International Journal of Multidisciplinary Research and Growth Evaluation.

Development of enhance reference monitor algorithm for software defined networking (SDN) controller for 5G security

Jamilu Usman Waziri [1*], **E Omokhuale** [2]

[1] Department of MIS/ICT, Federal University Gusau, Zamfara State, Nigeria

[2] Department of Mathematical Sciences, Federal University Gusau, Zamfara State, Nigeria

* Corresponding Author: **Jamilu Usman Waziri**

Article Info

ISSN (online): 2582-7138
Volume: 03
Issue: 03
May-June 2022
Received: 17-04-2022
Accepted: 03-05-2022
Page No: 226-241
DOI: https://doi.org/10.54660/anfo.2022.3.3.15

Abstract

The aim of this research is to develop an enhanced reference monitor algorithm for Software Defined Networking (SDN) controller for 5G security. It is anticipated that by 2025, the network infrastructure should be able to provide connectivity for almost everything. This is expected to bring over 50 billion connections which cannot be handled by the current 4G. While 4G networks main focus is ubiquitous mobile broadband, 5G technology characteristics will have to increase immensely. The flexibility provided by software is key to meeting the unforeseen future service requirement. In this regard, Software Defined Networking (SDN) has recently gathered momentum in the networking industry and specific standard is yet to be adopted on how to check security challenges on SDN for 5G. This work proposes to adopt 5-ENSURE framework of integrating Reference Monitor (RM) to SDN controllers in order to impose access control policy. The Study will also isolate and handle malicious packets in a distributed manner among nodes rather than only permitting or denying access based on access policy. This work would contribute in security standardization of SDN for 5G which is currently under study across the world.

Keywords: Reference Monitor, 5G, Software Defined Networking (SDN), cryptography

1. Introduction

Information and Communication network and services demand today is ever becoming more complex as virtually every systems and gadgets we use in our daily lives are becoming increasingly dependent on Internet connectivity, ranging from mobile phone, cars, smart meters, smart home appliances to critical infrastructures such as: energy, gas, water, transportation, health and military. The security of Internet connectivity and information is a critical infrastructure for societies and economies in creating a safer world. Addressing this has been a major challenge in developing countries and hence the researcher's aspirations and interest in the area of security in 5G network as the communication industry is moving towards the actualization of 5G.

It is anticipated that by 2025, the network infrastructure should be able to provide connectivity for almost everything: goods, people, processes, content, knowledge, information, things and computing centres in a flexible, truly mobile, and powerful way. This is expected to brings over 50 billion connections by 2020 which cannot be handled by the current 4G (Panwar, Sharma, & Singh, 2015) [36]. While 4G networks main focus is ubiquitous mobile broadband, 5G technology characteristics will have to increase immensely. 5G will serve many different purposes with respect to reliability, latency, throughput, data volume, and mobility (Horn & Schneider, 2015) [18]. The integration of all these characteristics implies a complex system that will be difficult to manage, operate, and adapt to changing demands when using current technologies.

1.1 Software Defined Networks (SDN)

Software-Defined Networking SDN has emerged as a new intelligent architecture for network architecture to reduce hardware limitations.

The main idea of introducing SDN is to separate the control plane outside the switches and enable external control of data through a logical software component called controller. SDN provides simple abstractions to describe the components, the functions they provide, and the protocols to manage the forwarding plane along with Mobile IP from a remote controller via a secure channel. In conclusion, the inability of mutual access between different parts of heterogeneous networks would be solved. This abstraction is used instead of the common requirements of forwarding tables for a majority of switches and their flow tables. Hence, the controller monitors network packets, publishes policy, or solves errors according to the monitoring results. A number of northbound interfaces (connection between the control plane and applications) that provide higher level abstractions to program various network-level services and applications at the control plane. The OpenFlow standard has been exploited as the dominant technology for the southbound interface (connection between the control plane and network devices). This scheme allows on-demand resource allocation, self-service provisioning, completely virtualized networking, and secures cloud services. Thus, the static network would be evolved into a truly flexible service delivery platform that can respond rapidly to the network changes such as: end-user and market needs, which greatly simplifies the network design and operation. Moreover, the devices themselves no longer need to understand and process thousands of protocol standards but they should be capable of understanding instructions from the SDN controllers (Akram & Berthou, 2020) [5]. Facing the rapidly growing needs of users, Internet service providers cannot afford huge upgrades, adaption, or building costs, as hardware elements are expensive. Therefore, another advantage of exploiting SDN is to make it easier to introduce and deploy new applications and services than the classical hardware-dependent standards (Malik & Campbell, 2020) [27]. The ultimate goal of SDN is to create a network that does not need any the design or adjustments of the administrator interference, so, the network can be implemented fully automated administration. The administrators can manage the network through the controller plane more easily with dictating the required policy to the routers and switches, while they have a fully function monitoring over the network. Software defined networking (SDN) is bringing about a paradigm shift in networking through the ideas of programmable network infrastructure and decoupling of network control and data planes. It promises simplified network management and easier introduction of new services or changes into the network. Use of SDN concepts in 4G/5G mobile cellular networks is also being seen to be beneficial (e.g., for more effective radio resource allocation through centralization, seamless mobility across diverse technologies through a common control plane:

1.2 Reference Monitor

One of the major landmarks in computer security research was the definition of the reference monitor as a central location for access control decisions, and the associated notion of a Trusted Computing Base (TCB) Liu, S. and Li, B., (2020) [25]. The classic definition of a reference monitor has three properties:

(1) It is always invoked (equivalently, is unbypassable).

(2) It is tamper-proof.

(3) It is verifiable.

Much research over the last thirty years has been done on the first (Anthony, 2019) [19] and second properties, but comparatively little research has appeared on the verifiability of reference monitors. Given the importance of reference monitor correctness, this state of affairs is highly regrettable. Bugs in the reference monitor have a high probability of directly compromising the security goals of the system, as they are in the control flow path for every access control decision. While a reference monitor for a file system or other resource is often part of the TCB, recent trends in extensible operating kernels M and proof-carrying code Open network Foundation (ONF), 2013 suggest that it may be possible to move some portions of traditional reference monitor functionality out of the TCB, as long as computations done outside the TCB can be checked within the TCB. Formal methods can be difficult and time-consuming to use, and are considered impractical for many applications. Because of the difficulty of using formal methods, alternative assurance techniques have been developed (Ali *et al*., 2019) [7]. Unfortunately, alternative techniques do not provide the same confidence in correctness as formal, machine-checked, proofs. That formal methods remain the "gold standard" can be seen, for example, in the Common Criteria. The paucity of general purpose operating systems evaluated at EAL7 (under any protection profile) shows the rarity of formally-verified reference monitors.

1.3 5G Network

The 5G networks, as a new wireless communication technology, experience a fast development in recent years. As shown in Figure 1, this technique has been widely used in every corner of our daily life. Compared with the outdated commercial 4G (LTE/WiMAX) system, this technology has advantages in high data rate, reduced latency, and massive device connectivity, making it a fundamental infrastructure module for wireless communication in the near future. Motivated by its significant advantages, many researchers have designed their own protocols (Colville & Spafford, 2021) [12] to make it fit the requirement of the real applications. In this work, software-defined networking (SDN) is one of the key design concepts. SDN is an approach to improve network performance and monitoring by facilitating network management and enabling programmatically efficient network configuration (Rafat Jahan, 2020) [37]. By separating data and control planes, SDN enables a wide range of new innovative applications from traffic engineering to data center virtualization, fine-grained access control, and so on (Navid *et al*., 2021E) [30]. It has a proven advantage in many commercial networks Ian F. et al. (2019) [20] and therefore is also a good choice in the field of 5G networks. Therefore, 5G is considered to provide adaptive and flexible centralized processing, which allows efficient management of an ultra-dense mobile network and enables more flexible dedicated software solutions across various technology. The flexibility provided by software is the key to enabling further innovation and to meeting the unforeseen future service requirement. In this regard, Software Defined Networking (SDN) has gathered momentum in the networking industry in the past few years and specific standard is yet to be adopted on how to check security challenges on SDN for 5G; The sophisticated control provided by SDN opens opportunities for better cloud security engineering as well as new vulnerabilities which are potentially exposed as new technologies are introduced

(Trivisonno *et al.*, 2015) [46].

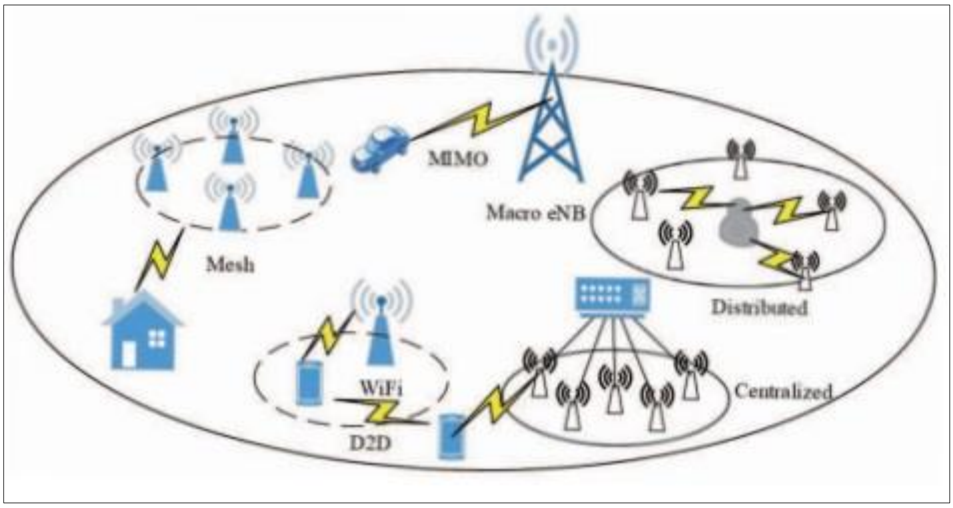


Fig 1: 5G Networks

1.4 Background of the Study

1.4.1 Origin of SDN and Architecture

The work towards isolation of control logic from data logic has a long history. But it came in limelight in 2006, when Martin Casado, a PhD understudy at Stanford University and group propose a new security design (SANE) which characterizes a unified control of security (rather than at the edge as typically done). It states that security should be checked at each entrance as well as main entrance in the network. Ethane sums it up to all arrangements providing ethane switches to provide a hybrid network environment as it was not possible to replace the whole existing network Guolin, *et al.*, (2020) [15]. The possibility of Software Defined Network happened from OpenFlow venture (ACM SIGCOMM 2008) (Aditya, *et al.*, 2018) [2]. In 2009 Stanford announced OpenFlow V1.0.0 specs and Martin Casado again helped and established Nicira in June 2009. In March 2011 Open Networking Foundation was framed and First Open Networking Summit was hung on October 2011. Numerous Industries Juniper, Cisco declared to consolidate. In July 2012 VMware purchases Nicira for $1.26B. SDN is based on the concept of data plane and control plane. A network can be viewed as constitute of data and control plane. The data plane is responsible for forwarding the data as per the flow rules and control plane defines the flow rules and control decisions necessary for the delivery of user data to right destination. In traditional networking this all comprises in a single box (e.g. Routers). In SDN the controlling part of the network has been decoupled from the inter- networking devices to a logically centralized controller and these network devices work as the general purpose data forwarding devices. For clarity, SDN is described in this article with the Open Networking Foundation (ONF) 2013 definition: “In the SDN architecture, the control and data planes are decoupled, network intelligence and state are logically centralized, and the underlying network infrastructure is abstracted from the applications.” SDN focuses on four key features:

a. Isolation of logical intelligence from the devices
b. A central place for all intelligence and control
c. APIs between the data logic and control logic i.e. controller and devices
d. Innovation through programmability
e. Increased Security and reliabilities with complete visibility and control over the network

