Internet of Things 12 (2020) 100289

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Review article

# A survey on the architecture, application, and security of software defined networking: Challenges and open issues

Kashif Nisar[a,b,f,*], Emilia Rosa Jimson[a], Mohd Hanafi Ahmad Hijazi[a], Ian Welch[b], Rosilah Hassan[c], Azana Hafizah Mohd Aman[c], Ali Hassan Sodhro[d], Sandeep Pirbhulal[e], Sohrab Khan[f]

[a] Faculty of Computing and Informatics, University Malaysia Sabah, Kota Kinabalu, Sabah, Malaysia
[b] School of Engineering and Computer Science, Victoria University of Wellington, New Zealand
[c] Centre for Cyber Security, Faculty of Information Science & Technology, Universiti Kembagsaan Malayasia, UKM, 43600 Bangi, Selangor, Malaysia
[d] Department of Computer and Information Science, Linkoping University, Linkoping 58183, Sweden
[e] Department of Information Security and Communication Technology, Norwegian University of Science and Technology, Norway
[f] Computer Systems Engineering Department, Balochistan University of Engineering and Technology, Khuzdar, Pakistan

ARTICLE INFO

*Article history:*
Received 25 February 2020
Revised 13 July 2020
Accepted 25 August 2020
Available online 9 September 2020

*Keywords:*
SDN
OpenFlow
Control plane
SDN security

ABSTRACT

Software Defined Networking (SDN) is a new technology that makes computer networks farther programmable. SDN is currently attracting significant consideration from both academia and industry. SDN is simplifying organisations to implement applications and assist flexible delivery, offering the capability of scaling network resources in lockstep with application and data. This technology allows the user to manage the network easily by permitting the user to control the applications and operating system. SDN not only introduces new ways of interaction within network devices, but it also gives more flexibility for the existing and future networking designs and operations. SDN is an innovative approach to design, implement, and manage networks that separate the network control (control plane) and the forwarding process (data plane) for a better user experience. The main differentiation between SDN and Traditional Networking is that SDN removes the decision-making part from the routers and it provides, logically, a centralised Control-Plane that creates a network view for the control and management applications. Through the establishment of SDN, many new network capabilities and services have been enabled, such as Software Engineering, Traffic Engineering, Network Virtualisation and Automation, and Orchestration for Cloud Applications. This paper surveys the state-of-the-art contribution such as a comparison between SDN and traditional networking. Also, comparison with other survey works on SDN, new information about controller, details about OpenFlow architecture, configuration, comprehensive contribution about SDN security threat and countermeasures, SDN applications, benefit of SDN, and Emulation & Tested for SDN. In addition, some existing and representative SDN tools from both industry and academia are explained. Moreover, future direction of SDN security solutions is discussed in detail.



* Corresponding author at: Faculty of Computing and Informatics, University Malaysia Sabah, Kota Kinabalu, Sabah, Malaysia.
*E-mail address:* kashif@ums.eud.my (K. Nisar).

https://doi.org/10.1016/j.iot.2020.100289
2542-6605/

## 1. Introduction

Software Defined Networking (SDN) is an emerging networking example that splits the network control plane from the data forwarding plane with the assurance to dramatically advance network resource utilisation, simplify network management, reduce operating. SDN has been mostly considered and progressively adapted for campus networks, data centre networks, Wide Area Networks (WANs), enterprise networks, and Internet exchange points [1]. The increase of cloud services resulting in the unprecedented growth of both public and private cloud services adds to the complexity. IT's planning for cloud services must be done in an environment of increased security, compliance, and auditing requirements, along with business reorganisations, consolidations, and mergers that can change assumptions overnight. Providing self-service provisioning, whether in a private or public cloud, requires elastic scaling of computing, storage, and network resources, ideally from a common viewpoint and with a common suite of tools [2]. The traffic patterns have obviously changed within the enterprise information centre. Today's applications access different databases and servers, creating a flurry of "east-west" machine-to-machine traffic before returning data to the end user device in the classic "north-south" traffic pattern. Besides that, users are changing network traffic patterns as they push for access to corporate content and applications from any type of device. The increase of personal devices has put the Information Technology (IT) under pressure in order to protect the corporate data and intellectual property in a delicate manner.

The details of the SDN architecture overview are explained as follows: There may be more than one SDN controller if the network is large-scale or a wide-area region network. The control layer globally regulates the network states via network policies in either a centralised or distributed manner. SDN is a technology that introduces a new network architecture, where the Control and Data Planes are decoupled [3]. This new technique has been adopted in several fields and companies including the environmental engineering for sustainability [4]. Due to the unrestricted access to global network elements and resources, such network policies can be updated timely to react to the current flow activities shown in Fig. 1.

SDN introduces an abstraction [188], where the Control-Plane and Data-Plane are decoupled [50]. In SDN, the forwarding state (in the Data-Plane) is controlled by the SDN controller [222]. This means the control functionality is removed from the network devices. The forwarding decisions are Flow-Based, and in the SDN/OpenFlow concept, a flow is a sequence of packets between a source and a destination [230]. The flow is defined by a set of packet field values acting as a match (filter) criterion and a set of actions (instructions). Each packet of flow receives identical service policies at the forwarding devices [51,52]. The OpenFlow Protocol [53] is a mechanism that allows the Control-Plane Layer to communicate with the Data-Plane Layer. This protocol is a standard protocol for communication over North-Bound [54] and South-Bound [55] APIs. The SDN controller [232,236] configures the forwarding devices with the help of the OpenFlow Configuration and Management

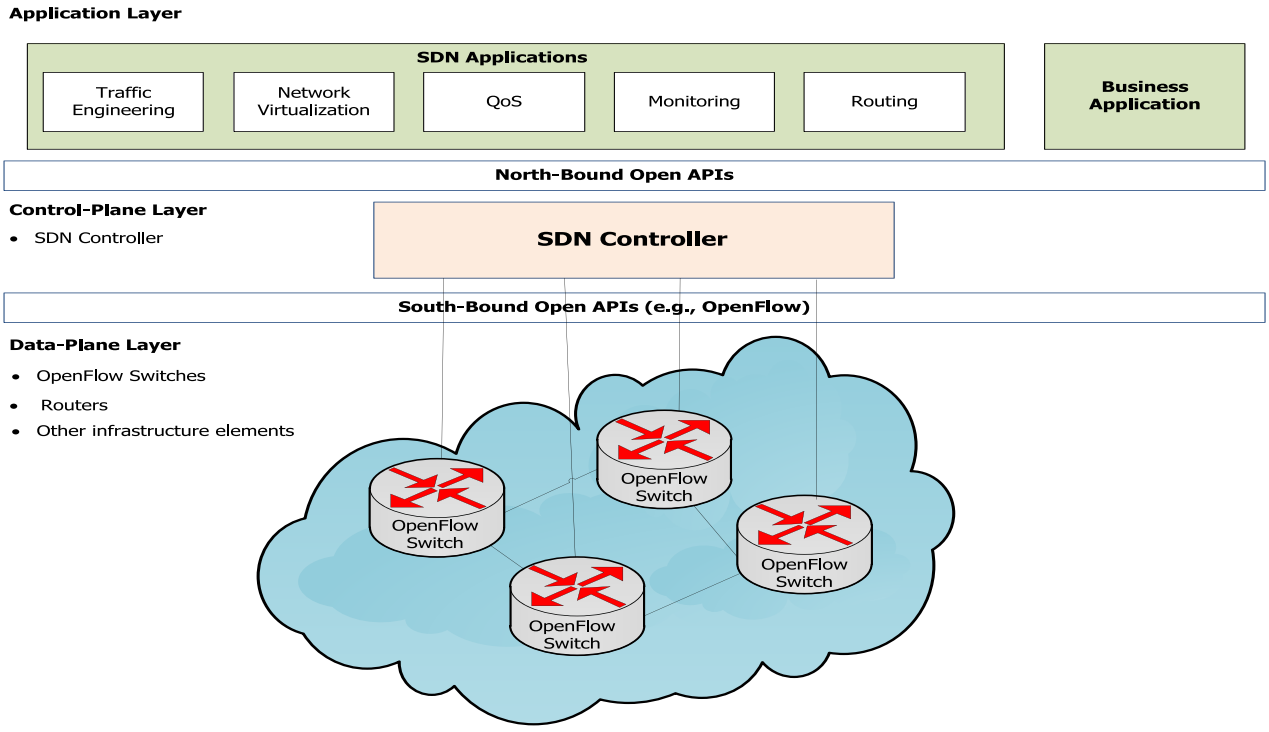


**Fig. 1.** Overview of software defined networking architecture.

Protocol (OF-Config) and the Open vSwitch Database Management Protocol (OVSDB). Besides that, the OF-Config and OVSDB also act as specific extensions to OpenFlow. Send Packet Out, Packet Received, and Modify [56–62].

### 1.1. Background of software defined networking

Open Networking Foundation (ONF) states that SDN is an "Emerging architecture that is dynamic, manageable, cost-effective, and adaptable, thus making it ideal for the high-bandwidth, dynamic nature of today's applications" [5]. The SDN architecture is divided into three (3) layers: the Application, Control, and Data Planes. SDN represents a rapid prominent networking model promising to ensure an end-to-end QoS promised by affording a more important network flexibility, abstraction, control, management, and programmability to network resources [6]. In 2007, SDN was invented at Stanford University by Martin Casado [7]. It was a collaboration between Martin Casado, a PhD student of Stanford University [8]; Professor Nick McKeown who was Martin Casado's academic supervisor in the Electrical Engineering and Computer Science department [9,10] and Scott Shenker, the professor of Computer Science at the University of California, Berkeley [11]. The idea of the SDN invention came from Martin Casado's PhD thesis entitled "Architectural Support for Security Management in Enterprise Networks". Through his research, a flow-based Ethernet switch controlled centrally from the outside was introduced, which is now known as "Ethane" [92]. He discovered the Ethane model to develop the OpenFlow Protocol, and developed a program software to accomplish a range of control applications. All these activities contributed to the basic concept of the SDN development.

Before SDN was invented, the intention to create a programmable network had long been thought of, for example the researchers in [12–18] supported high speed programmable packet processing. This obviously shows that the researchers had tried to create a programmable network. The team of these three doctors worked hard to make the network more programmable in order to solve technological problems. As an example, because the internet is a distributed network where individual communication devices exchange control information according to certain protocols, a new protocol needed to be created and all the communication nodes had to be redesigned to make sure each communication node would be compatible with the new protocol [19]. To solve the problem, they introduced SDN which made the communication nodes externally controllable by implementing abstraction based on flow tables for each communication node and introduced the right abstraction in the network control [20]. Besides that, they endorsed the concept of flexible software-based control of a whole network and clarified the layers at which to abstract external software control. Furthermore, several challenge are discussed [229]. The SDN term was first coined by the [21] article which discussed the OpenFlow project at Stanford University. This article was published by MIT Technology Review [22,23].The effort to create programmable networks has been carried out over the years. The main reason for programmable networking is to make the implementation of new network services easier, which leads to the process of service formation and deployment [24]. Table 1 shows early programmable network efforts that have become the SDN foundation. SDN started when its concepts were first explored in the Ethane project [25]. Although SDN has just been introduced, many IT experts expect that deployments of SDN will increase rapidly over the next few years and, at the same time, will face many challenges that come from different aspects. In this section, several challenges are discussed.

#### 1.1.1. Security

In terms of security, [231] the SDN faces challenges to develop SDN applications that improve network security and to secure the SDN infrastructure itself. The OpenFlow specifications do not define the certificate format to ensure data integrity; this is because only minimum security is specified in SDN. This makes the SDN security need a two factor authentication and mechanism to encrypt for recovery of packets from failure and to avoid hackers [26]. The researchers in [27] divided these security challenges into 3 main problems, which are to secure the SDN infrastructure, to integrate security appliances with network control, and to create languages and control methods that can enforce specified security policies (Table 2).

#### 1.1.2. Controller design

This controller is unable to single-handedly control the whole traffic. To increase scalability, reliability, and integrity, the centralised controller must be physically distributed [28]. Rexford et al. [29] proposed Kandoo to preserve scalability without changing switches, introduced Bottom Layer and Top Layer controls. The Bottom Layer consists of controllers without interconnections and does not have knowledge about the network-wide state whilst the Top Layer maintains the network-wide state and it centralises the controller logically.

#### 1.1.3. Application controller interaction

Fewer protocols have been defined for the communication between application and controller. The researchers in Pyretic [30], Procera [31], Frenetic [32], and FML [33] proposed the use of a network configuration language to express policies. Pyretic motivates programmers in the way to specify a network policy at a high-level abstraction through Pyretic support modular programming, which allows it to mix multiple policies together using Policy Composition Operators, Parallel Composition, and Sequential Composition. Procera uses the functional reactive programming principles that allow operators to express network policies based on reactive and temporal behaviours.

**Table 1**
Early programmable network efforts.

| Network | Overview |
|---|---|
| Active networking [26–38] | • David Tennenhouse called programmable network infrastructure "Active Networks" [38].<br>• Considered Programmable Switch and Capsule approach. The Programmable Switch approach provides a mechanism that will support the downloading of programs and retains the existing packet format [36]. The capsules contain program fragments that can be interpreted and processed via routers [37].<br>• Focus networking never brought widespread industry usage due to security and performance concerns [39] |
| Devolved control of ATM networks (DCAN) [40] | • The fundamental purpose of DCAN was to design and create infrastructure for ATM network management.<br>• SDN concept: DCAN removes control functions from the network device. Besides that, DCAN minimises the protocol between the network and the manager, such as the OpenFlow Protocol. |
| Forwarding and control element separation (ForCES) [41] | • The ForCES Network Element consists of Forwarding Elements and Control Elements. Both of these elements use the ForCES protocol to communicate with each other.<br>• The ForCES Network Element still presents the Forwarding Elements and Control Elements as a single network thing to the industry. |
| 4D project [24] | • 4D Project supported the division between the routing decision logic protocols that govern the communication from network devices.<br>• The Decision-Plane has a universal view of the network that is used to control a Data-Plane to forward traffic over network.<br>• Works like NOX [41] were inspired from the 4D Project. |
| Network configuration protocol (NETCONF) [15,41,223] | • NETCONF is a set of management protocols that modify the configuration of network devices.<br>• NETCONF is a beneficial tool protocol that can be used in parallel on mixed or hybrid switches to support solutions that enable programmable networking. The hybrid SDN, which is composed of SDN devices and traditional network equipment, will exist during a long time. |
| Ethane [42] | • Ethane proposed a centralised controller to control the security & policy over a network.<br>• Similar to SDN, Ethane has two (2) components:<br>• A Central Controller and Ethane Switches.<br>• The Central Controller contains the global network rule that decides the forwarding process, whilst the Ethane switch provides a flow table and a secure channel to the controller.<br>• The basics of the original OpenFlow comes from the switch design in Ethane. |

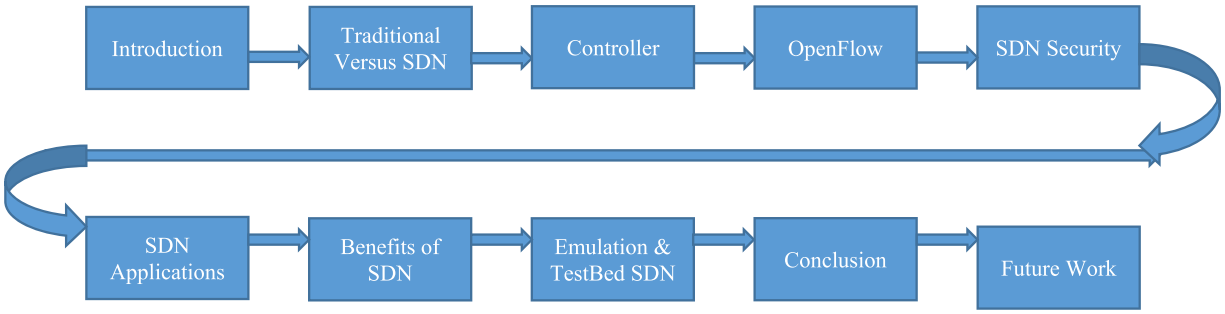


**Fig. 2.** Survey flow diagram.

*1.1.4. Implementation of SDN in traditional networks*

The implementation of SDN is still in the early phase of development [34]. The implementation of SDN in Traditional networks was considered as one of the toughest challenges. The researchers in [35] have identified that unpredictable interaction with other deployed networks, integration with the old networks that do not support the OpenFlow Protocol, architectural updates, deep changes in inter-domain routing protocols, and fundamental errors when emulating SDN under certain limits, are the main aspects of the challenges [43,44].

This paper provide a brief survey on some works that comparison with other survey works on SDN, new information about controller, details about OpenFlow architecture, configuration, comprehensive contribution about SDN security threat and countermeasures, SDN applications, benefit of SDN, and Emulation & Tested for SDN, survey on SDN is given in Fig. 2. As shown in the figure, paper distributes work in nine sections, Section 1 the introduction that how survey carried out the research. Section 2 about traditional networking versus SDN. Section 3 controller and usage in current internet paradigm.

