ISSN: 2321-8169
95 - 100

Software Defined Networking

Yogesh Jadhav, Shubhangi Shinde, Shubhangi Singh, Mansa Gontimukulwar, Mahesh Sorte
(D.B.N.C.O.E.T. Yavatmal, B.E, I.T, Final Year)
yogeshrjadhav969@gmail.com, shubhangis059@gmail.com, 555shubhangisingh@gmail.com, mansagontimukulwar@gmail.com, maheshsorte7@gmail.com

Abstract— A Network management is testing. The possibility of programmable networks has as of late re-increased impressive force because of the rise of the Product Defined Networking (SDN) worldview. SDN, frequently alluded to as a "radical new thought in networking", guarantees to significantly improve network management and empower development through network programmability. Additionally, for a radical change of worldview in the plan and operation of future media communications foundations. SDN is intended to address the way that the static engineering of conventional networks doesn't bolster the dynamic, versatile, processing and capacity needs of more cutting edge figuring environment, for example, server farms. In this paper, first we presented the more summed up ideas about SDN and distinguish issues with the present best in class network arrangement, management instruments to enhance different parts of network management. At that point we introduce the SDN design and the OpenFlow standard specifically, and inspect ebb and flow and future SDN applications, security utilizing SDN worldview, Particular concentration is set on the SDN significance for the future network and investigate promising examination bearings in light of the SDN worldview.

KeyTerms—*Software-Defined Networking, programmable networks, data plane, control plane, infrastructures.*

*****

1. INTRODUCTION

PC networks are regularly worked from a substantial number of network gadgets, for example, switches, switches and various sorts of center boxes (i.e., gadgets that control traffic for purposes other than bundle sending, for example, a firewall) with numerous intricate conventions executed on them. Network administrators are in charge of configuring strategies to react to an extensive variety of network occasions and applications. They need to physically change these abnormal state approaches into low-level configuration charges while adjusting to changing network conditions. What's more, regularly they have to achieve these extremely complex errands with access to exceptionally restricted devices. Accordingly, network management and execution tuning is very testing and in this way mistake inclined. The way that network gadgets are generally vertically-incorporated secret elements fuels the test network administrators and directors confront.

The possibility of "programmable networks" has been proposed as an approach to encourage network development. Specifically, Programming Defined Networking (SDN) is another networking worldview in which the sending equipment is decoupled from control choices. It guarantees to drastically improve network management and empower development and advancement. The primary thought is to permit programming engineers to depend on network assets in an indistinguishable simple way from they do on capacity and registering assets. In SDN, the network knowledge is sensibly incorporated in programming based controllers (the control plane), and network gadgets get to be distinctly basic bundle sending gadgets (the information plane) that can be customized by means of an open interface (e.g., Strengths [2], OpenFlow [3], and so forth). SDN is right now drawing in significant consideration from both scholarly community and industry. A gathering of network administrators, specialist co-ops, and merchants have as of late made the Open Network Establishment, a modern driven association, to advance SDN and standardize the OpenFlow convention [2]. On the scholarly side, the OpenFlow Network Examine Center [4] has been made with an emphasis on SDN inquire about. There have likewise been standardization endeavors on SDN at the IETF and IRTF and different standards delivering associations.

The term programming characterized networking (SDN) has been instituted as of late. Nonetheless, the idea driving SDN has been advancing since 1996, driven by the longing to give client controlled management of sending in network hubs. Usage by research and industry bunches incorporate Ipsilon (proposed General Switch Management convention, 1996), The Storm (a system for safe, asset guaranteed, programmable networks, 1998) and Web Building Team (IETF) Sending and Control Component Detachment, 2000, and Way Calculation Component, 2004. Most as of late, Ethane (2007) and OpenFlow (2008) have conveyed the usage of SDN nearer to reality. Ethane is a security management design consolidating basic stream based switches with a focal controller overseeing induction and directing of streams. OpenFlow [3] empowers sections in the Stream Table to be characterized by a server outside to the switch. SDN is not, be that as it may, constrained to any of these executions, however is a general term for the stage.

2. SDN AND OPENFLOW

Software-defined networking (**SDN**) is a way to deal with PC networking that permits network heads to oversee network benefits through reflection of lower-level usefulness. SDN is intended to address the way that the static design of customary networks doesn't bolster the dynamic, versatile figuring and capacity needs of more cutting edge registering situations, for example, server farms. This is finished by decoupling or disassociating the framework that settles on choices about where activity is sent (the control plane) from the hidden frameworks that forward movement to the chose goal (the information plane).