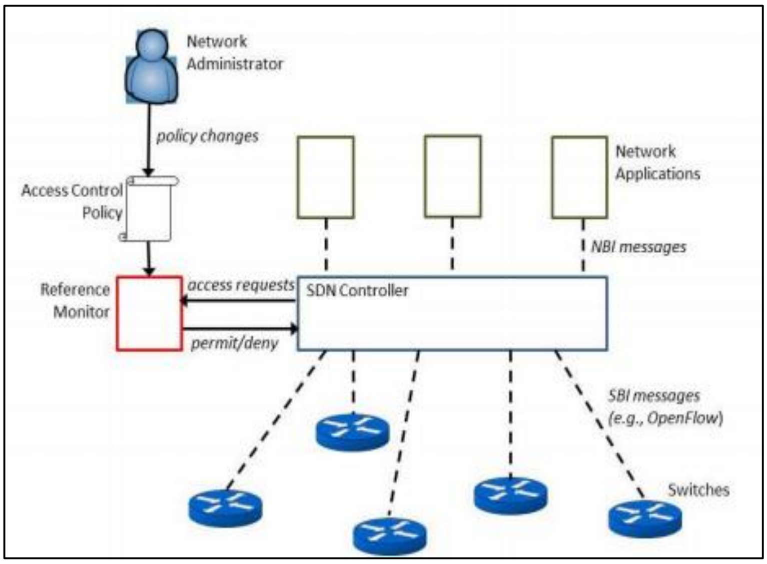


Fig 2: Architecture of SDN (Hyunwoo *et al.*, 2015)

Figure 2 shows the basic architecture of the SDN. The basic working of the SDN includes the communication of controller with data plane. The controller does this by using the open flow protocol. Open flow protocol works as a communication medium between the controller and forwarding devices and encourages the decoupling of control from the network devices. This is a flow based communication; each device in the data plane maintains a flow table which is managed by the controller. To maintain the communication over the network, an open flow controller adds and removes the forwarding rules in network switches. A forwarding rule is based on the match of the fields (packet header, incoming port etc.) e.g. source and destination IP addresses and related actions are performed e.g. forward or drop a packet. To configure a new policy in switch, the controller can modify relevant entries in the flow tables and this may also be done in real time.

1.4.2 Traditional Networking and SDN

In traditional networking the control plane and data plane resides inside the networking device. Every device (e.g. Routers) has its control plane and takes decisions as per the configured policy/protocol as shown in figure 3. Once the policies have been configured and flow has been defined it is very difficult to change the network behavior in response to changing traffic demands. The only way to make an adjustment is to change the configuration of all the devices. This leads to a bottleneck for the administrators who want to scale their network as per the demands. With the increase in use of the mobile devices, cloud computing and big data demand a great need of change the network behavior in the real time.

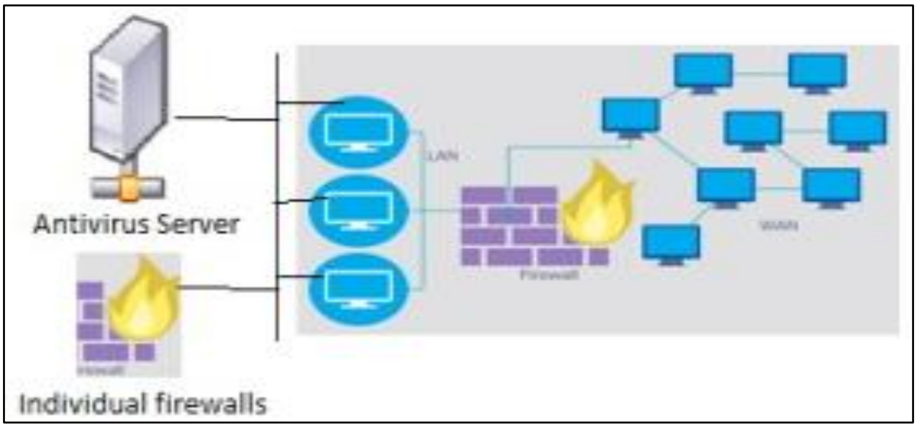


Fig 3: Traditional Security Architecture (Shin, and Gu, (2018) [41]

Figure 4 shows the conceptual design of SDN where controlling part of a network device has been separated to a logically centralized controller and networking devices are just switches which can work fast and efficiently. Security solutions in traditional networks use a lots complex mechanism to protect the network namely ACLs, VLAN, firewall, NAT etc. These policies are distributed on all the networking devices. The policies are topology based; address based and even port based which breaks as per the changes in network topology or user move.

A set of all security policies is put in one box i.e firewall and it is kept at the entry and exit point of the network as shown in figure 3. If an attacker makes it through the firewall it has all the access to the network. Distributed firewalls and antivirus are implanted on end users to mitigate this but it exhibits the complexity of the traditional network and placing all the trust in end users. A traditional firewall can only prevent threats to access your computer on internet while most of the viruses or Trojans are received via emails, through file sharing or through direct download of malicious programs. Traditional firewall cannot prevent this. In most of the firewalls packet filtering is done at network layer and transport layer generally. But nowadays there is a requirement of more enhanced version of firewalls which can even work at Application Layer. Some of the firewalls equip with this facility but they all depend on protocol specification related to particular applications. Proxy, IDS and IPS try to prevent the network attacks but traditional network architecture creates bottleneck having control distributed in devices which creates a lots of complexity for policy enforcement in these networks (Shin, and Gu, (2018) [40]. In SDN architecture, above the controller there is an application plane which introduces the concept of the programmability in the networks. Here we have different applications like traffic monitoring, security which can be directly programmed as per requirements. While in existing systems the network devices are closed boxes where there is no scope of programmability and innovation. The concept of network programmability is one of the prime implicates of the SDN. Until recently most modern network elements (e.g. routers, switches or firewalls) supported a small set of interfaces that were used to communicate with those elements. These typically included a proprietary command line interface (CLI), SNMP, CORBA and most recently NETCONF. Unfortunately, none of these languages are able to provide a complete common solution. They are very static in nature and require a priori data model design and declaration. SDN relies on having multiple managers, agents and controllers, all interacting in symphony of tightly coupled communication which leads to the optimizations and abilities which cannot be obtained by these old interfacing models. In order to realize this new era of communication and interaction, tightly coupled and bidirectional streaming interfaces are needed. Several application friendly interfaces come into consideration including JSON, Google buffers, Thrift and more recently the work in IETF's I2RS (Interface to Routing System).

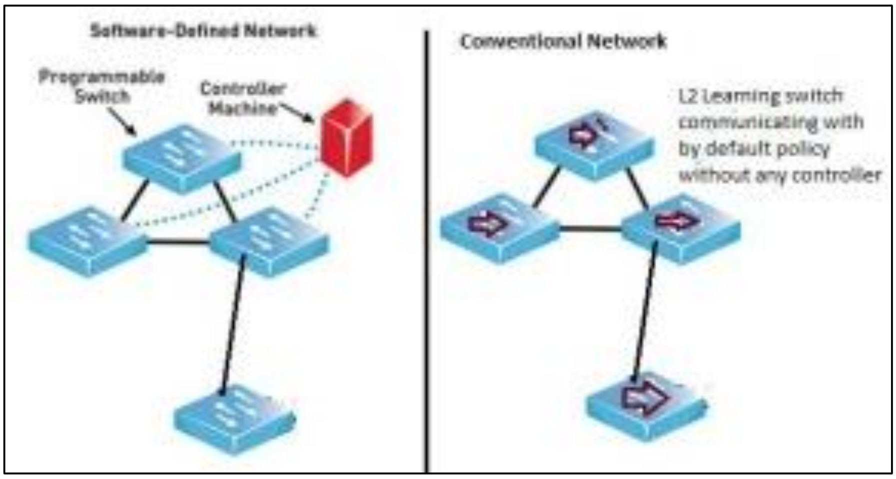


Fig 4: Isolation of Control from Devices in SDN

Figure 4 shows the conceptual design of SDN where controlling part of a network device has been separated to a logically centralized controller and networking devices are just switches which can work fast and efficiently. Security solutions in traditional networks use a lots complex mechanism to protect the network namely ACLs, VLAN, firewall, NAT etc. These policies are distributed on all the networking devices. The policies are topology based; address based and even port based which breaks as per the changes in network topology or user move.

1.5 Benefits of SDN

The separation of the control and data planes increases the flexibility of the network to adapt to evolving networks. One of the major benefits for operators and service providers is reduction in operation cost due to centralized management, efficiency in operations and existing hardware being fully utilized. The ability of the networking infrastructure to be programmable and manageable makes it scalable and more dynamic. Other expected benefits include increased network reliability and security discussed in this paper in addition to better user-experience due to SDN ability to adapt to dynamic user-needs. SDN is also expected to manage inflow of traffic from internet of things (IoT) devices by segmenting the traffic and helping to organize the data. Furthermore, SDN is expected to enable networks keep pace with the speed of 5G networks, as a new wireless communication technology, experience a fast development in recent years. As shown in Figure 5, this technique has been widely used in every corner of our daily life. Compared with the outdated commercial 4G (LTE/WiMAX) system, this technology has advantages in high data rate, reduced latency, and massive device connectivity, making it a fundamental infrastructure module for wireless communication in the near future. Motivated by its significant advantages, many researchers have designed their own protocols (Metzler, 2021) [29] to make it fit the requirement of the real applications. In this work, software-defined networking (SDN) is one of the key design concepts. SDN is an approach to improve network performance and monitoring by facilitating network management and enabling programmatically efficient network configuration (Kreutz *et al.*, 2019) [21]. By separating data and control planes, SDN enables a wide range of new innovative applications from traffic engineering to data center virtualization, fine-grained access control, and so on (Shirali and Ganjali, 2018) [43].

