International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 9, No.4 (July-2020)

http://dx.doi.org/10.12785/ijcds/090407

A Survey on Security Enhancements in the Internet of Things Using Software-Defined-Networking (SDN)

Omerah Yousuf[1] and Roohie Naaz Mir[1]

[1] *Department of Computer Science and Engineering, NIT Srinagar, Hazratbal – 190006, Jammu & Kashmir, India.*

Received 13 Jun. 2019, Revised 15 Jun. 2020, Accepted 24 Jun. 2020, Published 1 Jul. 2020

Abstract: Software Defined Networking (SDN) is an innovative, advanced, challenging and promising network technology that includes various features for ensuring security in the field of the internet of things (IoT). Since security is a major concern nowadays, numerous researchers believe that by decoupling the control plane from the data plane can greatly improve the security issue in different ways and provides various research opportunities. In this paper, we analyse the research works done by various researchers towards securing the IoT by SDN. The main motivation behind this survey is to identify and summarize the various security enhancements provided by the deployment of SDN in IoT. This paper reviews the various challenges in the development of SDN, current state of art, its applications and the architecture of IoT based on SDN. Finally, the paper highlights open challenges and the various future research directions in the field of IoT security provided by SDN.

Keywords: Software Defined Networking, Internet of Things, SDN based Architecture, Security

1. INTRODUCTION

The cyber-world of things (IoT) represents the current and future state of the Internet. IoT researchers contend that by 2020, IoT will develop altogether to cover all the objects in an environment engendering what they call the cyber world of everything (IoE). IoT has accomplished a worldwide success and is discovering its application in each field. IoT applications have experienced rapid development in recent years due to Radio Frequency Identification (RFID) and Wireless Sensor Networks (WSN) methods [1], [2]. The IoT has excellent potential for transforming today's way of living however security is the main problem in realizing entirely intelligent frameworks. Shortly, if security issues such as privacy, confidentiality, authentication, access control, end-to-end safety, trust management, worldwide policies, and norms are fully addressed, we can witness the transformation of everything by IoT. IoT-predicated frameworks need to deal with a plethora of information and the issue of giving security in the IoT is representing a significant test to the researchers in the present time. An assaulter might be intrigued in purloining the sensitive information, e.g. accounts, passwords, credit card numbers, patient information, etc. or may compromise the IoT components. IoT systems are at higher security risk due to the following reasons [3]:

- Lack of well-defined perimeters and user mobility.
- Due to presence of various heterogeneous devices, communication medium, and protocols.
- Work independently and can control other IoT devices.
- Incorporate "things" unfamiliar to the Internet.
- IoT systems might be physically intruded and controlled by different users.
- Permissions granted only to specific users due to a huge number of devices which is however not possible with smartphone applications, which require permissions for installations and interactions.

The idea of SDN has been advancing since 1996 and is developing remarkably. Various companies intend to use SDN for future network and are one of the most effective platforms for providing security enhancements.

E-mail:omerahyousuf@nitsri.net, naaz310@nitsri.net

SDN is aimed to conceal all involution in traditional system architectures by diverting all the controls and management operations from the basic contrivances and setting them up in a middleware layer, a software layer.

Open Networking Foundation (ONF) [4], [5] is an association devoted to the advancement and grasping of SDN through open measures improvement. As indicated by ONF the SDN is characterized as a system design where the control and data planes are decoupled, network astuteness and state are centralized, and the primary network infrastructure is separated from the applications.

SDN is proximately cognate to network function virtualization (NVF) [6], albeit both SDN and NFV aim at incrementing the limberness and flexibility of networks and decrementing intricacy and cost, they utilize different methods. In SDN, control planes are disunited from data planes, while as in NFV, network contrivances are superseded by software. Even though SDN and NFV are different concepts, but both can be advantageous to each other. The comparison between the SDN and NFV has been presented in Table I as under [7]:

TABLE I. COMPARISON BETWEEN SDN AND NFV

Features	***SDN***	***NFV***
Basic Idea	Separates data and control plane	Transfers network function to generic servers
Area of operation	Campus, data centers	Service provider network
Network servers	Servers and switches	Servers and switches
Initial Application Target	Cloud Orchestration and Networking	Routers, firewalls, WAN.
Protocol	OpenFlow	None
Supporting Organization	Open Networking Foundation (ONF)	ETSI NFV Working Group

A. Research Problem

The Internet of Things (IoT) paradigm has emerged as one of the most significant and groundbreaking technologies in recent times. It can make life easier in many respects in the day to day life. Each technology comes at the cost of something and IoT being no exception. One of the most fundamental challenges in the IoT is to provide the security of the data as well as the entities involved in exchanging the data, which at times is critical in nature. Security remains a top priority as per many of the surveys done in the field of IoT by the eminent researchers. A lot of ideas, architectures, and methods have been put forward to address the challenging issue of security in IoT. Among these architectures, Software-defined networking (SDN) is a novel network paradigm which has gained significant traction by many researchers in order to overcome the issue of IoT security. Provided the capabilities of SDN are efficiently exploited, the security challenges of IoT can be mitigated.

SDN based network security solutions are being broadly developed in the present era. The innovation empowers designers to legitimately program, control, and manages network resources through the SDN controller. The coherent centralization of system insight presents energizing difficulties and chances to improve security in such systems, including better approaches to prevent, detect and respond to threats, as well as new security services and applications are built using the abilities of SDN.

In this work, we performed an extensive review of security enhancements provided by SDN in the development of the Internet of Things (IoT). To the best of our knowledge, this work is quite novel and detailed in its methodology contrasted with different articles distributed on this subject. We identified the various challenges in the development of SDN and its applications in various domains. The paper highlights the various features of SDN for network security in IoT and the architecture of IoT based on SDN.

Most importantly, the paper discusses the need of SDN for IoT and explains how SDN is the most important solution to tackle the challenge of security in the field of IoT. The paper aims at classifying the various categories in order to capture the various network security enhancements based on the deployment of SDN in the field of IoT security.

Rest of the paper is organized as follows: Section 2 discusses the various challenges in the development of SDN. Section 3 discusses the need of SDN in IoT. Section 4 presents the architecture of IoT based on SDN and various applications of SDN are discussed in section 5. Various features of SDN for security in IoT are highlighted in section 6 and a detailed discussion of security enhancements provided by SDN in IoT in Section 7. An analysis of various SDN controllers is presented in Section 8. Finally, we conclude and provide future enhancement in section 9. References are listed in the end.

2. VARIOUS CHALLENGES IN SOFTWARE DEFINED NETWORK (SDN) AND ITS EXISTING SOLUTIONS

The various challenges and some of their existing solutions in the development of SDN are discussed as follows [8],[9],[10]:

Reliability

SDN is designed to make the networks more trustworthy. Due to the centralized architecture of SDN the whole network may breakdown as a result of single point of failure. SDN decouples the control and forwarding planes, thereby introducing the reliability issues in SDN. In order to increase the network reliability using SDN, the controller must be capable of redirecting the traffic through multiple paths from source to destination. Additionally, network reliability can be improved by replication of controller with the end goal that in the event that if one controller fails, another controller can assume control over the network.

The authors in [11] described the architecture of the SiBF (Switching with in-packet Bloom filters) information centre which eradicates the idea of a centralized controller and introduced an army of rack directors (RMs), one for every rack, as controllers. However, if the master controller fails, other stand-by controller (RM) will handle flow applications until the master controller becomes functional again. In the event of a switch failure, SiBF will install new mappings for each active entry in the ToR switches and will route the packets to their specified destinations on the alternative paths determined by the back-up entries.

