PROGRAMMABLE NETWORK SERVICES IN NEXT GENERATION SOFTWARE DEFINED NETWORKS TO PREVENT SECURITY ATTACKS

P. Anusha [@1] , *M. Rakesh* [#2]
@1 Asst. Professor, Balaji Institute of Technology and Science, Narsampet, Warangal
anusha.pasupunooti@gmail.com
#2 Asst. Professor, Balaji Institute of Technology and Science, Narsampet, Warangal
motherakesh@gmail.com

ABSTRACT

The Software Defined Networks is a recent research area that has the initiative in the programmable network technologies and standards developed around 2010. These technologies are associated with networking software, using open interfaces to connect resources. The software defined networks are used to control the entities of the network by centralizing the control plane. The functioning of the network and its security has to be checked by the administrator manually, which may leads to great burden on the network administrator. In this paper we list the various security attacks on software defined network controllers that violate the network topology and also mounted by compromised network entities like end hosts and soft switches. The SDN's are having the ability to provide network virtualization, greater control over network entities and dynamic network policy at reduced operational cost. There are some protocols like Open Flow to do this task. The next generation of research should involve in integration of all connectivity, processing resources and storage under new management interacting with controlling devices for on demand networking and services along with continuous updates and features. This brings into focus relatively key topics such as how to create the conditions for effective and continuous updating and changing the network functions without reinventing each time architectural aspects and related when we deal about security attacks.

This paper presents architecture and the key challenges of programmable enabled networks as the next generation Software Defined Networks (SDN). This paper also looks the problem of detecting security attacks on topology of the network and traffic.

Keywords—Programmable networks, Next generation SDN

I. INTRODUCTION

Softwarization is a technical transformation which will affect the design, Implementation, deployment and operations of infrastructures, deeply integrating network nodes and IT systems. It fully exploits the nature of software like flexibility and rapidity, for both network functions and services. This transformation will enable new architectural models, in turn implementing automated operations process while dealing with innovative information and communication technology.

In the existing networks, it is very difficult to integrate the new functionality or services without affecting already available functions and it is a cost effective process. These functionality comes with separate hardware components. To address this issue, network function virtualization is a recent initiative. It is transforming the way how operators architect their networks towards deploying network services onto virtualized industry standard servers. NFV [1] can be deployed in the network as required and is used to deliver the network functions as software that can run as virtualized instances, without installing hardware equipment for each new service. NFV is very useful to reduce the hardware cost and operational cost. The more flexible services are to be expected to deploy which can be scaled up and down very quickly. From the architecture perspective, the NFV can be seen as complementary to technologies such as SDN, Cloud computing, Virtualized network functions might run in an operators cloud environment.

SDN maintains and redirect the traffic by considering the assistance given by controller and this can be achieved by configuring routers and switches flow tables. The main functionality of the remote controller in the networking infrastructure is adding, updating and deleting flow rules. There are several opportunities for understanding flow configurations of physical topology [2] and can be achieved at the centralized controller who can configure flow paths for optimal utilization of network resources and to enhance the experience of the user.

In SDN architecture, the control layer is logically centralized to a software based controller which maintains the global view of the network. The controllers allows the flexibility to configure, manage, secure and to optimize network resources through automated and dynamic software programs. OpenFlow provides freedom to the users in exploiting new opportunities over the existing networking infrastructure. OpenFlow at the current stage will not support mobility [3].

The cloud community started to address networking requirements in OpenStack [4], but continues to put the priority on

intra datacenter networking. Cloud users can control and manage their applications but have no control on the connectivity and networking of their dedicated and distributed cloud services. Networking providers, users need more flexibility in deploying, configuring and instantiating cloud networking services to manage more easily and efficiently their resources. To provide control over the connectivity, SDN can be used as a solution [5] to handle inter cloud networking. SDN is a network architecture and design that decouples data plane forwarding from control and management plane functions.

The integration of internet and programmable infrastructures and traditional communication technologies has been always a challenge for network and service operators for service deployment and management.

