

North American Academic
Research | Volume 5 | Issue 3 |
March 2022 | Monthly Journal by
TWASP, USA | Impact Factor:
3.75 (2021)

North American Academic Research

Monthly peer reviewed Journal by **The
World Association of Scientists &
Professionals**
TWASP, United States



Advanced Study of Software-Defined Networking (SDN): Benefits, Recent Works, Challenges (Use Case for SDN-OpenFlow)

SARR Cheikh Ahmed Tidiane^{1*}, Jinhe WANG¹

¹School of Engineering, Huzhou Normal University, China



Accepted March 20, 2022

Published March 25, 2022

Copyright: © The Author(s); **Conflicts of Interest:** There are no conflicts to declare.

***Corresponding Author:** SARR Cheikh Ahmed Tidiane

Funding: None

How to cite this article: SARR Cheikh Ahmed Tidiane, Jinhe WANG (2022). Advanced Study of Software-Defined Networking (SDN): Benefits, Recent Works, Challenges (Use Case for SDN-OpenFlow). *North American Academic Research*, 5(3), 174-184. doi: <https://doi.org/10.5281/zenodo.6385667>

ABSTRACT

The Software Defined Networking (SDN) concept could be a means to deal with the demands and requirements of a rapidly increasing computer network infrastructure. Companies interested in expanding their network infrastructures are intrigued by the concept, but it's critical to consider the consequences of converting from a standard network to an SDN network. One strategy to mitigate the effects is to make a gradual transition from a traditional IP network to an SDN, resulting in a heterogeneous network.

Rather than completely replacing the network infrastructure and dealing with the consequences, the soft migration concept is to replace a portion of it with an SDN environment and evaluate its performance. The efficiency of a network consisting of a traditional IP network mixed with SDN will be examined in this study. Identifying the variations in performance between having a heterogeneous network and having a specialized standard IP network is critical during this process.

As a result, the problems addressed in this work will be how to construct such a heterogeneous network and how to quantify its performance in terms of throughput, latency, and packet drop.

We hope to achieve our objectives with this study work by employing an experimental approach and analyzing relevant works on SDN fundamentals to provide us and the audience with a better understanding.

Keywords

Software-Defined Networking (SDN); OpenFlow; network management; network programmability; Network Function Virtualization (NFV).

Introduction

Computer networks are inexorably increasing in today's electronic environment. Traditional networks are being pushed to meet more sophisticated requirements by the rapidly expanding infrastructure.

A new concept known as Software Defined Networking (SDN) was created to overcome the potential challenges ([D. Kreutz et al., 2015](#)).

Since the commencement of the OpenFlow project, the concept of a centralized controller has been discussed, and the name SDN was coined to reflect the concept of using a logical controller. Today, the term is used to describe a more flexible and scalable network solution, with SDN being viewed as a cutting-edge technology for network design, deployment, and management.

Because of the impact that SDN has when moving from a typical IP network infrastructure, firms are facing new issues such as managing new equipment, training current employees, and establishing new routines. One strategy to reduce the damage is to make a gradual transition from classical IP to SDN, resulting in a heterogeneous network.

Problem Formulation

The rapid expansion of SDN research has piqued the interest of companies looking to expand and develop their network infrastructure ([K. Kirkpatrick, 2013](#)). It is important to remember that this method is not as simple as it appears. While moving from standard IP networks to SDN may be a step forward, a new network infrastructure has lots of disadvantages, including:

- 1.1.1.** The network must be reconfigured, which may hinder the company from managing its operations for an indefinite period of time.
- 1.1.2.** Provide workers with expertise and information so they can handle the new architecture.
- 1.1.3.** New monitoring, administration, and maintenance technologies are also required for a new network.

