Review Article

Services and Security Threats in SDN Based VANETs: A Survey

Hammad Shafiq,¹ Rana Asif Rehman ,¹ and Byung-Seo Kim ²

¹Department of Computer Science, National University of Computer and Emerging Sciences, Chiniot-Faisalabad Campus, Chiniot 35400, Pakistan

²Department of Computer and Information Communication Engineering, Hongik University, Sejong City 30016, Republic of Korea

Correspondence should be addressed to Byung-Seo Kim; jsnbs@hongik.ac.kr

Received 25 October 2017; Revised 5 March 2018; Accepted 14 March 2018; Published 23 April 2018

Academic Editor: Jianhua He

Copyright © 2018 Hammad Shafiq et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As the number of vehicles is increasing, the number of road side accidents is also increasing rapidly. The majority of these accidents is caused by the negligence of the driver. For intelligent transportation services, new protocols and architecture are continuously being developed by researchers around the globe. Thus to ensure the safety of drivers many countries are now adopting and investing a lot on vehicular ad hoc network (VANET). On the other perspective, there are many issues related to this field that must be resolved before VANET technology is practically adopted. In the case of no or low-security, several attacks can occur that may affect the efficiency and the reliability of the system. To make VANET systems more efficient software defined networking (SDN) technology is introduced in it. This technique was shortly named as SDN-based VANET. SDN-based VANET system helps us to get rid of the restriction and the challenges that are present in the simple VANET systems. It helps us to decrease the overall load on the network by managing the overall network through a single remote controller. In this survey paper, we will elaborate the concept of SDN-based VANET, its working, applications, services, security threats, and benefits over the previous techniques.

1. Introduction

Because travel modes change over time, new techniques and mechanisms are introduced to increase their efficiency. At present, cars and other vehicles are used for traveling, and drivers are interested in having their own vehicles. However, as the number of private vehicles increases, the ratio of road accidents also increases. There are many reasons for road accidents, such as the negligence of the driver and mechanical faults in the vehicle. In the modern world, safe means of travel are required in order to prevent accidents that may cause serious damage to human lives.

To resolve these problems, the researcher investigated the wireless network domain. The wireless network provides the convenience of a wire-free environment to its hosts, so they have the ability to move freely, which leads to dynamic topologies. The main idea is to include the vehicles in a specific area in a communication network. However, this dynamic arrangement of nodes also causes unpredictability in network topologies. To achieve our goal, we use a mobile ad hoc network (MANET), which is related to the wireless ad hoc network [1]. MANETs are wireless ad hoc networks in

which every device is independent and moves freely in any direction. MANET is a self-configuring and infrastructure-less network that supports the mobility of devices. Each device frequently changes its links from one device to the other devices, which results in a highly dynamic and autonomous topology. Each device in this network plays the dual role of participant and router. The mobile ad hoc network is constructed as soon as the devices are connected to each other. Hence, the node that forwards data depends on the network connectivity, whereas, in wired networks, a fixed router performs the routing tasks. This system also differs from a wired infrastructure because in wireless networks, a specific node can act as an access point [2].

Within the field of MANET, scientists have discovered a new field known as the vehicle ad hoc network (VANET). It uses the same technology as MANET uses, that is, each node in VANET is free to move in any direction, which leads to frequent changes in the links between them. The nodes forward data to other nodes, which are then vehicles. The devices on the nodes must send data continuously to maintain the connection and the proper flow of data. These devices consist of efficient hardware that sends correct data without any delay