Another solution based on dynamic load-balancing multi-pathing strategy was defined by the authors in [12] and used distributed algorithms in case of controller failure. In the event of heavy traffic and unbalance in load, the switches are altered according to the algorithm on the desired routes.

Scalability

One of the important problems faced by the deployment of SDN is scalability that needs more attention. The three main controller scalability challenges are: (i) Latency-between a single controller and multiple nodes. (ii) Communication between various controllers. (iii) Maintenance of database size and operation. Another challenge to network scalability arises due to rise in the number of switches and bandwidth leading to overall controller overhead.

The authors in [13] suggested a fault-tolerant SDN architecture called' CORONET' to tackle the challenge of SDN scalability, which quickly recovers from faults and scale to huge network sizes. It utilizes several alternative routes in the network, operates with arbitrary networks and utilizes single controller plane to transmit choices.

The authors in [14] outlined an extensible SDN controller scheme called' McNettle' running on multicore shared memory servers and maintaining a straightforward, natural programming model for controller designers. Using a single controller with 46 cores, it can serve almost 5000 switches, accomplishing a throughput of more than 14 million flows for every second.

Security

IoT-based systems collect a huge amount of data from a heterogeneous environment having different security requirements which makes it difficult to ensure proper security to it. Brought together design of SDN makes it increasingly susceptible against different types of threats and therefore, makes it hard to guarantee the general security of the system. Therefore, it is becoming the need of an hour to append the appropriate security mechanisms to SDN in order to secure the communications.

The authors in [15] suggested FortNOX, a software extension providing the NOX OpenFlow controller with roles-based approval and implementation of safety constraints. It includes several elements required to enable safety applications in OF networks, including role-based authorization, reduction of rules, conflict assessment, policy synchronization and translation of the Security directive. FortNOX is a significant first step in enhancing network security, but lot of research remains to be done in developing a richer set of applications covering a broad range of security services.

Interoperability

Interoperability is considered to be a key to IoT success. Internet Engineering Task Force's Path Computation Element [16] is considered as a solution for the transition of a traditional network to SDN guaranteeing the interoperability with the current frameworks. However, standardization of northbound interface and lack of eastbound or westbound interfaces are still a main challenge for SDN.

The authors in [17] proposed a lightweight portability layer for SDN OS known as 'NOSIX' which is the initial move towards accomplishing flexibility and better performance across a broad range of switches. The primary thought behind this framework is to separate the applications prerequisites from the switch details and implementations. It doesn't endeavor to separate the application totally from the switch. Rather, it gives a switch that empowers applications to determine subtleties that they care about and overlook ones that are not significant.

Fault Tolerance

One of the most important challenges faced by SDN is fault tolerance caused due to the heterogeneous nature of IoT and is required at both the planes i.e., control plane (detecting link failure) and data plane (recovering from link failure). To address this issue, various fault tolerant SDN controllers has been designed by the researchers. FatTire [18] developed a new programming language for writing fault-tolerant programs in SDN based on regular expressions. Nevertheless, there are as yet many open issues associated with it, specifically the trade-off between consistency and performance in fault-tolerant SDN environment that need to be taken into account.

For tiny to medium-sized networks, the authors in [19] suggested a practical fault-tolerant SDN controller 'Smartlight.' The presented architecture is a recapitulated shared database that stores all information about the network status and is managed by one controller while the other controllers are used as safeguards in the event of failure. To ensure that the system isn't influenced by controller failure, we need to solve two issues-(i) Each controller must have the same standpoint across the whole network (ii) Choosing alternative controller to act as the main controller in the chance of failure. The suggested scheme therefore introduced the shared database to provide a worldwide perspective among all controllers and Zookeeper (coordination service) in order to handle these issues.

Congestion

A large number of IoT devices are connected to the Internet. Congestion of network links between the control plane and data plane may increase rapidly resulting in the reduction of quality of service (measurement of overall performance). There are also the chances that the SDN controller itself may get congested due to heavy load on the network and centralized nature of SDN. Researchers have proposed different congestion control mechanisms to reduce the congestion of the network by adjusting sending rate. It is possible to further make improvements in such mechanisms to increase the overall performance of the network.

The authors in [20] suggested a congestion control system for SDN-based TCP (SDTCP) by changing the TCP receive ACK packet window after a congestion message to the controller was triggered by OpenFlow switch. In OpenFlow-switch (OF-switch), a straightforward queue management system was created to cause a notification of congestion to the controller when the queue occupancy exceeds the K limit. Upon receiving this signal, the controller will pick a long-lived flow and press a change button to automatically change the receiving window of the ACK packet at OF-switch.

3. Need of Software Defined Networks for IoT

IoT comprises of many heterogeneous contrivances that are utilizing variants of protocols. Each protocol follows diverse access mechanisms and security measures. But a coalesced security mechanism is still not in place yet. Variegated traditional security mechanisms like Firewalls, Intrusion Detection, and Obviation Systems (IDPS), etc. were deployed to forfend the IoT contrivances from external attacks and such mechanisms are not adequate to provide security to the next generation Internet. The borderless architecture of SDN gives raise to secondary problems associated with the network access control and software authentication.

Researchers have identified sundry security challenges in IoT- network security, identity management, privacy, heterogeneity, scalability, etc. Among these, the heterogeneity and scalability are considered the most consequential challenges in the field of security in IoT. Since there are billions of contrivances that want to communicate in the environment of IoT and a substantial amount of data is being engendered by these contrivances poses the main reason behind why scalability and heterogeneity are the major challenges in IoT security. Consequently, to address such challenges SDN can act as a very utilizable implement.

Software Defined Networking (SDN) is the nascent networking paradigm for empowering advancement in networking research, improvement and offers numerous chances to rampart the system in a progressively secure and effective manner. By applying SDN configuration and management can be reduced greatly. Enterprises wide acknowledgment for SDN shows that SDN can build up a more tightly association inside the environment of IoT [4].

4. Architecture of IoT Based on SDN

Generally, the architecture of IoT based on SDN is divided into three components as shown in Figure1. The various components are discussed as follows [21]:

IoT Agent

- Responsible for analysing, sensing and collecting
- IoT devices need to be registered

IoT Controller

- Responsible for taking necessary actions
- Upon connection, a request will build forwarding rules
- Receiving destination objects and finds in the network as IoT agents are already registered with the IoT controller

SDN Controller

- Establishes the network path
- Gathers topology information from both IoT and SDN