The fundamental goal of this thesis is to create a better scenario in which an existing network infrastructure is gradually replaced with a new, more efficient, scalable, and centralized network infrastructure, such as SDN. Rather than embarking on large-scale projects to replace network infrastructure, we propose replacing elements of the network one at a time, resulting in what is now known as a “heterogeneous network”. As a result, in this thesis, we consider a network that has a classical IP component as well as a software-defined network component.

The objective is to avoid upgrading network infrastructure with new equipment all at once, opting instead for a gradual transition from a traditional network to an SDN.

The purpose of this work is to build a heterogeneous network and test the performance based on a set of parameters.

It is critical that we identify and measure the variations in network performance that will occur when using a heterogeneous network vs. a dedicated traditional network. As a result, during the course of this thesis, the following questions will be addressed:

- ✓ What is the best way to construct a heterogeneous network?
- ✓ In terms of performance, latency, and packet drop, how will the heterogeneous network perform?

Background

The fundamentals of networking and software-defined networking are presented in this section.

Control and Data Plane

The control plane, data plane, and management plane are the three types of network planes, each of which is responsible for various key tasks ([G. Whelan, 2016](#)). The following section will cover the control and data plane fundamentals, that are required for this search.

Control Plane

Routing protocols are used by routers and multi-layer switches to detect and construct neighborhoods. Neighborhood formation takes place in the control plane ([K. Karakus, A. Durresi, 2017](#)).

A network device's control plane maintains the network topology and other network devices it is aware of. The Forwarding Information Base (FIB) and Media Access Control (MAC) tables are also handled by the control plane. The FIB table is used to perform lookups and determine where packets will be forwarded, whereas the MAC table is used to maintain unique layer 2 addresses.

Using the control plane's tables, a network device can determine where to deliver packets.

Data Plane

The data plane, also known as the forwarding plane, is in charge of transporting data packets to end devices ([K. Karakus, A. Durresi, 2017](#)). It consists of packet forwarding networking equipment such as switches and routers that have been specialized and implemented. The data plane can fulfil its functions based on the routing decisions made by the control plane.

The data plane is responsible for packets that are not intended for the device but must be transmitted to other devices.

Traditional Networking Challenges in Future Communication Networks

Technology usage and demand are increasing at a breakneck speed. As network infrastructure grows, networks continue to develop and become more sophisticated. In addition, the network is constantly being expanded with new users and network services. New users and network services necessitate a massive increase in network resources, which is occurring at an exponential rate. Traditional network management procedures would be very inefficient as the size and complexity of networks grew. This puts a lot of pressure on network operators since they have to adopt a lot of different configurations and keep track of a lot of different things on the network.

Over time, the number of connected devices has exploded, far exceeding what was previously thought to be possible for data communication networks to handle in the near future.

A solution to combat this significant expansion that can be managed through a single point is critical for this operation. Such a solution must be capable of meeting future needs and demands while also making network management simple.

The key solution to the aforementioned issues is SDN. Network programmability is improved, and network elements can be operated remotely from a centralized controller, thanks to the concept of software-defined networking.

The parts that follow go into greater detail about software-defined networking.

Software Defined Networking

SDN (Software Defined Networking) is a network architecture that seeks to make network management and administration easier (J. Wickboldt et al., 2015). As seen in Figure 1, SDN creates a more centralized network by separating the Control Plane and Data Plane on network devices and shifting control functions to the controller. SDN allows for programmable access and gets more efficient as the complexity and size of the network grows. SDN applications enable customers to leverage Quality of Service (QoS), traffic engineering, security, routing, load balancing, virtualization, monitoring, and, most likely, additional developments in the future.

