INTERNATIONAL JOURNAL OF RECENT TRENDS IN ENGINEERING & RESEARCH
Impact Factor 4.101 ISSN -2455-1457
RECENT ADVANCES IN ELECTRONICS & COMMUNICATION ENGINEERING (RAECE-2K17)

A Survey Architecture Interface and Security Issues in Software Defined Networking

Banothu Gopala Rao
Lecturer, Government Institute Of Electroni cs,
East Marred Pally
Secunderabad

P.Madhavi
Lecturer, Government Institute Of Electronics,
East Marred Pally
Secunderabad

***Abstract*—With increasing number of mobile phones and smart devices, it has become hard to manage the networks proactively as well as reactively. Software Defined Networking (SDN) is an emerging technology that promises to solve majority of the challenges faced by the networks in current times. SDN is based on decoupling of data plane and control plane. SDN has a generalized control plane for all networking devices of the network which makes it simple and easy to configure devices on the fly. This paper surveys how Software Defined Networks evolved to be one of the most preferred technology of contemporary times. The architecture and working of all the planes of SDN have been discussed. SDN finds application in variety of areas, some of which have been highlighted in this paper. SDN faces many security threats in each of its planes. The major security challenges are also presented in detail at the end of the paper.**

***Index Terms*—Software Defined Networking (SDN), Network Virtualization, Openflow, Software Defined Wireless Networking (SDWN).**

I. INTRODUCTION

The proliferation of smart devices has led to exponential growth of Internet and we are in a computerized society where everything is connected and can be reached from anywhere. The increased dynamic behavior of nodes results in more control over traffic for a properly coordinated connection. Such multifold devices demand wireless connections with high mobility is also a necessity now. In addition to mobility and wireless connection, virtualization and agility with changing network configurations make network operation a challenging task.

Emerging trends in the world of internet and technology domain are pushing towards the integration of all devices, no matter how big or small. The network capability of every object demands collaboration of different technologies. The networks need to support heterogeneity and at the same time be able to handle huge influx of data coming from internet of everything. The privacy of users, devices and the data being transmitted also needs to be ensured. Despite the indefinite expansion and adoption in every dimension, traditional IP networks are rigid and quite difficult to manage. Static and inflexible nature of traditional networks makes it hard for network operators to incorporate latest technologies and hence is not ideal for emerging business ventures. The traditional networks are tightly coupled with hardware components which make it hard to configure them. Currently both wireless and wired networks support heterogeneity only for devices with similar network characteristics or protocol which results in poor Quality of Service and Quality of Experience [1]. A programmable automatic network architecture operating in a dynamic fashion can provide an ideal solution. Against backdrop of the above stated issues of traditional networks, Software Defined Networking (SDN) has evolved as a suitable solution for majority of the concerns.

SDN is an agile network architecture which offers a centralized control mechanism across the network domain [2]. SDN approach is based on the separation of control and data planes of network devices. SDN contains centralized single control plane for entire underlying physical network infrastructure. The control plane is created by integrating the control mechanism of all network devices and is placed on top of Network Operating System. The data plane is merely used for forwarding the data packets across the network. The only major task carried out t hardware level is forwarding of the packets and majority of control related jobs like policy enforcement are carried out at software level. That

actually is the essence of SDN, separation of the control from the network devices, allowing network operators to manage network dynamically [2]. The abstraction created by SDN has led to its popularity in the field of innovative technologies since it provides desired flexibility and virtualization in different networking environments.

SDN is seemingly going to benefit almost all the areas of computer technology predominantly Mobile networks, Wireless Sensor Networks, Cloud computing, big data Analytics, Internet of Things. Although SDNs is a promising network infrastructure that is going to solve most of the pressing issues faced by traditional networks, security has not been considered seriously as yet and remains an open issue. Technology comes first and security follows later. The security concerns in SDN are elevated with the centralization of network logic or control mechanism.

This paper is organized in six sections. The first two sections discuss about evolution and architecture of SDN. The next section is about the applications of SDN in various areas. The fifth and sixth sections focus on the security issues of SDN and measure that can be taken to improve the security in SDN.

1. SDN EVOLUTION

The design and management of networks is slowing changing with the onset of Software Defined Networking. Although seemingly a recent concept, SDN is based on older networking concepts which eventually led to its development.