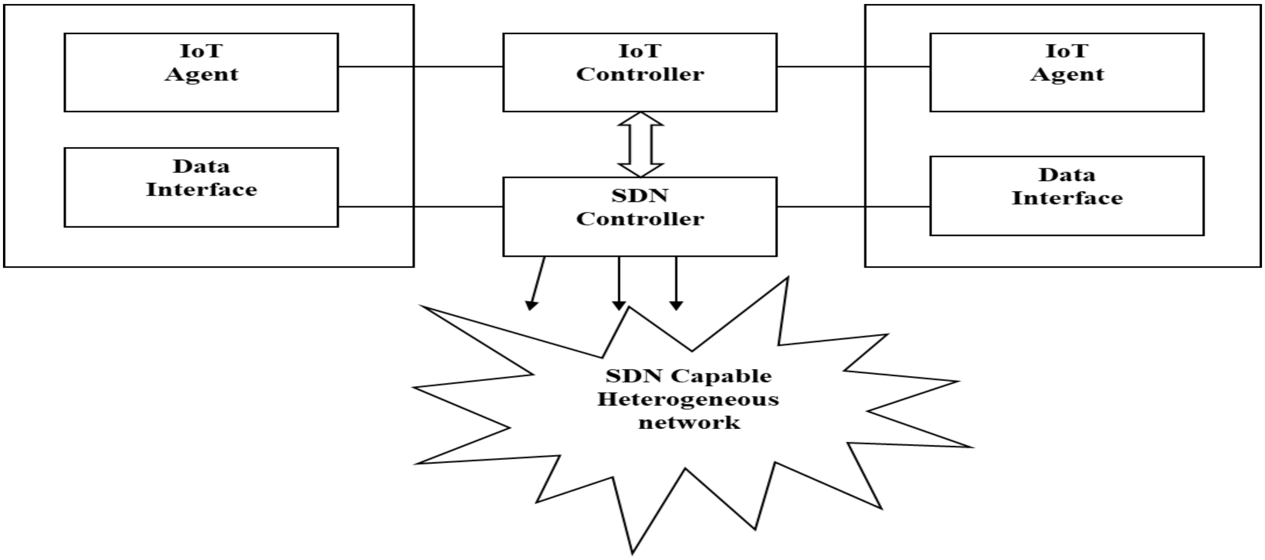

Figure 1. IoT Based on SDN

5. SDN APPLICATIONS

There is a wide range of applications of SDN provided in the different network environment as described below in Table II [22], [23] :

TABLE II. SDN APPLICATIONS

Area of Application	*Benefits of SDN*	*Examples*
Enterprise Networks	• Enforce network policies, monitoring network activity, and network performance • Simplify the network and provide unified control	• Ethane [24] • OpenFlow based updates [25]
Data Centers	• Energy Efficiency • Customized routing and traffic engineering • Increases fault tolerance, scalability and cost efficiency	• ElasticTree [26] • Honeyguide [27]
Security	• Improved Security of network systems • Better policies • Increased tolerance to various types of attacks	• Pedigree [28]
Virtualization	• Possible to create advanced virtual networks	• Flowvisor [29] [30]
Infrastructure-based wireless access networks	• Improved performance on handover events • Improved mobility management and load balancing	• OpenFlow Wireless [31] • OpenRoads [32]
Optical Networks	• Technology-agnostic unified control • Improving network control and management • Expanding new services by introducing virtualization	• OpenFlow based Wavelength path [33] • SDON [34]
Home and Small Business	• Better network management • Improved security	• Data Recorder [35] • Outsourcing [36]
Mobile Networks	• Flow-based model offers better tools • Enables openness, innovation, programmability	• MobileFlow [37]
Multimedia	• Optimization of multimedia management tasks • Increased Quality of experience	• Optimized path assignment [38] • QOE Fairness Framework [39]
Reliability and Recovery	• Global vision enables the customizing of recovery algorithms • Controller helps to find an alternative path if the failure occurs • Reliability can be increased by backup controllers	• Fast Failure Recovery [40] • CPRecovery [41]

MANET	• QoS Management • High flexibility • Offload cellular network • Recover from disruptions of computation-intensive MANETs.	• MANET oriented SDN [42]
VANET	• Self -adapting to dynamically changing topology • Better performance, storage , reduced performance cost • Satisfy the safety and non-safety requirements	• SDVN [43]
UWSN	• Improve flexibility • Reduce development risks • Solves redundancy of repeated deployment	• SoftWater [44] • AUV system [45]

6. Features of Software Defined Networks for Security in IoT

The various advantages of SDN features that can strengthen the network security in IoT are discussed as under [46], [47] :

Dynamic Flow Control

- Network flows controlled more efficiently and effectively by separating the network control and data plane
- Dynamically distinguish malign network flows from beneficent ones
- Examples include:

  (i) FlowVisor [48]- introduced the slicing layer between control and data plane.

  (ii) OpenVirtex [49]-Network virtualization platform.

Centralized Controller and global view of the network

- Enables users to receive all network status information
- Managed by a single centralized controller
- Helps in monitoring the network from security threats.
- Examples include:

  (i) CloudWatcher [50] helps in monitoring security services in cloud networks easily and efficiently

  (ii) FleXam [51] sampling extension for OpenFlow to access packet level information by a controller

Programmable Network

- Network programmability provided through the use of APIs
- Helps in developing security applications easily
- Examples include:

  (i) FRESCO [52] is an application development framework to deploy complex security services

  (ii) Nettle [53] permits programming OpenFlow networks in an exquisite, declarative style.

Simplified Data Plane

- Simplifies the data plane by separating control plane logic
- Security usage by adding new modules
- Examples include:

  (i) AVANTGUARD [54]→Data plan extension consisting of connection migration and initialize triggers

  (ii) OFX [55]→ Improves performance and deployment hurdles in various security applications

  (iii) OpenSDWN [56]→Flexible and fine-grained network management for Wi-Fi networks.

7. Security Enhancements provided by SDN

Abundant research has been done to provide essential network security by an SDN network. Based on the deployment of SDN, researchers have analyzed different categories in order to acquire the potential network security improvements done by SDN. In this section, these categories are discussed and the various solutions are outlined to offer enhanced network security in IoT by taking advantage of various SDN features [25]. Table III summarizes the problem and proposed solutions for network security enhancements using SDN.

A. Network management

SDN is the new paradigm in the network environment which clarifies the separation of data plane and control plane. SDN offers a centralized view and provides better management than traditional networks. Due to the global and flexible nature of SDN, it is possible to integrate new security applications easily [57].

The authors in [58] have designed and implemented, an event-driven network control framework based on SDN, named Procera. It is feasible and introduced a variety of network policies, exigently reduced the complexity of network management in a variety of network settings and for a range of network policies. It follows the SDN and makes all traffic forwarding decisions and updates using a controller which then translates the network policies to forwarding rules. Using OpenFlow protocol [59], the network controller establishes the connection to each OpenFlow switch which is then used to insert, delete or modify the packet forwarding rules. However, the main drawback of this framework is the immanent delay caused due to the communication between the control plane and the data plane and the packets have to wait until suitable forwarding rules are installed in the data plane. Due to the limitation of OpenFlow 1.0.0, the proposed framework can support only allow and drop of packets and is very difficult to support an affluent set of actions.

The authors in [60] have discussed two main challenges that occur during the deployment of a network. First- the deployment of middle-boxes in chokepoints and Second-dynamic updates of the network topology. In this paper, the authors have implemented a prototype of an encryption processing unit [61] which manipulates only the specific traffic that requires to be encrypted while the rest of the traffic is forwarded without passing through the middle-box. However, the main disadvantage of this prototype is the performance issue caused due to software used for encryption and the execution of OpenFlow controller applications on large scale networks.

The authors in [62] suggested a resilience management structure based on policy-controlled management models including the multiple processes to solve problems of traffic shaping, load balancing, detection of anomalies and classification of traffic. By identifying prevalent management patterns and implementing those using OpenFlow apps, the use of management patterns defines a way to orchestrate distinct resilience services. These models are used for particular network problems that can then be reused in various areas of the network. However, the scheme proposed does not support a broader variety of network problems

The authors in [63] provided an SDN-based architecture to secure wired as well as wireless infrastructure. This model is scalable with various SDN domains where a unique sort of controller known as' Border Controllers ' can be used to communicate between distinct domains. Using the grid safety idea, the suggested model can be used to address multiple safety problems in IoT and Ad-Hoc networks. In addition, the suggested design can be further explored to construct and test the architecture in actual settings.