**Table 2**
Our survey comparison with other survey works on software defined networks.

| Survey | Theme | General | Performance | Security | Energy | Rule placement | End System | Formulation | Hardware | Software |
|---|---|---|---|---|---|---|---|---|---|---|
| Shirmarz and Ghaffari [194]; Kreutz et al. [195]; Karakus and Durresi [196]; Ahmad et al. [197] | Overview | Yes | No | No | Yes | No | No | No | No | No |
| Zhang et al. [198] | Security | No | Yes | Yes | No | No | No | No | No | No |
| Memon et al. [199]; Nisar and Ibrahim [200] | Hybrid | No | No | No | No | No | No | No | No | Yes |
| Ndiaye et al. [201]; Nisar and Ibrahim [202]; Coronado et al. [203] | Wireless | Yes | Yes | No | No | No | No | No | No | No |
| Harada et al. [204]; Nisar et al. [[220] | Energy | No | Yes | No | Yes | N0 | Yes | No | Yes | No |
| Montoya-mu et al. [108]; Wang et al. [205]; Rangisetti et al. [206] | Load Balancing | No | No | No | No | No | No | No | No | No |
| Nguyen et al. [183]; Li et al. [207] | Rule Placement | No | Yes | No | Yes | No | Yes | No | No | No |
| Rawat and Reddy [208] | Security & EE | No | No | Yes | Yes | Yes | Yes | No | No | No |
| Baktir et al. [209] | Edge Computing | No | No | No | No | No | No | No | No | No |
| Our survey | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | No | Yes |

Section 4 gives a broad information about OpenFlow protocol. Section 5 discussed SDN security Threats and countermeasures in term of security rules, Illegal access, security attacks on IoT devices, Hardware Trojan Attack, etc., and give reader a comprehensive understanding about SDN approaches and their differences and similarities with each other. Section 6 explores more about SDN applications and gives reader a summarised comparison of SDN with its counterparts. Section 7 is benefits of SND which gives a researchers a new horizon to think about. Section 8 discussed about emulation tool and TestBed for SDN. Finally, conclusions are drawn in Section 9.

## 2. Traditional networking versus software defined networking

In this section, the different concepts between Traditional Networking and Software Defined Networking (SDN) are discussed. Based on [45], Traditional Networking is characterised by two main factors: (1) Most network functionality is implemented in a dedicated appliance, and (2) the dedicated appliance is implemented in dedicated hardware. Dedicated appliance refers to one or multiple switches, routers, and/or application delivery controllers. In traditional networking, each switch has its own Control and Data planes in network, which are known as closed systems. The administrator of traditional networking needs to update each switch inside the network in case he or she wants to deploy new services or protocols in the network. Besides that, in traditional networking, a longer time is needed when an IT administrator needs to make any modification in the network (e.g., adding or removing a single device in the network) since many steps involved in the configuration for each device is made manually. This tends to cause the IT administrator to make errors. After adding or removing a single device in the network, the next step is to update numerous configuration settings (e.g., ACLs, VLANs, QoS [47], 5G, QoE, Artificial Intelligence) by using device-level management tools [191–193]. Overall, the configuration process in traditional networking is complex. This causes an organisation using traditional networking to more likely encounter security breaches.

Nowadays, organisations are using devices from different vendors. In traditional networks, all these devices are placed in the same 'Zone' which contributes to the increased risks of external parties accessing the entire network. Besides that, the organisation faces difficulties in incorporating all these devices within the network in a safe and structured manner. To improve this traditional networking limitation, the SDN concept should be applied to the wired and wireless network [48,49].

### 2.1. Traditional network

In this survey section, the current network architecture is referred as traditional network. The design of the current network devices architecture is vertically integrated, where both the control plane and the data plane are existing in the network devices [195]. The tight integration between these two planes make the network management process complicated [50]. Besides that, since traditional network architecture have distributed control plane, any changes made in the network require the network administrator to configure each of the network devices individually. Manually modifying high-level policies into low-level configuration commands is very challenging and error-prone for a large scale network. This statement is supported by researchers [195], based on their finding network misconfiguration is a common problem experienced in current network architecture and according to finding from scholars Feamster and Balakrishnan [27] there are more than 1000 configurations errors have been detected in Border Gateway Protocol routers, and these errors result in undesirable network behavior such as service contract violations. According to researchers Hilmi et al. [212] the current network architecture unable to fulfill the future network needs since it lacks the ability to view the global network resource and lacks the ability to adapt with the real-time state of the network.

Fig. 3 shows the architecture of traditional network devices, the control plane and data plane are bundled together in hardware component. According to researchers [152] current network design is suffering from several limitations in terms of manageability, flexibility, and extensibility. From the review that made in this research, most of researchers agree that SDN is a solution to overcome the limitations faced in current network architecture. For example, the researchers [189] have stated that the SDN paradigm has been widely adopted in network industry such as telecom industry since it allows flexible network resources allocation, configuration, and management. In the next section, the SDN will be explained in more detail.

### 2.2. Software defined networking

Nowadays, there is a strong interest in the softwarisation of telecommunication networks and cloud computing infrastructure in both industry and academia [211]. The term of softwarisation is referring to the use of a software rather than traditional hardware to solve a particular problem. Softwarisation of the networks has been identified as the way to enable flexibility and simplicity in the network management. SDN has been identified by scholars [211] as a softwarisation enabler. This latest network architecture aims to ease the control and the management of a network. Besides that, this architecture also targets to address the limitation of traditional network architecture. According to definition of the term "Software Defined Networking" given by Open Networking Foundation (ONF) organisation, it is: "A network architecture where the network control plane is decoupled from the forwarding plane, and the control plane controls several devices [83]". The ONF is a non-profit organisation that has developed OpenFlow standard which is the first communication interface for the communication between the control plane and the data plane. Based on the definition given by the ONF the key idea of

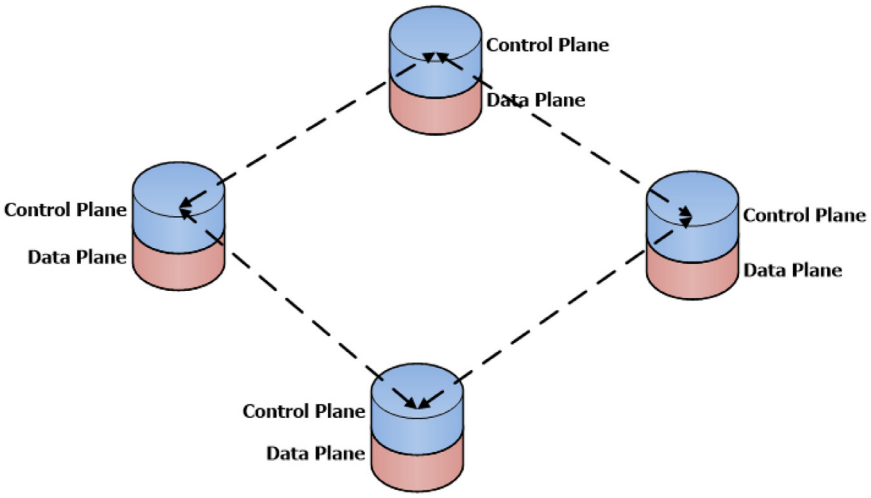


**Fig. 3.** The architecture of traditional network devices.

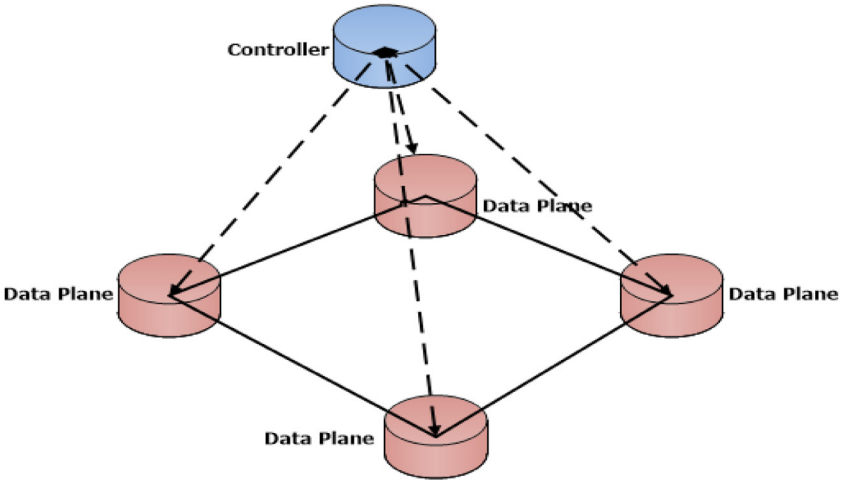


**Fig. 4.** The architecture of software defined networking devices.

SDN architecture is the removal of the control element from the hardware component and the centralisation of the control element in a software component which known as SDN controller. Fig. 4 shows the architecture of SDN devices. The SDN controller is the control intelligence in SDN, it is a software control program. The data plane is remained in the hardware component (e.g., routers and switches). The software-based control program is responsible for managing the forwarding information of the OpenFlow switch, while the hardware is responsible to handle the traffic forwarding according to the rules assigned by the software-based control program. The data plane responsibility is the same as in traditional network architecture the only difference is that the routing decisions are removed from this layer [212,227].

## 3. The controller

The controller can be seen as the “brain” of the SDN. This is because the whole network would be destroyed if it was attacked. Controllers give a global view of the SDN status to the Application layer/Management-Plane. The Control-Plane can be managed both by multiple controllers or a central controller. The Routing path of each new flow will be calculated by the controller. The SDN performance is greatly affected by the architecture of a Controller. There are three different Controller Architectures that have been newly proposed. These are: Multi-Core, Logically Centralised, and Completely Distributed Controllers. A single controller cannot generate the optimal routing solutions in case it needs to handle a large amount of flow that exceeds its capacity ability. If this situation happens, a bottleneck of the controller can occur for the whole week and partial control to dynamically optimise the flow routing [224]. The Multi-Core Controllers are the reasonable solution for the limited capacity of a single controller as it is unable to accommodate to the increasing scale of the internetwork. However,

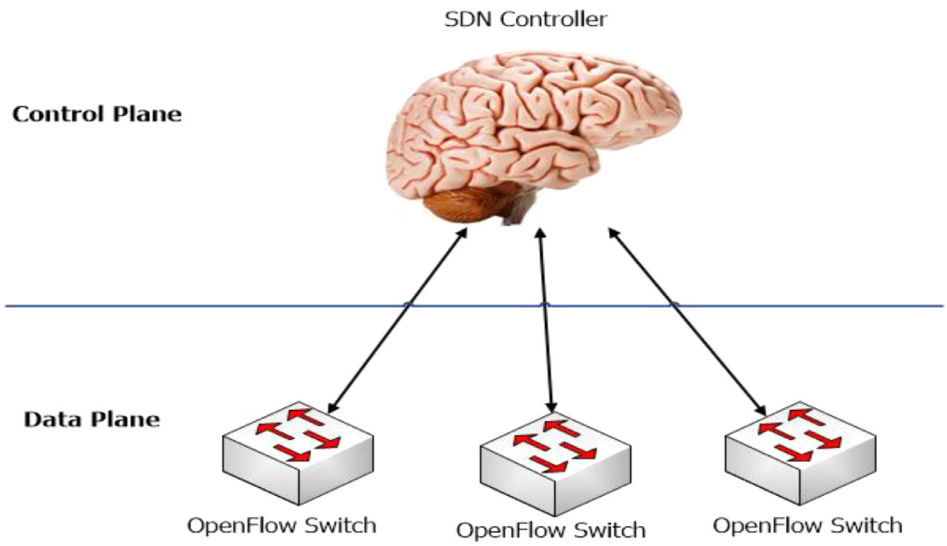


**Fig. 5.** Controller as the brain of SDN.

a Multi-Core Controller can also become a point of failure and it has limited scalability. The processing ability and resource of a controller becomes a big concern for network operators [225].

For Logically Centralised Controllers, the controllers have to share information with each other for a consistent view of the whole network. An example of a Logically Centralised Controller but Physically Distributed Controllers are ONOS [62,63] and OpenContrail [62,64]. Physically Distributed Controllers have to synchronise their conditions with others to make a global optimal solution and to maintain a global view of the whole network. They will synchronise the updated information with the other controllers when there is a local state of controller changes. The Logically Centralised Controller consists of lots of Distributed Controllers. With these, the response time for each flow request is shortened and the number of flow requests per second that the controller can handle is improved. However, the information exchange between the controllers consumes many network resources. The Completely Distributed Controllers achieve the logically Centralised Control-Plane by maintaining a global consistent network view. Each controller shares information through the synchronisation mechanism. In order to improve the scalability, work focuses on reducing the overload of state synchronisation and keeping the information consistent between the controllers must be done. The control logic in SDN is called as SDN controller. Some researchers such refer SDN controller as OpenFlow controller. Researchers refer the SDN controller as a “Network Brain” of SDN [210] Details are in as following Fig. 5.

This is because all processes of managing the activities and managing the network resources are done through this control logic [62]. The SDN controller gives global view of the SDN status to the application layer [221]. The SDN network can be managed either by multiple controllers or by a single controller. The Routing path of each new flow will be calculated by the controller. Based on the researcher in [62], the North-Bound Interface, South-Bound Interface, and East-West Interface are the three major interfaces of the controller that are important to enable the SDN components.

### *3.1. North-bound interface*

The North-Bound interface connects the SDN applications (in the Application-Layer) to the controller (in the Data-Plane Layer). The North-Bound Interface is essential to support the different networking applications and for developing applications. This interface abstracts the low-level instruction sets used by South-Bound interfaces to program forwarding devices.

### *3.2. South-bound interface*

The South-Bound Interface acts as a medium for the communication between the SDN controller and the forwarding devices (Data-Plane elements) [57,41]. For the purpose of establishing the channel between controllers and switches, the process of setting up of the switches and optimising the network management are all performed through this interface. The most famous protocol used in the communication between the forwarding elements and the controllers is the OpenFlow Protocol [65–70].

### *3.3. Types of controllers*

The researcher in [46] stated that the most relevant aspects to categorise the controllers is based on either the SDN having a centralised or distributed Controller. A Centralised Controller is a single entity which manages all the forwarding

**Table 3**
A list of OpenFlow controllers based on centralised and distributed controllers.

| Centralised controller | Distributed controller |
|---|---|
| Floodlight [68] | DISCO [77] |
| NOX-MT [75] | HP VAN SDN [79] |
| Beacon [76] | Fleet [81] |
| Maestro [79] | Onix [82] |
| Meridian [80] | yanc [83] |
| Rosemary [84] | HyperFlow [85] |
| ProgrammableFlow [86] | PANE [87] |

devices, whilst the Distributed Controller has multiple control elements that are distributed throughout the system. Table 3 shows the list of OpenFlow controllers based on the Centralised and Distributed Controllers.

## 4. OpenFlow protocol

The OpenFlow protocol is the most common interface between the Data-Plane and Control-Plane Layers in SDN [71]. The OpenFlow Protocol was proposed by Stanford University, enabling its concept to be implemented in both hardware and software. Through an OpenFlow Protocol, the existing hardware can be utilised to design new protocols and analyse their performance. In SDN, the controller (software) manages the collection of switches for traffic control. The communication between the controller and the OpenFlow switch is through the OpenFlow protocol. Besides that, the controller also manages the switch through this protocol. The SDN allows the abstraction and centralised management of the lower-level network functionalities by decoupling the network logic from the data forwarding devices into the logically centralised distributed controllers. However, this separation introduces new scalability and performance challenges in large-scale networks of dynamic traffic and topology conditions. Many research studies have represented that centralisation and maintaining the global network visibility over the distributed SDN controller introduce scalability concern [189].There are three types of tables in the OpenFlow switch, which are: (1) Flow Table – It specifies the actions that need to be performed on the packets and matches incoming packets to a specific flow. (2) Group Table – Triggers different types of actions that affect one or more flows. (3) Meter Table – Triggers different types of performance- related actions on a flow [72].

The SDN South-Bound interface is provided by the OpenFlow Protocol. In other words, the communication interface between the SDN controller (Control-Plane) and SDN switches (Data-Plane) is provided by this protocol. This protocol allows the SDN controller to configure and manage the SDN switches [2]. The OpenFlow protocol is compatible with the GENI standard. This enables a user to arbitrarily create slices without being aware of the physical network infrastructure [48]. An OpenFlow Protocol specifies how traffic should flow through the network by allowing the controller to access and manipulate the flow tables (forwarding rules) of the SDN Switch [228]. Each incoming packet is matched against a set of rules and the action list associated with the matching rule is executed. Forwarding or dropping packets is an example of action lists that are supported by an OpenFlow switch. Besides that, an OpenFlow supports rewriting of packet headers by the switch, adding or removing VLANS and MPLS tags and rewriting of MAC source and destination addresses over network.

### *4.1. OpenFlow architecture*

The basic concepts of the OpenFlow architecture are: (1) The network is developed by the OpenFlow-Compliant switches which compose the Data-Plane, (2) The Control-Plane has one OpenFlow or more OpenFlow Controllers, and (3) The switches connect with the Control-Plane through a secure control channel [183]. Controllers and forwarding devices are the main elements in the SDN/OpenFlow architecture. As mentioned earlier, forwarding devices are specialised in packet forwarding whilst controllers are responsible for decision making [48,73,74]. An OpenFlow-enabled forwarding device is based on the pipeline of the flow tables [46]. Each entry of a flow table has three parts: (1) A matching rule, (2) Actions (to be executed on matching packets), and (3) Counters (keep statistics of matching packets).

The OpenFlow architecture has three (3) main components [48,49], and [75]: Switches: There are three basic components of an OpenFlow switch, which are: Flow table, a communication channel, and an OpenFlow Protocol. Each flow table has action fields associated with each flow entry. As illustrated in Fig. 6, an OpenFlow switch can have multiple flow tables, a group table, and an OpenFlow channel [48,76,77], and [80]. The transmission of commands and packets between a controller and the switch is provided by the communication channel. The communication between the OpenFlow controller with any router or switches is enabled by the OpenFlow protocol. The switches are connected to each other by the OpenFlow ports over network. They [78] presented a dynamic routing scheme named DIFF that differentiates flows based on their impact on network resource and adaptively selects routing paths for them to mitigate the problems of flow-table overflow and inefficient bandwidth allocation. They [79] proposed Software-defined Adaptive Routing (STAR), an online routing scheme that efficiently utilises limited flow-table resources to maximise network performance

Controllers: the controller is responsible for updating (revise, add, delete) flow-entries from the flow table.
Flow-entries: each flow table contains flow-entries. The group table can configure the flow-entries.