SDN was normally connected with the OpenFlow convention (for remote correspondence with network plane components with the end goal of deciding the way of network bundles crosswise over network switches) since the last's development in 2011. Since 2012, nonetheless, many organizations have moved far from OpenFlow, and have grasped distinctive strategies. These incorporate Cisco Frameworks' Open Network Environment and Nicira's network virtualization stage. SD-WAN applies comparative innovation to a wide region network (WAN).For clarity, SDN is depicted in this paper with the Open Networking Establishment (ONF) [1] definition: "Programming characterized networking (SDN) is a way to deal with PC networking that permits network executives to oversee network benefits through reflection of lower-level usefulness."

SDN concentrates on four key elements:
- Division of the control plane from the information plane
- A brought together controller and perspective of the network
- Open interfaces between the gadgets in the control plane (controllers) and those in the information plane
- Programmability of the network by outside applications

Programming characterized networking has establishes in past network control frameworks, for example, RCP [5], 4D [6], and Ethane [7]. Late work has presented the thought of southbound and northbound interfaces. The southbound interface alludes to the interface and convention between programmable switches (SDN-fit switches) and the product controller. The northbound interface decides how to express operational errands and network approaches, furthermore how to make an interpretation of them into a frame the controller can get it. In Fig. 1, the convention between the controller and programmable switch layer is alluded to as southbound; northbound alludes to the upper part of the controller, including the arrangement layer. OpenFlow [3] is a standout amongst the most well-known southbound SDN interfaces. Numerous merchants, including HP, NEC, NetGear, and IBM, create OpenFlow-skilled network switches accessible in the market. The Open Networking Establishment (ONF) is in charge of standardizing the OpenFlow convention. There are an assortment of OpenFlow controllers, for instance, NOX [8], Floodlight, and Maestro. NOX is a system that permits designers to program their product program with C++ or Python, utilizing an arrangement of use programming interfaces (APIs) to cooperate with OpenFlow-fit switches, while Floodlight is a Java-based controller. Maestro concentrates on accomplishing better execution and adaptability in a brought together controller utilizing multithreading. In spite of the fact that there has been much review and modern exertion in characterizing, cleaning, and executing the southbound piece of SDN conventions, there has been generally little consideration on northbound interfaces and conventions. Procera is one push to characterize a northbound interface that gives the capacity to determine and actualize responsive approaches.

3. BACKGROUND: WHY SDN?

The central reason for the correspondence network is to exchange data starting with one point then onto the next. Inside the network the information traversed various hubs, and proficient and successful information exchange (sending) is upheld by the control gave by network applications/administrations.

4. NETWORKING: THE OLD WAY

In conventional networks, as appeared in Fig. 2, the control and information planes are joined in a network hub. The control plane is in charge of arrangement of the hub and programming the ways to be utilized for information streams. Once these ways have been resolved, they are pushed down to the information plane. Information sending at the equipment level depends on this control data. In this conventional approach, once the stream management (sending strategy) has been characterized, the best way to make a conformity to the arrangement is by means of changes to the setup of the gadgets. This has demonstrated prohibitive for network administrators who are quick proportional their networks in light of changing activity requests, expanding utilization of cell phones, and the effect of " big data."

5. NETWORKING: THE SDN WAY

From these administration focussed prerequisites, SDN has developed. Control is moved out of the individual network hubs and into the different, brought together controller. SDN switches are controlled by a Network Working Framework (NOS) that gathers data. utilizing the Programming interface appeared as a part of Fig. 2a and controls their sending plane, giving a dynamic model of the network topology to the SDN controller facilitating the applications. The controller can in this manner abuse finish information of the network to enhance stream management and bolster benefit client necessities of adaptability and adaptability. For instance, transfer speed can be powerfully distributed into the information plane from the application. In Fig. 3, once the main parcel of another stream lands at the change from the sender (step 1), the switch checks for a stream administer for this bundle in the SDN reserve (step 2). On the off chance that a coordinating section is found, the guidelines connected with the particular stream passage are executed (e.g., redesign counter, parcel/coordinate fields, activity set, metadata). Bundles are then sent to the beneficiary (stride 5). On the off chance that no match is found in the stream table, the parcel might be sent to the controller over a safe channel (step 3). Utilizing the southbound Programming interface (e.g., OpenFlow, Powers, PCEP), the controller can include, overhaul, and erase stream sections, both responsively (in light of bundles) and proactively. The controller executes the directing calculation, and adds another sending section to the stream table in the change and to each of the applicable switches along the stream way (step 4). The switch then advances the bundle to the proper port to send the parcel to the collector (step 5).