A novel architecture based on SDN paradigm for WSN was suggested by the authors in [64]. The controller is introduced in this architecture at the base station, which helps to address multiple problems in WSN such as energy-saving, mobility of sensor nodes, network management, the accuracy of location and topology discovery. However, the suggested architecture requires to be further explored in order to enhance the efficiency, reliability, safety of WSN and to be tested in realistic situations.

B.Attack Detection and Mitigation

SDN is an emerging networking paradigm that is still under development and allows the integration of network attack detection and mitigation methods. The main advantage of SDN is that it can be used to hide network topology by using Network Address Translation (NAT) and helps in monitoring of network services [57].

The authors in [65] developed the first SDN solution for providing security to the network known as Resonance which provides dynamic access control and was proposed mainly to secure enterprise networks. However, there are chances that a host may get compromised during normal operation. If, however, the controllers receive an alarm about the event, then it can be switched to an authenticated state. However, the researchers are trying to investigate the proposed solution in detail to provide support for more complex access control policies.

The authors in [66] proposed a new framework called AVANT-GUARD to secure interface between the control and data plane by using a technique known as connection migration for protecting a network from the saturation attacks. Also, the proposed framework tried to improve the impartiality of the network during the presence of threats by creating actuating triggers to increase the efficiency of the network. However, the authors didn’t provide any description for the wireless mobile environment and support for interoperability.

The authors in [67] proposed an intrusion detection framework (NICE: NIDS) for virtual network systems where the use of virtual machines (VMs) introduces a wide range of vulnerabilities including shared and storage resources. An OF-based network intrusion detection agent is used to control and evaluate network traffic and the various possible countermeasures have been presented to deal with the different types of vulnerabilities that arise in a network. The suspicious VMs can be isolated, examined and ultimately protected based on a set of possible countermeasures such as traffic isolation, port blocking, etc. To provide better attack detection, it included the attack graph analytical procedures which describe how an attacker can compromise with the network security. However, the authors only investigated the zombie attacks and should improve the detection accuracy by using host-based IDS solutions and need to explore the scalability of the proposed solution.

C. DoS/DDoS Protection

Distributed denial-of-service (DDoS) attacks are the most famous type of attacks caused due to IoT vulnerability and are a real threat to network, digital and cyber-infrastructure. It can be achieved in multiple ways: (i) Server flooding- refers to different ways in which a server can be flooded with multiple requests in order to crash the system (ii) Botnets- a network of infected bots or devices The main objective of a DDoS attack is to bring down the services of a target using multiple sources that are distributed across the network.

The authors in [68] introduced a lightweight strategy for detecting DDoS attack in IoT based on traffic flow characteristics, executed over a NOX based network, where OpenFlow (OF) switches maintain all information about all active Flows utilizing NOX controller [69]. It utilizes Self Organizing Maps, an unsupervised neural network, without the need of human intervention. The main advantage of this method is low overhead and can detect new types of attacks in IoT. However, this methodology doesn't permit the correspondence between various detectors from various system areas and utilized the factual strategies to dissect ports of switches so as to figure out which hosts are propelling assaults.

The authors in [70] proposed another structure for DDoS detection and mitigation in data centers which endeavors to identify assault traffic going from the low rate to high rate and long-lived to short-lived attacks using an SDN engine, executed over the Mininet Emulator which performs better than OpenFlow based QoS approach. However, the proposed system needs to be explored further since it doesn’t handle all types of DDoS attacks.

Using botnet-based OpenFlow APIs, the authors in [71] proposed a DDoS blocking scheme that could effectively block DDoS attacks. It includes protected servers that help to set up the safe communication channel with the DDoS blocking application (DBA-address pool) and redirect alternative server address. However, the proposed system must reduce the dependency between the SDN controller and the protected servers.

D. Authentication, Authorization, and Accounting

The authors in [72] presented an SDN-driven authentication, authorization, and accounting (AAA) system to empower network security for solving the problem of unauthorized access, an authentication and access control mechanism in the network. The main advantage of this system is that it can reuse the pre-existing authentication and account infrastructure, network hardware and guarantee the tight binding between user/device and topology. Here the access control framework is executed by changing the Floodlight controller to enlist and verify switches, validate hosts and link them to switches, to validate users, and to manage flows and flexibility. The suggested system was compatible with the traditional network structure and there was a possibility to incorporate SDN features progressively. However, as the network size increases, the system doesn’t perform well and need to be investigated further for increasing its performance.

The authors in [73] suggested AAA architecture based on certificates that provide a solid, safe, flexible and extensible mechanism for AAA. The system offers loose coupling between the architecture and the experimental network that makes it reusable and offers AAA services via well-defined interfaces. It also serves as a guide for a consistent migration route for current SDN services and can support various processes for authentication.

E. Secure and Scalable Multi-Tenancy

Another important characteristic of SDN is Multi-tenancy (single instance serves multiple tenants). The researchers have designed various solutions that can derive the benefits from the characteristics of SDN to provide security in a multi-tenant environment.

The authors in [74] proposed an Open virtual Network Management and Security (OpenvNMS) to support multi-tenancy in IoT. The main advantage of this system is that it resolved network and Virtual Tenant Network (VTN) scalability issues. It is an autogenous SDN architecture for supporting multi-tenancy and providing flexible isolation between various tenant networks. This method can be used to manage and explore OVSs in an efficient way, solving the issue of scalability in the SDN design infrastructure. Also, it provided high flexibility and performance for packet processing while maintaining the benefits of network control centralization. Here the authors have developed a mechanism at SDN control plane in order to provide substantial isolation at layer. The proposed Open vNMS architecture provided the self-management of virtual nodes, supporting network/VTN scalability and collaboration. However, the proposed system suffers from scalability issues and doesn't perform well under different scenarios.

The authors in [75] proposed Tualatin, a centralized architecture in order to provide network security services for tenant cloud infrastructures. Using SDN and OpenFlow, Tualatin offers solidified security protection for dynamically changing network topologies. Resource utilization can be enhanced by implementing traffic classification with OpenFlow and apply individually security policies to selected traffic flows. It is a hardware-software co-design satisfying variety of security requirements of VPC service model with more efficient resource utilization. However, evaluations have shown that the proposed system could deliver security provisions with agility and have little impact on data plane latency.

The authors in [76] suggested a collaborative network security model to solve network security challenges in multi-tenant data centers with a centralized cooperative system and a profound inspection system. It also built an intelligent packet verdict system for packet inspection to protect against the various kinds of assaults within data centers. However, the authors need to investigate the system further to include the concept of parallelization and detect the intrusion using unsupervised learning methods.

The authors in [77] proposed a network security strategy for SDN-based cloud situations by keeping a centralized database that contains all system/service security-related data. It improves the precision of particular systems by redirecting network traffic to more systems to collect extra data about the systems linked to security. For each cloud service, the suggested strategy describes strategies individually and utilizes forwarding choices to mitigate malicious traffic to and from the cloud setting. However, the writers need to further investigate the strategy of supporting large information sets and comparing the different safety facilities in the cloud.

F. Collect, Detect And Protect

The authors in [78] designed an intrusion detection system (IDS) for embedded mobile devices which can be deployed with no modifications to the devices in order to provide security to the devices. It relies on mathematical based model to detect anomalous traffic in the network. The proposed system addressed the issue of providing support for end user mobility and includes richer set of actions due to which this system can be used outside the enterprise networking environments which was not possible using the traditional IDS. Various rapid responses and network reconfigurations are available to handle the various types of attacks in real time.