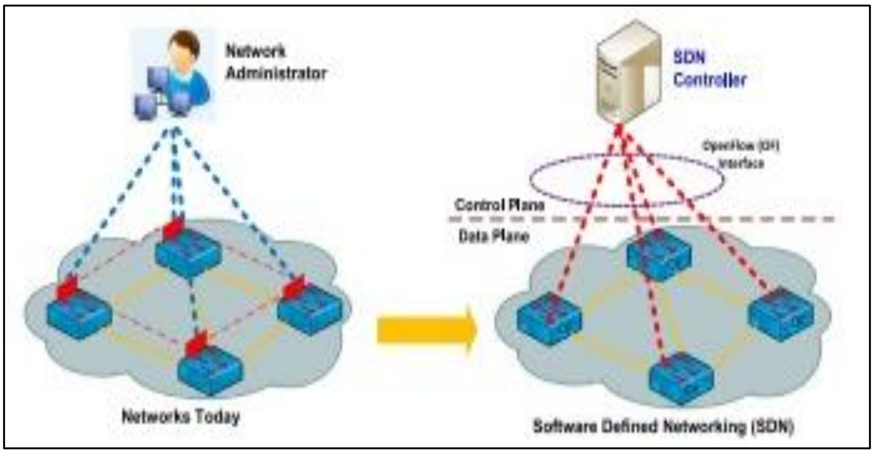


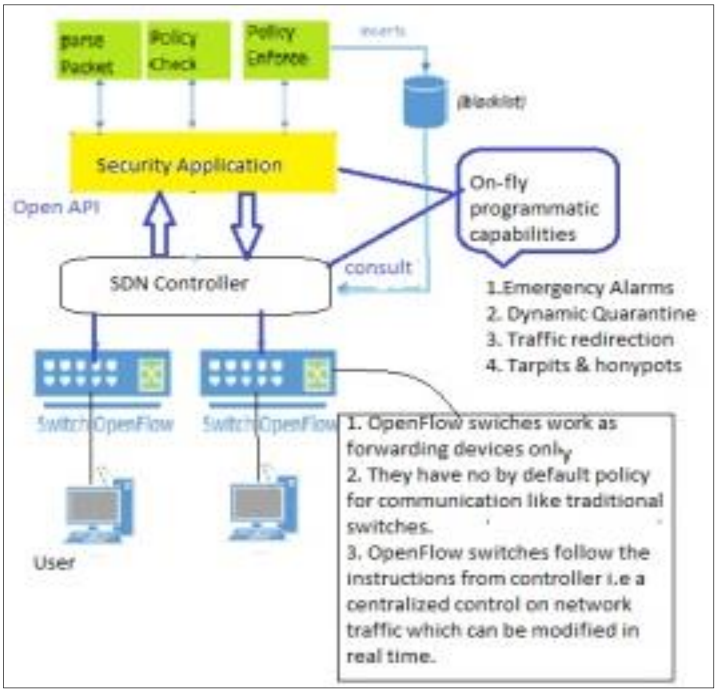
Fig 5: Comparison of Traditional Network and SDN

It has a Future Internet of 14 proven advantages in many commercial networks (Akhunzada, *et al.*, 2019) [4] and therefore is also a good choice in the field of 5G networks. Despite these advantages, forming such an SDN-based 5G network is not free, and there remains a lot of challenges when it comes to security. This is because the intelligence centralization of SDN is vulnerable for various attacks. Several research works on SDN have already investigated security applications built upon the SDN controller, with different aims in mind. Distributed Denial of Service (DDoS) detection and mitigation (Dabbagh,*et al.*, 2018) [13], as well as botnet (Wang, *et al.*, 2019) [20] and worm propagation (Ying-Dar *et al.*, 2017) [49], are some concrete use-cases of such applications: basically, the idea consists of periodically collecting network statistics from the forwarding plane of the network in a standardized manner (e.g., using OpenFlow), and then apply classification algorithms on those statistics in order to detect any network anomalies. If an anomaly is detected, the application instructs the controller how to reprogram the data plane in order to mitigate it. Another kind of security application leverages the SDN controller by implementing some moving target defense (MTD) algorithms. MTD algorithms are typically used to make any attack on a given system or network more difficult than usual by periodically hiding or changing key properties of that system or network. In traditional networks, implementing MTD algorithms is not a trivial task since it is difficult to build a central authority capable of determining for each part of the system to be protected which key properties are hidden or changed. In an SDN network, such tasks become more straightforward thanks to the centrality of the controller. One application can, for example, periodically assign virtual IPs to hosts within the network, and the mapping virtual IP/real IP is then performed by the controller (Li, 2020). Another application can simulate some fake opened/closed/filtered ports on random hosts in the network in order to add significant noise during the reconnaissance phase (e.g., scanning) performed by an attacker (Scott-Hayward *et at.*, 2019) [39]. Additional value regarding security in SDN enabled networks can also be gained using FlowVisor (Kreutz, *et al.*, 2019) [21] and FlowChecker (Shin, and Gu, 2018) [40], respectively. The former tries to use a single hardware forwarding plane sharing multiple separated logical networks. Following this approach, the same hardware resources can be used for production and development purposes as well as separating monitoring, configuration and internet traffic, where each scenario can have its own logical topology which is called slice. In conjunction with this approach, FlowChecker (Kreutz, Ramon and Verissimo 2019) [21] realizes the validation of new OpenFlow rules that are deployed by users using their own slice. SDN controller applications are mostly deployed in large-scale scenarios, which require comprehensive checks of possible programming errors. A system to do this called NICE was described in 2012. Introducing overarching security architecture requires a comprehensive and protracted approach to SDN. Since it was introduced, designers are looking at possible ways to secure SDN that do not compromise scalability. One architecture called SN-SECA (SDN+NFV) Security Architecture.

Table 1: Security Problems in 5G Networks

Channel Type	IP Spooling	MITM Attack	Replay Attack
Control Channel	yes	yes	yes
Data Channel	yes	yes	yes

Unlike the conventional network (2G, 3G, 4G), SDN separates the control plane from the data plane. The control plane composes of a (logically centralized) controller which interacts with the data plane component such as switches via its southbound interface (SBI); Network application such as network traffic routing applications interacts indirectly with the data plane components via controller's northbound interface (NBI) applications; Open Flow is the major communication protocol used by SDN (5G-ENSURE, 2016) [1]. This work proposes to adopt 5-ENASURE framework of integrating reference monitor to SDN controllers that handles every message that comes in and out of the controller; The reference monitor simply drops a malicious packet or allows access to network resources according to a given access control policy (5G-ENSURE, 2016) [1]. This research takes a step further to enhance the capability of the reference monitor through the design of an algorithm that isolate such packet and make its handling a distributed task among nodes. Figure 5 shows the SDN components and its interaction including the reference monitor where the proposed algorithm will be position.



1. OpenFlow swiches work as forwarding devices only
2. They have no by default policy for communication like traditional switches.
3. OpenFlow switches follow the instructions from controller i.e a centralized control on network traffic which can be modified in real time.

Fig 5: SDN Components (5G-ENSURE, 2016) [1]

1.6 Problem Statement

Software Defined Networking (SDN) has gathered momentum in the networking industry in the past few years and specific standard is yet to be adopted on how to check security challenges on SDN for 5G. Most recent works have focused in securing SDN controller because threat to the controller is considered to be a major threat to the entire network (Chen *et al.*, 2015) [10]. One of them is the work by 5G-Ensure, 2016 [1], in its framework called D3.2 5G-PPP security enablers open specifications (v1.0), that integrates reference monitor to SDN controller in order to handle messages in and out of the controller. The reference monitor simply drops any incoming malicious packet or allow it access to network resources according to a given access control policy. This research is proposed to improve on the capacity of the reference monitor in v1.0 as proposed by 5G-Ensure, through the design of an algorithm that isolates suspicious packets and handles it in a distributed manner across nodes.

1.7 Aim/Objectives

The aim of this research is to improve reference monitor of SDN controllers to be able to isolate suspicious packet in a distributed manner among nodes.

The Objectives are to:

1. Develop algorithms for the isolation of malicious packets and distribution point action in the reference monitor of SDN Floodlight controller
2. Simulate the developed algorithms in 1 using mininet tool
3. Deploy and validate the simulated algorithms in (2) in 5G and IoT Testbed 1 at Melbourne Innovation Districts (MID)
4. Evaluate the performance of the algorithms using False Positive and False Negative metrics and the controller using OFLOPS tool

1.8 Significant of study

This study would contribute in security standardization of SDN for 5G which is currently under study across the world. As the major task in securing SDN is to secure the control plane, the enhance reference monitor of the SDN controller in this research would improve security.

2. Literature Review

The controller is the core of the SDN architecture and if the controller is compromised, the entire network is compromised. To date, several approaches to SDN security have been proposed.

Sánchez *et al.*, (2014) [38] analyzed vulnerabilities in SDN and network function virtualization (NFV) for 5G and propose a self-healing framework. This work is centered on recovery of the network when abnormality is observe using measure of degradation (broken state) as performance metric. However, this approach is applicable when attack has successful tempered with the network. In order word the SDN controller might have been compromised before such measures would come in play.

Cho *et al.*, (2014) [11] proposed self-defined Radio (SDR) and SDN as the solution for high performance in 5G. They presented a cross layer architecture combining the characteristics of SDR and SDN to enhance network performance. However, security implication was not taken into consideration.

Hong, Xu, Wang, and Gu (2015) [6], in an effort to mitigate network topology poisoning attack on SDN, developed TopoGuard; an extension of to SDN controller that provide real-time automatic detection of such attack. However, TopoGuard only guards against topology poisoning of floodlight SDN controller and since reference monitor serves as gateway to the SDN controller, such mechanism could effectively be implemented there.