Active networking was one of the first attempts in making networks dynamic by programming individual network nodes. Active networks proposed two different methods for programming nodes: programmable switches and capsules. The programmable switch allowed the switches to download the programs with specific instructions whereas capsule programming proposed that small programs should be embedded in the packet. This programme should be executed at each node through which packet traverses. OpenFlow, Protocol Oblivious Forwarding (POF) and ForCES, represent the newer trends in the area of programmable networks. Compared to active networking, these focus on altering the network devices to support flow mechanism alone since program execution is carried out by other devices.

The concept of programming the networks got more attention when the idea of decoupling of the control and data planes was given by the Network Control Point (NCP). NCP concept was basically created by AT&T for improving upon management and control of telephone services. Other relevant technologies such as ForCES, Tempest, Path Computation Element (PCE) and Routing Control Platform (RCP) suggested decoupling of the control and data planes to have a better control and management in Ethernet, Asynchronous transfer Mode (ATM), Border Gateway Protocol (BGP), and Multiprotocol Label Switching (MPLS) networks, respectively. Later after the above technologies lost their impact; initiatives like Synchronous Active Network environments (SANE), Ethane also proposed separation of the control and data planes. These were different from earlier attempts in a way that these did not require much change on the working style of forwarding devices.

Attempts to build operating systems for network systems had been done for quite some time but it took a new shape with the introduction OpenFlow based Networking Operating Systems such as NOX and Open Network Operating System (ONOS). Cisco's IOS is one of the most widely used Network Operating Systems. Some other notable operating systems for networks are Junos and ExtremeXOS. With the advent of Network Operating Systems (NOS) a levelof abstraction Is created hiding the complexities underlying physical infrastructure from network operators or administrators. It has simplified the configuration, control and management of the network hardware and also made it easy to develop and deploy the applications and protocols.

Network Virtualization (NV) provides networking services through a virtual network. The virtual network is decoupled from the physical network infrastructure and works dependently In a hypervisor.

SDN has also been conducive for the concept called Network Virtualization. The early instancesof Networkvirtualization was first introduced by the Tempest project. The idea behind Tempest was to share the physical resources among multiple small switches (switchlets) under control of single ATM switch on top. Likewise, MBone was also created to introduce virtualization. Similar works carried out in the same context include Planet Lab, Global Environment for Network Innovations (GENI), Virtual Integrated Network Infrastructure (VINI) and FlowVisor being the latest initiative. FlowVisor introduces hypervisor basedvirtual network infrastructure which is much like a hypervisor solution used in computer memory and storage. Koponen et al. [3] suggested Network Virtualization Platform (NVP) for improving the management in multi-tenant datacenters.

Open signaling which was introduced in 90's proposed open and programmable interfaces by decoupling control and data signaling. Currently OpenFlow and SDN under Open Networking Foundation (ONF) [4] are progressing towards a similar concept that was given by open signaling. Feamster et al. [5] have presented a detailed history about the programmable networks in a chronological order.

II. SDN ARCHITECTURE

Computer networks can be viewed as three distinct

functional planes namely the data plane, control plane and management plane.

The data forwarding task is carried out by the data plane. The policies and protocols used by the network for filling up the routing tables are represented by control plane. The management plane provides additional services like provisioning and monitoring of the networks. The data plane forwards data as per the policy enforced by control plane and the policies are defined in the management plane. The three planes work together in a coordinated way. The traditional networks have combined control and data planes on each network device. Such purely decentralized approach with embedded control and data planes work well and provide good performance until now [6]. Due to the exponential growth of Internet there is a need for higher network flexibility and reliability which led to the evolution of SDN. Software-Defined Networking (SDN) is novel paradigm that is expected to address limitations of the current networking practices and ensure optimum levels of performance. Comparison between traditional networks and SDN is shown by the following figure - Figure.1a and 1b respectively. SDN network architecture is based on following points:

1. Separation of control and data functionality. The control plane is lifted up to a centralized controller. Switches become simple forwarding elements. Segregation of duties turns switches into forwarding hardware with minimal burden of policing. Few expensive controllers and many cheap switches make the network infrastructure cost effective.
2. The control logic is imposed on the network by integrated and logically centralized controller. The centralized control in a SDN simplifies configuration of networks and makes enforcement of rules and policies easy job.
3. The controller is the heart of the SDN which has made it possible to create an operating system for networks, much like the operating systems of computer systems.SDN controller disseminates the control logic across the network with support from corresponding software component called Network Operating System (NOS).
4. The NOS like any other operating system acts as an intermediary providing a environment conducive for programmability and abstraction.
5. Packets are forwarded as per the rules supplied by the controller compared to destination based traffic flow of traditional IP based networks. The forwarding decisions are based on flow rules. When a packet arrives at the ingress port of a switch, its header fields are matched with the flow entries in a table, if any entry matches, the statistics/counters are updated and corresponding actions are taken.
6. The basic characteristic of SDN is the programmable network (software defined) which is achieved by running software application on top of the SDN controller.
7. SDN provides a global view of the entire network and brings together different types of underlying network behavior by enabling flow abstraction and a centralized control.

