A Survey and a Comparative Study on Software-Defined Networking

2 authors:


Mohammed Al Salam
University of Technology, Iraq
33 PUBLICATIONS **17** CITATIONS


Afrah Salman Dawood
University of Technology, Iraq
4 PUBLICATIONS **2** CITATIONS

International Research Journal of Computer Science (IRJCS)
Issue 08, Volume 3 (August 2016)
ISSN: 2393-9842
www.irjcs.com

A Survey and a Comparative Study on Software-Defined Networking

Afrah Salman Dawood*, Mohammed Najm Abdullah*
Department of Computer Engineering, University of Technology, Baghdad, Iraq

Abstract — ***Software-Defined Networking (SDN) is a new networking paradigm in network technologies in which the data plane and network plane are separated. This new technology began to be widely used in last few years and studies started to increase on this new technology. The communication scheme of the network consists mainly of the controller and programmable OpenFlow switches. The centralized controller can be considered as the brain of the network which is basically responsible on determining the path of the incoming packet by informing the switches to route that packet in the right direction. In this paper, a complete definition and description have been shown and a survey study on the SDN has been presented based on different types of studies have been already made historically. An overview on the simulators used to implement these networks has also been shown.***

Keywords— Software-Defined Networking (SDN), programmable network, data plane, control plane, OpenFlow switches, controller, SDN simulators.

I. INTRODUCTION

Computer networks are implemented for widely different purposes ranging from small office connectivity to very large organizations distributed around the world. The networks are the basic links for organizations to be connected and they could be completely on-premises, cloud-based, or a hybrid of both [17]. These networks are built from a large number of different networking devices including routers, switches, middle-boxes and PCs [3]. There are numerous architectures of the networks. The old fashions of networks rely on the distribution of control and transport network protocols among routers and switches [8]..

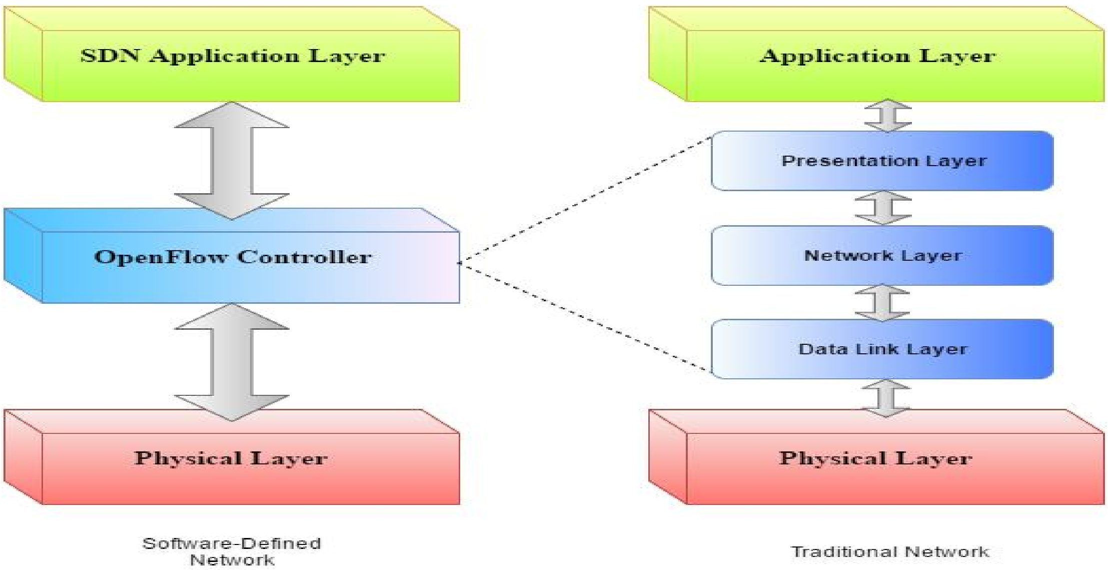

Fig. 1 General Comparison between Traditional Networks and Software-Defined Networks

Figure 1 shows a comparative view between traditional networks and software-defined networks. The distribution of control and data plane has its own advantages and disadvantages. This section will focus on the disadvantages and limitations that led to the movement from distributed control to the centralized control.