Liyanage *et al.*, (2015) [26], proposed a multitier security approach to secure Software Defined Mobile Network (SDMN) for 5G network. They employed Host Identity Protocol (HIP) and IPsec tunnel to secure communication channels. Also, access control was implemented at the mobile backhaul by policy based communication. This security mechanism successfully addressed spoofing and DoS attack. However, security was implemented on communication channels and more work is further needed on the control plane of the SDN controller.

Akyildiz, *et al.,* (2015), worked on a software defined networking architecture for 5G wireless system called SoftAir. The SoftAir leverage on network function cloudification and network virtualization to enhance flexibility, scalability and resilience. However, the control plane of the SDN controller needs to be enhanced.

In 2015, Adrian, Louis, Evangelos, George, & Nikolaos, identified network assets and the security threats associated with it; challenges and risks arising for these assets. They came up with 6 technical recommendations and 3 organization recommendations for SDN/5G.

2.1 Opportunities for security enhancement in SDN

SDN system-wide complete view of network, programmability through open application programming interfaces, and control of policies through a centralized entity controller provides various ways for security enhancement and threat mitigation. SDN opens up a new platform to create customized security algorithms (Feldmann *et al.*, 2021) [14]. SDN supported network proffer a central place for data collection from network devices and new security approaches assumes a centralized data model which was not possible in conventional networks. This is an extreme transformation which has positive ramification for various algorithms related to network monitoring, and firewall methodologies (ND Szabo et al., 2019) [31]. In this section we will analyze how SDN work with different terminologies like network monitoring, network verification & automation, threat detection and response, which can identify promising future research directions in these networks.

2.1.1 Network Monitoring

Network monitoring is the fundamental part for network security. Actually, suspicious traffic patterns can be found by collecting the real time data from the network and testing it for security breach through various anomaly detection algorithms, For example an attacker can use scanning tools to know the network behavior before doing attack operation. In this case network monitoring becomes more important. Network monitoring in SDN, based on open flow consists of collection of flow based data at controller side which is a natural open flow process in SDN. This can be achieved in two ways. One through the push operation, when a switch tells the controller about the flow that it is expired (Flow Removed Message). Another way is pull operation when controller asks the forwarding devices to know the status of flows through Flow Statistics Request and Flow Statistics Reply messages. Flow Sense (Zheng Ma *et al.*, 2020) [50] is an example of push operation.

2.1.2 Network Verification and Automation:

Manual policy configuration is always the error prone and there should be some automation techniques for configuration verification and consistency. A survey from Gartner points out that, in a passage of year 2010 to 2015, most of network blackouts affecting vital administrations are because of manual configurations and process related, and over half of them coming from policy changes i.e. reconfigurations and updates issues (H. Hung Cho et. al., 2014) [11]. In SDN when there are more than one controller, several applications and multiple users running concurrently in the same domain, this may lead to inconsistency and policy violation issues. This can cause several network faults like loops, blackholes and access control issues. Moreover in big networks where there are many switches, controllers need to install thousands of flows dealing with many flow tables, controller can install approximately 50000 new flows every second (Shin, *et al.*, 2018) [41], there should be brisk, efficient approach to guarantee security consistence, adaptation to non-critical failure, and quick failover. The good work around there, Flow Checker (Mehdi, *et al.*, 2019) [28] is property-based verifier tools that find different misconfiguration inside the network. Flow Checker uses Binary Decisions Diagrams and encodes switch flow-table configuration to create a state machine depicting the flow statistics of forwarding devices in the network. NICE (Rafat Jahan 2020) [37] is also another error finding tool in SDN configurations. Moreover except these solutions which are used before the network start or application installation, VeriFlow (Anthony, 2019) [9] is an on-fly arrangement which check network accuracy in real-time as the network advances progressively. NOX controller also has an inbuilt error checking solution called FORTNOX (Ying-Dar *et al.*, 2017) [49] which identify conflicting flow rules in real-time

2.1.3 Improvised Threat Detection

In SDN, controller provides a complete view of the devices which is very much favorable for threat detection. The open flow switches do not have by default communication policy as in L2 learning switches, OF switches follow the instructions from controller and controller can reprogram the data plane devices in the network to conduct analysis for suspicious data and malicious device in the network (Shin and Gu, (2018) [42]. Most of the traditional security systems provides security on Layer 3 and layer 4 and cannot detect the malicious payload at application level, in case of application level security in SDN there is need to send all the packets to controllers which create an overhead on controller and respective links. To avoid this situation propose an algorithm which is based on the number of unsuccessful connection attempts of fake request. It sends only those packets to the controllers which are suspicious based on the given algorithms. Microsoft is also using SDN solutions in its data centers for malicious traffic detection (Scott-Hayward *et al.*, 2019) [39]. With a very large infrastructure of Microsoft conventional packet inspection technology like port mirroring and switch port analyzer (SPAN) are not feasible which require a lots of physical ports and accounting arrangements. In SDN this can be easily configured through controller by using the virtual ports (Ying-Dar *et al.*, 2017) [49]. Radware has used the SDN platform for innovative security solution and provided Defence Flow for detecting malicious network attacks like DoS (Feldmann *et al.*, 2021) [14]. For research and development the open source version of the same has also been provided.

2.1.4 Dynamic Response to Threats

SDN system-wide complete view of network, programmability through open application programming interfaces, control of policies through a centralized entity controller bolsters the security providers as well researchers and opens up a new ways to provide a dynamic response to threats. Due to the lack of centralized control in legacy network the only response is to drop the malicious traffic but in case of SDN we can redirect the traffic for forensics by reprogram the switches dynamically through the controller. FRESCO (Scott-Hayward *et al.*, 2019) [39] and FORTNOX are the example of SDN enabled dynamic response to threats. Also Colville & Spafford 2021 [12] reveals that lack of integrated network control creates network management challenges and the error prone configuration process triggers network faults, bugs, and security lapses. Feldmann *et al.*, 2021 [14] suggest that because of inflexibility, network innovation has essentially stagnated. However, SDN model frontally addresses this challenge by separating the packet forwarding functionality of the forwarding devices or data plane from the control element or control plane (Shirali and Ganjali, 2018) [43]. The separation technique which is technically called decoupling remains a key feature of SDN. Decoupling spawns innovative network architecture where the network switches functions such as basic forwarding devices and the control logic is implemented in a logically centralized controller (Scott-Hayward *et al.*, 2019) [39]. Akhunzada *et al.*, 2019 [4], argue that the integrity and security of SDNs remain unproven regarding the placement of management functionality in a single centralized virtual server making it easier to compromise the whole network through a single point of failure. However, claim that SDN provides a unique opportunity for effectively detecting and containing network security problems in home and office networks. The research findings of, reveal four prominent track anomaly detection algorithms which can be implemented in an SDN framework using Open flow compliant switches and NOX (open source development platform for C++ based SDN control applications) as a controller. They further indicated that these algorithms are significantly more accurate in detecting malicious activities in the home networks in comparison to the Internet Service Provider (ISP) (Anon, 2019) [8]. SDN's major security issue is being self-secure. Kreutz *et al.*, 2018 [21] advocated incorporating security and dependability into the SDN architecture from the ground level up. According to them, SDN is susceptible to several threats such as forged traffic flow to attacking network entities; Denial of Service (DoS) attacks on switches, controllers and control plane communications (Malik *et al.*, 2020) [27]. Potential attacks on the interface between the controller and high-level applications, exploiting the weaknesses in Secure Socket Layer (SSL) and Transport Layer Security (TLS) protocol implementations in addition to switches in the network may be hijacked or exploited (Navid *et al.*, 2021) [30]. These are missing gaps that this study will attempt to address on the security issues in the evolution of SDN and its adoption by service providers in Nigeria. Kreutz *et al*. 2019 [22] proposed stringent authentication mechanisms and trust models which could counter common identity-based attacks as few of the potential solutions to the identified threats inherent in the current SDN is a monotony regime (Panwar *et al.*, 2015) [36]. Therefore, there is need to diversify the protocols, controllers, and tools employed and consequently reduce common implementation vulnerabilities, a major focus of this study. Shin *et al.*, 2018 [41] propose FRESCO, a security specific application development framework for OpenFlow networks for securing the design of SDN. FRESCO simplifies transferring of the application programming interface (API) scripts to enabling the development of threat-detection logic and security monitoring as programming libraries (Adrian et al., 2015) [3]. But Akhunzada *et al.*, 2019 [4] state that, FRESCO does not improve the security of the application and infrastructure layers of SDN. As alternatives, (Shirali-Shahrez and Ganjali 2018) [43] propose FleXam, a sampling extension for OpenFlow to enhance the security of SDN while Shing and Gu 2018 [40], propose Cloud Watcher, a framework for monitoring clouds. Kreutz *et al.*, 2018 [21] propose L-IDS, a learning intrusion detection system to protect mobile devices in a specific location which they regard as a prominent solution for security enhancement. Also, Wang *et al.*, 2019 [20] offer a systematic approach to detecting and resolving conflicts in an SDN firewall by checking firewall authorization space and flow space using ‘header space analyses’ to investigating the effectiveness and efficiency of this approach in addressing security analyses threats. Shin et.al. 2020 [42] suggest the use of connection migration, an extension to the data panel to reducing interactions between data and control panel to addressing DoS attacks on the southbound interface. This is like the approach proposed by Ying-Dar *et al.*, 2017 [49], for reducing the traffic overhead to the controller and providing NFV through an extended SDN architecture. Their evaluation show that in the extended SDN architecture, only 0.12 percent of the input traffic is handled by the controller extended, while 77.23 percent is handled on the controller in conventional architecture (Dabbagh *et al.*, 2018) [13]. Akhunzada *et al.*, 2019 [4] also claim that, OpenWatch, an adaptive method of flow counting to detect anomalies in SDN is a credible solution for security analyses and is expected to improve the overall security of Network protocols such as OpenFlow points out, that as cyber-threats continue to evolve and become more sophisticated, the potentials of a highly configurable network attack is catastrophic.

Furthermore, 5G-ENSURE (2016) [1] proposed a specification for 5G security architecture. Part of their specification is on Management and virtualization Isolation enablers open specification using SDN. They specify reference monitor (RM) to be integrated into SDN controllers to impose access control policies. However, this controller only permits or denies packets access according to access control protocol.

That is why this research is proposed to improve on the capacity of the RM proposed by 5G-ENSURE through the design of an algorithm that isolates suspicious packets and handles it in a distributed manner across nodes.