6. WHERE DOES SDN TAKE US?

SDN implementation opens up a means for new innovation and new applications. Dynamic topology control (i.e., adjusting switch usage depending on load and traffic mapping) becomes possible with the global network view. This introduces scope for network-wide access control, power management, and home networking, for which the network view is not beneficial but absolutely necessary. Furthermore, the network programmability possible in SDN allows seamless communication at all levels, from hardware to software and ultimately to end users (network operators). Programmability makes applications aware of the network and the network aware of applications. This enables greatly improved use of resources and opens up the potential for new applications with the associated potential for revenue generation (e.g., flow metering) in which cost plans can be defined based on a level of service provision.

7. THE NEED FOR A NEW NETWORK ARCHITECTURE

The blast of cell phones and substance, server virtualization, and appearance of cloud administrations are among the patterns driving the networking business to reevaluate customary network architectures.[22] Numerous ordinary networks are various leveled, worked with levels of Ethernet switches orchestrated in a tree structure. This plan appeared well and good when customer server figuring was prevailing, however such a static engineering is illsuited to the dynamic registering and capacity needs of today's undertaking server farms, grounds, and transporter situations. A portion of the key registering patterns driving the requirement for another network worldview include:

Changing Traffic Patterns

Inside the venture server farm, activity designs have changed altogether. Rather than customer server applications where the majority of the correspondence happens between one customer and one server, today's applications get to various databases and servers, making a whirlwind of "east-west" machine-to-machine movement before returning information to the end client gadget in the great "north-south" activity design. In the meantime, clients are changing network activity designs as they push for access to corporate substance and applications from a gadget (counting their own), associating from anyplace, whenever. At last, numerous undertaking server farms supervisors are pondering an utility processing model, which may incorporate a private cloud, open cloud, or some blend of both, bringing about extra movement over the wide region network.

The "consumerization of IT"

Clients are progressively utilizing portable individual gadgets, for example, cell phones, tablets, and scratch pad to get to the corporate network. IT is under weight to oblige these individual gadgets in a fine-grained way while securing corporate information and protected innovation and meeting consistence orders.

The rise of cloud services

Undertakings have excitedly grasped both open and private cloud administrations, bringing about phenomenal development of these administrations. Endeavor specialty units now need the readiness to get to applications, foundation, and other IT assets on request and individually. To add to the many-sided quality, IT's getting ready for cloud administrations must be done in a domain of expanded security, consistence, and evaluating necessities, alongside business redesigns, unions, and mergers that can change suppositions overnight. Giving self-benefit provisioning, whether in a private or open cloud, requires flexible scaling of processing, stockpiling, and network assets, in a perfect world from a typical perspective and with a typical suite of devices.

"Big data" means more bandwidth

Taking care of today's "Big data" or mega datasets requires gigantic parallel handling on a huge number of servers, all of which need guide associations with each other. The ascent of mega datasets is energizing a consistent interest for extra network limit in the server farm. Administrators of hyper scale server farm networks confront the overwhelming assignment of scaling the network to beforehand unfathomable size, keeping up any-to-any availability without becoming penniless.

8. SDN ARCHITECTURE

The accompanying chart clarifies about the SDN engineering. The accompanying rundown characterizes and clarifies the building components:[10]

SDN Application

SDN Applications are projects that expressly, straightforwardly, and automatically convey their network necessities and sought network conduct to the SDN Controller by means of a northbound interface (NBI). What's more they may devour a preoccupied perspective of the network for their inner basic leadership purposes. A SDN Application comprises of one SDN Application Rationale and at least one NBI Drivers. SDN Applications may themselves uncover another layer of preoccupied network control, consequently offering at least one more elevated amount NBIs through individual NBI specialists.

SDN Controller

The SDN Controller is a consistently incorporated element accountable for (i) interpreting the prerequisites from the SDN Application layer down to the SDN Datapaths and (ii) furnishing the SDN Applications with a conceptual perspective of the network (which may incorporate insights and occasions). A SDN Controller comprises of at least one NBI Operators, the SDN Control Rationale, and the Control to Information Plane Interface (CDPI) driver. Definition as an intelligently brought together element neither endorses nor blocks usage points of interest, for example, the league of different controllers, the various leveled association of controllers, correspondence interfaces between controllers, nor virtualization or cutting of network assets.