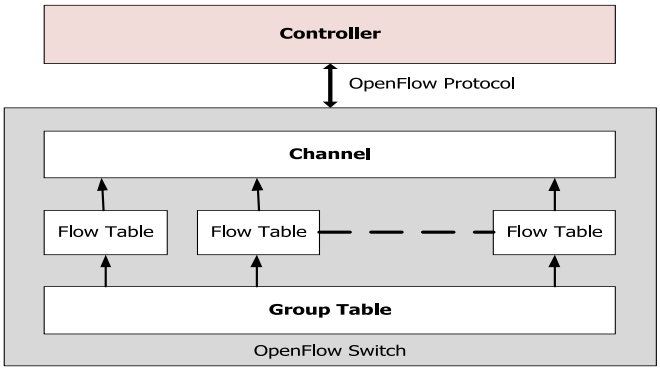


**Fig. 6.** OpenFlow model [48].

### *4.2. Configuration*

The lookup process will start as soon as a new packet arrives. The process starts in the first table and ends either with a match in one of the tables of the pipeline or with a miss. The rules follow the natural sequence number of the tables and the row order in a flow table. After an OpenFlow switch receives a packet, the header fields are compared to the flow table entries. If a match is found, the packet is treated according to the actions stored in the flow table. But, if the received packet does not match the flow table entries, it encapsulates the packets in an OpenFlow Packet-In message and sends it to the controller. The controller can install new forwarding rules in order to handle the packets of the new flow. After that, the controller will inform the switch either to make a new entry in the flow table to support the new flow or give instructions to drop the packet [81]. The SDN controller requests configuration information (including its active switch ports and corresponding MAC addresses) from the SDN switch by initialising a protocol handshake using an OpenFlow OFPT_FEATURES_REQUEST message to inform the controller about the existence of the switches in the network. An OpenFlow Packet-In (OFT_PACKET_IN) message is used to forward the data packets that it receives to the controller. An OpenFlow Packet-Out (OFT_PACKET_OUT) message is used by the controller to send a data packet to a switch with instructions of what to do with it in the form of an action list [2]. The OpenFlow Switch can be instructed to behave like a router, switch or firewall, or perform other roles, such as load balancer and traffic shaper which depends on the rules installed by a controller application [46]. There are four possible actions in a flow table: Forward packet to port(s), encapsulate and forward to the controller, drop the packet, and send to the normal processing pipeline. The OpenFlow Protocol has different versions. Each new version has new improvements and new features.

## 5. SDN security threats and countermeasures

The researchers in [82] classified the threats and its countermeasures into three groups according to the SDN layer (Application, Control-Plane, and Data-Plane) at which the corresponding attacks occur.

The Open Networking Foundation in its article entitled Principles and Practices for Securing Software-Defined Networks [83] has proposed principles for securing SDN. Table 4 proposed security principles are for all protocols, components, and interfaces of the SDN architecture. The security principles proposed by ONF. An attack in the SDN can occur through the central location for management, which is the Controller. Besides that, it can also occur through the Switches' flow tables that consist of information related to switching, routing, and access control. The North-Bound Interface, South-Bound Interface, and East-West Interface can also be attacked by tricking the controller to allow malicious applications to join the network and communicate with the controller, the network, and its traffic. Besides that, the channel between the controller and the switches can be also attacked (Table 5).

### *5.1. Illegal access*

The applications running on the controller have special rights to control the network behaviour and to enter remote information resources. This makes the Application Layer vulnerable to Illegal Access. The applications are also very flexible and are able to be extended. The culprit can steal the network information/resources and it can interfere with the network configuration by inserting malware computer programs into the application layer. Most of the applications running on the controller are developed by third-party organisations, not controller vendors, and this makes the authentication applications become the main challenge in SDN, which provides programmable networks [84]. Since SDN technology is still new in IT industries, there are limited standardised security mechanisms available for SDN applications. VeriCon [82,85] is responsible for verifying that an SDN program is performing in the correct way. It uses first-order logic, after that it implements classical Floyd-Hoare-Dijkstra deductive verification using Z3. The first-order logic helps the VeriCon to determine admissible network topologies and desired network-wide invariants. According to the researchers in [85], VeriCon is able to rapidly

**Table 4**
Principles for securing software-defined networks.

| | |
|---|---|
| **Principle 1** | Define Security Dependency and Trust Boundaries Clearly<br>• Security dependencies between must be defined.<br>• Avoid circular dependencies. |
| **Principle 2** | Ensure Strong Identity<br>• Strong identity must have the characteristics below:<br>Able to differentiate its owner from other entities, can be revoked, updated, and generated. Use strong cryptographic mechanisms to prevent impersonation. |
| **Principle 3** | Create Security based on Open Standards<br>• Avoid use of algorithms/protocols (e.g., MD5: MD5 Collision – Two files have the same hash, this makes the hash function become unreliable since the hash functions as evidence authentication (A. Sotirov), (Collisions, 2006)) that have been proved insecure by standard organisations. |
| **Principle 4** | Protect the Information Availability, Integrity, and Confidentiality<br>• The impact of the security control towards the overall SDN architecture should be evaluated.<br>• The security control that is constructed must not indicate the reduction of system Availability, Integrity, and Confidentiality. |
| **Principle 5** | Secure Operational Reference Data<br>• Unexpected system behaviour, such as loss availability, integrity, and confidentiality, are impacted by the integrity of the reference data.<br>• The reference data should be generated, processed, and transported securely. |
| **Principle 6** | Develop Secure Systems by Default<br>• The security control must be reconfigurable and can be disabled.<br>• The system must set a minimum level that most of the primary security is enabled by default.<br>• The security control must be configured with minimum properties. to make sure the control is effective. |
| **Principle 7** | Security Must Support Accountability and Traceability<br>• Security controls must be auditable over network. For auditing purposes, the logged data must have enough information.<br>• Make sure the audited data does not contain similar data and the auditing action does not lead to a violation of the security policy.<br>• Data must be protected from unauthorised modifications and access. |
| **Principle 8** | Characteristics of Manageable Security Controls. A strong identity must have the characteristics below:<br>• Security objectives and assumptions must be clearly stated.<br>• Security control must support installation from the lowest to the highest reference system without introducing complexity. New security must only introduce minimum complexity to the implementation.<br>• Security control must be easy to implement, maintain, and operate, and be based on well-defined standards. |

**Table 5**
Comparison of the proposed OpenFlow-SDN per-flow source routing mechanisms.

| Proposed approach | Rules placement | Method | Objectives |
|---|---|---|---|
| SlickFlow [75,90] | Reactive | Source Routing | • Utilises source routing method to reduce the distribution of network state in all the data forwarding nodes along the routing path. |
| JumpFlow [91] | Reactive | Source Routing | • Uses the available VLAN identifier (VID) of the flow entry's packet header to carry the routing information, and partitions the routing information and distribute them on a few selected contact forwarding nodes. |
| HPH [92] | Hybrid | Source Routing | • Reduces both the number of permanent flow entries and the number of configuration messages from the controller by dividing switches in to several regions. |

verify correctness and identify bugs in large-scale simple core SDN programs. NICE (No bugs In Controller Execution) [86,87] is a tool that is capable of testing the original controller programs written for the popular NOX platform by automatically generating carefully-crafted streams of packets under many possible event interleavings. It can also be used to verify generic correctness properties (e.g., No black holes and no forwarding loops). PermOF [82,88] isolates applications and checks their permissions. It also provides control rights to OpenFlow controllers and applications running on top of it. It specifies a set of permission categories which allow the network operator to sharply define the access privileges allowed to applications [93,94].

### 5.2. Security rules and configuration conflicts

SDN allows a network to be operated by many applications at the same time. However, these applications may not cooperate with each other due to a variety of software applications that uses the different programming languages being used in an Application Layer. This causes Security Rules and Configuration Conflicts. The application layer must have security programs to access the security interfaces of the controller as this is needed for the purpose of providing a wide range of network services. FLOVER [82,89] is a model checking system that has a function to make sure the flow policies deployed by an OpenFlow application do not violate the security policies of the network. It is implemented as an OpenFlow application that runs on an OpenFlow controller. Every time an OpenFlow switch receives a new flow rule from an OpenFlow controller, the FLOVER will verify a set of specified non-bypass characteristics against the updated flow rule set. There are two types of execution modes supported by FLOVER, which are: the in-line mode - this mode executes flow rule validation with each flow rule update and batch mode – in this mode the verification process is performed periodically to enhance the response time of the controller. The FLOVER can examine requested changes and respond to them. As below section aims to provide a comprehensive analysis of attack threats in IoT environments. Based on the IoT taxonomy presented in the previous section, the security threats are discussed in-depth for each domain.

### 5.3. Security attacks on IoT devices

Accounting for the constrained computation capabilities and limited energy supply of IoT devices, the adoption of conventional strong security mechanisms is not guaranteed, thus increasing the potential vulnerabilities. In addition, IoT devices can operate remotely and unattended by human intervention, thus making them vulnerable to physical attacks. We remark that attacks against the physical devices can be extremely dangerous in IoT systems, since the compromised nodes can generate altered measurements. As a result of the corrupted information, IoT control systems can be severely impacted, providing erroneous feedback information and wrong services. In the following, we describe the main attacks related to the IoT device layer.

1) *Hardware trojan attack*: trojans have emerged as a major security concern for IoT devices [20]. Trojan is a malicious modification of hardware, which allows the attacker to exploit the infected IoT device to gain access to either sensitive data or software running on that device. To this aim, the attacker alters the original circuitry during design or fabrication and inserts a triggering mechanism that activates the malicious behavior of the Trojan [20].
2) *Replication attack*: a malicious attacker can create a new node by replicating the sensitive identification information of a target device. Then, to allow the connectivity to the existing IoT system, the replicated node is faked as authorised, generating severe vulnerabilities in the IoT system. Indeed, the node can generate false data, making IoT applications returning erroneous feedback commands or providing wrong processed information. Furthermore, the replicated node can also enable the attacker to obtain security privileges, such as extracting cryptographic shared keys [58], and to revoke authorised nodes by carrying out node revocation mechanisms [59].
3) *Tampering attacks*: IoT devices can operate remotely and unattended by human intervention, thus making them vulnerable to tampering attacks. Tampering attacks refer to all scenarios where a malicious entity performs an unauthorised physical or electronic action against the device. The adversary can exploit the physical access to the device to gain full control, therefore known also as node capture attacks, causing intentional malfunction or sabotage [60]. In [61], an extended analysis of tampering attacks on sensor node is pro- vided, especially focusing on the malicious approaches which can be executed in the deployment area, without interruption of the regular node operation [190].

Anteater [82,95] is a tool for detecting invariants in data forwarding layers. It also diagnoses a broad, general class of network problems. Anteater performs checking functionality for the Data-Plane by analysing the contents of the forwarding tables contained in routers, switches, and other networking equipment. The researchers in [96] introduced NetPlumber. It is a real time policy detection tool. NetPlumber checks every event (e.g., Installation of a new rule, removal of a rule, port or switch up and down events). NetPlumber also can discover simple invariant violations, such as loops and reachability failures. An attacker can get sensitive data from communication between Application-Layer elements and SDN Controllers. To protect the application from information disclosure, Message Encryption [97] should be performed in communications between the application and the controller.

### 5.4. Threats to the control-plane layer

In the Control-Plane Layer, the controller will become the main focus for the attacker since the entire network will be affected if the OpenFlow controllers and their security are affected. This also gives a direct impact towards the switches on the Data-Plane Layer (Forwarding Layer). This is because, without receiving any forwarding rules from the controller the switch cannot forward packets.

### 5.5. DoS/DDoS attacks on the controller

Fig. 7 shows how the DoS/DDoS attack happens. In this type of attack, the attacker produces a large amount of traffic in a short period of time to the SDN-Enabled network using its own host. This consumes memory resources in both

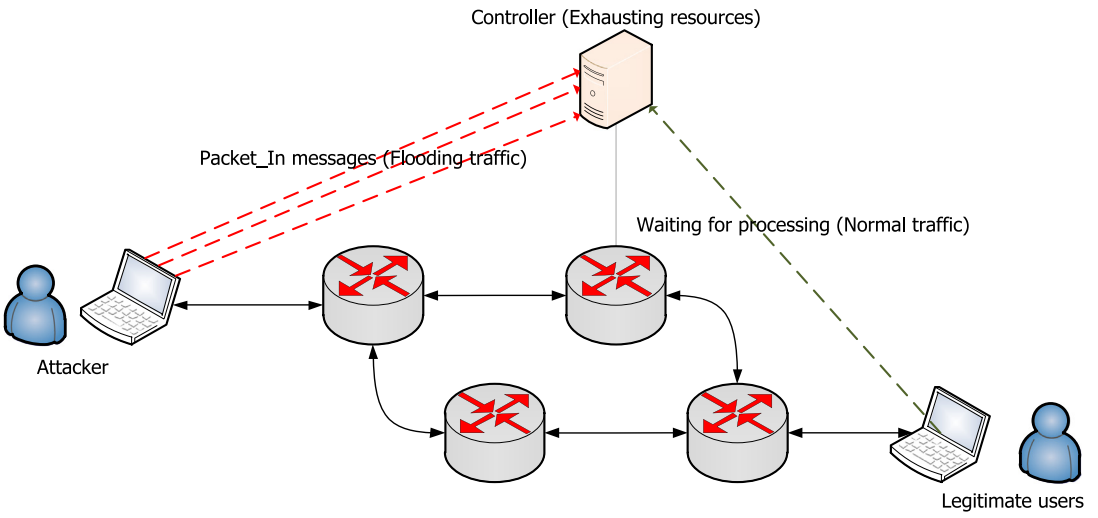


**Fig. 7.** DoS/DDoS attack on controller [82].

the Control-Plane and Data-Plane. DoS/DDoS attacks can result in the degradation of the network performance, drop of legitimate packets, and increase network delay [98,99,233]. This attack also makes the controller functions unavailable to legitimate users. The traffic that is flooded by the attacker will be mixed up with the original traffic that flows in the network [100,101]. This makes the network difficult to differentiate between the two traffic types.

The researchers in [102] explained that before an attacker attacks an SDN network using DoS/DdoS, it must first confirm that the network is an SDN network. This is achieved by identifying the processing times of the first packet and the next packets. After that, the response times between these two packets are compared. The controller in SDN only takes a short period of time to respond to a new flow entry for a new packet. After the attacker has successfully identified the SDN network, the attacker can just flood the network with fake packets. This type of attack can cause exhaustion of network resources that leads to degradation of network performance. The researchers in [103] introduced FloodGuard to overcome the DoS attack by using two new modules, which are: The Proactive Flow Rule Analyser and Packet Migration. After the FlowGuard detects the flooding packets, the Packet Migration redirects the table-miss packets in the OpenFlow switch to the Forwarding Layer cache. Packet Migration temporarily caches the flooding packets to protect the controller from being overloaded. After that, the round-robin scheduling and rate limit are used to submit the flooding packets to the OpenFlow controller. The Proactive Flow Rule Analyser identifies the different types of sensitive parameters by tracking the current network flow. It installs the flow rules inside the OpenFlow switches.

The researcher in [104] introduced the DDoS Blocking Application (DBA) to cope with Distributed Denial-Of-Service (DDoS) attacks. As mentioned earlier, the traffic that is flooded by the attacker will be mixed up with the normal traffic and the network will face difficulty in differentiating between the two traffic types. By using a DBA through the Locator/ID Separation Protocol (LISP) [82], the controller can differentiate between normal traffic and attack traffic. The DBA also provides a clue for the controller to discover the attack by notifying the controller of the location of a network element change. Besides that, the controller will consider attack traffic if the transmission rate of the traffic is more than a specific value that has been set. A Content-Oriented Networking Architecture (CONA) [82,105] can take a countermeasure against resource-exhaustive attacks, such as DDoS attacks. In order to reduce the harm of DDoS attacks, the CONA analyses and filters the content request messages from clients. If the level of the requested messages arriving at the content server is more than a specific value that has been set, a DDoS attack is considered to be in progress. Each relevant CONA agent will receive a notification from the controller to prevent the attacking traffic from spreading.

### *5.5.1. Threats from applications*

The threats to the controller can also come from the applications that run on the controller. For each type of application that has different functional requirements, the network needs to specify a specific security policy. For example, the intrusion detection application (IDS) must examine the header fields of packets. FRESCO [106] has been proposed to alleviate the rapid design and modular composition of OpenFlow-enabled detection. It intends to address issues that can speed up the composition of new OpenFlow-enabled security applications. FRESCO makes the management of complicated security services for OpenFlow networks simpler. It also provides different types of security software modules which perform several security functions, such as attack deflectors. The Security-enhanced Floodlight controller (SEFloodlight) [107] is able to detect attacks effectively by providing an audit subsystem that tracks all security events. In order to manage the permission of applications, SEFloodlight provides a programmable north-bound API. It provides a role-based function and authorisation

module which areused to specify different rights corresponding to the roles of the application. Besides that, SEFloodlight also verifies the integrity of the software modules by using application authentication modules.

*5.5.2. Tampering*

The packets that are exchanged during communication between the controller and other entities, such as devices, in the application layer can be altered by a malicious entity in the middle of the interaction process. Through the SDN Controller, the attacker will be able to modify the network topology information, and backup the flow table contents, policy, and log information. Strong Message Authentication Code (MAC) algorithms [108] and signatures can be used to protect the software and update packages' integrity and authority.