II. SECURITY ISSUES OF SDN

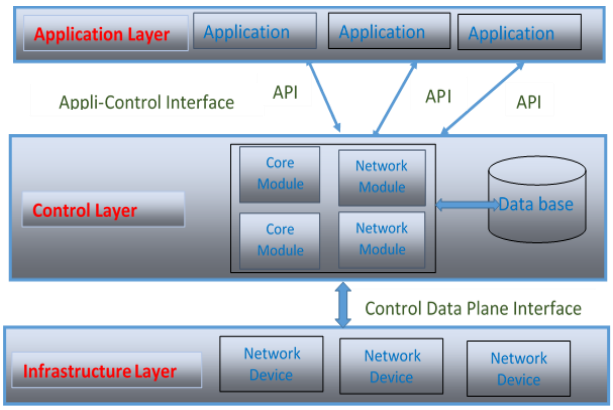
Integrity, confidentiality, availability, authentication are the minimum requirements for secure data transmission, so that the data transmitted across the network will be protected from malicious or unauthorized attacks. In general, SDN architecture control the network with the help of two individual components called a centralized controller and ethane switches. To enforce the global policy the first component is used and second one used to forward the packets depending on the rules available in flow table. For the programmability, the control plane and data plane to be separated by this simplified network. Ethane is a new architecture which is used especially in enterprise networks and it provides powerful and simple management method to provide strong security. Unlike the switches using today, the ethane has all the complex functionality including routing, naming and security checks are performed by centralized controller. The controller has to give permission for each flow in the network by verifying the flow whether it is permissible according to network policy. If the controller allows a flow, it computes a route for the flow to take and adds an entry in each of the switches along the path.

The SDN architectures which are using now are providing various services like Network function virtualization (NFV). It is possible to identify the various challenges related with application, control, data layer and their interfaces based on the security issues with SDN. The ethane architecture will be closely associated with SDN and OpenFLow, even then it is having several drawbacks [7]. The use of transport layer security when there is a mutual authentication between the switches and controllers are described by OpenFLow switch specification [8]. The authors are not specified the standard of TLS and they found that, due to less use of TLS leads to insertion of fake rule, and rule modification.

The IEEE software defined networking argues that one key area of this exercise is the introduction of new functional node as an intersection point of these functions in order to create a future proof architecture [11].

Most prior work has looked at development and analysis of SDN security applications and controllers and real time verification of network constraints separately. The solutions of these works are not effective against the threats in SDN due to compromised end hosts or switches, which can be used to control the entire network or part of it. The major problems identified in controlling the network are

- The operational semantics of OpenFlow based SDN lower the barrier for mounting sophisticated attacks on both control and data planes, since they allow any unmatched packets to be sent to the controller.
- Attacks that effect traditional networks may also affect SDN's because the traditional defenses assume switches to be intelligent, whereas separation of control and data planes forces SDN switches to be dumb forwarding entities that forward packets based on the rules installed by the SDN controller and which needs redesign.
- The enterprise network administrators often use programmable soft switches, like Open vSwitches [12] to provide network virtualization. These are like Hardware switches and must have direct connectivity to the controller to provide required functionality.



Functional Software Defined Network Architecture

Some authors try to present overall security of SDN with the help of high level analysis [9]. The results of this paper is due to the nature of the centralized controller and programmability of the SDN, there is a possibility for new threats which requires new responses from the network.

The research network and testbed ProtoGENI [10] has also analyzed by some authors and they discovered that the numerous attacks between users, flooding attacks to the internet were possible in the network.

Some authors tested most popular controllers like OpenDaylight [13], POX [14] and Floodlight [15] and found them vulnerable to diverse attacks originating within the software defined network. It is possible to implement defense against known attacks or specific vulnerabilities, such patching does not provide protection against unforeseen security threats. In this context we are proposing

self-managed network programmable services to detect security attacks on topology of the network and forwarding data.