The basic limitation of the traditional IP networks is the complexity and difficulty of the management and configuration distributed around almost all devices (i.e. the configuration manual must be implemented throughout all network devices) where the designers must identify the structure and the source code of the software running on switches so it cannot be easily modified; this phenomenon is known as network (or internet) ossification [1] and [9]. Another disadvantage is the coupling of the data plane and the control plane inside the networking devices. There are also the problem of the internet ossification where the internet has become very difficult to blossom in both the physical infrastructure and the protocols and performance [3]. Generally, when the routing device receives the packet, it uses a set of rules contained in its firmware to decide the routing path and the destination device for the received packet [6]. In recent years, an attempts to overcome these problems have been made through the use of a new paradigm in networking architectures by using the Software-Defined Networking (SDN). There are many definitions of SDN, the most common one is that software applications and SDN controllers control networks instead of traditional network management consoles and commands that are difficult to be managed and controlled [17]. This means that the application plane, control (i.e. network) plane and the data plane will be decoupled by directly programming and the interface will be opened between the controller and the forwarding element for communication [1], [2], [4], [14] [8], [18], [16] and [5]. Recently, SDN became a popular new trend in both the academic and industrial fields [2]. The basic architecture of software-defined networking can be viewed in Figure 2.

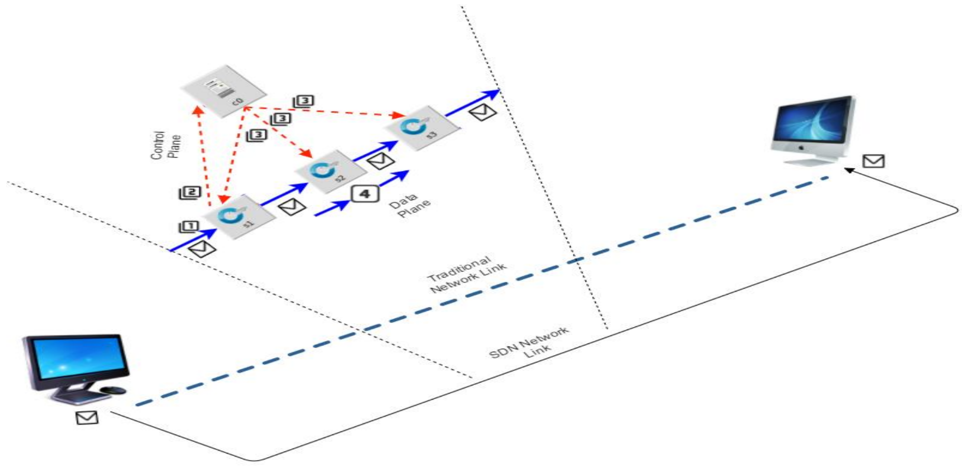

Fig. 2 The basic working of the Software-Defined Networking

II. BASICS OF SOFTWARE-DEFINED NETWORKING

In this section an overview on the SDN including background, definition, benefits and challenges will be described.

A. BACKGROUND

SDN has been presented as a solution to the significant limitations of the traditional networks which includes complexity, inconsistent policies and inability to scale. During the early appearance of the SDN on the technological field, there were stern ideas about the design and architecture of the SDN. The SDN focuses on being a platform capable of hosting a plentiful of IT workflow automation solutions that derive customers to their aim [17]. The SDN is a brand-new technology for networks; it is growing very fact due to its benefits, yet it has some challenges need to be solved for better performance [3].

B. DEFINITION OF SDN

While we are moving from traditional networks to programmed networks, many definitions have been occurred to describe SDN according to different applications and implementations. Mainly, SDN means the automatic and dynamic control and management for large number of network devices, services, traffic paths, etc.

According to [8], there are four distinct definitions which are:

1) *The most general definition is that the software-defined network is a networking scheme that discriminates the control plane form the forwarding plane [16] and this will lead to a simple (packet) forwarding elements [8].*
2) *The forwarding decisions are flow based instead of destination based [8], like that in traditional networks where the packet is forwarded hop-by-hop based on the destination IP address in its header field. The controller in SDN is responsible for deciding the path that the packet should follow to attain its destination.*
3) *The external controller (or a network OS with applications) is the brain of the SDN where it is liable on the control logic. The NOS is a key software element of the SDN network. It controls the network infrastructure components and network data flows. The OS of the SDN network determines features like performance, scalability and reliability.*
4) *SDN is a dynamically programmable network through software applications running on the external controller [8]. The network programmability is made possible through the decoupling approach, where Active Networking (AN) and Open Signaling (OpenSig) are the main approaches [1].*

In traditional networks, each switching element is responsible on packet forwarding logic based on rules specified in its own local software [1]. This scheme describes the decentralization of the control plane. On the other hand, the packet forwarding logic is accomplished in a centralized manner where the SDN controller is responsible for the decision making process of the path selection.

C. Benefits of SDN

This subsection describes the types of benefits provided by SDN to organizations [17]:

1) *SDN Automation Leads to Business Agility: The lightness and punctuality of the business objectives are achieved by greater degrees of infrastructure automation [17].*
2) *A New Approach to Network Policy: Business requirements is not aware in how the network is achieved, however, the way of application working is essentially relative to high-level business policies and objectives [17].*
3) *SDN provides better techniques for centralized dynamic management and control configurations for an improved automation, scalability and consistency [17] [18].*
4) *SDN provides adaptive resource management and control for simplifying the industrial and research communities [19].*

III. SDN AFFAIR AND RELATED RESEARCH DIRECTIONS

Several research trends in SDN have been implemented based on various aspects. Most researches focus on the layered taxonomy as those in [2] [3] [8] etc. In this section, an analytical survey and a comparative study will be produced based on the software and hardware trends. Software-Defined Networking is a radical new technology for network implementation and can be implemented according to various concepts. Figure 3 shows an overview of the classification of the basic reviewed research trends based on the proposed taxonomy.