*5.5.3. IP spoofing*

The attacker can use IP Spoofing to get unauthorised access to the SDN Controller by forwarding packets to the SDN Controller with a source address to indicate that the packet comes from a specific system or port. The attacker uses a fake identity in order to access and attack the SDN Controller. Identity authentication can be used to protect an SDN Controller from IP Spoofing. Mutual authentication must be performed before communication with the SDN Controller is established. A Source address verification can be performed at the SDN controller [109]. The researchers in [110] proposed a new security extension to the SDN controllers which is known as TopoGuard. The purpose of this security is to provide automatic and real-time detection of Network Topology Poisoning Attacks. Based on the evaluation made by the researchers, TopoGuard in the Floodlight controller successfully secured the network topology whilst introducing a minor impact on the normal operations of the OpenFlow Controller. The Intra-AS IP Source Address Validation Solution with OpenRouter (InSAVO) [111] is another mechanism that can effectively validate the IP source address of packets.

*5.5.4. Threats to the data-plane layer*

The Data-Plane Layer consists of thousands of switches that have the responsibility to forward packets. It is important to identify the possible threats and corresponding countermeasures in this layer since switches are the direct entry point of network access for end users. The attacker can attack the switch by attacking the link to a port of the switch. The possible security threats in this layer are Man-In-The-Middle Attacks that occur between the switch and the controller, and DoS attacks.

*5.5.5. Man-in-the-Middle attack*

In SDN, the Man-In-The-Middle Attack intercepts the communication between the switch and the controller by inserting an agent node between them. All switches and hosts connected directly to the switches on the communication path have a high tendency to be converted to agent nodes in this attack type. Session Hijacking and DNS Spoofing is an example of a Man-In-The-Middle Attack. The attacker that performs the Man-In-The-Middle Attack using DNS Spoofing provides fake information to the target media. The attacker uses Session Hijacking of the cookies and steals with the help of a Hyper Text Transfer Protocol (HTTP) [112]. This type of attack takes control of the network packet forwarding by intercepting it with the forwarding rules issued to the switch.

In order to prevent this attack from intercepting the communication between the switch and the controller, a secure channel needs to be created between them. The communication channel can be secured by Transport Layer Security (TLS). However, many vendors do not provide support for TLS in their OpenFlow switch. In addition to that, not all versions of OpenFlow provide TLS, since the configuration of the TLS is very complex. The implementation of the authentication with TLS into a single controller, controlling a set of network devices will provide enough security protection. However, the access control and authorisation becomes more complicated if TLS is applied to multiple controllers interacting with a single device or multiple control processes interacting with a single centralised controller [113]. However, the configuration of the TLS is not compulsory for the later version of OpenFlow specifications. The next countermeasure to this attack is FortNOX [82,114]. It provides an authentication security enhancement strategy. Collisions of various forwarding rules can be detected by FortNOX. It provides security constrains enforcement and role-based authorisation. After a flow rule is inserted into FortNOX, the OpenFlow program is inhibited from inserting flow rules into the OpenFlow network. This is how FortNOX prevents conflicts between these rules. It handles the conflict by conducting authorisation roles using digitally signed flow rules. VeriFlow [115] successfully improves the network performance during the network checking process by achieving low latency during the process. Besides that, it acts as a middle layer between the switches and the controller. It provides analysis for the numerous header fields and API for verification of custom invariants. VeriFlow only needs few hundred milliseconds to check any rule insertion or deletion activities.

*5.5.6. DoS attack (Overflows the flow table and flow buffer)*

By using a DoS attack, the attacker can produce numerous fake packets that will be sent to unknown network devices in a short time period. The legal traffic will not be forwarded correctly if the Flow Table has an overflow of irregular traffic. This is because the Flow Table does not have the capacity to insert new rules.

If the Flow Buffer is attacked by a DoS attack, it will result in an overflow of the Flow Buffer, since the switch needs to buffer numerous fake packets. Each packet that waits for the rule, searches or waits for the insertion of new rule results, which are buffered in the Flow Buffer before they are forwarded out. The deletion of packets in the Flow Buffer is treated

by using the First in First out (FIFO) concept to release the storage space. Since the storage volume of the Flow Buffer is limited, it makes the Flow Buffer not have enough space to store the legitimate packets. As a result, the legitimate packets will have to be dropped. FlowVisor is a network slicer [116] as it acts as a transparent proxy between the OpenFlow switches and controllers. The FlowVisor is able to create a different controller that has a different responsibility to handle the large network. Through the FlowVisor, a virtual black hole can be created. It can be used to absorb the affected traffic that is flooded by the DoS attack. In response to a DoS attack, the controller can create a rule to drop all UDP traffic, the FlowVisor then rewrites the rules to ensure the rules only affect the specific part of the network that has been specified to the controller. Only unaffected traffic will remain in the network after the FlowVisor rewrites the rules to drop all the UDP traffic. The attacker can use IP spoofing to perform DoS attacks. The main purpose of the Virtual source Address Validation Edge (VAVE) [82,117] is to prevent this attack. Through VAVE, the controller will carry out source address validation for a new packet that does not match any rule in the Flow Table. During this process, IP spoofing can be detected and to inhibit the specific flow from the source address, the controller creates a rule in the Flow Table. Table 7 summarises the Security Threats in SDN and the Countermeasures.

## 6. SDN applications

Rural Connection [48] is one of the SDN applications. The main problems faced by the network administrator for deploying network technology in the rural areas are the sparse population and resource constraints. The separation of the construction of the network and the configuration of the network in SDN enables the rural infrastructure deployment business to be performed in rural environments and the Internet Service Provider's (ISP) business to be performed remotely in cities. Through SDN, the management of rural networks is not necessary in the rural areas. The next application is Mobile Device Offloading. Each piece of data in the network requires a different level of security, for each application in the internet data privacy is the main concern. For example, in the authors used the Enterprise- Centric Offloading System (ECOS) to make sure that the applications that have security need only to be offloaded on approved machines. The SDN's responsibility is to select the network resources that meet the security needs. It will specify a device that provides the security requirements for offloading. Data is prevented to be offloaded from the mobile host if there is no device available. Any resources available can be used if energy savings is not a concern. The landscape of Internet of Things (IoT) is continuously evolving, attracting an increasing number of cybercriminals who aim at exploiting vulnerabilities of IoT systems to carry out malicious attacks on a potentially global scale [118]. Conventional security mechanisms have been revealed to be inefficient accounting for the heterogeneity, and mobility of IoT devices.

Date Centre Upgrading is another application of SDN. The researchers in [119] have discussed the use of the SDN concepts for solving the problems faced in cloud computing services, specifically in the Data Centre Network. Data centres are an integral part of many companies [120]. Google uses the SDN technology to interconnect its large number of data centres located at different geographical areas to ensure that the data can be provided quickly when requested. Through OpenFlow, the switches can be managed from a central location [121]. It helps Google to improve operational efficiency [122] and reduce the management costs. The researchers in [120] proposed a network infrastructure based on OpenFlow that can be used to interconnect data centre networks. The proposed network infrastructure improves the latency by moving the workload to underutilised networks. VMware NSX is a network virtualisation platform [123], VMware NSX is SDN-based. It simplifies the network management since through the NSX, the network administrator does not need to deal with VLANs and complex sets of firewall rules. Besides that, the NSX also provides network segmentation where each virtual network uses its own address space and this network is also isolated from the other virtual networks. SDN provides a solution for the challenges faced by Data-Centre Networks (DCNs). According to the research conducted by the Enterprise Strategy Group (ESG), the Data-Centre Networks' system has undergone rapid changes due to the aggressive alliances in the data centres, progressive use of virtualisation technology, and wide deployment of web applications [124]. The Vello system is SDN-based. It uses the OpenFlow standard to provide a basic set of management constructs in order to enable value-added capabilities for data centre local and wide area networks [125]. Besides that, the Vello system also allows a unified control of the global cloud for WAN [234] resource optimisation. The SDN-based Vello systems have proposed open and scalable network virtualisation solutions in order to connect the storage and compute the resources in data centres within public and private cloud platforms.

Switching with In-Packet Bloom Filters (SiBF) is another example of the DCN architecture which has been proposed by the researchers in [126]. SiBF introduces Rack Managers that act as OpenFlow Controllers that provides scalability and maintain the globally required state to provide fault-tolerance in the DCN. It proposes scalable forwarding services that are self-configurable and do not require endpoint modifications. It can also be seen as another choice of forwarding service parallel to other Ethernet styles. This proposed data-centre architecture also uses an encoding technique to provide load – balancing services. The researchers in [127] introduced the Energy-Aware Data-Centre Architecture based on an OpenFlow platform. The main purpose of introducing the architecture was to achieve the concept of "Green Data Centre" in the DCN. It provides guidelines to learn about the energy consumption in the DCN elements, such as switches, links, and ports. The DC power and energy consumption is measured in three important places, which are: at the utility meter, at the plug, and at the hardware computing load inside the box of the IT equipment itself [128]. Through the proposed architecture, the traffic load can be captured and monitored. This makes the process of analysing the connection between the traffic load and energy consumption possible. Through the proposed architecture, also, the different energy-aware topology optimisation and

routing algorithms can be deployed and analysed. The minimum power required by a network topology can be estimated through this proposed architecture. Through VAVE, the controller will carry out source address validation for a new packet that does not match any rule in the Flow Table. During this process, IP spoofing can be detected and to inhibit the specific flow from the source address, the controller creates a rule in the Flow Table. Table 6 summarises the Security Threats in SDN and the Countermeasures.

The researchers in [130] proposed the OpenFlow Switch Controller (OSC) in DCN to minimise the power consumption of the switches in the DCN by minimising the influence of carbon emissions in the DCs. This proposed OSC is able to work at different power saving modes since it receives control messages from: the OpenFlow controller and control switches, and links which are based on the programmable controller. The OSC together with a NetFPGA based OpenFlow switch [131] can be used for power-aware networking research. The proposed OSC also helps reduce the configuration time of network elements.

Next, the SDN Technology is also used to improve the DCN Metrics. The researchers in [132] introduced an integrated way to monitor the path load metric to administrate link layer multipathing [235] and congestion control. The integrated congestion control uses the link load information in edge switches that directly inform sources to control traffic admission. The proposed method integrates the Dynamic Load Balancing MultiPath (DLBMP) scheme with congestion control, where the routing intelligence is decoupled from the data transmission (SDN techniques) to minimise overhead and to speed up the update process. Through the proposed methods, the DCN will experience loss-less delivery and provides data sources which can respond rapidly to congestion. Besides that, the network throughput is also improved with a fine flow differentiation mechanism.

The SDN technique is also used to deal with the header redundancy problem in DCs. The researchers in [133] introduced "Scissors" that has the ability to make changes on packet headers in order to decrease DC traffic and network power consumption. From their investigation, header redundancies add up to 30–40% of the DC traffic which has an effect on the latencies and complexity of the processing which can increase power consumption in a DC. Through Scissors, the redundant header information is replaced with a shorter tag called a Flow ID and for packets that have the same flow. It is part of a group in the same ID. Hence, it improves network delay and power gains.

The researchers in [134] introduced a SDN-based network solution to improve DCN and deployed it in a multitenant experiment. Through the proposed prototype, the multiple OpenFlow switches are managed by the central controller and the responses to network updates are based on APIs. This makes the configuration update process become simpler. Through the SDN-based network solution, the DCN strategies have fulfilled the cloud service provider's needs which are: multitenant, low-cost, flexible, easy to operate, and configurable. Multiple data centres are placed in different geographical locations. CrossRoads [135] is a network fabric that facilitates live and offline virtual machine migration across multiple data centres. CrossRoads is OpenFlow-based. It provides support for East-West and North-South types of traffic for virtual machine migration. East-West traffic is used for virtual machine migration within data centres whilst North-South traffic is used for virtual machine migration of the external clients. Next, the researchers in [120] introduced a software middleware solution which is known as Network Infrastructure as a Service (IaaS). It is OpenFlow-based. Through this proposed solution, the connectivity interruption of virtual machines during the migration process is minimised. It also supports live virtual machine migration between different DCNs. Networking as a Service is a Cloud-based network architecture which is implemented by using the OpenFlow Protocol [136]. The proposed architecture evaluates the provision, delivery, and consumption of the Networking as a Service. It is composed of the NRP-network resource pool, the NOI-network operation interface, the NRE-network run-time environment, and the NPS-network protocol service. The cloud-based network architecture consists of the Control-Plane layer where the switching and routing processes occur and the Network Data-Plane layer where packet forwarding activities are conducted (Table 8).

The Software Defined Internet Exchange (SDX) [137,138] is another application of SDN. The SDX capabilities allow two networks' peers only for streaming video traffic which is known as application- specific peering. Besides that, the SDX has also created new programming abstractions which allow participating networks to develop/run the application that is able to behave correctly when it interacts with the border gateway protocol and the networks do not interfere with each other [139]. By deploying SDN at Internet Exchange Points (IXPs), it can perform many different actions on packets based on multiple header fields that enable inbound traffic engineering and wide-area server load balancing. OpenRoads (OpenFlow Wireless) [140,141] is a program for innovation implementation of services for the wireless networks. It creates an open program to explore different mobility solutions, routing protocols, and network controllers. Through OpenRoads, the researcher is able to control the data path using OpenFlow and handle the configuration of the device using the Simple Network Management Protocol (SNMP) [142]. The control abilities from OpenFlow and the SNMP make OpenRoads to easily manage the different wireless technologies, for example, WiFi and WiMAX. The researchers in [143] were truly inspired by OpenRoads. They tried to resolve specific needs and challenges to deploy a software defined cellular network.

The researchers in [144] proposed the Software Defined Optical Network (SDON) architecture and QoS-Aware Unified Control Protocol for optical burst switching in OpenFlow-Based Software-Defined Optical Networks. The main function of this architecture is to improve QoS for a different type of traffic. The effectiveness of the proposed protocol was evaluated by using the conventional GMPLS-Based distributed protocol. This proposed protocol successfully improves the QoS for a different type of traffic. The researchers in [145] developed OpenFlow-Based update mechanisms to support high-level abstractions. The main point of creating a set of high-level abstractions is to enable the administrator to update the whole network and to make sure each packet which crosses the network is processed by a single-fix global network configuration.

**Table 6**
Summarised security threats in SDN and the countermeasures.

| Layer | Threats | Countermeasures |
|---|---|---|
| **Threats to the Application Layer** | Illegal Access | • VeriCon [85]: Verifies that the SDN program is performing in the correct way.<br>• NICE [86]: Tests unmodified controller programs written for the well known NOX platform and automates.<br>• PermOF [88]: Isolates applications and checks their permissions, provides privilege control to OpenFlow controllers and specifies a set of permissions. |
| | Security Rules and Configuration Conflicts | • Flover [89]:Verifies that the flow policies deployed by an OpenFlow application do not breach the security policies, and it supports a batch and in-line mode.<br>• Anteater [95]: Detects and diagnoses a broad, general class of network problems and provides verification functionality for the data forwarding layer.<br>• NetPlumber [96]: Checks every event and detects simple invariant violations, such as loops and reachability failures. |
| | Information Disclosure | • Perform Message Encryption on communications between application and controller [129]. |
| **Threats to the Control-Plane Layer** | DoS/DDoS Attacks on the Controller | • FloodGuard [103]: Uses the Proactive Flow Rule Analyser and Packet Migration to overcome the DoS attack. Besides that, round-robin scheduling and rate limit are used to submit the flooding packets to the OpenFlow controller.<br>• DDoS Blocking Application (DBA) [104]: Uses the Locator/ID Separation Protocol (LISP) to differentiate between normal traffic and attack traffic, and provides a clue for the controller to discover the attack by notifying it of the location of network element changes.<br>• A Content-Oriented Networking Architecture (CONA) [105]: Reduces the harm of DDoS attacks by analysing and filtering the content request messages from clients. Moreover, the controller will send a message to each relevant CONA agent to prevent the attack from spreading. |
| | Threats from Applications | • FRESCO [106]: Provides different types of security software modules which perform several security functions, such as attack deflectors.<br>• SEFloodlight [107]: Detects attacks effectively by providing an audit subsystem that tracks all security events, provides a programmable north-bound API to manage the permission of the application, and verifies the integrity of the software modules by using an application authentication module. |
| | Tampering | • Uses Strong Message Authentication Code (MAC) algorithms [108] and a signature. |
| | IP Spoofing | • Performs Mutual Authentication before communication with the SDN Controller can be established.<br>• TopoGuard [110]:<br>• Provides automatic and real-time detection of Network Topology Poisoning Attacks.<br>• Intra-AS IP Source Address Validation Solution with OpenRouter (InSAVO) [111]:<br>• Validates IP source address of packets. |
| **Threats to the Data-Plane Layer** | Man-In-The-Middle Attack | • A secure channel needs to be created between the switch and the controller by using Transport Layer Security (TLS).<br>• FortNOX [114]: Provides an authentication security enhancement strategy and detects collisions of various forwarding rules.<br>• VeriFlow [115]: Responsible for the dynamic verification of network variables within the whole network, especially when a new forwarding rule is inserted. |
| | DoS Attack (Overflows the Flow Table and Flow Buffer) | • FlowVisor [116]: A network slicer, Creates a virtual black hole that is used to suck in all malicious traffic that is produced by a DoS attack. It rewrites the rules to ensure the rules only affect the specific part of the network that has been specified to the controller.<br>• VAVE [117]: Reduces DoS attacks caused through Internet Protocol spoofing. And, it conducts source address validation for a new packet that does not match any rule in the Flow Table over network. |

**Table 7**
Recent research done in cyber security and forensics in SDN.

| Research | Reference | Overview |
|---|---|---|
| Aggressive DDoS Attacks | [184] | • In this research, they proposed a correct-by-construction REsilient COntrol Network architecture (ReCON ) that is 40% more resilient against DDoS on FP and increases the OFA capacity by at least 2 times using SDN resources [184]. |
| Adversarial Network Forensics in SDN | [185] | • In this research, they demonstrate that flow rules in SDN networks can be predicted.<br>• Also, discussed real-world SDN application, and point out that the predictability of flow rules can open severe security leaks if exploited by attackers. |
| AEGIS & Verification for SDN | [186] | • AEGIS is the first attempt at automatically and flexibly generating permission model from the API description using NLP techniques.<br>• It is also the first study to completely separate the per- mission system from the SDN controller to facilitate its application for any controller without source-code modification. |
| SDN Novel Attacks and Practical Countermeasures | [187] | • In this paper, we evaluate the actual security of the existing mechanisms for network topology discovery in SDN.<br>• They presented 2 novel topology attacks, called Topology Freezing and Reverse Loop, that exploit vulnerabilities in the widely used Floodlight controller. |
| Mobility-Enabled Security for Optimising | [192] | • In this research, they found the security level, energy optimisation and reliability. The have proposed IoT systems because it delivers better reliability and mobility-enabled security. |

This is because a changeable configuration can cause security flaws and performance disruptions. OpenRadio [146] provides declarative programming interfaces through a programmable wireless Data-Plane which gives flexibility at the Physical Layer and MAC layers. Besides that, OpenRadio provides a modular interface that has the capability to execute traffic subsets using different protocols like WiFi and 3GPP LTE-Advanced. Odin [147] introduces programmability in enterprise wireless local area networks (WLANs) through the SDN concepts. WLANs must support authentication, mobility, load balancing, and interference management. Through Odin, the admin can implement enterprise WLAN services as a network application. It builds access point abstraction which simplifies client management.