III. ANALYSIS OF SECURITY ATTACKS

To find the required target in the network, the attackers use different techniques like virtual internet Protocol addressing. In this technique, the OpenFlow controller is used to manage a vector of IP addresses which are assigned to hosts with in the network and hiding the real IP addresses from the outside world.

The various layers and their interfaces of SDN are Application Layer, Application – Control interface, Control layer, Control-Data interface, Data layer. The layers and interfaces between the layers were affected based on the different security attacks.

Application Layer: This layer will be affected by unauthorized applications, insertion of fake rule and policy enforcements.

Application-Control Interface: This interface will be affected in case of unauthorized applications, insertion of fake rule and policy enforcements.

Control Layer: This layer will be affected by unauthorized controller access, unauthorized application, Flow rule modification to update packets, controller hijacking, insertion of fake rule, policy enforcements, controller switch communication flood and lack of TLS or other authentication technique adoption.

IV. PROGRAMMABLE NETWORK SERVICES

The architecture of SDN introduces programmable network services for the use in the network. These Intelligent programmable services should support the existing intrusion detection systems (IDS) and Prevention system (IPS). There is need of new methods and techniques to be introduced to enhance the programmability features in SDN by enabling the dynamic adjustments for managing, detection and prevention of attacks.

This architecture consists of network software formed by network applications, services which is different from traditional architecture. The Network applications and services can serve multiple controllers. It may consume one or more services to accomplish a certain task. The information available in individual controllers is also available in controllers of central network information base.

The network software module can used to customize the SDN controller based on the user needs. The different SDN controllers may have different features, depending on what modules and what services they choose. The programmable network service architecture provides various services and service chaining. The controller having the software program inside, will reduce the latency and over traffic.

Control –Data Interface: This layer will be affected in case of unauthorized controller access, Flow rule modification to update packets, controller hijacking, controller switch communication flood and lack of TLS or other authentication technique adoption

Data layer: This layer will be affected in case of unauthorized controller access, side channel attack on input buffer, packet process timing analysis, Flow rule modification to update packets, controller hijacking, controller switch communication flood, switch flow table flooding and lack of TLS or other authentication technique adoption.

A flow is a directed traffic pattern observed between two endpoints with distinct MAC addresses over specified ports. A flow graph is a graph representation of a traffic flow with edges as the flow metadata and switches being the nodes in the graph.

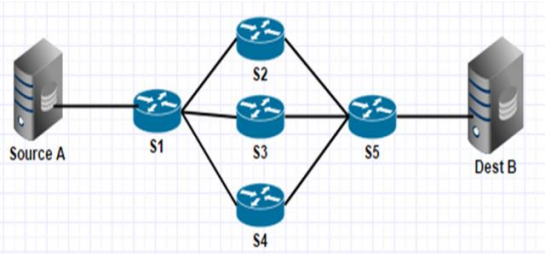


Fig : Flow Graph for flow from source A to Destination B

The next generation software defined network should be integrated with storage, processing the resources and various connectivity's under new management interacting with control systems to provide on demand networking and services with continuous update of features. A programmable network services will be deployed on the existing connection oriented and connection less networks and its devices. The interfaces and mechanisms that enable control and exchange of information between layers. The functional decomposition simplifies the implementation that is driven by the envisioned functionality, such approach is completely different from that of OpenFlow which does not decompose the layers into functional blocks. The key component of the programmable network services design is the description of services provided by each layer.

The physical resource or infrastructure layer has to sink with heterogeneous environments. It has two main functions. It provides a uniform view of different technological network and computational resources and it has intrinsic autonomic and programmable management of the resources, which provides a fast reaction time for management operations and facilities scalability of the programmable network solution in case of distributed management implementation.