The core of the SDN network, the programmable switch, can conduct as a router, switch, firewall, load balancer, etc. depending on polices of the controller application [8], though researches have been widely varied. Table I shows a comparative study related to the classification mentioned earlier in Figure 1. Resource management, energy consumption, storage, security, programming and interfacing, probability of error and several other factors are very important issues for the performance evaluation in almost all networks, however organization errands and strategy particulars are additionally critical components in SDN systems since the systems develop in size steadily and turn out to be substantially more unpredictable to oversee and keep up [17]. As for software-defined networking, the process of managing resources still a difficult issue since most researches focuses on the architecture or on implementing the OpenFlow protocol on the programmable switches. In the rest of the section we will describe factors that should be considered when designing such networks.

A. Resource Management: Resource management is the way toward utilizing organization's assets as a part of the most effective way imaginable. These assets incorporate substantial assets, monetary assets and work assets, for example, HR. Asset administration has thoughts, for example, ensuring that one has enough physical or programming assets for the one's organizations. The heterogeneous nature of the applications, advances and equipment that today's systems need to bolster has made the administration of such frameworks a complex task [43]. The Software-Defined Networking (SDN) worldview has risen as a promising answer for lessen this many-sided quality through the production of a brought together control plane free of particular vendor equipment. However, planning a SDN-based answer for system asset administration raises a few difficulties as it ought to display adaptability as in [6] [15] [11] and [35] scalability as in [11] [15] [25] [27] [28] and [35], and flexibility as in [6] [7] [11] [15] [27] [28]. According to table I, network resource management can be described best in several researches such as [6] where the chairmen have a remote control over the system and can change the system attributes, for example, administrations and availability taking into account the workload designs.

International Research Journal of Computer Science (IRJCS) ISSN: 2393-9842
Issue 08, Volume 3 (August 2016) www.irjcs.com

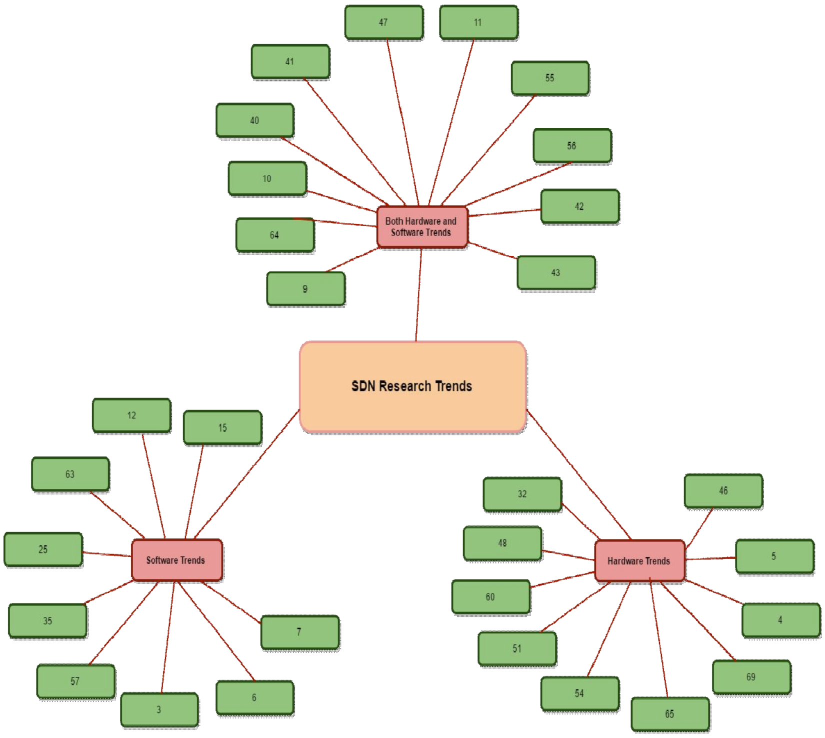

Fig. 3 SND Research Trends

B. ENERGY CONSUMPTION: Energy consumption [7] [11] [31] [38] is very critical issue to be considered while designing both traditional networks and Software-Defined Networks. To save energy, it is axiomatic to switch off the hardware to lessen traffic load; and since traditional networks combine the data plane and the control plane in the same devices then traffic load can't be evaluated correctly as they are running insensible of the traffic extent. Several energy-efficient algorithms have been proposed and implied in most cases to adapt resource usage [45] and some SDN researches focused basically on this concept like that in [44] where the outcomes demonstrated a recovery of up to 45% of the vitality utilization at evening time. Another example on energy consumption is the design explained in [45] of new energy-aware protocols and their deployment in SDNs. Specifically, both algorithmic and functional challenges considering diverse sorts of systems including spine, data center, framework based remote access (Cellular/Wi-Fi) or venture systems has been addressed.