3. Methodology

This section shows the methodology used in carrying out the research.

The overall framework and steps used in carrying out the research in different phases base on the research objectives would be achieved through the following steps is shown in Figure 6.

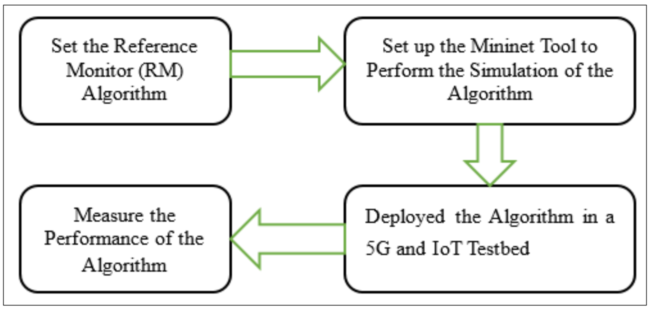

Set the Reference Monitor (RM) Algorithm
Set up the Mininet Tool to Perform the Simulation of the Algorithm
Deployed the Algorithm in a 5G and IoT Testbed
Measure the Performance of the Algorithm

Fig 6: Overview of the Methodology

Step 1-Objective (1) would be achieved through the following step

1) The RM would first analyze the incoming packet
2) If the packet is suspicious then

The RM isolate the packet and assign the further of the packet across nodes
If packet is malicious, then discard
Else, allow the packet to pass.

3) Else, allow packet passage

Figure 7 shows the flow chart of the proposed algorithm. The algorithm was implemented using floodlight controller. It has more features such as REST API and support non-OpenFlow domains. It also has generally good documentation that surpasses that of other controllers such as Beacon (Suomalainen *et al.*, 2014) [45].

Step 2: Objective (2) would be achieved using Mininet tool whose switches support distributed network topology and openflow protocol networks. In Mininet, network is created from a single, simple Python API and need for installation/configuration of multiple orchestration systems is avoided. Mininet’s experimental cluster support would be deployed to distribute the virtual testbed across multiple physical (or virtual) servers (Lantz & O'Connor, 2015) [23]. This would be used to monitor how malicious packets are isolated and handle across multiple nodes. To make creation of a virtual testbed more convenient, Lantz & OConnor (2015) [23] suggested the extension of Mininet’s Host class by the addition of a server class.

Step 3: To achieve objective (3), the simulated algorithm would be deployed in 5G and IoT Testbed 1 at Melbourne Innovation Districts (MID).

Step 4: Objective (4) which is performance evaluation would be measured using False Positive (FP), False Negative (FN), True Positive (TP) and True Negative (TN). These values would be obtained by calculating the sensitivity, specificity, Positive Predictive Value (PPV), Negative Predictive Value (NPV), False Positive Rate (FPR) and False Negative Rate (FNR) on the test packets. These terms are defined as:

1. False Positive (FP) is the number of packet isolated and handled accordingly that are not malicious
2. False Negative (FN) is the number of malicious packets the algorithm fails to isolate and handle accordingly.
3. True Positive (TP) is the number of malicious packet successfully isolated and handled accordingly.
4. True Negative (TN) is the number of non-malicious

packet identified as malicious and isolated and handled accordingly.

5. Sensitivity, also called True Positive Rate (TPR), is a measure of how well the algorithm correctly isolates and handles malicious packet.
6. Positive Predictive Value (PPV), also known as precision, is the probability that a positive prediction is correct.
7. Negative Predictive Value (NPV) is the probability that a negative prediction is correct.
8. False Positive Rate (FPR), is the measure of how much the algorithm fails to isolate and handle accordingly malicious packets.

False Negative Rate (FNR) measures of how much the algorithm isolate and handle accordingly non-malicious packets.

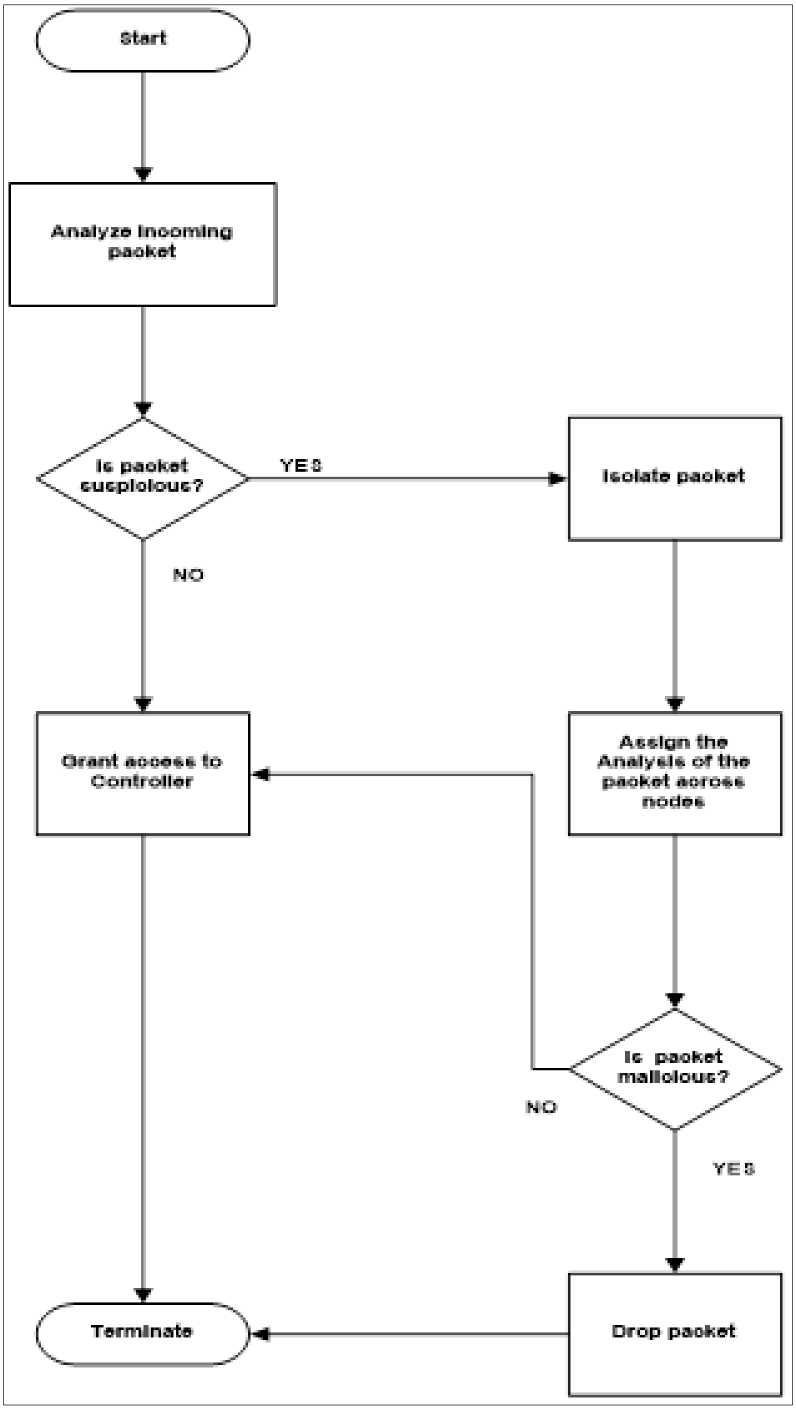


Fig 7: Proposed Algorithm Flowchart

3.1 Simulation Setup

The experimental setup for SDN consists of a controller, open flow switches and hosts as shown in figure 8. For conducting analysis on SDN we are using Mininet. Mininet is SDN network emulator based on Linux. It consists of miniEdit tool which is used for creating the network topology. First the setup is tested for defined topologies with hub code. The hub code is added with the functionalities L2 learning switch. Then setup is tested with openflow supported switch. For simulation purpose there are various tools which are used for analysis of SDN. A virtual image of mininet is provided by github that need to be imported in virtual box. This image does not support graphics so it is needed to use xming server on the host computer. Host computer is used to connect with mininet image. Several network analysis utilities have been used with mininet for conducting the experiments on the reference monitor. For checking the real time traffic patterns of the algorithms wireshark was used. Wireshark is a utility which is used for packet filtering and network analysis in the network.

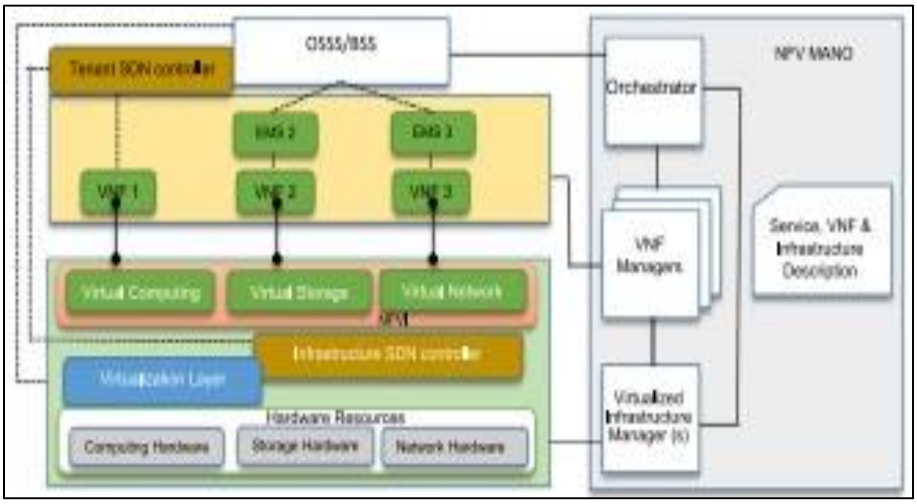
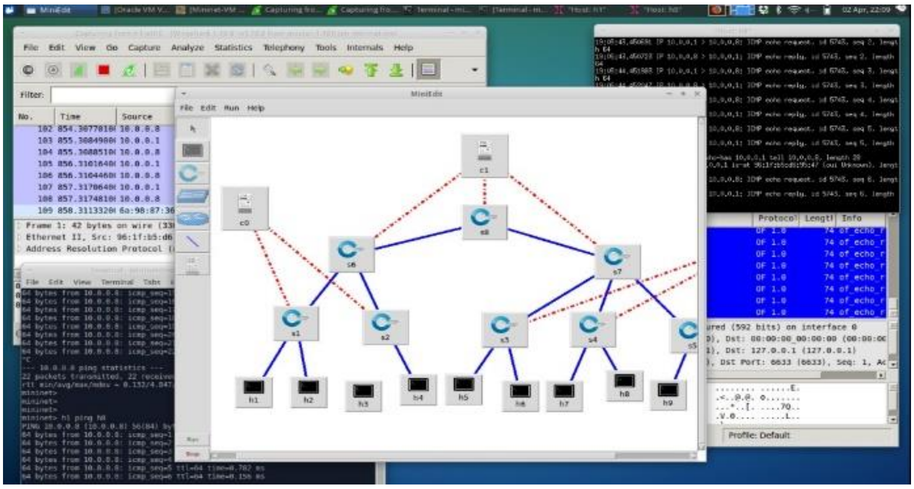