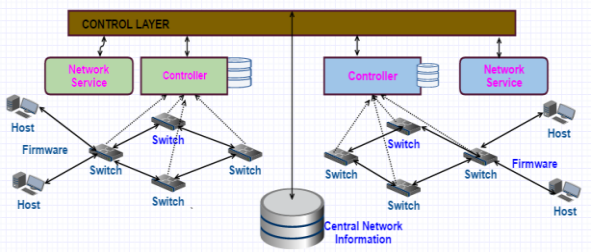


Fig: Improved Architecture of SDN with Programmable Network Services

The virtual networks which are utilizing physical resources can be created according to the virtual programmable network layer mechanisms. These virtual networks acquire their properties based on the needs of customers and services. The networks have embedded self-managed mechanisms that can control and monitor the underlying physical resources, through utilizing in an intelligent manner the low level control and monitoring components of the physical resource layer. The self-management operations include self-configuration, performance optimization, and self-healing. The performance optimization can be realized with efficient usage of physical resources and programmable virtual networks. All these facilities aid in the scalability of programmable network services.

The application providers and end users can use specific virtual networks according to their needs to provide high quality, personalized, QoS aware, and secure services. The programmable network layer would be able to create the virtual network and instantiate the requested user's functionalities at the required locations to provide the desired QoS.

It has to be noted that the aforementioned programmability and self-management of different layers of programmable network services requires the ability to send, execute and monitor the execution code and therefore the management operations should be extended appropriately. In order to get that we need centralized or distributed execution environment.

The services which can be deployed in programmable network services are

- New services may be rapidly deployed
- Incorporating the features of existing services
- Improving the network resources
- Scalability and cost reduction in network service management
- Independence of network equipment manufacturer.
- Integration of information networks and its service
- A need of coordination from the interfaces to provide inter domain communication by allowing networking function.
- Authentication for other operators

Service module mechanisms for communication and programmability deployed by different operators for the same service.

V. PROGRAMMABLE NETWORK SERVICES TO DETECT ATTACKS

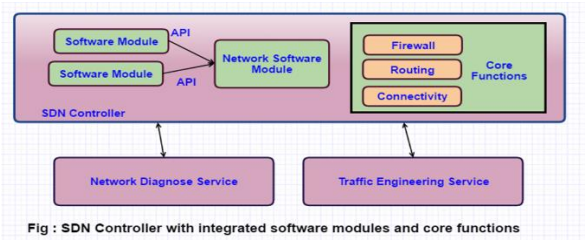


Fig : SDN Controller with integrated software modules and core functions

The present internet architecture has many problems to support, verify in detecting security attacks. The typical security systems are based on securing hosts or installing specialized network devices like middle boxes to detect anomalies into the network. These solutions become ineffective when the host require constant updates and effort from the network administrator like selecting the traffic to be filtered.

The software program in the SDN serves as a middleware connecting the network applications, services, central network information center and the various controllers in the middle layer called Network software and control layer.

The network hosting virtual environments and virtual Machines to overcome the problem of having several execution environments implemented in various technologies, and providing different abstractions, interfaces etc. The creation of required network of execution environments and set of virtual machines which are managed as one to realize and activate network software features. The virtual environment puts a common layer for management on top to support the installation and configuration of services code in various ways. This allows external clients to interact with services through the interface of the virtual environment in a generic way and interactions will be mapped to specific interfaces of the execution environments, the partitioning of resources. The service provider can manage its own virtual network to access the virtual environments available with respective service provider.

A virtual network will be formed by several virtual environments belonging to the same service provider but running on different network nodes to deploy services and make them available to customers. We also need to consider when we want to know the environments belongs to what virtual networks. i.e The approach of deploying and managing services which are not depending on the technology of the execution environment, how to manage nodes in easy manner for service providers as well as network providers, how to partition the resources among several service providers, account of resources usage per service provider and delegation of service management to the service providers.

Programmability for detecting attacks:

The new network services to detect security attacks in fast manner needs a dynamically deployed network devices such as routers, switches and application servers. Dynamic programming is used to create a new functionality at time by inserting executable code

into the network element. The basic idea is to enable third parties such as operators and service providers to inject application specific services into the network. Application may utilize this network support in terms of optimized network resources to become network aware. The network programming provider's unprecedented flexibility in various communications. The viable architectures for programmable networks must be engineered carefully to achieve suitable tradeoff between flexibility, performance, security and manageability.