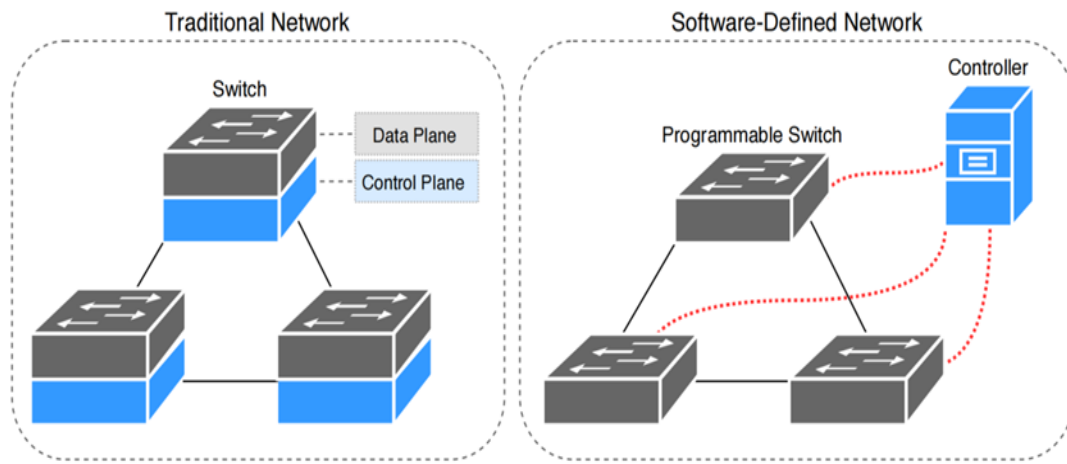


Figure 1: Control and Data Plane separation (SDN vs traditional network)

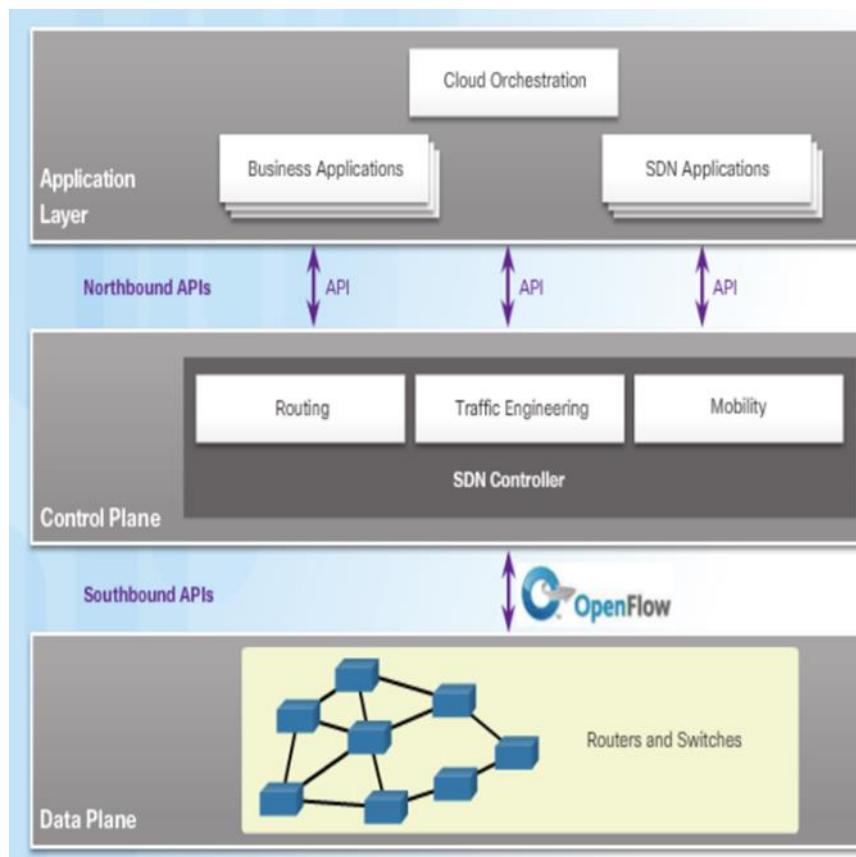


Figure 2: SDN Framework

Benefits

By adding programmability into the network, decoupling the control plane from the data plane allows for greater network management and flexibility. This topology allows network administration to take place on the control plane while data flows through the data plane remain unaffected. SDN has a number of advantages, including:

Enhanced Configurations

As the network grows in size and new devices are introduced, correct configurations must be established to ensure that the network runs smoothly. A typical technique is deemed laborious and fallible due to the heterogeneous nature of network devices. SDN connects several devices to a centralized control plane, allowing configuration and management to be done from a single location.