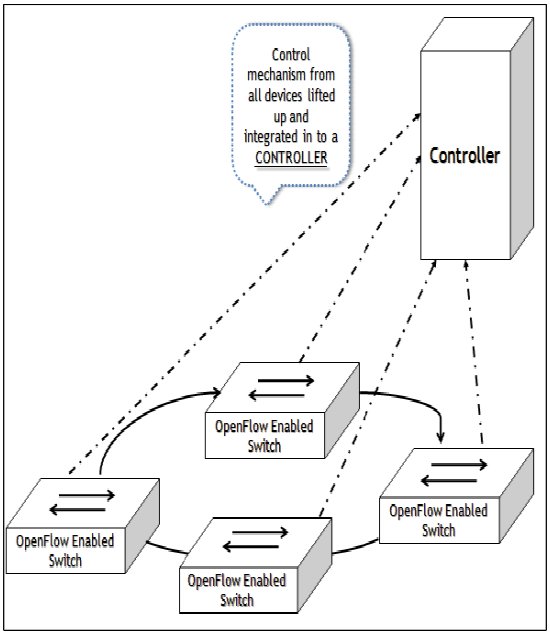

Fig.1(a). Control and Data planes in Traditional Networks

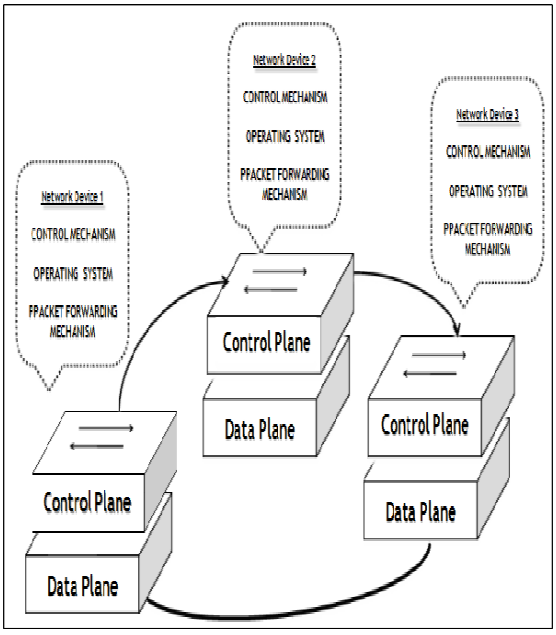

Fig.1(b). Control and Data planes in Software Defined Networks

The centralized control in SDN does not preclude the distributed physical existence of the SDN controllers. The motivation behind having programmable and centralized control logic is simpler and less error-prone network policies with high level languages and software support. A control program is capable of automatically reacting to abnormal behavior of the network. The global view of the network status aids flexibility and agility of the networks [6]. The working of different layer in SDN has been depicted in the following Figure 2:

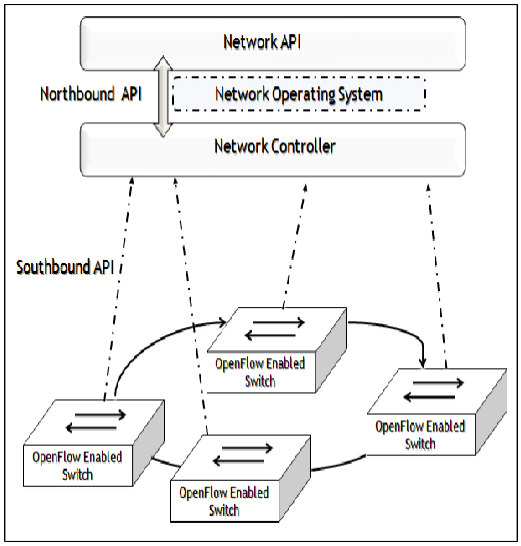

Fig.2. Simple SDN Architecture

A. *Application Layer*

The application plane is the top layer of SDN and it provides a suitable environment for creation of applications and services. The SDN applications are the software or programmable part of SDN which is used to share network behavior and requirements with the SDN controller via northbound APIs.