Fig 8: The Integration of SDN Controller

3.2 Implementation of the Algorithm

Proposed architecture has been implemented by using Mininet emulator tool, which is inexpensive and quickly configurable network emulator. Mininet is a standard Linux based networking emulator where virtual topologies like virtual host, switch and link can be created. It also supports OpenFlow protocol which can be used for computer network based SDN simulation. Mininet is also great way to enhance, share, and experiment with OpenFlow and Software-Defined Networking systems. By single command Mininet creates realistic virtual network, runs collection of end-hosts, switches, routers, and links on a single machine (VM, cloud or native). Mininet released under a permissive BSD Open Source license which is actively developed and supported.



```
102 854.3077818( 10.0.0.8
103 855.3084980( 10.0.0.1
104 855.3088510( 10.0.0.8
105 856.3101640( 10.0.0.1
106 856.3104460( 10.0.0.8
107 857.3170640( 10.0.0.1
108 857.3174810( 10.0.0.8
109 858.3113320( 6a:98:87:36
Frame 1: 42 bytes on wire (33
Ethernet II, Src: 96:1f:b5:d6
Address Resolution Protocol (
```

```
--- 10.0.0.8 ping statistics ---
22 packets transmitted, 22 receive
mininet>
mininet>
mininet> h1 ping h8
PING 10.0.0.8 (10.0.0.8) 56(84) by
64 bytes from 10.0.0.8: icmp_seq=5 ttl=64 time=0.782 ms
64 bytes from 10.0.0.8: icmp_seq=6 ttl=64 time=0.156 ms
```

```
OF 1.0    74 of_echo_r
OF 1.0    74 of_echo_r
OF 1.0    74 of_echo_r
OF 1.0    74 of_echo_r
OF 1.0    74 of_echo_r
OF 1.0    74 of_echo_r
OF 1.0    74 of_echo_r
ured (592 bits) on interface 0
0), Dst: 00:00:00_00:00:00 (00:00:0
1), Dst: 127.0.0.1 (127.0.0.1)
), Dst Port: 6633 (6633), Seq: 1, Ac
```

Fig 9: Virtual Network with end-hosts, switches, routers, and links on a single machine

3.3 Performance Evaluation

To test the performance and functionality of the automatically generated reference monitor, we manually wrote the code to link the reference monitor with cryptographic libraries and administrative interfaces. These components were then integrated into a HTTP server. While building the test system around the verified reference monitor was not conceptually challenging, it was necessary to make a number of modifications and extensions to the tools and utilities we used. In addition to providing a working system that confirms our belief in the potential for verified components of a trusted computing base, we also produced improvements in a standard web server and cryptographic library interface that will be useful for other projects.

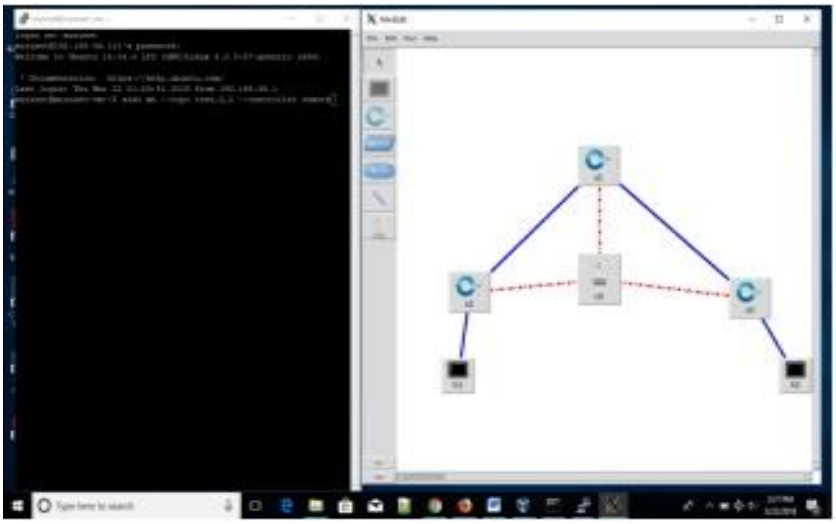


Fig 10: Implementation Scenario of SDN from Mininet

Figure 10, shows the implementation scenario of SDN based proposed security application with SDN/Openflow. When a source host send a data packet to destination, the openflow switch check for the matching entry in flow table if a match is found in switch flow table the related action is taken, i.e. the packet is dropped or send to the destination. If no match is found the packet is sent to the controller. The controller sends the packet to the security application policy analysis. The security application first parses the received packet, checks whether the incoming packet violates the security policies or not and enforces a flow rule based on the security policies. Finally this rule is delivered to switch by the SDN controller and switch update the rule in its flow table. Packet is blocked based on some event associated with an attack signature in the openflow network through Packet_event messages and further packets from this sender blacklisted by the security application. Moreover with the programmability in handling suspicious and malicious traffic can also be redirected to a sandbox or quarantine dynamically as per demand.

The novelty is

1. Network monitoring and reporting in SDN is more powerful with centralized view and control of the network through the controller.
2. Most of the security algorithms supports and work more efficiently on centralized environment as compare to distributed approach which is best fit for network threat detection in SDN.
3. With the help of programmability and control, we can generate dynamic responses to the network threats in a more effective way.

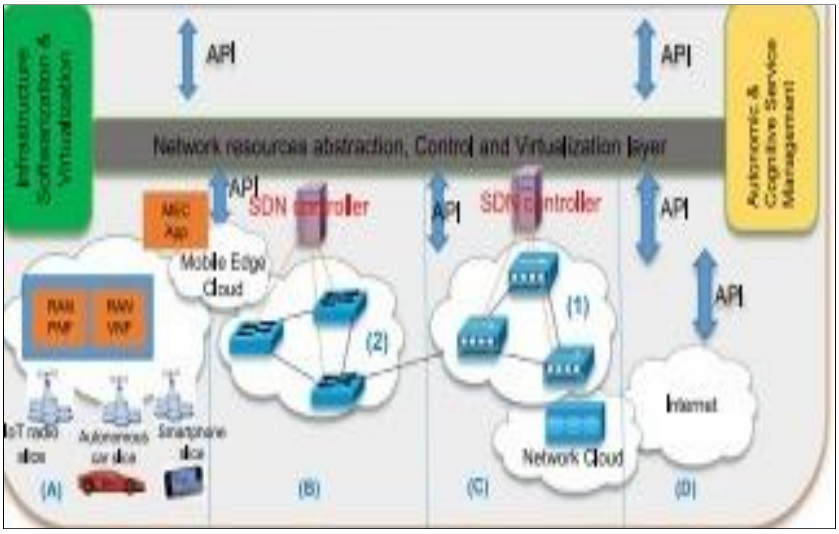


Fig 11: Centralized View of Network Monitoring and Reporting in SDN

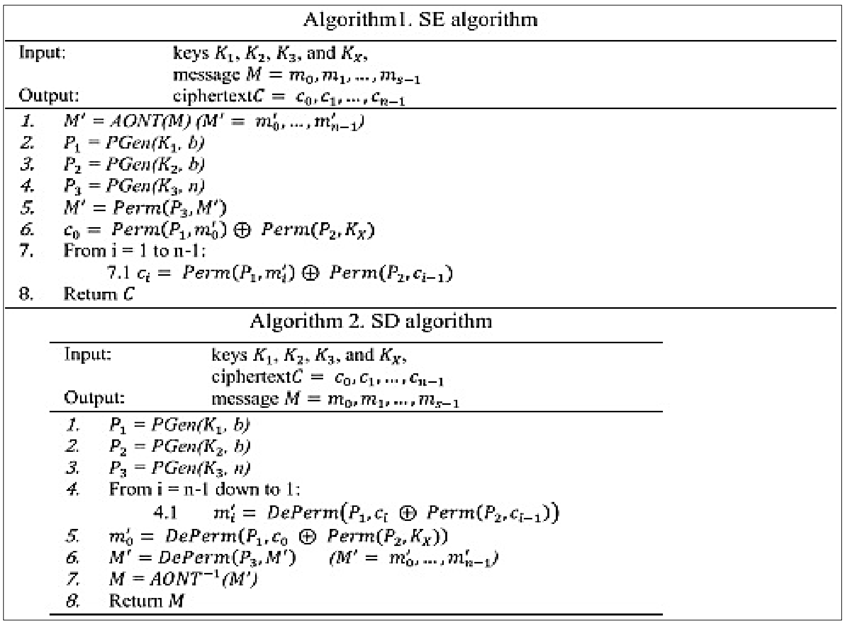
3.3.1 Symmetric Encryption Module

Symmetric Encryption Module (SEM) provides a symmetric key cryptosystem to encrypt resources in our system. There are three functions in this module:

1. Symmetric Key Generator (SKG): in this function three random k-bit keys K1, K2, and K3 and a random b-bits key KX are generated using RandGen function. Here, k is the size of key in Pgen algorithm and b is the size of each block of original message M in AONT algorithm. The resulted key is K = (K1, K2, K3, KX), which is used in SE function, SD function.

2. Symmetric Encryption (SE): the function encrypts message M using the keys K1, K2, K3, and KX. Algorithm 1 shows the SE algorithm.

3. Symmetric Decryption (SD): the function decrypts ciphertext C using the keys K1, K2, K3, and KX. The SD algorithm is shown in Algorithm 2.