Enhanced Performance

The main goal of SDN is to achieve this. The consolidated control plane provided by SDN allows for performance optimization. As a result, all performance optimization difficulties would become manageable. As a result, typical issues such as QoS support, end-to-end congestion, and so on can be developed and exploited to prove their efficacy in improving system performance ([Xia, Wen, Foh, Niyato, & Xie, 2015](#)).

Reduced Cost

The centralized control plane, which can be managed by a single controller, is responsible for managing and orchestrating the network with SDN. To put it another way, SDN takes network infrastructure management and control away from network devices and instead places it into software, lowering operational expenses.

Encouraging Innovation

It is difficult to predict and precisely match the demands of future network applications, resulting in the deployment of new applications, network services, and ideas. It provides a programmable network platform for experimenting and implementing new ideas, resulting in increased innovation.

Challenges of Software-Defined Networking

Despite the fact that SDN has been dubbed the "key solution" to the problems that the expanding network infrastructure is causing, it is still regarded to be in its infancy. Benefits such as lower costs, improved configuration, and improved performance, among many others, have been spelled out, but there are still a number of challenges that need to be addressed. As SDN gets more generally implemented and new solutions are offered, challenges will continue to arise. In this section, we'll look at four major SDN challenges:

- (1) scalability,

- (2) flexibility and performance,
- (3) security, and
- (4) interoperability.

Scalability

One of the key issues posed by SDN has been demonstrated to be this. This single obstacle can be divided into two "sub-issues": (1) controller scalability and (2) network-node scalability. Up to 6 million flows per second can be handled by a single controller. As a result, this demonstrates that a single controller or many controllers may manage control plane services for a large number of data forwarding nodes. Instead of working on a peer-to-peer basis, the conceptually centralized controller must be physically spread to boost scalability.

However, whether a distributed or peer-to-peer controller infrastructure is used, the obstacles faced by the controller as interaction happens will be shared among network devices.

Flexibility and Performance

One of the most fundamental challenges of SDN is figuring out how to effectively deal with high-level packet processing flows.

Flexibility and performance are two important factors to consider.

In this section, the term "flexibility" refers to a network's capacity to adapt to new and unusual features such as applications and network services.

The speed with which network nodes in the data plane handle information from the control plane is referred to as performance.

Application Specific Standard Products (ASSPs) create the foundation for high-performance networks, while general-purpose processors (CPUs/GPPs) provide the most flexibility. The restricted flexibility of ASSPs, on the other hand, is a drawback. Application-specific integrated circuits (ASICs) are custom-built vendor-specific devices manufactured by companies such as Cisco and Juniper that are used when standard products are unavailable and programmable solutions are insufficient to meet performance requirements.

ASICs, on the other hand, offer the best performance at the expense of the least flexibility. The characteristics of ASICs make them the best choice for implementing the SDN data plane, which is now being implemented into SDN devices. In terms of data processing technology performance specifications, it can be determined that a hybrid approach will give an efficient and effective solution for SDN technology.

Security

Security issues linked with SDN have been the subject of research up to this point. As SDN becomes more widely used and deployed, security considerations must be considered.

With this in mind, the Open Networking Foundation (ONF) formed a security working group with them. Authentication and authorization at the controller-application level are considered to be at the top of the list of security vulnerabilities. An effective security model must be implemented to support network protection.