C. STORAGE: Storage and memory units [7] [23] [37] [38] are very important issues especially in switches because they limit the number of flow entries allowed on switches. Some researchers focus on controlling the available memory on the network such as that in [46] where many switch/hardware challenges including storage have been discussed. The future managers still need to guarantee the greatest flow table size will fit their requirements. This hardware limitation still needs to be addressed and improved.

D. SECURITY: System Security is the way toward taking physical and programming precaution measures to shield the fundamental systems administration foundation from unapproved access, abuse, glitch, change, obliteration, or disgraceful revelation, in this manner making a protected stage for PCs, clients and projects to play out their allowed basic capacities inside a safe domain [47]. Over long time security made big effects on network design and with moving towards Software-Defined Networks it still very important issue. The benefits of SDN offers several new threats that need to be handled [46]. These threats include forged or faked traffic flows [5] [6] [9] [11] [15] [31] [33] susceptibilities in switches [4], [15], attacks on control plane communications [6] [11] [15] susceptibilities in controllers [4] [6] [15] [28] controller and application trust [6] [15] [31] [36] susceptibilities in administrative stations [5] [15] lack of forensics and remediation [15] and several others.

E. PROGRAMMING AND INTERFACING: SDN/OpenFlow programming languages have been studied in some projects [5] [6] [7] [10] [12], etc. as shown table I. The possibility of "programmable systems" has been proposed as an approach to encourage system development [3]. Different programming languages and interfacing are the most widely researched trends in these new technologies as SDN means that networks are programmed.

F. Programmable networking efforts [3] have been evolved during time and they include open signaling, active networking, DCAN, 4D project, NETTCONF and ethane. Interfacing in Software-Defined Networks is basically divided into southbound API, northbound API and SDN controller [11].

G. Probability of Error: It is the expectation value of the bit error ration. It is up to the OpenFlow protocol to dole out communicate to the focal controller for flow setup. Along these lines, the low-level switches need to communicate with the controller as often as possible to acquire instructions on the most proficient method to handle approaching packets, thus there is a possibility of errors [6]. Many methods have been discussed to enhance the performance of the controller to avoid errors. Not too many researches focus on the point of the probability of error; the meaning of this point almost found in [22] [23] [24] [35].

H. Policy and privacy Specification: Policies [6] [11] [12] [23] [24] [28] [34] [36] are the set of rules that directs the work flow of the network. Each policy is a set of conditions and a set of corresponding actions [5]. Policies are classified as static or dynamic according to the set of actions been fixed or dynamic. Dynamic updates are described in [5] [7] [10] and several others. Too many studies found on privacy and policy specifications as shown in table I; for example, in [3], a description on reactive and proactive policies has been discussed with different examples.

I. Performance Evaluation: Performance evaluation depends on several metrics and almost every network must be evaluated according to a set of specified metrics. These metrics are sensitive to the changes of network performance and efficiency of the design. With SDN, performance evaluation takes into account metrics like packet throughput [21] [32] forwarding probability [22] [30] Message Delivery Ratio (MDR) [29] rule activation time [30] control traffic overhead [30] and several other metrics.

J. Implementing Applications Using SDN: All traditional networks and Software-Defined networks are useless without using them in useful applications. SDN has applications in a wide assortment of networked environments [3] by decoupling the data plane from the control plane. Several researches showed how to implement different applications environments like data centers, enterprise networks, etc. using this brand new networking technology, referring to table I. More information can be found in [7] [9] [10] [11] [13] and several other researches.

TABLE I - A COMPARATIVE SURVEY ON SDN RESEARCHES

Grouping	Reference	Addressed Issue									Proposed Objectives and Solutions
		Resource Management and Monitoring	Energy Consumption	Storage	Security	Programming and Interfacing	Probability of Error	Policy and privacy Specification	Performance Evaluation	Implementing Applications using SDN	
Hardware Trends	Yan Luo, *et al* [21]	✓	✗	✗	✗	✓	✗	✗	✓	✗	The design options have been described and experiment results have been reported show that a 20% reduction on packet delay and the comparable packet forwarding throughput compared to conventional designs.
	B. Rais, *et al* [29]	✓	✗	✗	✗	✓	✗	✓	✓	✗	Through broad reenactments, advantages of MeDeHa++ have been illustrated, particularly as far as the amplified scope it gives and also its capacity to adapt to discretionarily enduring network disturbances. Another vital commitment of this work is to send and assess message conveyance structure on a genuine system testbed and additionally lead tests in "half and half" situations running somewhat on recreation and mostly on genuine hubs.