## 7. Benefits of SDN

Via the centralisation of the network controller, the SDN forwarding devices (switches) become simpler and cheaper compared with the traditional network devices. The network management and configuration are also simplified [2]. Compared with the current network architectures, SDN can be reconfigured faster to respond to the new business requirements. Through SDN, the network performance is improved globally [149]. Besides that, any new application, protocols, and policies can be easily implemented through an application running on the controller which controls the forwarding devices via well-defined APIs, such as the OpenFlow Protocol [2,150]. The researchers in [151] introduced an OpenFlow controller handling IP multicast that is deployed in the Control-Plane, and without making any changes to the forwarding devices, the control software installs the forwarding entries in the switches based on the multicast application. This is possible since the OpenFlow switches support the forwarding operations needed. Should the need arise, as in the case if the protocol needs other operations that are not provided by the OpenFlow specifications or the OpenFlow Data-Plane needs to be upgraded, FLARE [152] is the solution for the programmable Data-Plane. SDN also has the ability to provide network virtualisation via tools such as FlowVisor or OpenVirteX [2,153]. Network virtualisation is the process to combine hardware and software network resources and functionality into a single virtual network, where the SDN allows the network provider to integrate virtual and physical environments [154]. Through network virtualisation and by installing appropriate rules, a controller application can specify the SDN switches' functionalities widely, for example, firewalling, network address translation, and load balancing [2]. Below are some highlights of the specific benefits of SDN. Highlight of the Benefits of SDN are as below:

### *7.1. Content delivery*

Controlling data traffic is one of the primary advantages of SDN. The ability to direct and automate data traffic makes implementing Quality of Services (QoS) [226] for Voice over IP (VOIP), video and audio transmissions much easier [148]. Software defined networking provides a seamless experience for end-users streaming high quality audio and video.

### *7.2. Lessen capital expenditure*

By implementing SDN, businesses can easily optimise existing network devices. Existing hardware can be repurposed to follow the instructions of a SDN controller, and more cost-efficient hardware can be deployed with greater effect.

**Table 8**
Summarised SDN applications.

| SDN application | Overview |
|---|---|
| Rural Connections [48] | • SDN enables the rural infrastructure deployment business done in rural environments and the Internet Service Provider (ISP) business done remotely in cities. |
| Mobile Device Offloading [118] | • ECOS ensures that applications with additional security requirements are only offloaded on approved machines. |
| Date Centres Upgrading [122] | • Google uses SDN technology to interconnect its large number of data centres located at different geographical areas to ensure the data can be provided quickly when requested. |
| Network infrastructure service based on OpenFlow is utilised to connect data centre networks [120] | • Interconnects data centres and improves latency. |
| VMware's network virtualisation platform, NSX [123] | • Simplifies the network management since through the NSX, the network administrator does not need to deal with VLANs and complex sets of firewall rules. |
| Vello system [125] | • Provides a basic set of management constructs in order to enable value-added capabilities for data centre local and wide area networks. |
| In-Packet Bloom Filters (SiBF) [126] | • Introduces Rack Managers that acts as OpenFlow Controllers that provides scalability and maintain the globally required state to provide fault-tolerance in the DCN. |
| Energy-Aware Data-Centre Architecture [127] | • The main purpose is to achieve the concept of "Green Data Centre" in the DCN and provides guidelines to learn about the energy consumption in the DCN elements, such as switches, links, and ports. |
| OpenFlow Switch Controller (OSC) [130] | • Minimises the power consumption of switches in the DCN by minimising the influence of carbon emissions in the DCs. |
| Integrated way to monitor the path load metric. To improve the DCN Metrics [132] | • Introduces an integrated way to monitor the path load metric to administrate link layer multipathing and congestion control. |
| Scissors [133] | • Decreases DC traffic and network power consumption. |
| CrossRoads [135] | • Network fabric that facilitates live and offline virtual machine migration across multiple data centres and provides support for East-West and North-South types of traffic for virtual machine migration. |
| Network Infrastructure As a Service (IaaS) [120] | • Connectivity interruption of virtual machines during the migration process is minimised. |
| Networking as a Service [136] | • Evaluates the provision, delivery, and consumption of Networking as a Service. |
| Software Defined Internet Exchange (SDX) [137,138] | • Creates new programming abstractions which allow participating networks to develop/run the application that is able to behave correctly when they need to interact with the border gateway protocol and do not interfere with each other. |
| OpenRoads [140,141] | • Creates an open program to explore different mobility solutions, routing protocols, and network controllers.<br>• Researchers are able to control the data path using OpenFlow and handle the configuration of the device using the Simple Network Management Protocol (SNMP) |
| SDON [144] | • Improves QoS for a different type of traffic. |
| OpenFlow-Based update mechanisms [169] | • Supports high-level abstractions.<br>• Each packet across the network is processed by a single-fix global network configuration. |
| OpenRadio [146]; Odin [170] | • Provides declarative programming interfaces through a programmable wireless Data-Plane which gives flexibility at the Physical Layer and MAC layers, introduces programmability in WLANs through SDN concepts, and builds access point abstraction which simplifies client management. |

### 7.3. Centralisation

SDN offer a centralised view of an organisations entire network, making it easier to streamline enterprise management and provisioning. As VLANs become a more prominent part of physical LANs, the number of links and dependencies can easily create confusion. SDNs can speed service delivery and provide more agility for both virtual and physical network provisioning, all from a central location [155].

### 7.4. Management

In order to accommodate big data, businesses are constantly setting up new applications and virtual machines to handle processing requests. By implementing SDN, IT teams are able to change network configurations with no effect to the network.

Besides that, all applications can take action from any part of the network. All applications in the network have a global network view. This means all applications are able to access the same network information. The integration between different applications also becomes simpler (for example load balancing and routing applications can be combined sequentially) [46,156]. SDN enables innovation, it allows organisations to rapidly deploy new types of services and applications that can provide new income streams and more value from the network because SDN introduces orchestration that enables a large number of devices to be managed automatically with higher network resource utilisation rates and lower capital costs [157]. SDN also reduces the need to buy ASIC-Based networking hardware and purpose-built [158].

## 8. Emulation tools and testbeds for SDN

### *8.1. Simulation/emulation tools*

The Mininet emulator [159] is used to emulate the OpenFlow networks. It is an open source that has permits to deal with SDN networks. Mininet allows the whole OpenFlow network to be emulated on a single machine by creating a realistic virtual network, running a real kernel, switch, and application code using a single command [160]. Mininet is able to create SDN elements, customise and share them with other networks, and perform interactions [161] (e.g., Hosts, Switches, Controllers, and Links).Through Mininet, it is possible to create a customised network by using Python APIs or directly building some simple network topologies through the Command-Line Interface (CLI). Besides that, Mininet can also work with several different SDN controllers, for example, Floodlight controllers. It allows researchers to rapidly test new algorithms and protocols in a built-in environment since the performance of the emulator depends on the available resources supplied by the host. The Mininet CE [162] and the SDN Cloud-DataCentre [163] are extensions to Mininet to enable wide-scale simulations. The main goal of the Mininet CE is to create upper-level software over Mininet which can combine separate instances of Mininet into one Cluster. EstiNet [164,165] is an OpenFlow network simulator and emulator. One thing that makes EstiNet different from other network simulators/emulators is that it has the capability to enable the unmodified real application to run on simulated hosts since it uses kernel re-entering methodology. This makes the simulation results of the EstiNet simulator accurate and equal with the result obtained from an emulator.

Besides that, EstiNet uses its own simulation clock to manage the simulation event execution order. Because of the kernel re-entering simulation methodology used in EstiNet, the real-life OpenFlow controller programs, such as NOX/POX [166], Floodlight [167] and Ryu [168], can directly run on a simulated host to control simulated OpenFlow switches without making any changes. EstiNet also supports multiple hosts through a single kernel. It is also able to simulate multiple OpenFlow switches.

The Mininet HiFi [171] is a Container-Based Emulation (CBE). It modifies the original Mininet architecture by adding a process for performance segregation, provisioning, and monitoring for performance fidelity. The original Mininet uses lightweight, OS-Level Virtualisation to emulate network links and switches that follow the Imunes [172] system approach. The Mininet-HiFi is suitable for experiments that benefit from flexible routing and topology configuration.

OMNeT++ [173,174], and [175] is a simulator that supports large-scale simulation. It is possible to generate input and output files through commonly available software tools. This simulator supports OpenFlow version 1.2.0 through a plugin.

### *8.2. SDN testbeds*

The California OpenFlow Testbed Network (COTN) [163] is dedicated to OpenFlow Research. It deploys OpenFlow-enabled switches into the backbone of CENIC's CalREN networks and interconnects with other OpenFlow testbeds within the national research network, for example, the Internet2 [176] testbed. The Future Internet Testbed Experimentation between Brazil and Europe (FIBRE) [177] provides large-scale services for future internet research. Its infrastructure mixes the different technologies and heterogeneous physical resources, which includes optical and wireless communication and OpenFlow. The topologies used in the evaluations are fat-tree, BCubic, CERNET, German, SDNLib generated. Controller column indicates the name of the controller used in the experiment (NOX, POX, FL-Floodlight, RY-RYU, and OD-OpenDaylight). SNDLib the network topology and traces provided on SNDLib are widely used data sets in measuring energy efficiency in SDN [213]. The OpenFlow in Europe: Linking Infrastructure and Applications (OFELIA) [179] is based on an OpenFlow which allows virtualisation and controls the network environment through secure interfaces. Through OFELIA, different experiments can run in parallel [180] ElasticTree is a power management solution for data centre networks which is implemented on a testbed consisting of OpenFlow switches. The idea is to turn off links and switches based on the amount of traffic load [214]. The Open-Access Research Testbed for next-generation wireless Networks (ORBIT) [181] is a two-tier wireless network emulator. The main goal of this testbed is to achieve reproducible experimentation, and supports realistic evaluation of protocols and applications. It is used to test the protocols in real-world settings which include the OpenFlow-based network.

Research Infrastructure for Large-Scale Network Experiments (RISE) [182] is a wide area OpenFlow testbed. This testbed is based on the JGN-X network in Japan. In the technique of Carrier Grade, the focus is the energy efficiency and resilience characteristics of carrier grade networks. Related to energy efficiency, it is demonstrated that OpenFlow can reduce network wide network energy consumption and improve scalability [215]. MLTE is implemented together with energy saving mechanisms such as controlled adaptive line rates at the switches [216]. For the resilience, it is shown that OpenFlow can

handle failures at the switches and the controller, and perform recovery with flow restoration. Similar to ElasticTree, energy savings of up to 50% are achieved. REsPoNse is a framework that allows network operators to automatically identify energy-critical paths [217]. It investigates the possibility to pre-compute a few energy-critical paths that, when used in an energy-aware fashion, can continuously produce close-to optimal energy savings over long periods of time. REsPoNse identifies energy-critical paths by analysing the traffic matrices, installs them into a small number of routing tables (called always-on, on-demand, and fail-over), and uses a simple, scalable online traffic engineering mechanism to deactivate and activate network elements on demand. The network operators can use REsPoNse to overcome power delivery limits by provisioning power and cooling of their network equipment for the typical, low to medium level of traffic. FLOWP attempts to achieve both power reduction and QoS for fat-tree topology. An IP formulation for power efficient flow scheduling and the corresponding heuristics is proposed [218]. GreenRE uses redundancy removal (RE) to achieve energy efficiency where the motivation is stated as follows. Networks exhibit several redundant links while users access similar contents. Even though redundancy increases reliability, it also degrades the performance of the network. Instead of sending the same data through many different paths repeatedly, sending it through a single link increases the throughput and in effect reduces the load in the links [219] Global Environment for Network Innovations (GENI) [178] provide a large-scale experiment infrastructure by giving the user access to hundreds of distributed resources including network resources (e.g., WiMaz base station, links, and switches) and computer resources (e.g., Virtual machine). Through the GENI, the user can program both ends, a host of an experimental network and switches in the core of a network, which allows the user to experiment with novel IP-routing algorithms.

## 9. Conclusion

SDN is becoming popular due to the interesting features it offers that unlock innovation in how we design and organise networks. However, there are still important challenges to be solved before realising successful SDN. In this survey, we introduce existing SDN-related technologies and also argue that data plane programmability needs to be considered in the SDN definition. SDN created an opportunity for solving the Traditional Network problems. For example, in Traditional Networks, the Control and Data Planes are vertically integrated. This causes each of the elements in the network to have their own specific configuration and management interface. This makes the management of the network become complex. Through the SDN, the network management becomes simpler because SDN allows dynamic programmability in forwarding devices (Control-Plane elements) since the Control and Data Planes are decoupled. Besides that, SDN provides a global view of the network by logical centralisation of the Control-Plane elements. This paper surveys the state-of-the-art contribution such as a comparison between SDN and traditional networking. Also, comparison with other survey works on SDN, new information about controller, details about OpenFlow architecture, configuration, comprehensive contribution about SDN security threat and countermeasures, SDN applications, benefit of SDN, and Emulation & Tested for SDN. In future, we will discuss about SDN security and applications.

## Declaration of Competing Interest

None.

## Acknowledgement

The survey paper collaboration among University Malaysia Sabah, Victoria University of Wellington, New Zealand, and Universiti Kebangsaan Malaysia (UKM). The authors would like to thanks Professor Dr. Yong-Jin Park (IEEE Life member) Former Director IEEE Region 10 for his expertise in SDN, IoT & Future Networks, his valuable comments and suggestions to improve the quality of the paper. This work was partially supported by under the grant scheme GUG0072-SG-2/2016 and the Faculty of Computing and Informatics, University Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia.

## References

[1] Z. Guo, W. Chen, Y. Liu, Y. Xu, Z. Zhang, Joint switch upgrade and controller deployment in hybrid software-defined networks, IEEE J. Select. Areas Commun. 37 (5) (2019) 1012–1028, doi:10.1109/JSAC.2019.2906743.
[2] E. Jimson, K. Nisar, M. Hijazi, Bandwidth management using software defined network and comparison of the throughput performance with traditional network, in: Proceedings of the International Conference on Computer and Drone Applications (ICONDA), Kuching, Malaysia, 2017, pp. 71–76, doi:10.1109/ICONDA.2017.8270402.
[3] I.A. Lawal, A.M. Said, K. Nisar, P.A. Shah, A.A. Mu'azu, A distributed model to analysed QoS parameters performance improvement for fixed WiMAX networks, Advances in Computer Science and its Applications, Springer Link, Lecture Notes in Electrical Engineering Book series, 279, LNEE, 2013, pp. 695–701.
[4] T. Hu, Z. Guo, P. Yi, T. Baker, J. Lan, Multi-controller based software-defined networking: a survey, IEEE Access 6 (2018) 15980–15996.
[5] CT. Yuan, X. Huang, M. Ma, J. Yuan, Balance-based SDN controller placement and assignment with minimum weight matching, in: Proceedings of the IEEE International Conference on Communications (ICC), 2018, pp. 1–6.
[6] A. Binsahaq, T. R. Sheltami, K. Salah, A survey on autonomic provisioning and management of QoS in SDN networks, IEEE Commun. Surv. Tutor. 7 (2019) 73384–73435.
[7] G. Wang, Y. Zhao, J. Huang, Y. Wu, An effective approach to controller placement in software defined wide area networks, IEEE Trans. Netw. Serv. Manag. 15 (1) (2018) 344–355.
[8] B.P.R. Killi, S.V. Rao, Capacitated next controller placement in software defined networks, IEEE Trans. Netw. Serv. Manag. 14 (3) (2017) 514–527.

[9] N.F. Ali, A. Said, K. Nisar, I. Aziz, A survey on software defined network approaches for achieving energy efficiency in wireless sensor network, in: Proceedings of the IEEE Conference on Wireless Sensors (ICWiSe), Miri, Malaysia, 2017, pp. 28–33, doi:10.1109/ICWISE.2017.8267157.

[10] K. Nisar, E.R. Jimson, M. Hijazi, A.A.A. Ibrahim, Y.J. Park, I. Welch, A new bandwidth management model using software-defined networking security threats, in: Proceedings of the 13th IEEE International Conference on Application of Information and Communication Technologies (AICT), Baku, Azerbaijan, 2019, pp. 1–3. https://ieeexplore.ieee.org/document/8981784.

[11] R.D.R. Fontes, C. Campolo, C.E. Rothenberg, A. Molinaro, From theory to experimental evaluation: resource management in software-defined vehicular networks, IEEE Access 5 (2017) 3069–3076.