The authors in [79] merged OpenFlow and sFlow to develop an effective and scalable anomaly detection system using entropy-based algorithm and SDN mitigation using OF. By eliminating the collection of flow statistics through forwarding tables, it decreases overloading of the control plane. It minimizes false positive rates, detects anomalies while traffic is high and can manage real-time traffic effectively. The writers, however, need to concentrate on the interaction of inter-domain OF controllers to manage assaults close their source.

The authors in [80] have created an' OrchSec' orchestrator-based architecture that uses network monitoring and SDN control features to create safety apps that can be used to mitigate assaults such as DoS and scanning. It decouples tasks of control and tracking, thus reducing overhead on the SDN controller. Using sFlow-RT functionalities involving activities such as speed limiting and traffic drop, it was used to identify attacks. The authors must, however, support other versions of OpenFlow and a wider range of security services.

TABLE III. SUMMARIZES THE PROBLEMS AND PROPOSED SOLUTIONS FOR NETWORK SECURITY ENHANCEMENTS USING SDN

Problem	Proposed Solution	Advantages	Limitations
Network Management	Procera [58]	• Richer set of network policies • Reduces complexity • Network controller makes all traffic forwarding decisions and updates flow table entries.	• Implicit delay caused by the control plane and the data plane • Uses OpenFlow specification version 1.0.0 • Doesn't support throughput, limit and QoS
	Encryption based Architecture [60]	• Deployment of middleboxes in choke points efficiently • Dynamic updates of network topology efficiently • Network appliance manipulates only specific traffic • Obtained traffic isolation dynamically	• Performance issues due to use of common use software for encryption • Difficult to implement existing OpenFlow controller applications on large scale
	Network Resilience management [62]	• Support for orchestration • Includes mechanisms for attack detection and anomalies • Common management relationships • Management patterns can be reused in different parts of network	• Doesn't provide support for wider range of network challenges
	SDN based Ad-Hoc network [63]	• Scalable with multiple SDN domains • Enhance forwarding capabilities of ad-hoc devices • Prevent attacks using grid security • Border controllers for guaranteeing independence in each domain	• Support for real environments

	Smart WSN [64]	• Global view helps to overcome inherent weakness • Integration of controller at base station provides efficient network management • Energy saving	• Improvement in performance, reliability and security to support realistic scenarios
Attack Detection and Mitigation	Resonance [65]	• Provides dynamic access control and authentication • Securing enterprise networks	• Doesn't support more complex access control policies
	AVANT-GUARD [66]	• Guaranteeing security between the control plane and data plane by connection migration technique • Improving responsiveness by creating triggers • Offering optimization by expanding feasibility at the data plane	• Doesn't offer specification for the wireless mobile environment • Doesn't favour exchangeability
Attack Detection and Mitigation	NIDS [67]	• Prevents Zombie VMs • Analyzing network traffic using an OF-based network intrusion detection • Suspicious VMs protected by traffic isolation, port blocking, etc. • Incorporates attack graph analytical procedures	• Addresses Zombie Attacks only • Need to improve detection accuracy, • Must include host-based IDs solutions • Scalability
DoS/DDoS Protection	Lightweight method [68]	• Depends on properties of traffic flow • Performed using NOX based network • Derives features with minimum overhead • Able to monitor more than one observation point • Uses Self Organizing Maps • Detection rate is remarkably good	• Doesn't allow communication among various detectors • Not analysed ports of OF switches using statistical methods
	FlowTrApp [70]	• Based on two parameters i.e., flow rate and flow duration • Aimed to identify malicious traffic ranging from low to high and long-lived to short-lived • Categorize the network traffic Based on FTT, a type of attack can be detected	• Doesn't handle all types of DDoS attacks
	Blocking Scheme [71]	• Determines if a given Flow is compromised or not • Prevent the botnet-based DDoS attacks efficiently • Uses standard OpenFlow APIs • Redirects traffic to new address	• Doesn't provide transparent protection to servers inside • Presumes IP address Spoofing • Dependency between SDN controller and servers
Authentication, Authorization, and Accounting	AAA [53]	• Likely to utilize the pre-existing authentication and account infrastructure • Provide strong authentication, authorization, accounting, and policy management	• Scalability
	Certificate-based AAA [73]	• Secure and Flexible • Well-structured privilege system • Reusable • Variety of use cases	• Doesn't provide support for legacy systems

Secure and Scalable Multi-tenancy	OpenvNMS [74]	• Providing elastic isolation between tenants networks • Scalability, flexibility, performance and control centralization • Possibility to run multiple slices	• Doesn't perform well under different scenarios
Secure and Scalable Multi-tenancy	Tualatin [75]	• Offers fine-grained security protection • Insignificant effect on data plane latency • Significantly high throughput	• Challenging to protect overall network infrastructure • Needs adjustments to accommodate virtualized environments
	vCNSMS [76]	• Security in multi-tenant datacenters • Simplify security rule management • Efficient packet inspection to defend against various types of attacks • Scalability and flexibility	• Detecting network policy violations and intrusions using artificial intelligence • Lack of Parallelization
	NetSecCloud [77]	• Centralized database providing security related information • Reconfiguration process to redirect traffic • Flexible and faster reaction time • Well defined policies forwarding rules	• No support for huge data sets
Collect, Detect And Protect	IDS [78]	• No modifications required while extending pre-existing on-device security • Widespread quick responses and network reconstructs available	• Performance issues not addressed
	Combining OpenFlow/sFlow [79]	• Efficient data gathering for anomaly detection • Eliminates collection of Flow statistics, thus avoids control plane overloading • Detects network anomalies during high traffic	• Doesn't provide support for inter-domain OF controllers communication
	OrchSec [80]	• Reduces overhead on SDN controller by enhancing network security • Increases performance using single SDN controller • Detect attacker traffic using sFlow-RT	• Lack of flexibility • Doesn't provide support for every attack mitigation actions

8. Software defined Network (SDN) controllers

SDN controllers are considered as the "brains" of the network [81], an application in SDN architecture that manages flow control in order to improve network management and application performance. Depending upon the programming language, performance, time, purpose, etc. various SDN controllers were developed till now. Examples includes [82], [83]: POX, NOX, Ryu, ONOS, Floodlight, OpenDayLight. Selecting the best controller is a Multi-Criteria Decision Making problem and choosing the best controller for the given problem, we must consider the following requirements: (i) TLS Support and Virtualization (ii) Open source [84].

A. *Features of SDN Controllers*

Various features of SDN Controllers are discussed as follows [85]:

Cross-Platform Compatibility: Most commonly used languages for SDN controllers are python, C++ and Java. Among these languages, Java is considered best suited to run cross-platform, allowing multithreading and have good modularity.

Southbound Interfaces: SDN controller communicates with switches, routers through southbound APIs, e.g. OpenFlow, NETCONF, OF-Config, Opflex. These APIs can be used to control the entire network and make dynamic changes to forwarding rules.

Northbound Interfaces: Mainly used by the application layer to communicate with the controller and provides opportunities to build new applications. Most commonly used northbound API is REST API and provide support for a variety of applications.

OpenFlow Support: Software-defined networks must provide support for OpenFlow protocol which helps to manipulate the forwarding planes. Different versions of OpenFlow protocol has been developed such as v1.0, v1.3 or v1.4, etc.

Network programmability: SDN controllers provide support for network programmability that allows users to manage network efficiently. It provides users a good graphical interface and various applications can be deployed on the top of the controller to perform different management functions.

Efficiency: Efficiency of controller refers to various parameters – performance, scalability, reliability, and security. Since SDN controllers use the concept of centralization, this presents a serious challenge to network performance and reliability, thus some effective measures must be incorporated to deal with such issues.