	Sushant Jain, *et al* [35]	✓	✗	✗	✗	✗	✓	✗	✓	✗	The design, implementation, and evaluation of B4, a private WAN connecting Google's data centers across the planet, have been presented. The objective function of the proposed system is to deliver max-min fair allocation to applications.
	P. Dely, *et al* [30]	✓	✗	✗	✗	✗	✗	✗	✓	✗	A simple solution has been implemented to solve the issue of customer portability in a WMN which handles the quick movement of customer locations between Mesh APs and the collaboration with re-steering without the requirement for burrowing.
	M. Mendonca, *et al* [31]	✓	✓	✓	✓	✗	✗	✗	✗	✗	In this paper, spurred by the vision that future online worlds will include infrastructure–based and infrastructure–less systems, the creators investigate the utilization of the Software–Defined Networking (SDN) worldview in these so–called "heterogeneous" arranged situations.
	Pat Bosshart, *et al* [38]	✓	✓	✓	✗	✗	✗	✗	✗	✗	The solid outline illustrates, in spite of worries inside the group, that adaptable OpenFlow equipment switch executions are achievable at no extra cost or power.
	White Paper [42]	✓	✗	✗	✗	✗	✗	✗	✓	✗	This white paper investigates SDN and NFV with an accentuation on the advantages, use cases and difficulties that must be overcome to push ahead.
	Nishtha, *et al* [4]	✓	✗	✗	✓	✗	✗	✗	✗	✗	This paper concentrates on a large portion of the issues that exists in SDNs and OpenFlow
	Celio Trois, *et al* [5]	✗	✗	✗	✓	✓	✗	✓	✓	✓	This paper introduces a pragmatic view on up-to-dated OpenFlow-based SDN languages. The methodology depends on a scientific categorization including every single noticeable element found in those dialects. Cases are talked about to show the crucial deliberations. In conclusion, all assembled data is condensed, talking about the principle progressing research endeavors and challenges.
	Marcelo R. Nascimento, *et al* [27]	✓	✗	✗	✗	✗	✗	✗	✗	✗	RouteFlow methodology was proposed, a novel point to in the configuration space of product directing arrangements with broad usage towards virtual switches and IP systems as an administration.
Software Trends	Yuefeng Wang, Ibrahim Matta [12]	✓	✗	✗	✗	✓	✗	✓	✗	✗	General regular architecture for SDN system and configuration prerequisites of the administration layer that is at the center of the design have been recognized. The open issues and shortcoming of existing SDN administration layers have likewise been distinguished.
	Technical white paper [36]	✗	✗	✗	✓	✓	✗	✓	✗	✓	This white paper gives a review of programming characterized systems administration and how HP is utilizing SDN to convey the Virtual Application Networks technique.

	Qiao Yan, *et al* [15]	✓	✗	✗	✓	✗	✗	✗	✓	✓	New patterns and attributes of DDoS attacks on SDN and how to make full utilization of SDN's focal points to crush DDoS attacks in distributed computing situations, how to keep SDN itself from turning into a casualty of DDoS attacks and available solutions have been examined.
	Nick McKeown, *et al* [20]	✓	✗	✗	✗	✓	✗	✗	✓	✗	The objective is to permit analysts to assess their thoughts in true activity setting, support sending of OpenFlow in proposed vast scale testbeds like GENI as a helpful grounds part.
	M. Jarschel, *et al* [22]	✗	✗	✗	✗	✗	✓	✗	✓	✗	This paper was the initial move towards OpenFlow versatility and execution. Essential model was inferred for sending velocity and blocking likelihood of an OpenFlow switch joined with an OpenFlow controller and accepting it utilizing a reproduction.
	N. Foster, *et al* [34]	✓	✗	✗	✗	✓	✗	✓	✗	✗	The authors outlined a straightforward and instinctive reflections to program to the three principle phases of system administration: observing system activity, indicating and forming parcel sending approaches, and redesigning strategies reliably.
	Bruno Astuto A. Nunes, *et al* [3]	✓	✗	✗	✗	✓	✗	✓	✗	✓	A notable point of view of programmable systems was given. At that point the SDN architecture and the OpenFlow standard were displayed.
	Fei Hu, *et al* [6]	✓	✗	✗	✓	✓	✓	✓	✓	✓	This overview can help both industry and the educated community research and development individuals to comprehend the most recent advancement of SDN/OpenFlow plans. Some critical unsolved examination issues have likewise been pointed in this energizing field.
	Wenfeng Xia, *et al* [7]	✓	✓	✓	✗	✓	✗	✓	✓	✓	This paper studies most recent advancements in this dynamic exploration territory of SDN and its design.
Both Trends	Adrian Lara, *et al* [9]	✓	✗	✗	✓	✗	✗	✗	✓	✓	A study has been given about OpenFlow and difficulties confronting the expansive scale arrangement of OpenFlow-based systems and no usage gave.
	The Open SDN Architecture [37]	✓	✗	✗	✗	✓	✗	✗	✗	✓	This white paper provides a study about Big Switch Networks Open SDN Suite architecture that provides unmatched network agility, choice in network hardware, and optimized network operations.
	Raj Jain and Subharthi Paul [10]	✗	✗	✗	✗	✓	✗	✓	✗	✓	The vision is to plan another session-layer reflection called OpenADN that permits ASPs to express and authorize application activity administration approaches and application conveyance requirements at the granularity of use messages and parcels.
	T. Koponen, *et al* [23]	✓	✗	✓	✗	✓	✓	✗	✓	✓	Onix was presented to address the control paradigm.