[12] D.R. di Lallo, et al., Leveraging SDN to monitor critical infrastructure networks in a smarter way, in: Proceedings of the IFIP/IEEE Symposium on Integrated Network and Service Management (IM), 2017, pp. 608–611, doi:10.23919/INM.2017.7987341.

[13] M.B. Anwer, M. Motiwala, M. Tariq, N. Feamster, "Switchblade: a platform for rapid deployment of network protocols on programmable hardware,", in: Proceedings of the ACM SIGCOMM Computing, 2010.

[14] H.I. Kobo, A.M. Abu-Mahfouz, G.P. Hancke, A survey on software-defined wireless sensor networks: challenges and design requirements, IEEE Access 5 (2017) 1872–1899.

[15] X. Jia, Y. Jiang, Z. Guo, Incremental switch deployment for hybrid software-defined networks, in: Proceedings of the IEEE 41st Conference on Local Computer Networks (LCN), Dubai, 2016, pp. 571–574, doi:10.1109/LCN.2016.

[16] J. Yang, Z. Yao, B. Yang, X. Tan, Z. Wang, Q. Zheng, Software-defined multimedia streaming system aided by variable-length interval in-network caching, IEEE Trans. Multimedia 21 (2) (2018) 494–509.

[17] N.I. Sarkar, A. X.-M. Kuang, K. Nisar, and A. Amphawan, "Hospital Environment Scenarios Using WLAN Over OPNET Simulation Tool", Healthcare Administration: Concepts, Methodologies, Tools, and Applications, Information Resources Management Association Chapter 40, pp. 789-804, doi: 10.4018/978-1-4666-6339-8.ch040

[18] K. Nisar, E.R. Jimson, M. Hijazi, S.K. Memon "A Survey, Architecture, security threats and application of SDN", J. Ind. Electron. Technol. Appl. 2 (1) (2019) 64–69 Daegu University, Republic of KoreaeISSN: 2635-635X http://jieta.org/v2n101/ .

[19] S. Shenker, Software-defined Networking (SDN), University of California, Berkeley, 2014.

[20] K. Nisar, E.R. Jimson, M. Hijazi, S.K. Memon, Software defined network and comparison of the throughput performance with traditional network, J. Ind. Electron. Technol. Appl. 3 (4) (2019) 298–310 Daegu University, Republic of KoreaeISSN: 2586-0852http://jiita.org/v3n402/.

[21] E.R. Jimson, K. Nisar, M. Hijazi, The State of the art of software defined networking (SDN) issues in current network architecture and a solution for network management using the SDN, Int. J. Technol. Diffus. 10 (3) (2019) 33–48 IGI Global Publishers, Hershey, PA, USA, doi:10.4018/IJTD.2019070103.

[22] N. Feamster, J. Rexford, E. Zegura, The road to SDN. An intellectual history of programmable networks, ACM SIGCOMM Comput. Commun. Rev. 11 (12) (2013) 1–14.

[23] I. PepeInjak, How Did Software Defined Networking Start?, ipSpace, 2013 http://blog.ipspace.net/2014/01/how-did-software-defined-networking.html.

[24] N. Bizanis, F.A. Kuipers, SDN and virtualization solutions for the Internet of Things: a survey, IEEE Access 4 (2016) 5591–5606.

[25] B. Yan, Y. Xu, H.J. Chao, Adaptive wildcard rule cache management for software-defined networks, IEEE ACM Trans. Netw. 2 (2) (2018).

[26] A. Hakiri, A. Gokhale, P. Berthou, D.C. Schmidt, T. Gayraud, Software-defined networking: challenges and research opportunities for future Internet, Comput. Netw. 75 (2014) 453–471.

[27] N. Feamster, J. Rexford, E. Zegura, The past, present, and future of software defined networking, Commun, ACM, 2013.

[28] P. Ranjan, R. Oswal, R. Bedi, P. Pande, Z. Qurani, A Survey of past, present and future of software defined networking, Int. J. Adv. Res. Comput. Sci. Manag. Stud. 2 (4) (2014) 238–248.

[29] J. Rexford, A. Greenberg, G. Hjalmtysson, D. a. Maltz, A. Myers, G. Xie, J. Zhan, H. Zhang, Network-wide decision making: toward a wafer-thin control plane, HotNets III (2004) 1–5.

[30] J. Reich, C. Monsanto, N. Foster, J. Rexford, D. Walker, "Modular SDN programming with pyretic, USENIX 38 (2013) 40–47.

[31] S. Lange, et al., Specialized heuristics for the controller placement problem in large scale SDN networks, in: Proceedings of the IEEE 27th International Teletraffic Congress (ITC), 2015, pp. 210–218.

[32] L. Lee, Y. Wei, A.A.A. Ibrahim, K. Nisar, Z. Iswandono, A. Ismail, I. Welch, Survey on geographic visual display techniques in epidemiology: taxonomy and characterization, J. Ind. Inf. Integr. 18 (2) (2020) Elsevier01-14, Impact Factor 3.30, doi:10.1016/j.jii.2020.100139.

[33] E.R. Jimson, K. Nisar, M. Hijazi, The state of the art of software defined networking (SDN): network management solution in current network architecture using the SDN, Int. J. Inf. Commun. Technol. Hum. Dev. 10 (4) (2018) 44–60, doi:10.4018/IJICTHD.2018100104.

[34] M. Kobayashi, S. Seetharaman, G. Parulkar, G. Appenzeller, J. Little, J. Van Reijendam, P. Weissmann, N. McKeown, Maturing of OpenFlow and software-defined networking through deployments, Comput. Netw. 61 (2014) 151–175.

[35] R. Horvath, D. Nedbal, M. Stieninger, A literature review on challenges and effects of software defined networking, Procedia Comput. Sci. 64 (2015) 552–561.

[36] E. Lakiotakis, C. Liaskos, X. Dimitropoulos, Application-network collaboration using SDN for ultra-low delay teleorchestras, in: Proceedings of the IEEE Symposium on Computers and Communications, Pediswesa, 2017, pp. 70–75.

[37] W. Li, H. Qi, K. Li, I. Stojmenovic, Lan, Joint optimization of bandwidth for provider and delay for user in software defined data centers, IEEE Trans. Cloud Comput. 5 (2) (2017) 331–343.

[38] P. Bellavista, C. Giannelli, T. Lagkas, P. Sarigiannidis, Multi-domain SDN controller federation in hybrid FiWi-MANET networks, Eurasip J. Wirel. Commun. Netw. 1 (2018) 103.

[39] P. Xiao, Z.-Y. Li, S. Guo, H. Qi, W.-Y. Qu, H.-S. Yu, A K self- adaptive SDN controller placement for wide area networks, Front. Inf. Technol. Electron. Eng. 17 (7) (2016) 620–633.

[40] K. Nisar, A. Saudi, Smart home: multisensor information fusion towards better healthcare, Adv. Sci. Lett. 24 (3) (2018) 1896–1901 American Scientific Publishers, USAISSN 1936-6612, doi:10.1166/asl.2018.11184.

[41] "RFC 5810 - Forwarding and Control Element Separation (ForCES) Protocol Specification", Tools.ietf.org, 2010.

[42] J. Shuja, R.W. Ahmad, A. Gani, A.I.A. Ahmed, A. Siddiqa, K. Nisar, S.U. Khan, A.Y. Zomaya, Greening emerging IT technologies: techniques and practices, J. Internet Serv. Appl. (2017) 1–11 Springer, London, United Kingdom, Vol. 8.9, doi:10.1186/s13174-017-0060-5.

[43] Migrating to SDN: Planning for a Smooth Transition, 1st ed., Brocade Communications Systems, Inc., 2014, pp. 1–2.

[44] "CDW LLC. People Who Get ITTM, "The Future Of Networking Arrives," 2015. <http://webobjects.cdw.com/webobjects/media/pdf/solutions/Networking/The-Future-of-Networking-Arrives-MKT2862.pdf>".

[45] H. Hasbullah, A. Said, K. Nisar, The effect of echo on voice quality in VoIP network, Adv. Comput. Sci. Eng. (2009) 95–100 Phuket, Thailand. 2009.

[46] D. Kreutz, F.M.V Ramos, P. Verissimo, C.E. Rothenberg, S. Azodolmolky, S. Member, S. Uhlig, Software-defined networking : a comprehensive survey, Proc. IEEE 103 (1) (2015) 14–76.

[47] K. Nisar, H. Hasbullah, The effect of panoramic view of a digital map on user satisfaction, in: Proceedings of the International Symposium on Information Technology (ITSim), Kuala Lumpur, Malaysia, IEEE, KLCC, 2008, pp. 1–4, doi:10.1109/ITSIM.2008.4631613.

[48] F. Hu, Q. Hao, K. Bao, A survey on software defined networking (SDN) and OpenFlow: from concept to implementation, IEEE Commun. Surv. Tutor. 16 (2014).

[49] B.A.A. Nunes, M. Mendonca, X. Nguyen, K. Obraczka, T. Turletti, A survey of software-defined networking : past, present , and future of programmable networks, IEEE Commun. Surv. Tutor. 16 (3) (2014) 1617–1634.

[50] X. Foukas, M. K. Marina, and K. Kontovasilis, " Software Defined Networking Concepts," 2014.

[51] M.H.A. Hijazi, T.C. Beng, J. Mountstephens, Y. Lim, K. Nisar, Malware classification using ensemble classifiers, Adv. Sci. Lett. 24 (2) (2018) 1172–1176 American Scientific Publishers, USA, doi:10.1166/asl.2018.10710.

[52] O. N. Narmanlioglu, E. Zeydan, Software-defined networking based network virtualization for mobile operators, Comput. Electr. Eng. 57 (2017) 134–146.
[53] T.Q. Ngo, Z. Yan, J. Katto, Y.-J. Park, H. Nakazato, W. Kameyama, K. Nisar, A.A.A. Ibrahim, Mobility Support for Content-oriented Publish/Subscribe System, The Institute of Electronics, Information and Communication Engineers (IEICE), Tokyo, Japan, 2015, pp. 31–34. IEICE Technical ReportVol..
[54] S. Raza and D. Lenrow, " North Bound Interface Working Group (NBI-WG) Charter Final Version: V 1.1," 2013.
[55] "What are SDN Southbound APIs? - Where They Are Used.", SDxCentral. [Online]. Available: https://www.sdxcentral.com/sdn/definitions/southbound-interface-api/. [Accessed: 18- Aug- 2016]".
[56] Open Networking Foundations (ONF), OpenFlow management and configuration protocol, OF-CONFIG 1.2, OpenFlow Manag. Config. Protoc. (2014) 1–44.
[57] M. Oswalt, "[SDN Protocols] Part 3 - OVSDB", Keepingitclassless.net, 2014. [Online]. Available: [Accessed: - ].," 2014.
[58] Open Networking Foundation, "SDN Architecture," no. 1, p. 1–68, 2014.
[59] Open Networking Foundation, "SDN Architecture Overview Version 1.0," p. 1–5, 2013.
[60] SDN and the Future of Service Provider Networks, 1st ed., Fujitsu Network Communications Inc, 2016 pp. Sec1:1.
[61] K. Nisar, A.M. Said, H. Hasbullah, Enhanced performance of IPv6 packet transmission over VoIP network, in: Proceedings of the 2nd IEEE International Conference on Computer Science and Information Technology, 2009, Beijing, China, ICCSIT, 2009, pp. 500–504, doi:10.1109/ICCSIT.2009.5234652.
[62] J. Xie, D. Guo, Z. Hu, T. Qu, Control plane of software defined networks: a survey, Comput. Commun. (2015) 1–15 vol. 00.
[63] Kashif Nisar, Abas MD Said, Halabi Hasbullah, Internet call delay on peer to peer and phone to phone VoIP network, in: Proceedings of the International Conference on Computer Engineering and Technology (ICCET), Singapore, IEEE, 2009, pp. 517–520, doi:10.1109/ICCET.2009.258.
[64] Q. Gao, W. Tong, S. Kausar, L. Huang, C. Shen, S. Zheng, Congestion-aware multicast plug-in; for an SDN network operating system, Comput. Netw. 125 (2017) 53–63.
[65] K. Nisar, A.M. Said, H. Hasbullah, Enhanced performance of WLANs packet transmission over VoIP network, in: Proceedings of the IEEE 24th International Conference on Advanced Information Networking and Applications, Workshops, (AINA), Perth, Western Australia, 2010, pp. 485–490, doi:10.1109/WAINA.2010.76.
[66] M. Smith, R. Adams, Y. Laribi, V. Pandey, and P. Garg, " OpFlex Control Protocol," 1, p. 1–24, 2016.
[67] K. Nisar, G. Chen, A. Sarrafzadeh, A review: software-defined networking implementation and testing, in: Proceedings of the Asia-Pacific Advanced Network (APAN) Network Research Workshop, Fukuoka, Japan, 2015, pp. 1–09. Vol. 39.
[68] G. Bianchi, M. Bonola, A. Capone, C. Cascone, OpenState: programming platform-independent stateful OpenFlow applications inside the switch, ACM SIGCOMM Comput. Commun. Rev. 44 (2) (2014) 1–7.
[69] M. Sune, V. Alvarez, T. Jungel, U. Toseef, K. Pentikousis, An OpenFlow implementation for network processors, in: Proceedings of the 3rd European Workshop Software-Defined Networks, EWSDN, 2014.
[70] B. Belter, A. Binczewski, K. Dombek, A. Juszczyk, L. Ogrodowczyk, D. Parniewicz, M. Stroinski, I. Olszewski, Programmable abstraction of datapath, in: Proceedings of the Third European Workshop Software-Defined Networks, 2014.
[71] R. Masoudi, A. Ghaffari, Software defined networks: a survey, J. Netw. Comput. Appl. 67 (2016) 1–25.
[72] W. Stallings, Software-defined networks and OpenFlow, Internet Protoc. J. 16 (1) (2013).
[73] T.-Y. Mu, A. Al-Fuqaha, K. Shuaib, F.M. Sallabi, J. Qadir, SDN flow entry management using reinforcement learning, ACM Trans. Auton. Adapt. Syst. 13 (2) (2018) 1–23.
[74] Y.R. Chiang, C.H. Ke, Y.S. Yu, Y.S. Chen, C.J. Pan, A multipath transmission scheme for the improvement of throughput over SDN, in: Proceedings of the IEEE International Conference on Applied System Innovation for Modern Technology, ICASI, 2017, pp. 1247–1250.
[75] S. Fichera, M. Gharbaoui, P. Castoldi, On Experimenting 5G: Testbed Set-up for SDN Orchestration Across Network Cloud and IoT Domains. In: Network Softwarization (NetSoft), pp. 1–6, 2017.
[76] P.T. Congdon, P. Mohapatra, M. Farrens, V. Akella, Simultaneously reducing latency and power consumption in OpenFlow switches, IEEE ACM Trans. Netw. 22 (3) (2014) 1007–1020.
[77] A. Khan, N. Dave, "Enabling hardware exploration in software-defined networking: a flexible, portable openflow switch, in: Proceedings of the 21st Annual International IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM), 2013.
[78] Z. Guo, Y. Xu, R. Liu, A. Gushchin, K.-Y. Chen, A. Walid, H.J. Chao, Balancing flow table occupancy and link utilization in software-defined networks, Future Gener. Comput. Syst. 89 (2018) 213–223.
[79] Z. Guoa, R. Liub, Y. Xuc, A, Gushchind, A. Walid, H.J. Chaoc, STAR: preventing flow-table overflow in software-defined networks, Comput. Netw. 125 (2017) 15–25.
[80] M Hoffmann, et al., SDN and NFV as enabler for the distributed network cloud, Mob. Netw. Appl. 23 (3) (2017) 521–528.
[81] A. Metzler, "Ten Things to Look for in an SDN Controller," p. 1–14, 2013.
[82] Security in software-defined networking: threats and countermeasures, Mob. Netw. Appl. (2016) 1–13.
[83] Open Networking Foundation, "Principles and Practices for Securing Software - Defined Networks," p. 1–27, 2015.
[84] A. Akhunzada, A. Gani, N.B. Anuar, A. Abdelaziz, M.K. Khan, A. Hayat, S.U. Khan, Secure and dependable software defined networks, J. Netw. Comput. Appl. 61 (2016) 199–221.
[85] T. Ball, N. Bjørner, A. Gember, S. Itzhaky, A. Karbyshev, M. Sagiv, M. Schapira, A. Valadarsky, VeriCon: towards verifying controller programs in software-defined networks, in: Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design Implementation, 2014.
[86] M. Canini, D. Venzano, P. Perešíni, D. Kostić, J. Rexford, A nice way to test OpenFlow applications, in: Proceedings of the 9th USENIX Conference on Networked System Design Implementation, 2012.
[87] P. Perešínia, M. Kuzniara, M. Canini, D. Venzanoc, D. Kostić, J. Rexforde, Systematically testing OpenFlow controller applications, Comput. Netw. 92 (2015) 270–286.
[88] S.T. Ali, V. Sivaraman, A. Radford, S. Jha, Securing networks using software defined networking : a survey, IEEE Trans. Reliab. 64 (3) (2013) 1–12.
[89] S. Son, S. Shin, V. Yegneswaran, P. Porras, G. Gu, Model checking invariant security properties in OpenFlow, in: Proceedings of the IEEE International Conference on Communications, 2013.
[90] R.M. Ramos, M. Martinello, C.E. Rothenberg, SlickFlow: resilient source routing in data center networks unlocked by OpenFlow, in: Proceedings of the Conference on Local Computer Networks (LCN), 2013, pp. 606–613.
[91] Z. Guo, Y. Xu, M. Cello, J. Zhang, Z. Wang, M. Liu, H.J. Chao, JumpFlow: reducing flow table usage in software-defined networks, Comput. Netw. 92 (2015) 300–315.
[92] N. Kitsuwan, S. Ba, E. Oki, T. Kurimoto, S. Urushidani, Flows reduction scheme using two MPLS tags in software-defined network, IEEE Access 5 (2017) 14626–14637.
[93] O. N. F. (ONF), "OpenFlow Switch Specification Version 1.3.0," p. 1–105, 2012.
[94] O. N. F. (ONF), "OpenFlow Switch Specification Version 1.4.0," p. 1–206, 2013.
[95] H. Mai, A. Khurshid, R. Agarwal, M. Caesar, P.B. Godfrey, S.T. King, R. Agarwal, M. Caesar, P.B. Godfrey, S.T. King, Debugging the data plane with anteater, ACM SIGCOMM Comput. Commun. Rev 41 (2011) 290.
[96] P. Kazemian, M. Change, H. Zheng, Real time network policy checking using header space analysis, in: Proceedings of the USENIX Symposium on Networked Systems and Design Implementation, 2013.
[97] Z. Ali, A. Aman and R. Hassan, "Cloud Query Processing Analysis: Encryption," 3C Tecnología, no. October, pp. 28-39, 2019.
[98] I. Alsmadi, D. Xu, Security of software defined networks: a survey, Comput. Secur. 53 (2015) 79–108.
[99] A.A. S., R. Hassan, N.E. Othman, A.N. I., Y. Kenish, Impacts evaluation of DoS attacks over IPv6 neighbor discovery protocol, J. Comput. Sci. 15 (5) (2019) 702–727.