Partnership: SDN controllers must be maintained under the supervision of a good and experienced partnership that enhances the network for a longer time.

B. Types of SDN controllers of SDN Controllers

In the recent past, a lot of research have been done to compare various SDN controllers by researchers and it is very difficult to decide which controller will perform better under which scenario. Table IV summarizes the top five most popular SDN controllers including the advantages and disadvantages that provide a better understanding of SDN controllers [81],[86].

TABLE IV. SUMMARIZES VARIOUS SDN CONTROLLERS

SDN Controller	**Language**	**Developer**	**Advantages**	**Limitations**
NOX	C++	Niciria Networks	• Primarily used for scientific purposes, manifestations or experimentation	• Low performance of Network • Supports only OpenFlow version 1.0 • No support for distributed environment
Ryu	Python	NTT Labs	• Supports several versions of OpenFlow and other protocols • Works in a distributed manner	• Low performance
ONOS	Java	Linux Foundation	• Open source and distributed NOS • Supports OpenFlow and other protocols • Good Performance • Promotes OpenFlow versions 1.0 and 1.3	• Doesn't activate applications dynamically
Floodlight	Java	Big Switch Networks	• Good Documentation • Support for Multi-Tenant clouds	• Steep Learning Curve
OpenDaylight	Java	Linux Foundation	• Greatly supported by the networking industry • High performance • Robust	• Difficult to learn and develop applications

9. Conclusion and Future Work

Security is one of the most important challenges that require serious attention of researchers nowadays while developing the Internet of things (IoT). In this survey paper, we introduced Software Defined Networking (SDN) as the most important tool to motivate new innovations in the field of network security in IoT. This work specifically attempted to do the following:

- Identified the various challenges while deploying SDN.
- Highlight the various security features of SDN and need of SDN for IoT.
- Discussed the architecture of IoT based on SDN and its applications.
- Help researchers to understand better the advantages and disadvantages of various existing security enhancements provided by SDN.

The future research directions mainly compromise of introducing new security solutions for the integration of SDN and IoT. It is recommended to identify the unexplored areas in the space of security of IoT by SDN. Researchers also need to focus on the various security concerns caused by the architecture of SDN itself and prevent the SDN from different types of threats caused by attackers.

References

[1] Kumar, M. and Goyal, N., 2014. Reviewing underwater acoustic wireless sensing networks. International Journal of Computer Science and Technology, 5(2), pp.95-98.

[2] Goyal, N. and Gaba, A., 2013. A review over MANET-Issues and Challenges. In Dept. of computer science and engineering.

[3] Bertino E, Islam N. Botnets and internet of things security. Computer. 2017 Feb 1(2):76-9.

[4] Bilal T, Faiz Z, Shah MA. Software defined networks: An analysis on robust security practices. In 2017 23rd International Conference on Automation and Computing (ICAC) 2017 Sep 7 (pp. 1-5). IEEE.

[5] Sezer S, Scott-Hayward S, Chouhan PK, Fraser B, Lake D, Finnegan J, Viljoen N, Miller M, Rao N. Are we ready for SDN? Implementation challenges for software-defined networks. IEEE Communications Magazine. 2013 Jul; 51(7):36-43.

[6] Kreutz D, Ramos FM, Verissimo P, Rothenberg CE, Azodolmolky S, Uhlig S. Software-defined networking: A comprehensive survey. Proceedings of the IEEE. 2015; 103 (1):14-76.

[7] Jammal M, Singh T, Shami A, Asal R, Li Y. Software defined networking: State of the art and research challenges. Computer Networks. 2014 Oct 29; 72: 74-98.

[8] Dayal N, Maity P, Srivastava S, Khondoker R. Research trends in security and DDoS in SDN. Security and Communication Networks. 2016 Dec; 9(18):6386-411.

[9] Nayyer A, Sharma AK, Awasthi LK. Issues in Software-Defined Networking. In Proceedings of 2nd International Conference on Communication, Computing and Networking 2019 (pp. 989-997). Springer, Singapore.

[10] Jammal, M., Singh, T., Shami, A., Asal, R. and Li, Y., 2014. Software defined networking: State of the art and research challenges. *Computer Networks*, *72*, pp.74-98.

[11] C.A.B. Macapuna, C.E. Rothenberg, M.F. Magalhaes, In-packet bloom filter-based data-center networking with distributed OpenFlow controllers, in: IEEE 2010GLOBECOM Workshops, 6–10 December, 2010, pp. 584–588.

[12] S. Fang, Y. Yu, C.H. Foh, K.M.M. Aung, A loss-free multipathing solution for data center network using software-defined networking approach, IEEE Trans. Magn. 49 (6) (2013) 2723–2730.

[13] H.J. Kim, J.R. Santos, Y. Turner, M. Schlansker, J. Tourrilhes, N.Feamster, CORONET: fault tolerance for software-defined networks, in: Proceedings 2012 20th IEEE International Conference on Network Protocols (ICNP), October 30–November 2, 2012, pp. 1–2.

[14] A. Voellmy, J.C. Wang, Scalable software-defined network controllers, in: Proceedings, ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, 2012, pp. 289–290.

[15] M.L. Yu, J. Rexford, M.J. Freedman, J. Wang, Scalable flow-based networking with DIFANE, in: Proceedings, ACM SIGCOMM 2010 Conference (SIGCOMM ‘10), New York NY, 2010, pp. 351–362.

[16] A. Farrel and J. Ash, “No Title,” pp. 1–40, 2006.

[17] M. Raju, A. Wundsam, M. Yu, NOSIX: a lightweight portability layer for the SDN OS, ACM SIGCOMM Comput. Commun. Rev.44(April)(2014)28–35. <http://www1.icsi.berkeley.edu/~andi/nosix_ons13-

extabstract.pdf>.

[18] Reitblatt M, Canini M, Guha A, Foster N. Fattire: Declarative fault tolerance for software-defined networks. In Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking 2013 Aug 16 (pp. 109-114).

[19] Botelho, F., Bessani, A., Ramos, F. and Ferreira, P., 2014. Smartlight: A practical fault-tolerant SDN controller. *arXiv preprint arXiv:1407.6062*.

[20] Lu, Y. and Zhu, S., 2015, December. SDN-based TCP congestion control in data center networks. In *2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC)* (pp. 1-7). IEEE.

[21] Vandana C. Security improvement in iot based on software defined networking (sdn). International Journal of Science, Engineering and Technology Research (IJSETR). 2016 Jan; 5(1):2327-4662.

[22] Nunes BA, Mendonca M, Nguyen XN, Obraczka K, Turletti T. A survey of software-defined networking: Past, present, and future of programmable networks. IEEE Communications Surveys & Tutorials. 2014 Feb 13; 16(3):1617-34.

[23] Valdivieso Caraguay AL, Benito Peral A, Barona Lopez LI, Garcia Villalba LJ. SDN: Evolution and opportunities in the development IoT applications. International Journal of Distributed Sensor Networks. 2014 May 4; 10 (5):735142.

[24] Casado M, Freedman MJ, Pettit J, Luo J, McKeown N, Shenker S. Ethane: Taking control of the enterprise. In ACM SIGCOMM Computer Communication Review 2007 Aug 27 (Vol. 37, No. 4, pp. 1-12). ACM.

[25] Reitblatt M, Foster N, Rexford J, Schlesinger C, Walker D. Abstractions for network update. ACM SIGCOMM Computer Communication Review. 2012 Sep 24;42(4):323-34.