The programmable network services to detect security attacks should cope with heterogeneous environments providing uniform view of different technological networks and computational resource.

The programmability checks whether the packet matching a flow rule or not. If it is not matching a flow rule must be sent by the switch to the controller. This opens up the possibilities for malicious hosts to tamper with network topology and data plane forwarding. The malicious data can forge packet data that would then be relayed by switches as messages and frequently processed by the controller or implement denial of service (DoS) attacks on the controller and switches and channel mechanisms to extract information about flow rules. The soft switches can not only initiate all the host based attacks but also trigger dynamic attacks on traffic flows passing through the switch.

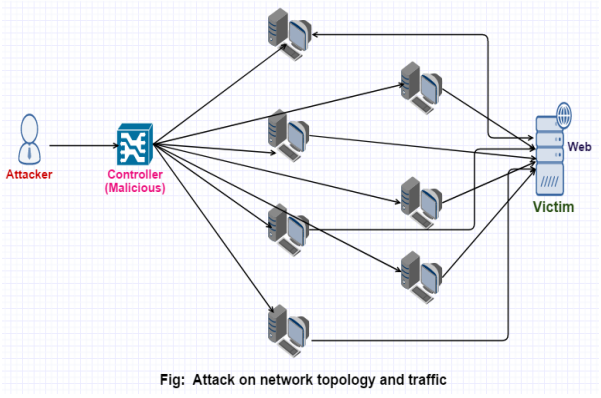


Fig: Attack on network topology and traffic

SDN controllers process a variety of protocol packets like LLDP, IGMP, ARP, sent by switches to construct its view of the network topology. Controllers process LLDP messages for topology discovery and IGMP messages to maintain multicast groups, where as it forwards ARP requests and replies enabling end hosts to build up ARP caches facilitating network communication.

If any fake topology attack can be launched on an SDN controller to poison its view of the network by sending any sort of message. These malicious messages could be generated by untrusted switches themselves or by end hosts, which can send the message across network links between the switches. When the controller tries to route traffic over these links, it results in packet loss, and if this link is on critical path, it could even lead to a blackhole.

Topological constraints like both network invariants as well as administrator specified, can be verified using the metadata. Once the default invariants have been verified, the metadata are compared against all applicable policies, and any deviant behavior is flagged. All such verification is deterministic and fast due to incremental flow graphs, which allows verification to proceed over the last edge or metadata that was added to the graph.

Malicious hosts and switches can mount DoS by flooding the network with traffic to arbitrary hosts to exhaust resources on vulnerable switches and/or the SDN controller, thereby affecting in the data plane. In general the flow rules are stored in switch's content addressable memory (SCAM), which is the fastest associative memory. Malicious hosts may target a switch's content addressable memory to perform directed DoS attacks against other hosts. Malicious hosts may send arbitrary traffic and force the controller into installing a large number of flow rules, there by exhausting the switch's content addressable memory. Meanwhile no other flow rules can be installed on this switch, until the installed flows expire. If this switch is on a critical path in the network, then it may result in significant latency or packet drops.

Verification of forwarding constraints in the data plane required the validation of both packet and flow level metadata, which may be either deterministic or probabilistic depending on the nature of constraints involved.

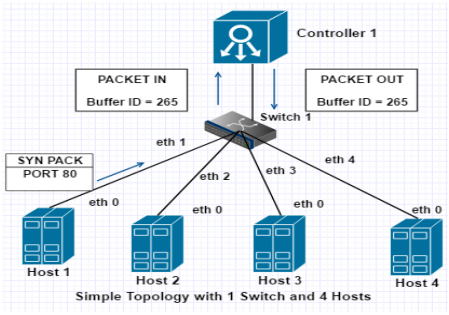
The attackers often break into the network to leverage internal ventage points, and subsequently launch attacks on the internal network. Our goal is to verify onset of attacks on the network topology, data plane forwarding and detect violations of policies within SDN's.