B. *Northbound SDN interface*

The northbound interface abstracts the physical network infrastructure and enables communication, between controller and application layer. There is no standard for northbound SDN interface at present.

C. *Control plane*

The network is controlled by integrated and logically centralized controller which lies in the control plane. The controller is the heart of the SDN which has made possible to create an operating system, for networks, much like the operating systems of computer systems. The centralized control in a SDN simplifies configuration of networks and makes enforcement of rules and policies an easy job. The controller's job is to formulate the flow table entries and send those to the SDN switches via Southbound Interface.

D. *Southbound Interface*

The southbound interface provides a communication link between the controller and SDN switches. The Southbound Interface supplies the flow rules, generated by the controller, to SDN switches. OpenFlow(OF) is most popular and common protocol used to carry out southbound communication.

E. *Data Plane*

The underlying network infrastructures and is known as data plane. This layer consists of the forwarding devices responsible for forwarding packets as per the flow rules provided by controller
via Southbound Interface. The data plane is responsible for enforcing management policies in the SDN hardware. It also gets the information from switches and sends back to the controller.

I. APPLICATION OF SDN IN VARIOUS AREAS

The advent of software defined networking (SDN) has led to many opportunities and paved the way for many innovations in variety of networking areas. By virtue of network programmability and virtualization, SDN promises to overcome many of the issues faced in current networking technologies. In this section some of the areas which can benefit from SDN have been highlighted.

A. *SDN in Data Center*

The dynamic assignment of resources between tenants of the data center and to the general public makes SDN and Network virtualization is the new trend in data centers. The ever rising density of servers need for better networking speed and bandwidth make it necessary for data centers to hold more and more information. A cost effective approach to overcome this issue is to merge many data centers into a single data center with more capacity, and virtualization with such larger physical density would serve as one of the best solutions. Virtualization has lesser power requirements and makes efficient use of hardware and is also capable of creating, deleting and expanding or reducing the applications or services in lesser time.

Koulouzis et al [7] have proposed a SDN based approach to file transfer services. Programmable switches have been used for large data transfers between Research infrastructures (RI)s since RIs play a major role in the dissemination of knowledge and technology. SDN based models provide a better base for the explosion in the size and number of data centers. With increased size and scale of data centers, recovery from failure has become a tough task. A global view of the network provided by SDN may make recovery from failure easier.

Although the SDN for data centers is in development stage, there are organizations that have started implementing SDN into production. Google has implemented Open SDN for managing WAN connections, Microsoft Azure uses overlay creating tens of thousands of virtual networks, eBay has implemented public cloud virtual networks with VMware's Nicira switches [8].

B. *SDN for WAN*

There is no certain traffic forwarding decision whenever failover occurs in network because of lack of a global view of the network. The global view enables network operators to see all paths as well as other capabilities from one fixed location. Compared to traditional IP networks, SDN controller can provide such a central view of the network and can
compute optimal paths [8].

Hongyu et al [9] have come up with a software defined network (SDN) based routing strategy called Backbone Networks Energy Saving Strategy (BNESS) for energy saving in backbone networks. Kotronis et al [10] proposed an idea involving a new routing model based on inter-domain centralization for Border Gateway Protocol (BGP). An emulation framework has been made publicly available by the authors for experimenting on the proposed idea of hybrid BGP–SDN inter-domain routing. The proposed idea has been evaluated as a use case.

Qadir et al [11] have given a description of various SDN based architectures that can be beneficial for creating futuristic programmable wireless networks. The distinct scenarios of software-defined radio, cognitive radio networking, software-defined networking and programmable wireless processors are presented with a common motive of building Software Defined Wireless Networks (SDWN). In same context, a framework of software-defined cognitive wireless networking has been introduced that uses SDN and Cognitive Radio Networking (CRN) to create new use cases for wireless networks.

C. *SDN for IoTs*

With centralized control, flexibility, scalability and abstraction, SDN offers enormous benefits to network control. Same features of SDN can be used in the Internet of Things (IoTs). Some of the major problems that are faced by IoTs are heterogeneity of devices, huge volume of data from numerous devices and the security issues. The challenges faced in making IoTs a reality can be largely evaded by SDN.