[26] Heller B, Seetharaman S, Mahadevan P, Yiakoumis Y, Sharma P, Banerjee S, McKeown N. Elastictree: Saving energy in data center networks. In Nsdi 2010 Apr 28 (Vol. 10, pp. 249-264).

[27] Shirayanagi H, Yamada H, Kono K. Honeyguide: A vm migration-aware network topology for saving energy consumption in data center networks. IEICE TRANSACTIONS on Information and Systems. 2013 Sep 1; 96 (9):2055-64.

[28] Ramachandran A, Mundada Y, Tariq MB, Feaster N. Securing enterprise networks using traffic tainting. In Proc. SIGCOMM 2009 Oct (pp. 1-2).

[29] Sherwood R, Chan M, Covington A, Gibb G, Flajslik M, Handigol N, Huang TY, Kazemian P, Kobayashi M, Naous J, Seetharaman S. Carving research slices out of your production networks with OpenFlow. ACM SIGCOMM Computer Communication Review. 2010 Jan 7; 40 (1):129-30.

[30] Sherwood R, Gibb G, Yap KK, Appenzeller G, Casado M, McKeown N, Parulkar G. Flowvisor: A network virtualization layer. OpenFlow Switch Consortium, Tech. Rep. 2009 Oct 14;1: 132.

[31] Yap KK, Sherwood R, Kobayashi M, Huang TY, Chan M, Handigol N, McKeown N, Parulkar G. Blueprint for introducing innovation into wireless mobile networks. In Proceedings of the second ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures 2010 Sep 3 (pp. 25-32). ACM.

[32] Yap KK, Kobayashi M, Sherwood R, Huang TY, Chan M, Handigol N, McKeown N. OpenRoads: Empowering research in mobile networks. ACM SIGCOMM Computer Communication Review. 2010 Jan 7; 40(1):125-6.

[33] Liu L, Tsuritani T, Morita I, Guo H, Wu J. OpenFlow-based wavelength path control in transparent optical networks: A proof-of-concept demonstration. In 2011 37th European Conference and Exhibition on Optical Communication 2011 Sep 18 (pp. 1-3). IEEE.

[34] Patel AN, Ji PN, Wang T. Qos-aware optical burst switching in openflow based software-defined optical networks. In 2013 17th International Conference on Optical Networking Design and Modeling (ONDM) 2013 Apr 16 (pp. 275-280). IEEE.

[35] Calvert KL, Edwards WK, Feaster N, Grinter RE, Deng Y, Zhou X. Instrumenting home networks. ACM SIGCOMM Computer Communication Review. 2011 Jan 22;41 (1):84-9.

[36] Feaster N. Outsourcing home network security. In Proceedings of the 2010 ACM SIGCOMM workshop on Home networks 2010 Sep 3 (pp. 37-42). ACM.

[37] Pentikousis K, Wang Y, Hu W. Mobileflow: Toward software-defined mobile networks. IEEE Communications magazine. 2013 Jul;51 (7):44-53.

[38] Kassler A, Skorin-Kapov L, Dobrijevic O, Matijasevic M, Dely P. Towards QoE-driven multimedia service negotiation and path optimization with software defined networking. In SoftCOM 2012, 20th International Conference on Software, Telecommunications and Computer Networks 2012 Sep 11 (pp. 1-5). IEEE.

[39] Georgopoulos P, Elkhatib Y, Broadbent M, Mu M, Race N. Towards network-wide QoE fairness using openflow-assisted adaptive video streaming. In Proceedings of the 2013 ACM SIGCOMM workshop on Future human-centric multimedia networking 2013 Aug 16 (pp. 15-20). ACM.

[40] Ganchev I, Van den Berg H, Van der Mei RD. Autonomous control for a reliable internet of services methods, models, approaches, techniques, algorithms, and tools (2018).

[41] Fonseca P, Bennesby R, Mota E, Passito A. A replication component for resilient OpenFlow-based networking. In 2012 IEEE Network operations and management symposium 2012 Apr 16 (pp. 933-939). IEEE.

[42] Bellavista, P., Dolci, A. and Giannelli, C., 2018, June. MANET-Oriented SDN: motivations, challenges, and a solution prototype. In *2018 IEEE 19th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)* (pp. 14-22). IEEE.

[43] Chahal, M., Harit, S., Mishra, K.K., Sangaiah, A.K. and Zheng, Z., 2017. A survey on software-defined networking in vehicular ad hoc networks: Challenges, applications and use cases. *Sustainable cities and society*, *35*, pp.830-840.

[44] Akyildiz, I.F., Wang, P. and Lin, S.C., 2016. SoftWater: Software-defined networking for next-generation underwater communication systems. *Ad Hoc Networks*, *46*, pp.1-11.

[45] Fan, R., Wei, L., Du, P., Mc Goldrick, C. and Gerla, M., 2016, November. A SDN-controlled underwater MAC and routing testbed. In *MILCOM 2016-2016 IEEE Military Communications Conference* (pp. 1071-1076). IEEE.

[46] Fajar AP, Purboyo TW. A Survey Paper of Distributed Denial-of-Service Attack in Software Defined Networking (SDN). International Journal of Applied Engineering Research. 2018;13(1):476-82.

[47] Shin S, Xu L, Hong S, Gu G. Enhancing network security through software defined networking (SDN). In 2016 25th International Conference on Computer Communication and Networks (ICCCN) 2016 Aug 1 (pp. 1-9). IEEE.

[48] Sherwood R, Gibb G, Yap KK, Appenzeller G, Casado M, McKeown N, Parulkar GM. Can the production network be the testbed?. InOSDI 2010 Oct 4 (Vol. 10, pp. 1-6).

[49] Al-Shabibi A, De Leenheer M, Gerola M, Koshibe A, Parulkar G, Salvadori E, Snow B. OpenVirteX: Make your virtual SDNs programmable. In Proceedings of the third workshop on Hot topics in software defined networking 2014 Aug 22 (pp. 25-30). ACM.

[50] Shin SW, Gu G. Cloudwatcher: Network security monitoring using openflow in dynamic cloud networks. In Network Protocols (ICNP) 2012 2012 Oct 30 (pp. 1-6). IEEE.

[51] Shirali-Shahreza S, Ganjali Y. FleXam: flexible sampling extension for monitoring and security applications in openflow. In Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking 2013 Aug 16 (pp. 167-168). ACM.

[52] Shin SW, Porras P, Yegneswara V, Fong M, Gu G, Tyson M. Fresco: Modular composable security services for software-defined networks. In20th Annual Network & Distributed System Security Symposium 2013 Feb 26. NDSS.

[53] Voellmy A, Hudak P. Nettle: Taking the sting out of programming network routers. In International Symposium on Practical Aspects of Declarative Languages 2011 Jan 24 (pp. 235-249). Springer, Berlin, Heidelberg.

[54] Shin S, Yegneswaran V, Porras P, Gu G. Avant-guard: Scalable and vigilant switch flow management in software-defined networks. In Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security 2013 Nov 4 (pp. 413-424). ACM.

[55] Sonchack J, Smith JM, Aviv AJ, Keller E. Enabling Practical Software-defined Networking Security Applications with OFX. InNDSS 2016 Feb 21 (Vol. 16, pp. 1-15).

[56] Schulz-Zander J, Mayer C, Ciobotaru B, Schmid S, Feldmann A. OpenSDWN: Programmatic control over home and enterprise WiFi. InProceedings of the 1st ACM SIGCOMM symposium on software defined networking research 2015 Jun 17 (p. 16). ACM.

[57] Schehlmann L, Abt S, Baier H. Blessing or curse? Revisiting security aspects of Software-Defined Networking. In10th International Conference on Network and Service Management (CNSM) and Workshop 2014 Nov 17 (pp. 382-387). IEEE.