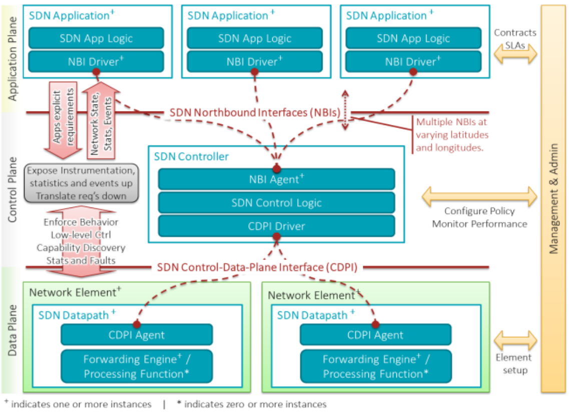
Application Plane
SDN Application⁺
SDN App Logic
NBI Driver⁺
SDN Application⁺
SDN App Logic
NBI Driver⁺
SDN Application⁺
SDN App Logic
NBI Driver⁺
Contracts SLAs
Apps explicit requirements
Network State, Stats, Events
SDN Northbound Interfaces (NBIs)
Multiple NBIs at varying latitudes and longitudes.
Control Plane
Expose Instrumentation, statistics and events up Translate req's down
SDN Controller
NBI Agent⁺
SDN Control Logic
CDPI Driver
Configure Policy Monitor Performance
Management & Admin
Enforce Behavior Low-level Ctrl Capability Discovery Stats and Faults
SDN Control-Data-Plane Interface (CDPI)
Data Plane
Network Element⁺
SDN Datapath⁺
CDPI Agent
Forwarding Engine⁺ / Processing Function*
Network Element⁺
SDN Datapath⁺
CDPI Agent
Forwarding Engine⁺ / Processing Function*
Element setup
⁺ indicates one or more instances | * indicates zero or more instances

Fig. SDN Architecture components.

SDN Datapath

The SDN Datapath is an intelligent network gadget that uncovered perceivability and uncontested control over its publicized sending and information handling abilities. The coherent representation may envelop all or a subset of the physical substrate assets. A SDN Datapath involves a CDPI operator and an arrangement of at least one movement sending motors and at least zero activity handling capacities. These motors and capacities may incorporate straightforward sending between the datapath's outer interfaces or inner activity handling or end capacities. At least one SDN Datapaths might be contained in a solitary (physical) network component—a coordinated physical blend of interchanges assets, oversaw as a unit. A SDN Datapath may likewise be characterized over numerous physical network components. This consistent definition neither endorses nor blocks usage subtle elements, for example, the sensible to physical mapping, management of shared physical assets, virtualization or cutting of the SDN Datapath, interoperability with non-SDN networking, nor the information handling usefulness, which can incorporate OSI layer 4-7 capacities.

SDN Control to Data-Plane Interface (CDPI)

The SDN CDPI is the interface characterized between a SDN Controller and a SDN Datapath, which gives in any event (i) automatic control of all sending operations, (ii) abilities commercial, (iii) insights reporting, and (iv) occasion notice. One estimation of SDN lies in the desire that the CDPI is executed in an open, seller impartial and interoperable way.

SDN Northbound Interfaces (NBI)

SDN NBIs are interfaces between SDN Applications and SDN Controllers and ordinarily give dynamic network sees and empower coordinate articulation of network conduct and prerequisites. This may happen at any level of deliberation (scope) and crosswise over various arrangements of usefulness (longitude). One estimation of SDN lies in the desire that these interfaces are actualized in an open, merchant nonpartisan and interoperable way.

9. SECURITY USING THE SDN PARADIGM

SDN engineering may empower, encourage or upgrade network-related security applications because of the controller's focal perspective of the network, and its ability to reconstruct the information plane whenever. While security of SDN design itself remains an open question that has as of now been concentrated a few circumstances in the examination community,[21][22][23] the accompanying sections just concentrate on the security applications made conceivable or returned to utilizing SDN.

A few research takes a shot at SDN have as of now explored security applications based upon the SDN controller, in view of various points. Circulated Foreswearing of Administration (DDoS) identification and mitigation,[24][25] and also botnet[26] and worm engendering, are some solid utilize instances of such applications: fundamentally, the thought comprises in intermittently gathering network insights from the sending plane of the network in a standardized way (e.g. utilizing Openflow), and after that apply arrangement calculations on those measurements to identify any network irregularities. In the event that an irregularity is identified, the application trains the controller how to reinvent the information plane to moderate it.