Arbiza et al [12] presented the use of a SDN-based middleware for monitoring health of patients in their own homes, which results in improved network management for smart networks. Information about devices communication is provided by OpenFlow counters.

Wen et al [13] present a representational state transfer (REST) framework for IoT based on software defined network (SDN). Separate control and data planes for IoTs are introduced in the framework suggested and a gateway model is also proposed to evaluate the functioning of the framework. The proposed architecture is supposed to be good for raw data collecting and has ability to provide a sharing mechanism for IoT.

D. *SDN in Other Environments*

Currently IT industry dealing with many issues related to networking devices like management, scalability, security and flexibility. In traditional networking all the complex tasks are carried out manually which result in poor management of networks. SDN can be foreseen as a solution for majority of such problems. Incorporating SDN not only simplifies the networks problems but it also reduces the operational cost considerably. Some of the important features of SDN which makes it interesting include: handling variety of devices, improved configuration, logically centralized control, global view of the network, granularity and flexibility. With such qualities SDN is applicable in most of the technologies for simplification and easier operation. Some of the areas where SDN has proved to be beneficial have been highlighted below:

IV. SDN SECURITY

Software Defined Networking has the potential to provide scalable, properly managed and flexible networking environments in future. The separation of control and date planes, the programmability of networks and centralized control have already made SDNs very popular in industry [14]. In spite of many advantages, security in the architecture was not considered initially. The logically centralized network control raises more security concerns. Security enforcement in SDN architecture is must in order to secure all resources associated with the network. The security issues concerning each SDN plane have been discussed in this section.

A. *Application Programming Interface*

Programmed control is the main objective of SDNs. The applications that run on top of the Network operating System control the network infrastructure making it easy to manage the network resources yet these applications can result in serious security challenges. The application plane related threats include:

1. Trust Model: Security in SDN cannot be enforces via physical infrastructure. It needs complete trust on SDN applications for network services as well as security policing. If applications are compromised whole network has to bear the consequences.
2. Chained applications: Many applications are capable of running in a chain. Such execution which includes nested applications can result in security challenges. In addition to that a chained execution nested application can bypass the access control mechanism leading to unauthorized access.
3. Altering SDN Database: Applications running on top of SDN have certain privileges to access the internal storage which they can exploit and manipulate the internal database of an SDN controller. Access to database can lead to manipulation of network behavior.[15]
4. Third-party applications: Third-party applications are provided by a vendor other than the manufacturer of the device. Interoperability issues and conflict in security policies can be serious security issues arising from third party controls.
5. Misuse of resources: Fraudulent applications can exhaust the vital system resources like memory and CPU. Such consumption seriously affects the performance of other applications, including the SDN controller. Such attacks have

been verified [16].

B. *Northbound Interface*

The Northbound interface is targeted for following security vulnerabilities:

1. Standardization: There is no standard for northbound Interface in SDN. Apart from that open source development face lot of security challengers from hacker since these do not follow any standard.
2. Improper Interface Design: Any northbound interface can be exploited easily by SDN applications if it is not designed properly. A malicious application can use poorly configured northbound interface to arbitrarily unsubscribe a target application[18]. It can fetch important subscribed control messages this way and use those for hacking.

C. *Control plane*

SDN controller plays a vital role in management, control of the entire network and thus can be primary target of the attacker. Some of the control plane related security issues faced today are:

1. Packet-in attacks: When a packet arrives at the SDN switch with no match that packet is directed to the controller. And such packets are known as packet-in, switches can send infected packet-in messages that can corrupt the controller [15].
2. Directed Denial of Service (DoS): The packet-ins can be flood a SDN controller and place it in an uncertain state. Such attacks are termed as Directed DoS attacks and are a serious security concern for SDN controllers. [17]
3. Blocked view of the controller: Controller view or monitoring mechanism can be affected when address resolution protocol (ARP) packet is relayed as a packet-in message The Link Discovery Packets (LDP) can be sent as packet-in messages to create false network topologies [17].
4. Side-channel attacks: Side channel attack is launched on the basis of information that the attacker gains from physical characteristics of a cryptograph system such as timing information and energy consumption. Such attacks can be caused by exploitation of packet-ins and important information can be gained. [17].
5. Visibility features of the SDN controller: The visibility features of SDN can also be vulnerable to harvesting of network intelligence of the underlying architecture for further exploitation [18].
6. Exploiting controller's ability to examine and authenticate applications: A challenging task to be accomplished by a controller is to authenticate, authorize and audit applications for access and use of network resources.
7. Apart from above mentioned, threats in a centralized environment, SDN controller are viewed as a single point of failure i.e. if SDN is halted whole network goes down *by* the switch to the controller directly.