[100] A.A. S., R. Hassan, Denial of service attack over secure neighbor discovery (SeND), Int. J. Adv. Sci. Eng. Inf. Technol. 8 (5) (2018) 1897–1904.
[101] A.A. M., R. Hassan, A proposed technique to detect DDoS attack on IPv6 web applications, in: Proceedings of the 4th International Conference on Parallel Distributed and Grid Computing, Waknaghat, India, 2016.
[102] K. Nisar, A. Amphawan, S. Hassan and N. I. Sarkar, "A comprehensive survey on scheduler for VoIP over WLANs," J. Netw. Comput. Appl., Norman, OK, USA, Vol. 36, No. 2, pp. 933-948, 01, March. 2013, ISSN: 1084-8045 (Indexing in Elsevier, Impact Factor 5.273) 10.1016/j.jnca.2012.07.019
[103] H. Wang, L. Xu, G. Gu, FloodGuard: A DoS attack prevention extension in software-defined networks, in: Proceedings of the International Conference on Dependable Systems and Networks, 2015.
[104] S. Lim, J. Ha, H. Kim, Y. Kim, S. Yang, A SDN-oriented DDoS blocking scheme for botnet-based attacks, in: Proceedings of the International Conference on Ubiquitous and Future Networks, ICUFN, 2014.
[105] K. Nisar, A.M. Said, H. Hasbullah, Enhanced performance of packet transmission using system model over VoIP network, in: Proceedings of the International Symposium on Information Technology (ITSim), IEEE, KLCC, Kuala Lumpur, Malaysia, 2010, pp. 1005–1008, doi:10.1109/ITSIM.2010.5561593.
[106] S. Shin, P. Porras, V. Yegneswaran, M. Fong, G. Gu, M. Tyson, FRESCO : modular composable security services for software-defined networks, in: Proceedings of the ISOC Network and Distributed System Security Symposium, 2013.
[107] W.R.S. Osman, K. Nisar, A.M.M. Altrad, Demonstrate broadband over power line network in Malaysia, in: Proceedings of the 2014 IEEE International Conference on Consumer Electronics, Shenzhen, China, 2014, pp. 1–6, doi:10.1109/ICCE-China.2014.7029864.
[108] A.I. Montoya-mu, D.M. Casas-velasco, F. Estrada-solano, A. Ordonez, D. De Telem, A Yang model for a vertical SDN management plane, in: Proceedings of the Communications and Computing (COLCOM), 2017, pp. 1–6.
[109] K. Nisar, I.A. Lawal, K. Abualsaud, T.M. El-Fouly, A new WDM application response time in WLAN network and fixed WiMAX using distributed, in: Proceedings of the 11th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA), Doha, Qatar, 2014, pp. 781–787, doi:10.1109/AICCSA.2014.7073280.
[110] S. Hong, L. Xu, H. Wang, and G. Gu, "Poisoning Network Visibility in Software-defined Networks: New Attacks and Countermeasures," in Ndss '15, 2015.
[111] W.R.S. Osman, K. Nisar, A.M.M. Altrad, Evaluation of broadband PLC technology over Malaysia's indoor power line network, in: Proceedings of the 2nd International Conference on Electronic Design (ICED), Penang, Malaysia, 2014, pp. 275–280, doi:10.1109/ICED.2014.7015813.
[112] K. Nisar, W.R.S. Osman, A.M.M. Altrad, Modeling of broadband over in-door power line network in Malaysia, in: Proceedings of the 10th International Conference on Computing and Information Technology, Advances in Intelligent Systems and Computing, 265, Angsana Laguna Phuket, Thailand, Springer, 2014, pp. 213–222, doi:10.1007/978-3-319-06538-0_21. Verlag in Lecture Notes in Electrical Engineering). ISBN 978-3-319-06538-0.
[113] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao, " Are We Ready for SDN ? Implementation Challenges for Software-defined Networks," p. 36–43, 2013.
[114] P. Porras, S. Shin, V. Yegneswaran, M. Fong, M. Tyson, G. Gu, A security enforcement kernel for OpenFlow networks, in: Proceedings of the First Workshop Hot Topics in Software Defined Networks (HotSDN), 2012.
[115] A. Khurshid, X. Zou, W. Zhou, M. Caesar, P.B. Godfrey, Veriflow: verifying network-wide invariants in real time, in: Proceedings of the ACM SIGCOMM Computing, 2012.
[116] W. You, K. Qian, X. He, and Y. Qian, "Towards Security in Virtualization of SDN," vol. 8, no. 8, p. 1386–1389, 2014.
[117] G. Yao, J. Bi, P. Xiao, Source address validation solution with OpenFlow/NOX architecture,", in: Proceedings of the International Conference on Network Protocols, ICNP, 2011.
[118] I. Farris, T. Taleb, Y. Khettab, J. Song, A Survey on emerging SDN and NFV security mechanisms for IoT systems, IEEE Commun. Surv. Tutor. 21 (1) (2019) 812–837 Firstquarter, doi:10.1109/COMST.2018.2862350.
[119] M. Jammal, T. Singh, A. Shami, R. Asal, Y. Li, "Software defined networking : state of the art and research challenges, Comput. Netw. 72 (2014) 74–98.
[120] B. Boughzala, R. Ben Ali, M. Lemay, Y. Lemieux, O. Cherkaoui, OpenFlow supporting inter-domain virtual machine migration, in: Proceedings of the IEEE Conference Publications, 2011.
[121] S. Chaudhary, A. Amphawan, K. Nisar, Realization of free space optics with OFDM under atmospheric turbulence, Optik 125 (18) (2014) 5196–5198 ISSN: 0030-4026, Impact Factor I.914, doi:10.1016/j.ijleo.2014.05.036.
[122] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, A. Vahdat, B4: Experience with a globally-deployed software defined WAN, in: Proceedings of the ACM SIGCOMM 2013 Conference, 2013.
[123] M Karakus, A Durresi, Quality of service (QoS) in software defined networking (SDN): a survey, J. Netw. Comput. Appl. 80 (2017) 200–218.
[124] J. Oltsik, B. Laliberte, "ESG Brief IBM and NEC Bring SDN / OpenFlow to Enterprise Data Center Networks," 2012.
[125] "Optimizing Cloud Infrastructure With Software-defined Networking".
[126] A. Amphawan, V. Mishra, K. Nisar, B. Nedniyom, Real-time holographic backlighting positioning sensor for enhanced power coupling efficiency into selective launches in multimode fiber, J. Mod. Opt. 59 (20) (2012) 1745–1752 Taylor & Francis Group 4 Park Square Milton Park Abingdon Oxfordshire OX14 4RN United KingdomIndexed by Scopus, Impact Factor 1.657, doi:10.1080/09500340.2012.739713.
[127] N.H. Thanh, P.N. Nam, T.-H. Truong, N.T. Hung, L.K. Doanh, R. Pries, Enabling experiments for energy-efficient data center networks on OpenFlow-based platform, in: Proceedings of the IEEE Conference Publications, 2012.
[128] J. Koomey, J. R. Stanley, and K. G. Brill," Four Metrics Define Data Center ' Greenness," 2007.
[129] O. N. F. (ONF), "Threat Analysis for the SDN Architecture," 2016.
[130] T.H. Vu, P.N. Nam, T. Thanh, N.H. Thanh, L.T. Hung, L.A. Van, N.D. Linh, T.D. Thien, Power aware OpenFlow switch extension for energy saving in data centers, in: Proceedings of the IEEE Conference Publications, 2012.
[131] G. Gibb, J.W. Lockwood, J. Naous, P. Hartke, N. Mckeown, NetFPGA – an open platform for teaching how to build gigabit-rate network switches and routers, IEEE Trans. Educ. (2008) 1–22.
[132] S. Fang, Y. Yu, C. Heng Foh, K. Mi Mi Aung, A loss-free multipathing solution for data center network : using software-defined networking approach, IEEE J. Mag. 49 (6) (2013) 2723–2730.
[133] S. Chaudhary, A. Amphawan, K. Nisar, Realization of free space optics with OFDM under atmospheric turbulence, Optik 125 (18) (2014) 5196–5198 Impact Factor 1.914, doi:10.1016/j.ijleo.2014.05.036.
[134] L. Sun, K. Suzuki, C. Yasunobu, Y. Hatano, H. Shimonishi, A network management solution based on OpenFlow towards new challenges of multitenant data center, in: Proceedings of the IEEE Conference Publications, 2012.
[135] V. Mann, A. Vishnoi, K. Kalapriya, S. Kalyanaraman, CrossRoads : seamless VM mobility across data centers through software defined networking, in: Proceedings of the IEEE Conference Publications, 2012.
[136] T. Feng, J. Bi, H. Hu, and H. Cao, "Networking as a Service : A Cloud-based Network Architecture," vol. 6, no. 7, p. 1084–1090, 2011.
[137] A Shirmarz, A Ghaffari, An adaptive greedy flow routing algorithm for performance improvement in software defined network, Int. Numer. Model Electron. Netw. Dev. Fields (2019) 1–21 Wiley online Libr..
[138] N. Feamstery, J. Rexford, S. Shenkerz, D. Levin, R. Clarky, and J. Bailey, "SDX: A Software Defined Internet Exchange," 2014.
[139] N.I. Sarkar, A.X.-M. Kuang, K. Nisar, A. Amphawan, Hospital environment scenarios using WLAN over OPNET simulation tool, Int. J. Inf. Commun. Technol. Hum. Dev. 6 (2014) 69–90.
[140] N.I Sarkar, K. Nisar, Performance of VoIP in wired-cum-wireless Ethernet network, Int. J. Interdiscip. Telecommun. Netw. 4 (4) (2012) 1–25 IGI Global Publishers, Hershey, PA, USA, doi:10.4018/jitn.2012100101.
[141] K. Yap, M. Kobayashi, R. Sherwood, T. Huang, M. Chan, N. Handigol, N. Mckeown, OpenRoads : empowering research in mobile networks, ACM SIGCOMM Comput. Commun. Rev. 40 (1) (2010) 125–126.
[142] A.M.M. Altrad, W.R.S. Osman, K. Nisar, Modelling of remote area broadband technology over low voltage power line channel, Int. J. Comput. Netw. Commun. 4 (5) (2012) 187–201, doi:10.5121/ijcnc.2012.4512.

[143] K. Nisar, N.I Sarkar, Modeling and performance studies of ATM networks over voice & video, Int. J. Technol. Diffus. 3 (3) (2012) 47–56 IGI Global Publishers, Hershey, PA, USA, doi:10.4018/jtd.2012070105.
[144] A.N. Patel, P.N. Ji, T. Wang, QoS-aware optical burst switching in OpenFlow based software-defined optical networks, Opt. Netw. Des. (2013) 275–280.
[145] N.I Sarkar, K. Nisar, Modelling and performance studies of ATM networks over email & FTP, Int. J. Adv. Pervasive Ubiquitous Comput. 4 (2) (2012) IGI Global Publishers, Hershey, PA, USA, doi:10.4018/japuc.2012040102.
[146] M. Bansal, J. Mehlman, S. Katti, and P. Levis, "OpenRadio: A Programmable Wireless Dataplane," in HotSDN, 2012.
[147] N.I. Sarkar, K. Nisar, L. Babbage, Performance studies on campus-wide focus on FTP, video and VoIP Ethernet network, Int. J. Adv. Pervasive Ubiquitous Comput. 4 (1) (2012) 49–59 IGI Global Publishers, Hershey, PA, USA, doi:10.4018/japuc.2012010106.
[148] M. Algarni, V. Nair, D. Martin, and S. Shirgaonkar, "Software-Defined Networking Overview and Implementation".
[149] W. Xia, Y. Wen, C. Foh, D. Niyato, A survey on software-defined networking, Surv. Tutor. 17 (1) (2015) 27–51.
[150] K. Nisar, Fourth stage of voice priority queue for VoIP over WLANs, Int. J. Interdiscip. Telecommun. Netw. 4 (2) (2012) 48–63 IGI Global Publishers, Hershey, PA, USA, doi:10.4018/jitn.2012040104.
[151] K. Nisar, N.I. Sarkar, Y. Dole, Performance studies of voice and video conferencing over ATM and gigabit Ethernet backbone networks, Int. J. Technol. Diffus. 3 (1) (2012) 22–32 IGI Global Publishers, Hershey, PA, USA, doi:10.4018/jtd.2012010103.
[152] K. Nisar, S. Hassan, Mobility of mobile station of voice priority queue for VoIP over WLAN networks, Int. J. Comput. Sci. Eng. Inf. Technol. 2 (3) (2012) 28–37, doi:10.20454/jcce.2012.249.
[153] A. Al-Shabibi, M. De Leenheer, M. Gerola, A. Koshibe, W. Snow, G. Parilkar, OpenVirteX: a network hypervisor, Open Netw. Summit (2014) 1–2.
[154] K. Nisar, N. Sarkar, S. Hassan, Di Wu, Performance studies of VoIP over Ethernet LANs, Int. J. Comput. Internet Manag. 19 (3) (2011) 1–07 (IJCIM0, Bangkok, Thailand http://www.ijcim.th.org/past_editions/2011V19N3/Paper_1-v19n3-p1-7.pdf .
[155] L. Hewlett-Packard Development Company, "Software-defined Networking and Network Virtualization[Technical white paper]," 2014.
[156] M. Casado, N. Foster, A. Guha, Abstractions for Software-defined Networks, Stanford University, 2013.
[157] Sonus, "SDN Orchestration Explained," 2015.
[158] "What's Software-defined Networking (SDN)?", SDxCentral. [Online]. Available: https://www.sdxcentral.com/sdn/definitions/what-the-definition-of-software-defined-networking-sdn/. [Accessed: 18- Aug- 2016]."
[159] B. Lantz, B. Heller, N. McKeown, A network in a laptop: rapid prototyping for software-defined networks, Work. Hot Top. Netw. (2010).
[160] C. Shaohua, T. Mengzhu, Z. Lv, D. Jiang, A study on application-towards bandwidth guarantee based on SDN, in: Proceedings of the IEEE Globecom Workshops, 2016, pp. 1–6.
[161] R.L.S. de Oliveira, C.M. Schweitzer, A.A. Shinoda, L.R. Prete, Using Mininet for emulation and prototyping software-defined networks, in: Proceedings of the IEEE Conference Publications, 2014.
[162] V. Antonenko, R. Smelyanskiy, Global network modelling based on Mininet approach, in: Proceedings of the Second ACM SIGCOMM Workshop Hot Topics in Software Defined Networks, 2013.
[163] J. Teixeira, G. Antichi, D. Adami, A. Del Chiaro, S. Giordano, and A. Santos, "Datacenter in a Box: Test Your SDN Cloud-datacenter Controller at Home."
[164] S. Wang, "Comparison of SDN OpenFlow network simulator and emulators : EstiNet vs. Mininet," in Proceedings of the IEEE Symposium on Computers and Communications
[165] S.-Y. Wang, C.-L. Chou, C.-M. Yang, EstiNet OpenFlow network simulator and emulator, IEEE J. Mag. 51 (9) (2013) 110–117.
[166] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, S. Shenker, NOX: towards an operating system for networks, SIGCOMM Comput. Commun. Rev. 38 (3) (2008) 105–110.
[167] S. Shin, Y. Song, T. Lee, et al., Rosemary: a robust, secure, and high-performance network operating system, in: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, ACM, 2014, pp. 78–89.
[168] P. Kampanakis, H. Perros, T. Beyene, SDN-based solutions for moving target defense network protection, in: Proceedings of the IEEE 15th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), IEEE, 2014, pp. 1–6.
[169] M. Reitblatt, N. Foster, J. Rexford, C. Schlesinger, D. Walker, Abstractions for network update, ACM SIGCOMM Comput. Commun. Rev. 42 (2012) (2012) 1–11.
[170] K. Nisar, S. Hassan, … M.M. Kadhum, A novel voice priority queue (VPQ) scheduler and algorithm for VoIP over WLAN network, J. Telecommun. Electron. Comput. Eng. 3 (2) (2011) 79–94 UTeM, Melaka, Malaysia.
[171] K. Nisar, A.M. Said, H. Hasbullah, An efficient voice priority queue (VPQ) scheduler architectures and algorithm for VoIP over WLAN networks, Int. J. Comput. Sci. Lett. ISSR J. 2 (2) (2010) 1–13.
[172] A.A.A. Ibrahim, K. Nisar, Y.K. Hzou, I. Welch, Review and analyzing RFID technology tags and applications, in: Proceedings of the 13th IEEE International Conference on Application of Information and Communication Technologies (AICT), Baku, Azerbaijan, 2019, pp. 1–4. https://ieeexplore.ieee.org/abstract/document/8981779.
[173] A. Varga, R. Hornig, An overview of the OMNeT++ simulation environment, in: Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems, 2008, pp. 1–10.
[174] D. Klein, M. Jarschel, An OpenFlow extension for the OMNeT++ INET framework, in: Proceedings of the Sixth International Conference on Simulation Tools and Techniques, 2013.
[175] K. Nisar, A.A.A. Ibrahim, Y.J. Park, Y.K. Hzou, S.K. Memon, N. Naz, I. Welch, Indoor roaming activity detection and analysis of elderly people using RFID technology, in: Proceedings of the IEEE International Conference on Artificial Intelligence and DATA Sciences (IEEE AiDAS), Perak, Malaysia, 2019, pp. 174–179, doi:10.1109/AiDAS47888.2019.8970780. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8970780.
[176] L.X. Wee, Z. Yan, Y.J. Park, Y.-B. Leau, K. Nisar, A.A.A. Ibrahim, ROM-P: route optimization management of producer mobility in information-centric networking", Intelligent Transport Systems, From Research and Development to the Market Uptake, INTSYS, 2018 Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 267. Springer, Cham, pp 81-91, 22 February 2019 (Scopus Index), doi:10.1007/978-3-030-14757-0_7.
[177] "The California OpenFlow Testbed Network", Cenic.org. [Online]. Available: http://cenic.org/network/cotn. [Accessed: 18- Aug- 2019]."
[178] C.B. Tan, M.H.A. Hijazi, Y. Lim, Y.-J. Park, K. Nisar, Performance efficiency analysis on Slepian-Wolf based proof of retrievability and its variants for cloud storage, in: Proceedings of the IEEE International Conference on Artificial Intelligence in Engineering and Technology, IICAIET, Kota Kinabalu, Malaysia, 2018, pp. 48–54, doi:10.1109/IICAIET.2018.8638472. ISBN: 978-153867813-8.
[179] "OFELIA", Fp7-ofelia.eu, 2016. [Online]. Available: http://www.fp7-ofelia.eu/. [Accessed: 18- Aug- 2016]."
[180] S. Salsano, N. Blefari-Melazzi, A. Detti, G. Morabito, L. Veltri, Information centric networking over SDN and OpenFlow: architectural aspects and experiments on the OFELIA testbed, Comput. Netw. 57 (16) (2013) 3207–3221.
[181] "Orbit", Orbit-lab.org. [Online]. Available: http://www.orbit-lab.org/. [Accessed: 18- Aug- 2016]."
[182] Y. Kanaumi, S.I. Saito, E. Kawai, S. Ishii, K. Kobayashi, S. Shimojo, RISE: a wide-area hybrid OpenFlow network testbed, IEICE Trans. Commun. (1) (2013) 108–118 Vols. E96-B.
[183] A. Nguyen, E. K. Cetinkaya, and J. P. G. Sterbenz, "Introduction to Network Simulation With NS-3," 2016.
[184] F. Gillani., E. Al-Shaer, and Q. Duan, "In-design Resilient SDN Control Plane and Elastic Forwarding Against Aggressive DDoS Attacks," MTD '18, pp. 80-89, 2018, Toronto, ON, Canada.
[185] S. Achleitner., T. Porta, T. Jaeger, and P. McDaniel, "Adversarial Network Forensics in Software Defined Networking," SOSR'17, pp. 8-20, Santa Clara, CA
[186] H. Kang., T. Porta., S. Shin., and V. Yegneswaran., S. Ghosh., and P. Porras., "AEGIS: An Automated Permission Generation and Verification System for SDNs," SecSoN'18, pp. 20-26, 2018, Budapest, Hungary
[187] E. Marin., N. Bucciol., and M. Conti., "An In-depth Look Into SDN Topology Discovery Mechanisms: Novel Attacks and Practical Countermeasures," CCS '19, pp. 1101-1114, 2019, London, United Kingdom.