	Tootoonchian and Y. Ganjali. Hyperflow [24]	✗	✗	✗	✗	✗	✓	✓	✓	✓	Preparatory assessment demonstrates that, accepting adequate control data transmission, to bind the window of irregularity among controllers by a variable of the deferral between the most remote controllers, the system changes must happen at a rate lower than 1000 occasions for every second over the system.
	Christian Esteve Rothenberg, *et al* [28]	✓	✗	✗	✓	✓	✗	✓	✗	✗	A controller-driven hybrid administration model and present the configuration of the RFCP along the model usage of an all-inclusive unique BGP steering administration were proposed in this paper.
	Daniel F. Macedo, *et al* [11]	✓	✓	✗	✓	✓	✗	✓	✓	✓	The authors endorse the convergence of the (SDN, SDR and virtualization) technologies. They portray programmable systems, where programmable gadgets execute particular code, and the system is isolated into three planes: information, control, and administration planes. They close with last contemplations, open issues and future difficulties.
	A. Detti, *et al* [32]	✓	✗	✗	✗	✗	✗	✗	✓	✗	The authors proposed a solution to integrate SDN functionality in a Wireless Mesh, trying to face the reliability concerns related to this environment. The proposed wmSDN approach integrates "ready-to-market" technologies.
	white paper [33]	✓	✗	✗	✓	✓	✗	✗	✗	✗	This white paper presents the Cisco perspective on SDN network programming and several Cisco products have OpenFlow-capable images available.
	Soheil Hassas Yeganeh and Yashar Ganjali. Kandoo [25]	✓	✗	✗	✗	✓	✗	✗	✓	✓	Kandoo methodology was stretched out to bolster new classifications of control applications that are not as a matter of course neighborhood but rather that have a constrained extension.
	Bob Lantz, *et al* [26]	✗	✗	✗	✗	✓	✗	✗	✓	✗	Mininet-based contextual analyses winnowed from more than 100 clients at 18 organizations, who have created SDN.

IV. SDN SOFTWARE SIMULATORS

As there are several simulation programs that support traditional networks, software-defined networks came up with several simulators (or emulators) to support different architectures implemented with SDN. Basically, there two basic simulators available for such networks which are NS3 and Mininet. Both Mininet and NS3 are equally good SDN simulators available currently for implementation. While the usage varies widely and the choices are to be made according to requirements. If OpenFlow is the most significant part of the work then it is better to use Mininet. Otherwise if checking large network behavior is as important as OpenFlow, go for NS3. Mininet can be considered far easier than NS3 but NS3 gives more control.

A. NS3 [39]: it supports OpenFlow switches which are restricted to be simulation only. OpenFlow switches are configurable via the OpenFlow API, and also have an MPLS extension for quality-of-service and service-level-agreement support. By extending these capabilities to ns-3 for a simulated OpenFlow switch that is both configurable and can use the MPLS extension, NS3 simulations can accurately simulate many different switches.

B. MININET [40]: Network emulation software that allows you to launch a virtual network with switches, hosts and an SDN controller all with a single command. Mininet supports research, development, learning, prototyping, testing, debugging, and any other tasks that could benefit from having a complete experimental network on a laptop or other PC.
Mininet creates a realistic virtual network, running real kernel, switch and application code, on a single machine (VM, cloud or native), in seconds, with a single command. Some of characteristics that guide the creation of Mininet are flexibility, applicability, interactivity, scalability, realistic, and share-able prototypes with other collaborators [41].

C. **CHALLENGES WITH MENTIONED SIMULATORS:** Basically, in NS3 we have an OpenFlow switch model that does not speak the actual switch - controller protocol, but instead, it talks to a sole object that implements the controller behavior. There have been discussions around fixing this so that one may run a controller inside a VM, connect the VM to an NS3 node using a tap-bridge device, and then run NS3 in emulation mode. This would allow switching the controller logic from simulation to emulation and then to actual ordeal. In Mininet-based systems, the CPU or data transfer capacity accessible on a solitary server can't (presently) be surpass. Non-Linux-compatible OpenFlow switches or applications likewise can't (at present) be run; in any case, this has not been a noteworthy issue by and by.