D. *Southbound interface*

The link between controller and switch uses Open- flow (OF) for communication. Transport Layer Security (TLS) is enabled for channel security in the latest versions of Open-Flow but TLS is not reliable in all the cases [19]. The optional TLS in OF does not support TCP level protection. The different types of attacks that can occur in Southbound Interface are:

1. Man-in-the-middle attacks: The southbound Interface can be exploited for man-in-the-middle attacks. An adversary can modify the control messages communicated between the controller and the SDN switches to corrupt network behavior [20].
2. Traffic Analysis: The southbound interface can also be exploited for traffic analysis. By sniffing the control messages an attacker can get access to sensitive information.

E. *SDN Switches*

Although SDN switches are simple forwarding components with minimal functionality, these can be used by attacker for launching high intensity security attacks. SDN switch when compromised can cause variety of attacks. Some of the securities challenges faced by SDN switches are:

Identification of correct flow rules: Switch flow rules are used the forwarding of traffic. The controller has to find out whether the switch- generated flow rules are correct or not. A switch which under

While as in distributed SDN scenario it takes time to converge the network which affects the performance. Tantar et al [22] have proposed that security in SDNs can be enhanced using cognitive algorithms and have presented a cognitive module implemented in the application plane to overcome security failures in SDN.

V. SECURITY MEASURES

A. *Security Application*

Having a specific application on API for enforcement of security can result in a secure SDN. With the help of a security application security policies can be coupled to SDN policies and rules. Such mechanism can validate SDN flow against the security policy, ensure that Security Policy is implemented for all traffic and monitor the network behavior.

B. *Secure development of Applications*

Security application development framework called FRESCO has been developed to address

several security issues in the OF enabled SDN. FRESCO provides a suitable environment for developing secure applications. FRESCO framework modules are the basic units of operation and also the most important element of FRESCO. All the security functions are provided by FRESCO modules. FRESCO has a library of 16 basic reusable modules for security enforcement, [OFsec]. These modules can be combined to make more sophisticated modules.

C. Security policy

Enforcing security policy can make SDN secure to a large extent. FLOVER is a mechanism to verify the flow that finds out whether the flow table is functioning as per the security policy of the network. FLOVER works efficiently except that it cannot handle the dynamic flow rules in SDN. One more tool for security enhancement is No bugs In Controller Execution (NICE)[18].NICE checks for inconsistencies in the Openflow applications and finds errors in Openflow application code. Other notable techniques used to check the correct functioning of flow table are FlowChecker, VeriFlow and NetPlumber.

VI. CONCLUSION

The complexity and rigidity in currents networks makes them hard to manage. Each product comes with propriety rights which make upgradation of new firmware slow and complex. Such vendor lock-in has also restricted innovation and interoperability of devices. Software-Defined Networking (SDN) with concept of network programmability promises to remove many of the prevailing issues in current network. The paper throws some light on the history of SDN. The components of software-defined networking are also discussed and a comparison is drawn between the traditional networking concepts and SDN. The paper also discusses how SDN is going to affect various areas in the field of computer science and technology. Finally the security issues in the SDNs are briefly discussed.

.

REFERENCES

[1] Mao Yang, Yong Li, Depeng Jin , Lieguang Zeng, Xin Wu, Athanasios V. Vasilakos "Software-Defined and Virtualized Future Mobile and Wireless Networks: A Survey", Mobile Network Applications, vol. 20, **(2015)** pp.4–18.

[2] Adam Drescher, "A Survey of Software-Defined Wireless Networks", (2014). http://www.cse.wustl.edu/~jain/cse 574-14/ftp/sdwn/index.html

[3] T. Koponen, K. Amidon, P. Balland, M. Casado, A. Chanda, B. Fulton, I. Ganichev, J. Gross, P. Ingram, E. Jackson, A. Lambeth, R. Lenglet, S.-H. Li, A. Padmanabhan, J. Pettit, B. Pfaff, R. Ramanathan, S. Shenker, A. Shieh, J. Stribling, P. Thakkar, D. Wendlandt, A. Yip, and R. Zhang, "Network virtualization in multi- tenant datacenters," NSDI 14, Seattle, WA, (2014), pp. 203–216.