[58] Kim H, Feaster N. Improving network management with software defined networking. IEEE Communications Magazine. 2013 Feb; 51(2):114-9.

[59] McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, Shenker S, Turner J. OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review. 2008 Mar 31; 38 (2):69-74.

[60] Lara A, Kolasani A, Ramamurthy B. Simplifying network management using software defined networking and OpenFlow. In 2012 IEEE International Conference on Advanced Networks and Telecommunciations Systems (ANTS) 2012 Dec 16 (pp. 24-29). IEEE.

[61] Gibb G, Zeng H, McKeown N. Initial thoughts on custom network processing via waypoint services. In WISH-3rd Workshop on Infrastructures for Software/Hardware co-design, CGO 2011 Apr.

[62] [60] Smith, P., Schaeffer-Filho, A., Hutchison, D. and Mauthe, A., 2014, May. Management patterns: SDN-enabled network resilience management. In *2014 IEEE Network Operations and Management Symposium (NOMS)* (pp. 1-9). IEEE.

[63] Flauzac, O., González, C., Hachani, A. and Nolot, F., 2015, March. SDN based architecture for IoT and improvement of the security. In *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops* (pp. 688-693). IEEE.

[64] De Gante, A., Aslan, M. and Matrawy, A., 2014, June. Smart wireless sensor network management based on software-defined networking. In *2014 27th Biennial Symposium on Communications (QBSC)* (pp. 71-75). IEEE.

[65] A. K. Nayak, A. Reimers, N. Feaster, and R. Clark, "Resonance: dynamic access control for enterprise networks," in Proceedings of the 1st ACM workshop on Research on enterprise networking. ACM, 2009, pp. 11–18.

[66] Shin S, Yegneswaran V, Porras P, Gu G. Avant-guard: Scalable and vigilant switch flow management in software-defined networks. In Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security 2013 Nov 4 (pp. 413-424). ACM.

[67] Chung CJ, Khatkar P, Xing T, Lee J, Huang D. NICE: Network intrusion detection and countermeasure selection in virtual network systems. IEEE transactions on dependable and secure computing. 2013 Jul;10 (4):198-211.

[68] Braga R, de Souza Mota E, Passito A. Lightweight DDoS flooding attack detection using NOX/OpenFlow. In LCN 2010 Oct 10 (Vol. 10, pp. 408-415).

[69] Gude N, Koponen T, Pettit J, Pfaff B, Casado M, McKeown N, Shenker S. NOX: towards an operating system for networks. ACM SIGCOMM Computer Communication Review. 2008 Jul 1;38 (3):105-10.

[70] Buragohain C, Medhi N. FlowTrApp: An SDN based architecture for DDoS attack detection and mitigation in data centers. In 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN) 2016 Feb 11 (pp. 519-524). IEEE.

[71] Lim, S., Ha, J., Kim, H., Kim, Y. and Yang, S., 2014, July. A SDN-oriented DDoS blocking scheme for botnet-based attacks. In *2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN)* (pp. 63-68). IEEE.

[72] Kuliesius F, Dangovas V. SDN enhanced campus network authentication and access control system. In 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN) 2016 Jul 5 (pp. 894-899). IEEE.

[73] Toseef, U., Zaalouk, A., Rothe, T., Broadbent, M. and Pentikousis, K., 2014, September. C-BAS: Certificate-based AAA for SDN experimental facilities. In *2014 Third European Workshop on Software Defined Networks* (pp. 91-96). IEEE.

[74] Ahmed MF, Talhi C, Pourzandi M, Cheriet M. A software-defined scalable and autonomous architecture for multi-tenancy. In 2014 IEEE International Conference on Cloud Engineering 2014 Mar 11 (pp. 568-573). IEEE.

[75] Wang X, Liu Z, Li J, Yang B, Qi Y. Tualatin: Towards network security service provision in cloud datacenters. In 2014 23rd International Conference on Computer Communication and Networks (ICCCN) 2014 Aug 4 (pp. 1-8). IEEE.

[76] Chen, Z., Dong, W., Li, H., Zhang, P., Chen, X. and Cao, J., 2014. Collaborative network security in multi-tenant data center for cloud computing. *Tsinghua Science and Technology*, *19*(1), pp.82-94.

[77] Seeber, S. and Rodosek, G.D., 2014, November. Improving network security through SDN in cloud scenarios. In *10th International Conference on Network and Service Management (CNSM) and Workshop* (pp. 376-381). IEEE.

[78] Skowyra R, Bahargam S, Bestavros A. Software-defined ids for securing embedded mobile devices. In 2013 IEEE High Performance Extreme Computing Conference (HPEC) 2013 Sep 10 (pp. 1-7). IEEE.

[79] Giotis, K., Argyropoulos, C., Androulidakis, G., Kalogeras, D. and Maglaris, V., 2014. Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments. *Computer Networks*, *62*, pp.122-136.

[80] Zaalouk, A., Khondoker, R., Marx, R. and Bayarou, K.M., 2014, May. OrchSec: An orchestrator-based architecture for enhancing network-security using Network Monitoring and SDN Control functions. In *NOMS* (pp. 1-9).

[81] Stancu AL, Halunga S, Vulpe A, Suciu G, Fratu O, Popovici EC. A comparison between several software defined networking controllers. In2015 12th International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS) 2015 Oct 14 (pp. 223-226). IEEE.

[82] Kaur S, Singh J, Ghumman NS. Network programmability using POX controller. In ICCCS International Conference on Communication, Computing & Systems, IEEE 2014 Feb (Vol. 138).

[83] Sheikh MN. SDN-Based Approach to Evaluate the Best Controller: Internal Controller NOX and External Controllers POX, ONOS, RYU. Global Journal of Computer Science and Technology. 2019 Feb 7.

[84] R. Khondoker, A. Zaalouk, R. Marx, and K. Bayarou, “Feature-based Comparison and Selection of Software Defined Networking (SDN) Controllers,” *2014 World Congr. Comput. Appl. Inf. Syst.*, pp. 1–7, 2014.

[85] O. Salman, I. H. Elhajj, A. Kayssi, and A. Chehab, “SDN Controllers : A Comparative Study,” *2016 18th Mediterr. Electrotech. Conf.*, no. 978, pp. 1–6, 2020.

[86] N. Feaster, “Software Defined Networking Lesson Overview”, Available: https://www.usebackpack.com/resources/ 7575/download? 1454314572.

Omerah Yousuf is a Ph.D Scholar in the Department of Computer Science & Engineering at NIT Srinagar, J&K (India). She received B.Tech in Computer Science & Engineering from Islamic University of Science & Technology Awantipora, Pulwama (India) in 2011, M.Tech in Computer Science & Engineering from Visvesvaraya Technological University Belagavi, Karnataka (India) in 2014. Her current research interests includes Security in Internet of Things and Wireless Sensor Networks.

Roohie Naaz Mir is a Professor and HoD in the Department of Computer Science & Engineering at NIT Srinagar, INDIA. She received B.E. (Hons) in Electrical Engineering from University of Kashmir (India) in 1985, M.E. in Computer Science & Engineering from IISc Bangalore (India) in 1990 and Ph.D from University of Kashmir, (India) in 2005. She is a Fellow of IEI and IETE India, senior member of IEEE and a member of IACSIT and IAENG. She is the author of many scientific publications in international journals and conferences. Her current research interests include reconfigurable computing and architecture, mobile and pervasive computing, security and routing in wireless adhoc and sensor networks.