Another sort of security application influences the SDN controller by executing some moving target safeguard (MTD) calculations. MTD calculations are normally used to make any assault on a given framework or network more troublesome than expected by occasionally covering up or changing key properties of that framework or network. In customary networks, actualizing MTD calculations is not an unimportant assignment since it is hard to construct a focal power capable of deciding - for every part of the framework to be ensured - which key properties are stowed away or changed. In a SDN network, such assignments turn out to be clearer on account of the centrality of the controller. One application can for instance occasionally appoint virtual IPs to have inside the network, and the mapping virtual IP/genuine IP is then performed by the controller. Another application can mimic some fake opened/shut/separated ports on arbitrary has in the network with a specific end goal to include huge clamor amid surveillance stage (e.g. filtering) performed by an attacker. [29]

Extra esteem with respect to security in SDN empowered networks can likewise be picked up utilizing FlowVisor[30] and FlowChecker[30] separately. The previous tries to utilize a solitary equipment sending plane sharing different isolated coherent networks. Taking after this approach a similar equipment assets can be utilized for creation and improvement purposes and in addition isolating observing, setup and web activity, where every situation can have its own intelligent topology which is called cut. In conjunction with this approach FlowChecker[30] understands the approval of new OpenFlow decides that are conveyed by clients utilizing their own particular cut.

SDN controller applications are generally conveyed in extensive scale situations, which requires exhaustive checks of conceivable programming blunders. A framework to do this called Decent was depicted in 2012. Presenting an overall security design requires a far reaching and extended way to deal with SDN. Since it was presented, planners are taking a gander at conceivable approaches to secure SDN that don't trade off adaptability. One engineering called SN-SECA (SDN+NFV) Security Design.[33]

10. SDN APPLICATIONS

Programming defined networking has applications in a wide assortment of networked situations. By decoupling the control– and information planes, programmable networks empower redid control, a chance to dispense with middle boxes, and additionally simplified improvement and arrangement of new network administrations and conventions. Underneath, we list diverse situations for which SDN arrangements can be proposed or actualized.

1. Enterprise Networks
2. Data Focuses
3. Infrastructure-based Remote Get to Networks
4. Optical Networks
5. Home and Private Company

11. CONCLUSION

In this paper, we gave a review of programmable networks and, in this specific circumstance, inspected the developing field of Programming Defined Networking (SDN). We take a gander at the historical backdrop of programmable networks, from early thoughts until late advancements. Specifically we portrayed the SDN engineering in detail and additionally the Open Flow [2] standard. We introduced current SDN usage and testing stages and analyzed network administrations and applications that have been produced in light of the SDN worldview. We finished up with a dialog of future bearings empowered by SDN going from support for heterogeneous networks to Information Centric Networking (ICN).

12. REFERENCES

[1] ONF, "Software-Defined Networking: The New Norm for Networks," white paper, https://www.opennetworking.org

[2] A. Doria, J. Hadi Salim, R. Haas, H. Khosravi, W. Wang, L. Dong, R. Gopal, and J. Halpern. Forwarding and Control Element Separation (ForCES) Protocol Specification. RFC 5810 (Proposed Standard), March 2010.

[3] N. McKeown et al., "OpenFlow: Enabling Innovation in Campus Networks," ACM Comp. Commun. Rev., Apr. 2008.

[4] Open Networking Research Center (ONRC). http://onrc.net.

[5] N. Feamster et al., "The Case for Separating Routing from Routers," ACM SIGCOMM Wksp. Future Directions in Network Architecture, Portland, OR, Sept. 2004.

[6] A. Greenberg et al., "A Clean Slate 4D Approach to Network Control and Management," ACM Comp. Commun. Rev., vol. 35, no. 5, 2005, pp. 41–54.

[7] M. Casado et al., "Rethinking Packet Forwarding Hardware," Proc. 7th ACM SIGCOMM HotNets Wksp., Nov. 2008.

[8] N. Gude et al., "NOX: Towards an Operating System for Networks," ACM SIGCOMM Computer Commun. Rev., vol. 38, no. 3, July 2008, pp. 105–10.

[9] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 38(2):69–74, 2008.

[10] "SDN Architecture Overview" (PDF). Openetworking.org. Retrieved 22 November2014

[11] Elizabeth Miller Coyne (23 September 2016). "Huawei Exec: SDN's Become a 'Completely Meaningless Term'". Light Reading. Retrieved 25 September 2016.