We also considered the following issues to detect the security attacks in programmable network environments

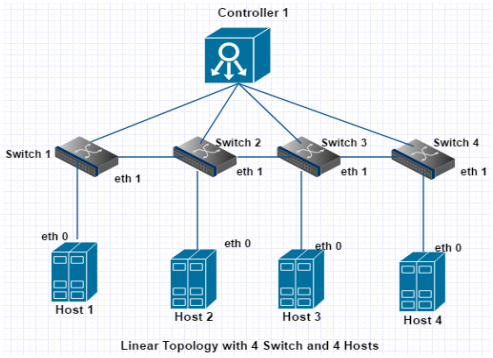
i) Control of Virtual wireless resources
ii) Mapping virtual resources to the wireless resources
iii) Control of virtual wireless resources
iv) Control of virtual resources for smart objects
v) Mapping virtual resources to smart objects resources
vi) Uniform automatic and optimized management
vii) Scalable programmable delivery infrastructure as systems inter orchestration for big data and service networks
viii) Energy management and optimization

VI. CONCLUSION & FUTURE WORK

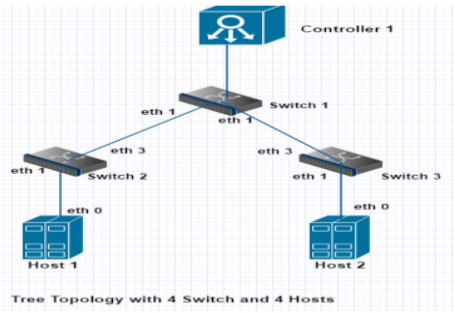
We considered the network topologies like simple topology with one switch and four hosts, linear topology with four switches and four hosts, finally we considered tree topology with 3 switches and two hosts. The programmable network services are enabled to test the security attacks on the topologies and data transmitted. Due to the limited internet facilities we limited to the topology and data traffic.



Simple Topology with 1 Switch and 4 Hosts



Linear Topology with 4 Switch and 4 Hosts



Tree Topology with 4 Switch and 4 Hosts

VII. EVALUATION

The possibility of debugging and testing new designs in a real environment is difficult and expensive. Even through the current network simulators are very expensive and some are not supportive for simulation, the network simulation is the first step to evaluate the implementation. However, the current network simulators are expensive and don't offer support to simulate.

Tools used to evaluate the performance are Mininet Emulator, Omnet, Wireshark 2.2.5, OpenFlow Controller.

We consider an enterprise SDN setup with minimum traffic across network entities. We assume a trusted controller but not on either switches or end hosts. i.e. the switches can lie about everything except their own identity, since the switches connect with the controller over separate connections. More security attacks were detected when the packets are transmitted in tree topology when we compared with other two topologies, simple and linear topologies.

The following image shows that the four terminals are running in parallel.

In first window floodlight controller is running with the command bellow.

$ java –jar target/floodlight.jar

In the second window we created topology in mininet. The third and fourth windows are used to run Sflow virtualization.