Algorithm1. SE algorithm

```
Input:    keys $K_1, K_2, K_3$, and $K_X$,
          message $M = m_0, m_1, \dots, m_{s-1}$
Output:   ciphertext$C = c_0, c_1, \dots, c_{n-1}$
1. $M' = AONT(M)$ $(M' = m_0', \dots, m_{n-1}')$
2. $P_1 = PGen(K_1, b)$
3. $P_2 = PGen(K_2, b)$
4. $P_3 = PGen(K_3, n)$
5. $M' = Perm(P_3, M')$
6. $c_0 = Perm(P_1, m_0') \oplus Perm(P_2, K_X)$
7. From i = 1 to n-1:
       7.1 $c_i = Perm(P_1, m_i') \oplus Perm(P_2, c_{i-1})$
8. Return $C$
```

Algorithm 2. SD algorithm

```
Input:    keys $K_1, K_2, K_3$, and $K_X$,
          ciphertext$C = c_0, c_1, \dots, c_{n-1}$
Output:   message $M = m_0, m_1, \dots, m_{s-1}$
1. $P_1 = PGen(K_1, b)$
2. $P_2 = PGen(K_2, b)$
3. $P_3 = PGen(K_3, n)$
4. From i = n-1 down to 1:
       4.1   $m_i' = DePerm(P_1, c_i \oplus Perm(P_2, c_{i-1}))$
5. $m_0' = DePerm(P_1, c_0 \oplus Perm(P_2, K_X))$
6. $M' = DePerm(P_3, M')$     $(M' = m_0', \dots, m_{n-1}')$
7. $M = AONT^{-1}(M')$
8. Return $M$
```

3.3.2 Asymmetric Encryption Module

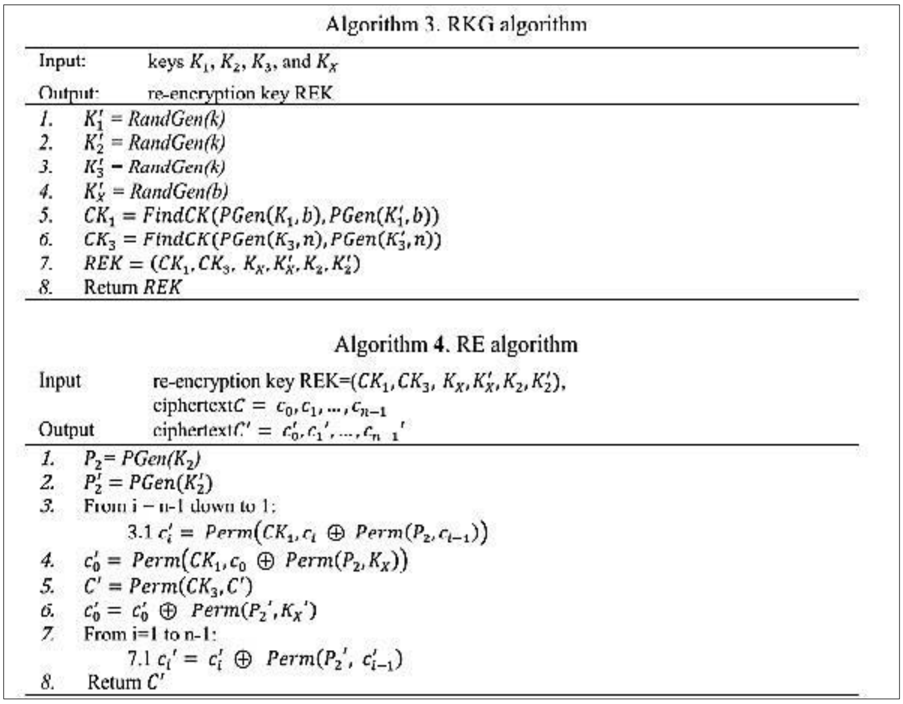
The aim of Asymmetric Encryption Module (AEM) is to provide a public-key cryptosystem used for sharing the keys of resources. It contains three functions including, Asymmetric Key Generator (AKG), Asymmetric Encryption (AE), and Asymmetric Decryption (AD). The module can be implemented using a public-key cryptosystem such as RSA.

3.3.3 Re-Encryption Module

Re-Encryption Module (REM) consists of two functions to provide a proxy re-encryption mechanism for access control policy updates:

1. Re-encryption Key Generator (RKG): the function generates re-encryption keys sent to the server when policy is updated. It is implemented using Algorithm 3.

2. Re-Encryption (RE): the function is used when the data owner needs to update his policies by re-encrypting the cipher-text with the new key. Details of this function are shown in Algorithm 4.



Algorithm 3. RKG algorithm

```
Input:    keys $K_1, K_2, K_3$, and $K_X$
Output:   re-encryption key REK
1. $K_1' = RandGen(k)$
2. $K_2' = RandGen(k)$
3. $K_3' - RandGen(k)$
4. $K_X' = RandGen(b)$
5. $CK_1 = FindCK(PGen(K_1, b), PGen(K_1', b))$
6. $CK_3 = FindCK(PGen(K_3, n), PGen(K_3', n))$
7. $REK = (CK_1, CK_3, K_X, K_X', K_2, K_2')$
8. Return $REK$
```

Algorithm 4. RE algorithm

```
Input     re-encryption key REK=$(CK_1, CK_3, K_X, K_X', K_2, K_2')$,
          ciphertext$C = c_0, c_1, \dots, c_{n-1}$
Output    ciphertext$C' = c_0', c_1{}', \dots, c_{n-1}{}'$
1. $P_2 = PGen(K_2)$
2. $P_2' = PGen(K_2')$
3. From i – n-1 down to 1:
       3.1 $c_i' = Perm(CK_1, c_i \oplus Perm(P_2, c_{i-1}))$
4. $c_0' = Perm(CK_1, c_0 \oplus Perm(P_2, K_X))$
5. $C' = Perm(CK_3, C')$
6. $c_0' = c_0' \oplus Perm(P_2{}', K_X{}')$
7. From i=1 to n-1:
       7.1 $c_i{}' = c_i' \oplus Perm(P_2{}', c_{i-1}')$
8. Return $C'$
```

4. Expected Outcome/Contribution to Knowledge

At the end of this research, integrated reference monitor to SDN controllers should be able isolate and handle malicious packets in a distributed approach among nodes rather than permitting or denying access based on access policy. This would facilitate SDN to have Intrusion Detection System (IDS) capabilities rather than depending on expensive IDS.

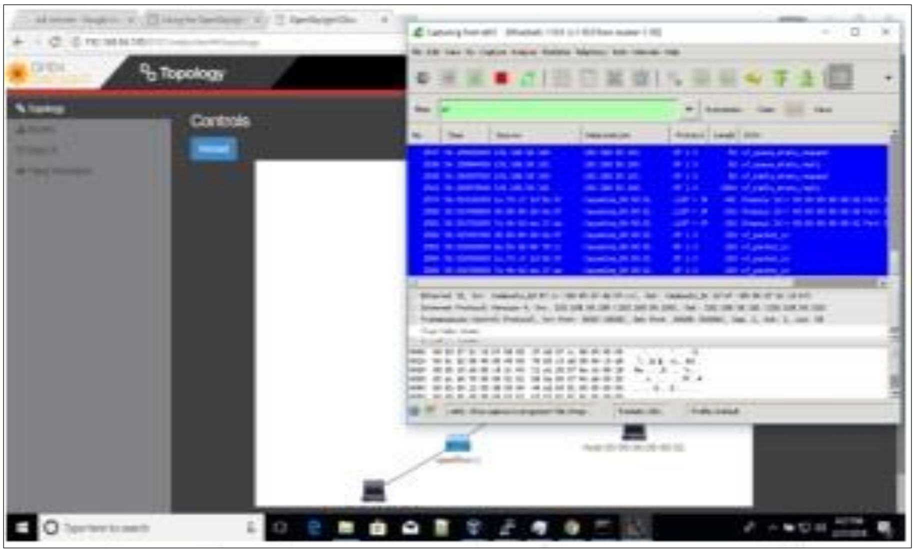


Fig 12: Handling Packets based on Access Policy

In a typical selective encryption based access control enforcement mechanism, changing policy may need the update of some resources' encryption keys. Therefore, these resources should be re-encrypted using new encryption keys. Reaching this purpose necessitates conducting the three steps of receiving the resource from the server, decrypting it with the old key and re-encrypting it with the new key, and finally, sending the encrypted resource to the server.

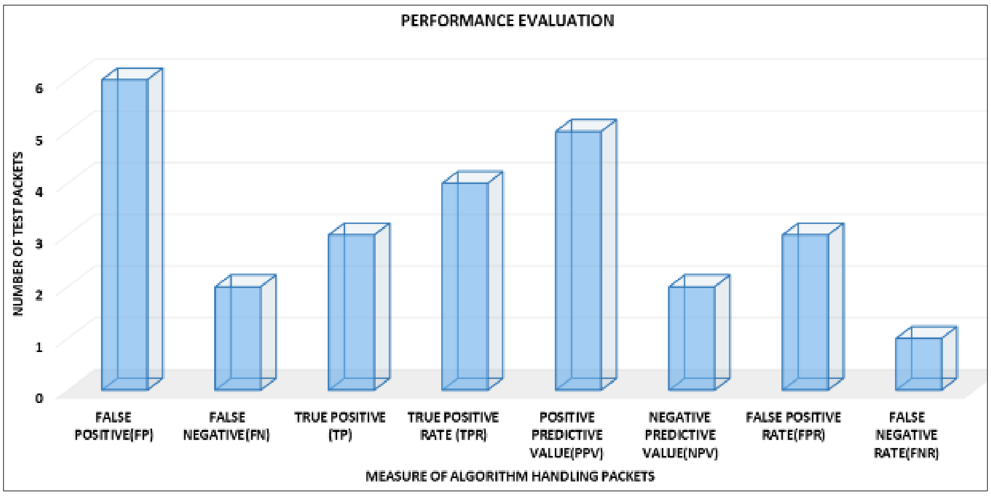


Fig 13: Performance Evaluation Chat

An attempt has been made to find the number of packets isolated and handled accordingly that are not malicious which is 6 and the number of non-malicious packet identified as malicious and isolated and handled accordingly is 2, Sensitivity, also called True Positive Rate (TPR), is the measure of how well the algorithm correctly isolates and handles malicious packet to be 4. We measure precision to be 5 that is the Positive Predictive Value (PPV) as the probability that a positive prediction is correct, but the Negative Predictive Value (NPV) is 2 which is the probability that a negative prediction is correct. Then we measure False Positive Rate (FPR) and False Negative Rate (FNR) to be 3 and 1 respectively that indicates the measure of how much the algorithm fails to isolate and handle malicious packets accordingly and the measures of how much the algorithm isolate and handle non-malicious packets accordingly. It is concluded from Fig.13 that False Positive has the highest number of packets isolated (without malicious attack) have more in PPR as compared to TPR with number of malicious attack because under attacks it does not allow the more packets to pass to their neighbor nodes. As Fig. 13 indicates that False Negative Rate (FNR) decreases drastically with the least number of malicious node and drops less packets because it doesn't allow the packet to flow further.