REFERENCES

[1] F. A. Lopes, M. Santos, R. Fidalgo, and S. Fernandes, “A Software Engineering Perspective on SDN Programmability,” IEEE Communications Surveys & Tutorials, vol. 18, no. 2, second quarter 2016.
[2] J. Yosr, M. Taous, and D. Mourad, “A Survey and a Layered Taxonomy of Software-Defined Networking,” IEEE Communication Surveys & Tutorials, vol. 16, no. 4, fourth quarter 2014.
[3] Y. Jarraya, T. Madi, and M. Debbabi, “A Survey of Software-Defined Networking past Present and Future of Programmable Networks,” IEEE Communications Surveys & Tutorials, vol. 16, no. 3, third quarter 2014.
[4] Nishtha and M. Sood, “A Survey on Issues of Concern in Software Defined Networks,” 2015 Third International Conference on Image Information Processing.
[5] C. Trois, M. D. Del Fabro, L. C. E. de Bona, and M. Martinello, “A Survey on SDN Programming Languages towards a Taxonomy,” This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2016.2553778, IEEE Communications Surveys & Tutorials.
[6] F. Hu, Q. Hao, and K. Bao, “A Survey on Software-Defined Network and OpenFlow from Concept to Implementation,” IEEE Communication Surveys & Tutorials, vol. 16, no. 4, fourth quarter 2014.
[7] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, “A Survey on Software-Defined Networking,” IEEE Communication Surveys & Tutorials, vol. 17, no. 1, first quarter 2015.
[8] J. Esch, “Software-Defined Networking: A Comprehensive Survey,” Proceedings of the IEEE | Vol. 103, No. 1, January 2015.
[9] A. Lara, A. Kolasani, and B. Ramamurthy, “Network Innovation using OpenFlow A Survey,” IEEE Communications Surveys & Tutorials, vol. 16, no. 1, first quarter 2014.
[10] R. Jain and S. Paul, “Network Virtualization and Software Defined Networking for Cloud Computing a Survey,” IEEE Communications Magazine • November 2013.
[11] D. F. Macedo, D. Guedes, L. F. M. Vieira, M. A. M. Vieira, and M. Nogueira, “Programmable Networks from Software-Defined Radio to Software-Defined Networking,” IEEE Communication Surveys & Tutorials, vol. 17, no. 2, second quarter 2015.
[12] Y. Wang and I. Matta, “SDN Management Layer Design Requirements and Future Direction,” 2014 IEEE 22nd International Conference on Network Protocols.
[13] T. Chen, M. Matinmikko, X. Chen, X. Zhou, and P. Ahokangas, “Software Defined Mobile Networks Concept Survey and Research Directions,” IEEE Communications Magazine • November 2015.
[14] Y. Li and M. Chen, “Software-Defined Network Function Virtualization A Survey,” 2015 IEEE. Translations and content mining are permitted for academic research only.
[15] Q. Yan, F. R. Yu, Q. Gong, and J. Li, “Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments A Survey Some Research Issues and Challenges,” IEEE Communications Surveys & Tutorials, vol. 18, no. 1, first quarter 2016.
[16] A. Blenk, A. Basta, M. Reisslein, and W. Kellerer, “Survey on Network Virtualization Hypervisors for Software Defined Networking,” IEEE Communications Surveys & Tutorials, vol. 18, no. 1, first quarter 2016.
[17] B. Underhahl and G. Kinghorn, “Software Defined Networking for Dummies A Wiley Brand,” Cisco Special Edition, Copyright c 2015 by John Wiley & Sons, Inc., Hoboken, New Jersey.
[18] A. Lara and B. Ramamurthy, “OpenSec: Policy-based Security Using Software-defined Networking,” This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2016.2517407, IEEE Transactions on Network and Service Management.
[19] D. Tuncer, M. Charalambides, S. Clayman, and G. Pavlou “Adaptive Resource Management and Control in Software Defined Networks,” IEEE Transactions on Network and Service Management, vol. 12, no. 1, first quarter, 2015.
[20] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “OpenFlow: enabling innovation in campus,” ACM SIGCOMM Computer Communication Review, Volume 38, Number 2, April 2008.