The controller is a target for threats in the SDN architecture, especially if it is vulnerable to unauthorized access. Because the controller is in charge of the entire network, attacks on it can cause major network damage. Furthermore, an attacker might pose as a controller and carry out nefarious actions. TLS (Transport Layer Security) is a security technique that aims to eliminate these dangers by allowing controllers and switches to authenticate each other. When implemented with a single controller overseeing a group of network nodes, TLS would offer the requisite security. Nonetheless, when a group of controllers interacts with a single node or vice versa, permission and authentication become more complicated.

A high-level security mechanism is supported by the SDN architecture. It can help with security policy changes, network forensics, and security service infiltration, among other things.

Even as the risk of unauthorized access increases, several threat mitigation solutions will emerge. To effectively achieve network protection, companies should create an efficient high-level security policy.

Interoperability

In this part, we'll look at how SDN solutions can be integrated into existing networks. We concentrate on the difficulties encountered during the transition from a traditional to an SDN approach. It would be sufficient to deploy an entirely new infrastructure based on SDN technology, with all network elements and devices being SDN-enabled ([Sezer et al., 2013](#)). Furthermore, there is a massive network infrastructure supporting critical systems and organizations, and migrating these networks to a new architecture is impossible because it is solely targeted at infrastructure-based networks like campus networks and data centers.

It is clear that migrating to SDN necessitates the use of both SDN and older equipment ([Sezer et al., 2013](#)).

The IETF Path Computation Element (PCE) could help with a smooth transition to SDN ([Paolucci, Cugini, Giorgetti et al., 2013](#)). PCEP is a protocol that makes it easier for network elements to communicate with one another.

Despite the fact that PCE is unable to provide complete SDN, the SDN controller can allow complete path calculation for data flow across many network devices.

To achieve a hybrid SDN architecture, research and development must be done, allowing traditional, SDN-enabled, and hybrid network nodes to operate together. As a result, in order to achieve interoperability, a protocol that is compatible with both SDN communication interface requirements and current traditional network protocols must be considered. Numerous industries working groups, like the IETF, ONF, and others, are continually proposing and developing standards and rules to make the transition from a traditional to an SDN model easier, and their efforts must be coordinated.

Implementation Tools for SDN

Many simulation tools, such as OMNET++ and Mininet, that I've used during my research to test different scenarios are available to test SDN performance. The other modeling instruments are the Ns-3 and Estinet.

These techniques have their own set of capabilities. Table 1 (Cisco Systems, 2012) shows the comparison between the various simulation tools. The purpose of this study was to provide an overview of SDN, including its description, architecture, benefits, and problems. We also compared and contrasted the SDN networking paradigm design with the open research challenges, revising some of the work done for each issue, such as scalability, security, dependability, and performance. Furthermore, numerous specific difficulties in SDN still require significant research to avoid inherited concerns from legacy networks, such as standardizing SDN modules and developing new SDN-specific procedures.

Table 1: The Simulation Tools Comparison

Features Tools	OMNET++	NS-3	Estinet	Mininet
Simulation Support	✓	✓	✓	x
Emulation Support	x	x	✓	✓
Capability to use an actual controller	x	x	✓	✓
Repeatable Outcomes	✓	✓	✓	x
Correctness of results outcome	No Real Controller	No Real Controller	✓	Performance relies on resources
Supporting GUI	Only for	only Monitoring, C++	Only for monitoring	only Monitoring, Python

Conclusion

SDN is a new networking paradigm that allows for uniform network behavior control through programming. SDN has been utilized to create many solutions to old network problems because it is a modern approach to networking. Several challenges remain difficult to solve. SDN addresses rising network complexity by enabling efficient and autonomous network control, addressing the requirement for rising network complexity as well as many other software domains.

This article analyzes the SDN networking paradigm concept and related open study difficulties, as well as some of the work done on each obstacle, such as scalability, security, reliability, and performance. Furthermore, numerous specific difficulties in SDN still require significant research to avoid inherited concerns from legacy networks, such as standardizing SDN modules and developing new SDN-specific procedures.

