Denial of Service (DDoS) Attacks in Control System", KEPCO Journal on Electric Power and Energy, Sept., Vol.**1**, No.**1**, **2015.**

[3]. Dutta Sai Eswari , P.V.Lakshmi, "A Survey On Detection Of Ddos Attacks Using Machine Learning Approaches", Turkish Journal of Computer and Mathematics Education Vol.**12**, No.**11**, pp.**4923-4931, 2021.**

[4]. J. Kaur, "Mac Layer Management Frame Denial of Service Attacks,'' 2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE), Ghaziabad, India, pp.**155-160, 2016.**

[5]. Mohammad Alhisnawi, Aladdin Abdulhassan, Mahmood Ahmadi, "A Comprehensive Survey on Software-Defined Network Controllers", Next Generation of Internet of Things, pp.**199-231.**

[6]. Mahdi Nsaif Jasim, "POX Controller and Open Flow Performance Evaluation in Software Defined Networks (SDN) Using Mininet Emulator", 3rd International Conference on Sustainable Engineering Techniques (ICSET 2020), IOP Conf Series: Materials Science and Engineering 881, 012102, **2020.**

[7]. F. Bannour, S. Souihi and A. Mellouk, "Distributed SDN Control: Survey, Taxonomy, and Challenges,'' in IEEE Communications Surveys & Tutorials, Vol.**20**, No.**1**, pp.**333-354, 2018.**

[8]. J. K. Chahal, P. Kaur and A. Sharma, "Distributed Denial of Service (DDoS) Attacks in Software-defined Networks (SDN)," 2021 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT), Mysuru, India, pp.**291-295, 2021.**

[9]. T. Huang, F. R. Yu, C. Zhang, J. Liu, J. Zhang and Y. Liu, "A Survey on Large-Scale Software Defined Networking (SDN) Testbeds: Approaches and Challenges,'' in IEEE Communications Surveys & Tutorials, Vol.**19**, No.2, pp.**891-917**, **2017.**

[10]. Hoang, D. B., & Pham, M. "On software-defined networking and the design of SDN controllers". 2015 6th International Conference on the Network of the Future (NOF), **2015.**

[11]. Bawany, N.Z., Shamsi, J.A. & Salah, K. "DDoS Attack Detection and Mitigation Using SDN: Methods, Practices, and Solutions", Arab J Sci Eng **42**, pp.**425–441, 2017.**

[12]. K. M. Sudar, M. Beulah, P. Deepalakshmi, P. Nagaraj and P. Chinnasamy, "Detection of Distributed Denial of Service Attacks in SDN using Machine learning techniques,"2021 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, pp.**1-5, 2021.** doi: 10.1109/ICCCI50826.2021.9402517..

[13]. B. Isong, R. R. S. Molose, A. M. Abu-Mahfouz and N. Dladlu, "Comprehensive Review of SDN Controller Placement Strategies," in IEEE Access, Vol.**8**, pp.**170070-170092, 2020.**

[14]. Mahadev, Vinod Kumar "Behaviour Analysis of DDoS Attack and Its Detection" International Journal of Computer Sciences and Engineering, May, Vol.**6**, Issue.**5**, **2018**. E-ISSN: 2347-2693

[15]. Aakriti Aggarwal, Ankur Gupta "Detection of DDoS Attack Using UCLA Dataset on Different Classifiers", International Journal of Computer Sciences and Engineering, Vol.3, Issue.8, E-ISSN: 2347-2693.

AUTHORS PROFILE

R. Karthikeyani Pursuing a Master's degree in Computer Science and Currently as a Research Scholar at Government Arts College Udumalpet indicates a Strong dedication to Academic and research pursuits. Passion for technology is evident, particularly in areas such as Advanced Networking and Python programming.

Dr.E.Karthikeyan working as a Head of the Department of Computer Science in Government Arts College, Udumalpet, Tamil Nadu. He completed Ph.D. from Gandhigram Institute of Technology, Dindigul in the year 2008. He is having 25 years of teaching and research experience. His area of research is Advanced Networking, Cyber Security. He completed research projects funded by UGC(Major Research Project), TANSCHE, TANSCST. He has published more than 60 papers in various national and international journals. He delivered lectures in many colleges and chaired in many conferences. He organized so many seminars/workshops/conferences and received funds from agencies like UGC, CSIR, TANSCHE, TANSCST. He also guided 10 Ph.D. students and guiding 4 students. He published a book entitled "A Text Book on C: Fundamentals, Data Structures and Problem Solving" by PHI. He is an Editor-in-Chief of International Journal of Advanced Networking and Applications (IJANA). He is a reviewer in IEEE, Springer, Elsevier, Wiley, Inderscience etc.