[21] Y. Luo, P. Cascon, E. Murray, and J. Ortega, “Accelerating OpenFlow switching with network processors,” In Proc. 5th ACM/IEEE Symp. On Architectures for Networking and Commun. Syst., ANCS ’09, New York, NY, USA, 2009. ACM.
[22] M. Jarschel, S. Oechsner, D. Schlosser, R. Pries, S. Goll, and P. Tran- Gia, “Modeling and performance evaluation of an OpenFlow architecture,” In Teletraffic Congress (ITC), 2011 Sept.
[23] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, and S. Shenker, Onix: A distributed control platform for large-scale production networks,” OSDI, Oct, 2010.
[24] A. Tootoonchian and Y. Ganjali. Hyperflow, “A distributed control plane for openflow,” In Proc. 2010 internet network management conf. on Research on enterprise network, USENIX Association, 2010.
[25] S. H. Yeganeh and Y. G.. Kandoo, “A Framework for Efficient and Scalable Offloading of Control Applications,” In Proc. 1st workshop on Hot topics in software defined networks, New York, NY, USA, 2012. ACM.
[26] B. Lantz, B. Heller, and N. McKeown, “A network in a laptop: rapid prototyping for software-defined networks” In Proc. 9th ACM SIGCOMM Workshop on Hot Topics in Network, 2010.
[27] M. R. Nascimento, C. E. Rothenberg, M. R. Salvador, C. N. A. Corrˆea, S. C. de Lucena, and M.F. Magalh ˜aes, “Virtual routers as a service: the routeflow approach leveraging Software defined networks,” In Proc. 6th Int. Conf. on Future Internet Technol., New York, NY, USA, 2011. ACM.
[28] C. E. Rothenberg, M. R. Nascimento, M. R. Salvador, C. N. A. Corrˆea, S. C. de Lucena, and R. Raszuk, “Revisiting routing control platforms with the eyes and muscles of software-defined networking,” In Proc. 1st workshop on Hot topics in software defined networks, New York, NY, USA, 2012. ACM.
[29] B. Rais, M. Mendonca, T. Turletti, and K. Obraczka, “Towards truly heterogeneous internets: Bridging infrastructure-based and infrastructureless networks,” In 2011 3rd Int. Conf. Commun. Syst. and Netw. (COMSNETS), IEEE, 2011.
[30] P. Dely, A. Kassler, and N. Bayer, “Openflow for wireless mesh networks,” In Proc. 20th Int. Conf. on Computer Communication and Network (ICCCN), IEEE, 2011.
[31] M. Mendonca, B. Astuto A. Nunes, K. Obraczka and T. Turletti, "Software defined networking for heterogeneous networks," HAL Id: hal-00838709, 2013.
[32] A. Detti, C. Pisa, S. Salsano and N. Blefari-Melazzi,"Wireless mesh software defined networks (wmSDN)," IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2013. “Cisco white paper,” © 2013 Cisco and/or its affiliates. All rights reserved.
[33] N. Foster, M. J. Freedman, A. Guha, and R. Horrison, “Languages for software-defined networks,” IEEE Commun. Mag. Feature Topic Softw. Defined Netw., vol. 51, no. 2, Feb. 2013.
[34] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart and A. Vahdat, “B4 Experience with a Globally Deployed,” Google, Inc., 2013.
[35] Technical White Paper, “Realizing the power of SDN with HP virtual application networks,” © Copyright 2012 Hewlett-Packard Development Company, L.P. “The Open SDN Architecture,” Copyright 2013 Big Switch Networks, Inc. All rights reserved.
[36] P. Bossharty, G. Gibbz, H. Kimy, G. Varghesex, N. McKeownz, M. Izzardy, F. Mujicay, M. Horowitzz, “Forwarding Metamorphosis: Fast Programmable Match-Action Processing in Hardware for SDN,” ACM Digital Library, 2013.https://www.nsnam.org/docs/release/3.16/models/html/openflow-switch.html,
[37] http://mininet.org/overview/
[38] F. Deti and S. Askar, “Emulation of Software Defined Networks Using Mininet in Different Simulation Environments,” 2015 6th International Conference on Intelligent Systems, Modelling and Simulation
[39] S. Perrin, and S. Hubbard, “Practical Implementation of SDN and NFV in the WAN,” Heavy Reading, on behalf of HP and Intel, White Paper, October 2013.
[40] http://www3.imperial.ac.uk/newsandeventspggrp/imperialcollege/engineering/computing/eventssummary/event_14-1-2016-10-7-24
[41] A. Markiewiz, P. N. Tran, and A. Timm-Giel, “Energy Consumption Optimization for Software Defigned Networks Considering Dynamic Traffic,” 2014 IEEE3rd International Conference on Cloud Networking (CloudNet).
[42] F. Giroire and D. M.L. Pacheco, “Energy-efficient Software-Defined Networks,” Networking (CloudNet), 2014 IEEE 3rd International Conference on Luxembourg.
[43] S. D. Elliott, “Exploring the Challenges and Opportunities of Implementing Software-Defined Networking in a Research Testbed,” A thesis submitted to the Graduate Faculty of North Carolina State University in partial fulfillment of the requirements for the Degree of Master of Science, Raleigh, North Carolina, 2015.https://www.sans.org/network-security/