[12] "Software-Defined Networking (SDN) Definition". Opennetworking.org. Retrieved26 October 2014.

[13] "WhitePapers". Opennetworking.org. Retrieved 26 October 2014.

[14] "SDNArchitectureOverview" (PDF). Openetworking.org . Retrieved 22 November2014.

[15] "OpenFlow: Proactive vs Reactive". NetworkStatic.net. Retrieved 2014-07-01.

[16] "Reactive, Proactive, Predictive: SDN Models | F5 DevCentral". Devcentral.f5.com. Retrieved 2016-06-30.

[17] Liyanage, Madhusanka (2015). Software Defined Mobile Networks (SDMN): Beyond LTE Network Architecture. UK: John Wiley. pp. 1–438. ISBN 978-1-118-90028-4.

[18] Jose Costa-Requena, Jesús Llorente Santos, Vicent Ferrer Guasch, Kimmo Ahokas, Gopika Premsankar, Sakari Luukkainen, Ijaz Ahmed, Madhusanka Liyanage, Mika Ylianttila, Oscar López Pérez, Mikel Uriarte Itzazelaia, Edgardo Montes de Oca, SDN and NFV Integration in Generalized Mobile Network Architecture , in Proc. of European Conference on Networks and Communications (EUCNC), Paris, France. June 2015.

[19] Madhusanka Liyanage, Mika Ylianttila, Andrei Gurtov, Securing the Control Channel of Software-Defined Mobile Networks , in Proc. of IEEE 15th International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM), Sydney, Australia. June 2014.

[20] "SD-WAN: What it is and why you'll use it one day". networkworld.com. 2016-02-10. Retrieved 2016-06-27.

[21] Kreutz, Diego; Ramos, Fernando; Verissimo, Paulo (2013). "Towards secure and dependable software-defined networks". Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking. pp. 50–60.

[22] Scott-Hayward, Sandra; O'Callaghan, Gemma; Sezer, Sakir (2013). "SDN security: A survey". Future Networks and Services (SDN4FNS), 2013 IEEE SDN for. pp. 1–7.

[23] Benton, Kevin; Camp, L Jean; Small, Chris (2013). "Openflow vulnerability assessment".Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking. pp. 151–152.

[24] Giotis, K; Argyropoulos, Christos; Androulidakis, Georgios; Kalogeras, Dimitrios; Maglaris, Vasilis (2014). "Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments". Computer Networks.62: 122–136.

[25] Braga, Rodrigo; Mota, Edjard; Passito, Alexandre (2010). "Lightweight DDoS flooding attack detection using NOX/OpenFlow". Local Computer Networks (LCN), 2010 IEEE 35th Conference on. pp. 408–415.

[26] Feamster, Nick (2010). "Outsourcing home network security". Proceedings of the 2010 ACM SIGCOMM workshop on Home networks. pp. 37–42.

[27] Jin, Ruofan & Wang, Bing (2013). "Malware detection for mobile devices using software-defined networking". Research and Educational Experiment Workshop (GREE), 2013 Second GENI. 81-88.

[28] Jafarian, Jafar Haadi; Al-Shaer, Ehab; Duan, Qi (2012). "Openflow random host mutation: transparent moving target defense using software defined networking".Proceedings of the first workshop on Hot topics in software defined networks. pp. 127–132.

[29] Kampanakis, Panos; Perros, Harry; Beyene, Tsegereda. SDN-based solutions for Moving Target Defense network protection (PDF). Retrieved 23 July 2014.

[30] Sherwood, Rob; Gibb, Glen; Yap, Kok-Kiong; Appenzeller, Guido; Casado, Martin; McKeown, Nick; Parulkar, Guru (2009). "Flowvisor: A network virtualization layer".OpenFlow Switch Consortium, Tech. Rep.

[31] Al-Shaer, Ehab & Al-Haj, Saeed (2010). "FlowChecker: Configuration analysis and verification of federated OpenFlow infrastructures". Proceedings of the 3rd ACM workshop on Assurable and usable security configuration. pp. 37–44.

[32] Canini, Marco; Venzano, Daniele; Peresini, Peter; Kostic, Dejan; Rexford, Jennifer; et al. (2012). A NICE Way to Test OpenFlow Applications. NSDI. pp. 127–140.

[33] Bernardo and Chua (2015). Introduction and Analysis of SDN and NFV Security Architecture (SA-SECA). 29th IEEE AINA 2015. pp. 7