[188] T. Das, V. Sridharan, M. Gurusamy, A survey on controller placement in SDN, IEEE Commun. Surv. Tutor. 22 (1) (2020) 472–503 Firstquarter, doi:10.1109/COMST.2019.2935453.
[189] M. Alsaeedi, M.M. Mohamad, A.A. Al-Roubaiey, Toward adaptive and scalable OpenFlow-SDN flow control: a survey, IEEE Access 7 (2019) 107346–107379, doi:10.1109/ACCESS.2019.2932422.
[190] M.H. Ohn, S. Yusof, M.G. Lansing, B. Ravindran, K. Nisar, I. Mchucha, Z. Iswandono, N.P. Luen, K.M. Ohn, Gamified online active learning theory, in: Proceedings of the IEEE International Conference on Artificial Intelligence in Engineering and Technology, IICAIET 2018, Kota Kinabalu, Malaysia, 2018, pp. 122–125, doi:10.1109/IICAIET.2018.8638463. ISBN: 978-153867813-8.
[191] A.H. Sodhro, S. Pirbhulal, G.H. Sodhro, A. Gurtov, M. Muzammal, Z. Luo", A joint transmission power control and duty-cycle approach for smart healthcare system, IEEE Sens. J. 19 (19) (2018) 8479–8486.
[192] S. Pirbhulal, W. Wu, K. Muhammad, I. Mehmood, G. Li, V. Albuquerque, Mobility enabled security for optimizing IoT based intelligent applications, IEEE Netw. 34 (2) (2020) 72–77.
[193] A.H. Sodhro, Z. Luo, G.H. Sodhro, M. Muzamal, J.J. Rodrigues, V.H.C. de Albuquerque, Artificial Intelligence based QoS optimization for multimedia communication in IoV systems, Future Gener. Comput. Syst. 95 (2019) 667–680.
[194] A. Shirmarz, A. Ghaffari, Performance issues and solutions in SDN-based data center: a survey, J. Supercomput. (2020), doi:10.1007/s11227-020-03180-7.
[195] D. Kreutz, F.M. Ramos, P.E. Verissimo, C.E. Rothenberg, S. Azodolmolky, S. Uhlig, Software-defined networking: a comprehensive survey, Proc. IEEE 103 (1) (2015) 14–76.
[196] M Karakus, A Durresi, Quality of Service (QoS) in software defined networking (SDN): a survey, J. Netw. Comput. Appl. 80 (2017) 200–218.
[197] W. Ahmad, S.K. Memon, K. Nisar, G. Singh, Learning the required entrepreneurial best practices using data mining algorithms, in: Proceedings of the Fifth International Conference on Computational Science and Technology (ICCST), 29-30, Kota Kinabalu, Malaysia, August 2018, pp. 461–470, doi:10.1007/978-981-13-2622-6_45. the Lecture Notes in Electrical Engineering Springer book series (LNEE, volume 481ISBN: 978-981132621-9.
[198] H. Zhang, Z. Cai, Q. Liu, Q. Xiao, Y. Li, C.F. Cheang, A survey on security-aware measurement in SDN, Secur. Commun. Netw. (2018).
[199] S.K. Memon, K. Nisar, W. Ahmad, Performance evaluation of densely deployed WLANs using directional and omni-directional antennas, in: Proceedings of the Fifth International Conference on Computational Science and Technology (ICCST), Kota Kinabalu, Malaysia, 2018, pp. 369–378, doi:10.1007/978-981-13-2622-6_36. the Lecture Notes in Electrical Engineering Springer book series (LNEE, volume 481)ISBN: 978-981132621-9.
[200] K. Nisar, A.A.A. Ibrahim, A smart home model using android application, in: Proceedings of the 7th International Conference on Kansei Engineering and Emotion Research (KEER), 3-10, Kuching, Malaysia, 2018, pp. 19–22, doi:10.1007/978-981-10-8612-0_1. Springer Book Series (AISC, Vol. 739).
[201] M. Ndiaye, G.P. Hancke, A.M. Abu-Mahfouz, Software defined networking for improved wireless sensor network management: a survey, Sensors 17 (2017) 1031.
[202] K. Nisar, A.A.A. Ibrahim, A Model new for smart home technologies knee monitor and walking analyser, in: Proceedings of the 7th International Conference on Kansei Engineering and Emotion Research (KEER), Kuching, Malaysia, 2018, pp. 501–509, doi:10.1007/978-981-10-8612-0_52. Springer Book Series (AISC, Vol. 739)ISBN: 978-981108611-3.
[203] E. Coronado, R. Riggio, J. Villalon, A. Garrido, Joint mobility management and multicast rate adaptation in software-defined enterprise WLANs, IEEE Trans. Netw. Serv. Manag. 15 (2) (2018) 625–637.
[204] S. Harada, Z. Yan, Y.-J. Park, K. Nisar, A.A.A. Ibrahim, Data aggregation in named data networking, in: Proceedings of the IEEE Region 10 Conference (TENCON), Penang, Malaysia, 2017, pp. 1839–1842, doi:10.1109/TENCON.2017.8228157.
[205] R. Wang, S. Mangiante, A. Davy, L. Shi, B. Jennings, QoS-aware multipathing in datacenters using effective bandwidth estimation and SDN, in: Proceedings of the 12th International Conference on Network and Service Management (CNSM), 2017, pp. 342–347.
[206] A.K. Rangisetti, T.V. Pasca, B.R. Tamma, QoS aware load balance in software defined LTE networks, Comput. Commun. 97 (2017) 52–71.
[207] Z. Li, Z. Deng, T. Zhang, Research on the optimal task scheduling algorithm based on SDNarchitecture, Int. J. Opt. 9 (10) (2017) 221–230.
[208] D.B. Rawat, S.R. Reddy, Software defined networking architecture, security and energy efficiency: a survey, IEEE Commun. Surv. Tutor. 19 (1) (2017) 325–346.
[209] A.C. Baktir, A. Ozgovde, C. Ersoy, How can edge computing benefit from software-defined networking: a survey, use cases, and future directions, IEEE Commun. Surv. Tutor. 19 (4) (2017) 2359–2391.
[210] Z. Yan, G.G. Gang, Y.-J. Park, H. Nakazato, K. Nisar, A.A.A. Ibrahim, On-demand DTN Communications in heterogeneous access networks based on NDN, in: Proceedings of the 85th IEEE Vehicular Technology Conference (VTC Spring), Sydney, NSW, Australia, 2017, pp. 1–2, doi:10.1109/VTCSpring.2017.8108641.
[211] F.De Turck, P. Chemouil, R. Boutaba, M. Yu, C.E. Rothenberg, K. Shiomoto, Guest editors' introduction: special issue on management of cloud services, IEEE Trans. Netw. Serv. Manag. 13 (3) (2016) 362–365, doi:10.1109/TWC.2011.09.001-tnsm1403-editorial.
[212] E.E. Hilmi, C. Seyhan, T.A. Murat, An optimization framework for QoS-enabled adaptive video streaming over OpenFlow networks, IEEE Trans. Multimedia 15 (3) (2013) 710–715.
[213] K. Nisar, A.A.A. Ibrahim, L. Wu, A. Adamov, J. Deen, Smart home for elderly living using wireless sensor networks and an android application, in: Proceedings of the 10th IEEE International Conference on Application of Information and Communication Technologies (AICT), Azerbaijan, Baku, 2016, pp. 1–8, doi:10.1109/ICAICT.2016.7991655. ISBN: 978-150901840-6.
[214] Heller, B., Seetharaman, S., Mahadevan, P., Yiakoumis, Y., Sharma, P., Banerjee, S., McKeown, N., ElasticTree: Saving Energy in Data Center Networks. In: NSDI, 2010
[215] D. Staessens, S. Sharma, D. Colle, M. Pickavet, P. Demeester, Software defined networking: meeting carrier grade requirements, in: Proceedings of the 18th IEEE Workshop on Local Metropolitan Area Networks (LANMAN), 2011, pp. 1–6.
[216] X. Li, S. Harada, Z. Yan, Y.-J. Park, W. Kameyama, K. Nisar, A.A.A. Ibrahim, Performance analysis of proxy based producer mobility in named data networking, in: Proceedings of the IEICE Society Conference, Sendai, Japan, 2015 The Institute of Electronics, Information and Communication Engineers (IEICE), BS-6-8,pp. S-30-S-31B., Vereecken, W., Colle, D., Pickavet, M., Demeester, P., Multilayer traffic engineering for energy efficiency. Photonic Netw. Commun. 21 (2), 127–14, 2011.
[217] M. Canini, D. Venzano, P. Perešíni, D. Kostić, J. Rexford, Identifying and using energy-critical paths, in: Proceedings of the Seventh Conference on Emerging Networking Experiments and Technologies, CoNEXT. ACM, 12, 2011, p. 18. 1–18.
[218] Y. Hu, T. Luo, N.C. Beaulieu, W. Wang, An initial load-based green software defined network, Appl. Sci. 7 (5) (2017) 459.
[219] F. Giroire, F. Havet, J. Moulierac, Compressing two-dimensional routing tables with order, Electron. Notes Discrete Math. 52 (2016) 351–358.
[220] K. Nisar, M.H.A. Hijazi, A.L Ibrahim, "A new model of application response time for VoIP over WLAN and fixed WiMAX, in: Proceedings of the Second International Conference on Computing Technology and Information Management (ICCTIM2015), Johor, Malaysia, Universiti Tun Hussein Onn Malaysia, 2015, pp. 174–179, doi:10.1109/ICCTIM.2015.7224613.
[221] T. Das, V. Sridharan, M. Gurusamy, A survey on controller placement in SDN, IEEE Commun. Surv. Tutor. 22 (1) (2020) 472–503 Firstquarter, doi:10.1109/COMST.2019.2935453.
[222] M. Alsaeedi, M.M. Mohamad, A.A. Al-Roubaiey, Toward adaptive and scalable OpenFlow-SDN flow control: a survey, IEEE Access 7 (2019) 107346–107379, doi:10.1109/ACCESS.2019.2932422.
[223] X. Jia, Y. Jiang, Z. Guo, G. Shen, L. Wang, Intelligent path control for energy-saving in hybrid SDN networks, Comput. Netw. 131 (2018) 65–76.
[224] P. Sun, J. Li, Z. Guo, Yang Xu, J. Lan, Y. Hu, SINET: enabling scalable network routing with deep reinforcement learning on partial nodes, in: Proceedings of the ACM SIGCOMM 2019 Conference Posters and Demos, 2019, pp. 88–89, doi:10.1145/3342280.3342317.
[225] Z. Guo, S. Zhang, W. Feng, W. Wua, J. Lan, Exploring the role of paths for dynamic switch assignment in software-defined networks, Future Gener. Comput. Syst. 107 (2020) 238–246.

[226] C. Yu, J. Lan, Z. Guo, Y. Hu, DROM: optimizing the routing in software-defined networks with deep reinforcement learning, IEEE Access 6 (2018) 64533–64539, doi:10.1109/ACCESS.2018.2877686.
[227] Z. Guo, S. Hui, Y. Xu, H.J. Chao, Dynamic flow scheduling for power-efficient data center networks, in: Proceedings of the IEEE/ACM 24th International Symposium on Quality of Service (IWQoS), Beijing, 2016, pp. 1–10, doi:10.1109/IWQoS.2016.7590399.
[228] X. Jia, Y. Jiang, Z. Guo, Z. Wu, Reducing and balancing flow table entries in software-defined networks, in: Proceedings of the IEEE 41st Conference on Local Computer Networks (LCN), Dubai, 2016, pp. 575–578, doi:10.1109/LCN.2016.96.
[229] X. Jiaa, Q. Li, Y. Jianga, Z. Guoc, J. Suna, A low overhead flow-holding algorithm in software-defined networks, Comput. Netw. 124 (2017) 170–180.
[230] T. Hu, P. Yi, Z. Guo, J. Lan, J. Zhang, Bidirectional matching strategy for multi-controller deployment in distributed software defined networking, IEEE Access 6 (2018) 14946–14953, doi:10.1109/ACCESS.2018.2798665.
[231] J. Ai, Z. Guo, H. Chen, G. Cheng, Improving the routing security in software-defined networks, IEEE Commun. Lett. 23 (5) (2019) 838–841, doi:10.1109/LCOMM.2019.2901486.
[232] T. Hu, P. Yi, Z. Guo, Julong Lan, Y. Hu, Dynamic slave controller assignment for enhancing control plane robustness in software-defined networks, Future Gener. Comput. Syst. 95 (2019) 681–693.
[233] T. Wang, Z. Guo, H. Chen, W. Liu, BWManager: mitigating denial of service attacks in software-defined networks through bandwidth prediction, IEEE Trans. Netw. Serv. Manag. 15 (4) (2018) 1235–1248, doi:10.1109/TNSM.2018.2873639.
[234] Z. Guo, W. Feng, S. Liu, W. Jiang, Y. Xu, Z. Zhang, RetroFlow: maintaining control resiliency and flow programmability for software-defined WANs, in: Proceedings of the IEEE/ACM 27th International Symposium on Quality of Service (IWQoS), Phoenix, AZ, USA, 2019, pp. 1–10, doi:10.1145/3326285.3329036.
[235] J. Ai, H. Chen, Z. Guo, G. Cheng, T. Baker, Improving resiliency of software-defined networks with network coding-based multipath routing, in: Proceedings of the IEEE Symposium on Computers and Communications (ISCC), Barcelona, Spain, 2019, pp. 1–6, doi:10.1109/ISCC47284.2019.8969591.
[236] T. Hu, Z. Guo, J. Zhang, J. Lan, Adaptive slave controller assignment for fault-tolerant control plane in software-defined networking, in: Proceedings of the IEEE International Conference on Communications (ICC), Kansas City, MO, 2018, pp. 1–6, doi:10.1109/ICC.2018.8422598.