A lot of progress has been made in terms of network management, but more on the wireless network side with this technology of centralizing the control of some WIFI (Unifi, for example).

The management of wireless equipment is nowadays simpler than the management of equipment such as routers and switches, for example.

With the advent of SDN, I anticipate that this same advancement will lead to simpler management that will benefit engineers, as well as more benefits in terms of time and money for businesses, without forgetting security.

References

- [1] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015. (Cited on pages 1, 2, 7, 27, 36, 65, and 131.) 12
- [2] K. Kirkpatrick, "Software-defined networking," *Commun. ACM*, vol. 56, no. 9, pp. 16–19, Sep. 2013. [Online]. Available: <http://doi.acm.org.ep.bib.mdh.se/10.1145/2500468.250047>
- [3] G. Whelan. (2016-11-15) Software defined networking fundamentals part 1: Intro to networking planes. Accessed: 2019-05-02. [Online]. Available: <https://www.linux.com/learn/software-defined-networking-fundamentals-part-1-intro-networking-planes>
- [4] K. Karakus and A. Durresi, "A survey: Control plane scalability issues and approaches in software-defined networking (sdn)," *Computer Networks*, vol. 112, pp.279,293, 2017-01-15.
- [5] J. Wickboldt, W. Jesus, P. Heleno Isolani, C. Both, J. Rochol, and L. Granville, "Software-defined networking: Management requirements and challenges," *IEEE Communications Magazine*, vol. 53, pp. 278 – 285, 01 2015.
- [6] Prajapati A, Sakadasariya A, Patel J. Software defined network: Future of networking. In 2018 2nd International Conference on Inventive Systems and Control (ICISC). 2018;1351-1354.

- [7] Open Networking Foundation OpenFlow Switch Specification, Version 1.5.0", WhitePaper, <https://www.opennetworking.org/wp-content/uploads/2014/10/openflowswitch-v1.5.1.pdf> , Dec. 2014. xv, xvii, 19, 20, 24, 67, 86, 87, 116
- [8] Open Networking Foundation. Software-Defined Networking: The New Norm for Networks", WhitePaper, <https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdnnewnorm.pdf> , Apr 2012. xvii,14, 15, 53, 54
- [9] Cisco Systems Inc. OpFlex: An Open Policy Protocol White Paper", White Paper, <http://www.cisco.com/c/en/us/solutions/collateral/data-centervirtualization/application-centric-infrastructure/white-paper-c11-731302.html> , Apr. 2012. xvii,15, 18
- [10] S. Kuklinski, Programmable management framework for evolved sdn, in Network Operations and Management Symposium (NOMS), 2014 IEEE, pp. 18, May 2014. xvii, 30
- [11] D. Evans, The internet of things: How the next evolution of the internet is changing everything. White Paper, 13
- [12] http://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf , Apr 2011. 2, 6
- [13] Xia, Wen, Foh, Niyato, & Xie, 2015, A Survey on Software-Defined Networking <https://ieeexplore.ieee.org/document/6834762>
- [14] Sezer, S., Scott-Hayward, S., Chouhan, P. K., Fraser, B., Lake, D., Finnegan, J., . . . Rao, N. (2013). Are we ready for SDN? Implementation challenges for software-defined networks. IEEE Communications Magazine, 51(7), 36-43.
- [15] Paolucci, F., Cugini, F., Giorgetti, A., Sambo, N., & Castoldi, P. (2013). A survey on the path computation element (PCE) architecture. IEEE Communications Surveys & Tutorials, 15(4), 1819-1841.



© 2022 by the authors. Author/authors are fully responsible for the text, figure, data in above pages. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>)

Author(s) have identified their affiliated institutions or organizations, along with the corresponding country or geographic region. NAAR, TWASP remains neutral with regard to any jurisdictional claims.