[4] https://www.opennetworking.org

[5] Nick Feamster, Jennifer Rexford, and Ellen Zegura, "The Road to SDN" Queue 11, 12, (2013), 21 page.

[6] Diego Kreutz, Fernando M. V. Ramos, Paulo Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig, "Software-Defined Networking:A Comprehensive Survey" (2014).

[7] Spiros Koulouzis, Adam S.Z. Belloum, Marian T. Bubak , Zhiming Zhao, Miroslav Živković, Cees T.A.M. de Laat," SDN-aware federation of distributed data" Future Generation Computer Systems vol.56, (2016) pp. 64–76.

[8] Peng Hongyu, Wang Weidong, Wang Chaowei, Chen Gang, Zhang Yinghai, "QoS-guaranteed energy saving routing strategy using SDN central control for backbone networks", vol.5, (2015) pp. 92–100.

[9] Vasileios Kotronis, Adrian Gämperli, Xenofontas Dimitropoulos, "Routing centralization across domains via SDN: A model and emulation framework for BGP evolution" vol.92, (2015) pp. 227–239.

[10] http://www.ietf.org/rfc/rfc5810.txt

[11] Junaid Qadir1, Nadeem Ahmed and Nauman Ahad, "Building programmable wireless networks: an architectural survey" EURASIP Journal on Wireless Communications and Networking, vol. 172 (2014) http://jwcn.eurasipjournals.com/content/2014/1/172.

[12] Lucas Mendes Ribeiro Arbiza, Liane Margarida Rockenbach Tarouco, Leandro Márcio Bertholdo and Lisandro Zambenedetti Granville, "SDN-Based "Software-defined networking: a survey". Computer Networks, vol. 81(0), (2015) page no. 79-95.

[13] Service Delivery in Smart Environments" Intelligent Distributed Computing, (2016) IX, DOI 10.1007/978-3-319-25017- 5_45.

[14] Zhigang Wen, Xiaoqing Liu, Yicheng Xu, Junwei Zou, "A RESTful framework for Internet of things based on software defined network in modern manufacturing" Int J Adv Manuf Technol, Springer, (2015) DOI 10.1007/s00170-015-8231-7.

[15] Adnan Akhunzada, Abdullah Gani, Nor Badrul Anuar, Ahmed Abdelaziz ,Muhammad Khurram Khan, Amir Hayat , Samee U. Khan, "Secure and Dependable Software Defined Networks" Journal of Network and Computer Applications- Elsevier (2015) (Article in Press).

[16] Seungwon Shin, Vinod Yegneswaran, Phillip Porras and Guofei Gu,"AVANT-GUARD: scalable and vigilant switch flow management in software-defined networks ", In: Proceedings of the ACM SIGSAC conference on comp & comm. Security (2013).

[17] Xitao Wen, Yan Chen, Chengchen Hu, Chao Shi, Yi Wang, "Towards a secure controller platform for openflow applications, Proceedings of the ACM second ACM SIGCOMM workshop on Hot topics in software defined networking; (2013).

[18] Mohan Dhawan, Rishabh Poddar, Kshiteej Mahajan and Vijay Mann, "SPHINX: Detecting Security Attacks in Software-Defined Networks", Proceedings of the Network and Distributed System Security (NDSS) Symposium, (2015).

[19] Adnan Akhunzada, Abdullah Gani, Nor Badrul Anuar , Ahmed Abdelaziz ,Muhammad Khurram Khan , Amir Hayat , Samee U. Khan, "Secure and Dependable Software Defined Networks" Journal of Network and Computer Applications- Elsevier (2015) (Article in Press).

[20] Diego Kreutz, Fernando M. V. Ramos and Paulo Verissimo, "Towards secure and dependable software- defined networks" Proceedings of the second ACM SIGCOMM workshop on hot topics in software defined networking (2013).

[21] Kevin Benton, L. Jean Camp, Chris Small , "Openflow vulnerability assessment", Proceedings of the second ACM SIGCOMM workshop on hot topics in software defined networking; (2013).

[22] Hamid Farhady, HyunYong Lee, Akihiro Nakao. Engel, "Cognition: A Tool for Reinforcing Security in Software Defined Networks" Advances in Intelligent Systems and Computing 288, DOI: 10.1007/978-3-319-07494-8_6, (2014) Springer.