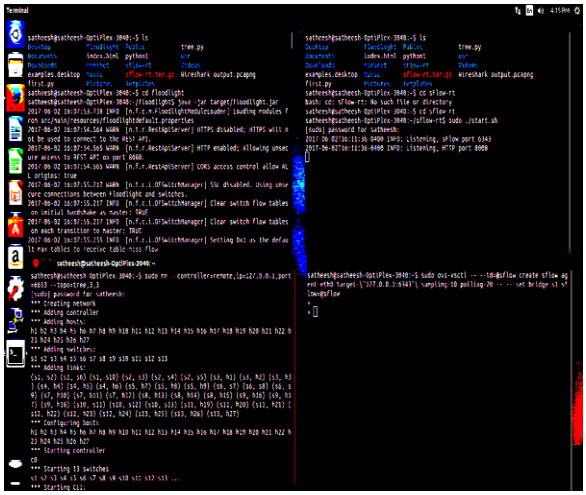


Fig R1: Four terminals for running floodlight controller, mininet and sflow

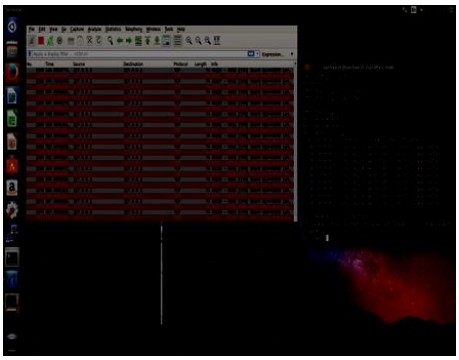


Fig R2: Attacks on the topology and traffic

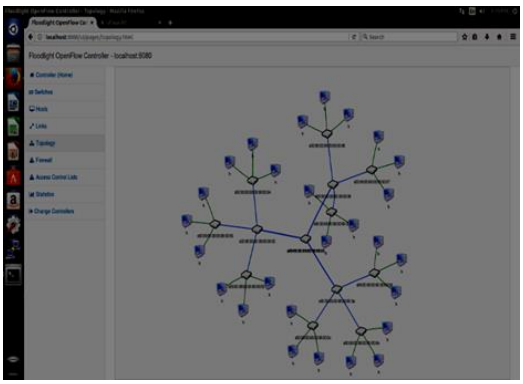


Fig R3: Simulation result for topology created in mininet

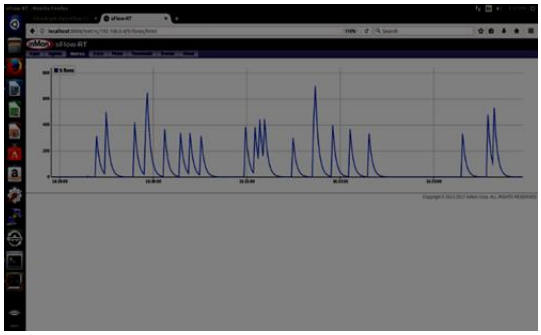


Fig R4: Figure showing flow analysis

REFERENCES

1. ETSI 'Software-aware and Management-aware SDN" initiative presented at 3rd ETSI Future Networks Workshop 9-11 April 2013
2. Network Function Virtualization, white paper, SDN and OpenFlow World Congress, Darmstadt, Germany, October 22-24, 2015.
3. Galis, A., Denazis, S., Brou, C., Klein, C. (ed) – "Programmable Networks for IP Service Deployment" ISBN 1-58053-745-6, pp450, June 2004, Artech House Books,
4. Signaling is growing 50% faster than data traffic, technical white paper, Nokia Siemens Networks, 2012.
5. I. Houidi, et al., "Virtual Network Provisioning Across Multiple Substrate Networks", Computer Networks, Vol. 55, N. 4, Special Issue on Architectures and Protocols for the Future Internet, March 2011, pp. 1011-1023.
6. M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, "Ethane: Taking control of the enterprise," in ACM SIGCOMM Computer Communication Review, vol. 37, no. 4. ACM, 2007, pp. 1–12.
7. "OpenFlow Switch Specification Version 1.3.2," Open Networking Foundation. [Online]. Available: https://www. opennetworking.org
8. D. Kreutz, F. Ramos, and P. Verissimo, "Towards secure and dependable software-defined networks," in Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking. ACM, 2013, pp. 55–60.
9. D. Li, X. Hong, and J. Bowman, "Evaluation of Security Vulnerabilities by Using ProtoGENI as a Launchpad," in Global Telecommunications Conference (GLOBECOM 2011). IEEE, 2011, pp. 1–6.
10. Ad
11. Ahmed, A., Ahmed, E., A Survey on Mobile Edge Computing - 10th IEEE International Conference on Intelligent Systems and Control, (ISCO 2016).
12. "Open vSwitch," http://openvswitch.org/.
13. "OpenDaylight," http://www.opendaylight.org/.
14. "POX," http://www.noxrepo.org/pox/about-pox/.
15. "Project Floodlight," http://www.projectfloodlight.org/floodlight/.