5. Conclusion

Both academia and industry are embracing SDN and NFV at unprecedented speed as technologies to overcome the challenge of management and orchestration of resources in 5G networks and meet different vertical's requirements. SDN and NFV promise to provide and implement new capabilities and solutions for enabling future 5G networks control and management to be adaptable, programmable and cost-effective. As it was shown, many different researches are trying to provide faster and more reliant base for 5G wireless networks. SDN as the main component of providing the virtualization, gained increasingly attraction. In this paper, a survey among different recent papers in this area has been carried out, and the main goal of each technique to improve different parts of this scheme has been reviewed. Meanwhile, the proposed architectures and basic rules of each paper have been extensively clarified. However, there can be more ways to develop these schemes, as the 5G is still at the middle stages of researches. The 5G network will consist of a huge number of devices, applications and technologies. Sharing the spectrum and the bandwidth over each single LAN domain among larger and larger number of users is an avoidable concepts. Providing more flexible network with high rate throughput is still needed to be more investigated. Also a number of different algorithms and a considerable amount of empirical have done to compute least path in existing network. The d implementations of minimum path algorithms have remained important research topics within related disciplines such as operations research and computer science. This paper implements reference Monitor Algorithm for Software Defined Networking controller and Mininet emulator. For future works, we will implement shortest path algorithm in SDN and compare the result with each other to compute the best minimum path finding algorithm.

Acknowledgment

This research work was supported by the TETfund Institutional Based Research (IBR) through the Center for Research and Development (CRD) of Federal University Gusau, Zamfara State, Under the TETF/DR&S/CE/UNIV/GUSAU/IBR/2020/VOL.1 grant allocation.

References

1. 5G-ENSURE. Deliverable D3.2 5G-PPP security enablers open specifications; c2016. p. 1.0. Available from: https://5g-ppp.eu/5g-ensure-publishes-its-5g-ppp-security-enabler-open-specifications-v1/. Accessed 2016 Jun 12.
2. Gudipati A, Perry D, Li LE, Katti S. Soft RAN: software defined radio access network. Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking; c2013.
3. Adrian BM, Louis M, Evangelos R, George S, Nikolaos P. Threat landscape and good practice guide for software defined networks/5G. European Union Agency for Network and Information Security (ENISA); c2015. DOI: 10.2824/67261.
4. Akhunzada A, Ahmed E, Gani A, Khan MK, Imran M, Guizani S. Security and privacy in emerging networks: securing software defined networks: taxonomy, requirements, and open issues. IEEE Communications Magazine. 2019; 57(1):34-44.
5. Hakiri A, Berthou P. Leveraging SDN for the 5G networks: trends, prospects and challenges. arXiv. 2020;1506.02876.
6. Akyildiz IF, Wang P, Lin SC. SoftAir: a software defined networking architecture for 5G wireless systems. Computer Networks. 2015;85:1-18.
7. Ali ST, Sivaraman V, Radford A, Jha S. A survey of securing networks using software defined networking. IEEE Transactions on Reliability. 2019;68(3):879-896.
8. Anonymous. Software-Defined Networking (SDN) definition; c2019. Available from: http://www.opennetworking.org. Accessed 2007 Mar 5.
9. Anthony L. Security risks in SDN and other new software issues. RSA Conference. Frost and Sullivan; c2019.
10. Chen M, Qian Y, Mao S, Tang W, Yang X. Software-defined mobile networks security. Mobile Networks and Applications; c2015 .p. 1-15.
11. Cho HH, Lai CF, Shih TK, Chao HC. Integration of SDR and SDN for 5G. IEEE Access. 2014;2:1196-1204.
12. Colville J, Spafford G. Configuration management for virtual and cloud infrastructures. Gartner Inc; c2021.
13. Dabbagh M, Hamdaoui B, Guizani M, Rayes A. Software-defined networking security: pros and cons. IEEE Communications Magazine, Communications Standards Supplement; c2018.
14. Feldmann A, Kind M, Maennel O, Schaffrath G, Werle C. Network virtualization - an enabler for overcoming ossification. Future Internet Technology. European Community in Information Technology (ERCIM) News; c2021.
15. Sun G, Liu F, Lai J, Liu G. Software defined wireless network architecture for the next generation mobile communication: proposal and initial prototype. Journal of Communications; c2020-2021.
16. Hong S, Xu L, Wang H, Gu G. Poisoning network visibility in software-defined networks: new attacks and countermeasures. Paper presented at the NDSS; c2015.
17. Horn G, Schneider P. Towards 5G security; c2015.
18. Nam H, Calin D, Schulzrinne H. Intelligent content delivery over wireless via SDN. IEEE WCNC; c2019.
19. Kreutz D, Ramos FMV, Verissimo P. Towards secure and dependable software defined networks. Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking. ACM; c2018 .p. 55-60.
20. Kreutz D, Ramos FMV, Verissimo P. Software-defined networking: A comprehensive survey. Proceedings of the IEEE. 2019;103(1):55-60.
21. Lantz B, O'Connor B. A Mininet-based virtual testbed for distributed SDN development. Paper presented at the Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication; c2015.
22. Li Y. Computer networks. 2020;72:74-98.
23. Liu S, Li B. On scaling software-defined network in wide-area networks. Tsinghua Science and Technology. 2020;20(3):221-232.
24. Liyanage M, Ahmed I, Ylianttila M, Santos JL, Kantola R, Perez OL, Jimenez C. Security for future software defined mobile networks. Paper presented at the Next Generation Mobile Applications, Services and Technologies, 2015 9th. International Conference on; c2015.
25. Malik MS, Montanari M, Huh JH, Bobba RB, Campbell RH. Towards SDN enabled network control delegation in clouds. 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN); c2020.
26. Mehdi SA, Khalid J, Khayam SA. Revisiting traffic anomaly detection using software defined networking. Proceedings of 14th International Symposium on Recent Advances in Intrusion Detection (RAID). 2019;6961:161-180.
27. Metzler J. Understanding software-defined networks. Information Week Reports; c2021 .p. 1-25.
28. Nikaein N, Marina MK, Manickam S, Dawson A, Knopp R, Bonnet C. OpenAirInterface: A flexible platform for 5G research. ACM SIGCOMM Computer Communication Review. 2014;44(5):33-38.
29. Szabo ND, Nemeth F, Sonkoly B, Gulyas A, Fitzek FHP.

Towards the 5G revolution: a software defined network architecture exploiting network coding as a service. Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication; c2019.

30. NetWorks. Advanced 5G network infrastructure for the future Internet; c2013. Available from: https://5g-ppp.eu/wp-content/uploads/2014/02/Advanced-5G-Network-Infrastructure-PPP-in-H2020_Final_November-2013.pdf. Accessed 2016 Jun 13.
31. Giorgetti A, Sgambelluri A, Casellas R, Morro R, Campanella A, Castoldi P. Control of open and disaggregated transport networks using the Open Network Operating System (ONOS). Journal of Optical Communications and Networking. 2020;12(2):A171-A181.
32. Vilalta R, Manso C, Yoshikane N, Casellas R, Martinez R, Tsuritani T, Morita I, Munoz R. Experimental evaluation of control and monitoring protocols for optical SDN networks and equipment [Invited Tutorial]. Journal of optical communications and networking. 2021;13(8):D1-D2.
33. Open Network Foundation. Open Flow-enabled mobile and wireless networks. ONF Solution Brief; c2013.
34. Panwar N, Sharma S, Singh AK. A survey on 5G: the next generation of mobile communication. Physical Communication; c2015.
35. Jahan R. Unlocking the true potential of 5G: techniques for latency reduction; c2020.
36. Sánchez J, Yahia B, Grida I, Crespi N, Rasheed T, Siracusa D. Softwarized 5G networks resiliency with self-healing. Paper presented at the 5G for Ubiquitous Connectivity (5GU), 2014 1st International Conference on; c2014.
37. Scott-Hayward S, Natarajan S, Sezer S. A survey of security in software defined networks. IEEE Communications Surveys & Tutorials. 2015;18(1):623-654.
38. Shin S, Gu G. Cloud Watcher: network security monitoring using OpenFlow in dynamic cloud networks. Springer; c2018 .p. 92-103.
39. Shin S, Porras P, Yegneswaran V, Fong M, Gu G, Tyson M. FRESCO: modular composable security services for software-defined networks. NDSS Network and Distributed System Security Symposium; c2018.
40. Shin S, Yegneswaran V, Porras PA, Gu G. AVANT-GUARD: scalable and vigilant switch flow management in software-defined networks. Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security; c2018 .p. 413-424.
41. Shirali-Shahrez S, Ganjali Y. FleXam: flexible sampling extension for monitoring and security applications in OpenFlow. Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking; c2018 .p. 167-168.
42. Son S, Shin S, Yegneswaran V, Porras PA, Gu G. Model checking invariant security properties in OpenFlow. Proceedings of IEEE ICC; c2018 .p. 74-79.
43. Suomalainen L, Nikkhouy E, Ding AY, Tarkoma S. Open source platforms, applications and tools for software-defined networking and 5g research. Department of Computer Science, University of Helsinki, Technical Report, Series of Publications C-2014-2; c2014.
44. Trivisonno R, Guerzoni R, Vaishnavi I, Soldani D. SDN-based 5G mobile networks: architecture, functions, procedures and backward compatibility. Transactions on Emerging Telecommunications Technologies. 2015;26(1):82-92.
45. Chin WH, Fan Z, Haines R. Emerging technologies and research challenges for 5G wireless networks. IEEE Wireless Communications. 2014;21(2):106-112.
46. Wang Y, Wen X, Chen Y, Hu C, Shi C. Towards a secure controller platform for OpenFlow applications. Proceedings of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking; c2019 .p. 171-172.
47. Lin YD, Lin PC, Yu CH, Wang YC, Lai YC. An extended SDN architecture for network function virtualization with a case study on intrusion prevention. IEEE Network; c2017.
48. Ma Z, Zhang ZQ, Ding ZG, Fan PZ, Li HC. Key techniques for 5G wireless communications: network architecture, physical layer, and MAC layer perspectives. Science China Information Sciences. 2020;58(4):1-20.