and may be connected to the internet. At any point in time, many devices may be connected to a single node that is a device on the vehicle. For the purpose of communication, all the components of VANET communicate wirelessly, which regulates the various aspects of the communication, such as security, latency, and the data transmission range. In wireless communication, different routing protocols are used. Protocols are the combination of rules for communication between two or more entries using any physical medium. Moreover, protocols are rules that are used to define the syntax of the communication and recovery method in case of any fault. Protocols are implemented in software and hardware and may be a combination of both. Several routing protocols are similar in MANETs and VANETs, but when they are compared according to behavior and characteristics, many of these protocols are dissimilar. A key difference between VANET and MANET is the production cost. The network topology of VANET is fast and highly dynamic because of the high speed of cars, whereas MANET is slow. In addition, more network bandwidth is required in VANET than in MANET [3]. Delivering accurate data is now considered a key task because of frequent signal disruption, contact opportunities in VANET, and the fast changes in topology. However, in VANET, the transmission of the signals is a main challenge because as the vehicle changes its position, the possibility of disconnection increases. Other challenges are the failed transmission of data because of the inadequate quality of the wireless links between different nodes and the absence of a link between the source and the destination. To address the problem of disconnection between nodes, dedicated short-range communication (DSRC) protocols are used for communication in VANET. DSRC provides a high data transmission rate and is used in safety applications. Through this form of communication, vehicles can share secure information to prevent any mishap as well as in post-accident investigation [4]. Information that is not relevant is discarded, which saves processing time. The aim of sharing the information is to deliver an alert message to the driver about the expected risk, which decreases the possibility of accidents while driving.

VANET has attracted researchers in different fields to develop protocols, applications, and simulation tools. However, researchers and developers still face several challenges. Peoples from different countries are contributing to get rid of these challenges by developing new communication protocols, advanced hardware, data security, and privacy techniques [5, 6].

To increase the efficiency of the entire network, a software defined network (SDN) technique was introduced in the field of VANET system. The SDN is innovative in the network field, and it is now considered a new, alternative technique for controlling the flow of the entire network in a programmable and systematic way. The SDN system also defines the data plane in data forwarding and the network control plane in controlling the entire network. Previously, all applications related to SDN-based systems used wires, such as Facebook, Google, and Cisco. However, because of its potential flexibility, the SDN system is considered an alternative method for mobile wireless networks. The separation of the control and

data planes makes this network simple and easy to manage even when the number of nodes in the network increases rapidly.

This new SDN system has been introduced in VANET to manage the communication throughout the network. In a simple VANET system, all communication is controlled with the help of routing protocols. However, because of the increase in the demand for the VANET system, it has become difficult for the routing protocols to manage the communication, privacy, and security of the system. Nevertheless, this problem can be solved easily with the help of the SDN system. Collaboration is developed between a simple VANET and an SDN-based system to control the overall communication in the system. The introduction of SDN in VANET has simplified the management of the overall behavior of the network.

This paper is organized as follows. Section 2 describes the background and related work, Section 3 presents the SDN-based VANET, and Section 4 presents the services of SDN-based VANET. Section 5 describes the security threats and challenges related to the SDN-based VANET. Section 6 describes the evolution of the SDN controller architecture for VANET, and Section 7 presents the applications of the SDN-based VANET. The final section concludes the paper.

2. Background and Related Work

Several devices are used for communication in computer networks, such as routers and switches. Network operators, which are responsible for the events that occur in the system configure these devices. Previously, network operators manually installed policies in the devices, which sometimes required frequent changes. Hence, it became very difficult to manage communication in the network, which could lead to its failure. Furthermore, internet-based applications and services have become increasingly complex, and it has become necessary to resolve many issues to make them work efficiently. To resolve this problem, a programmable network was introduced, which was further defined as the SDN [7].

2.1. Software Defined Network. The core concept of the SDN system is the separation of the data plane and the control plane as shown in Figure 1.

This feature facilitates the deployment of new routing protocols and management by eliminating the previous need to impose different types of policies and protocols on every device that is connected to the network [9]. The SDN system has two architectures: forwarding and control element separation (ForCES) and OpenFlow. However, these architectures are designed differently. ForCES is a strategy in which the network devices have separate control and forwarding units, and a single device contains both units. ForCES includes two entities, the forwarding element (FE) and the control element (CE), both of which implement the ForCES protocol to communicate. The FC is used to provide per-packet handling by utilizing hardware [10]. The CE is used to control the functionality of the ForCES protocols. The CE model works as the master, and the FE works as the slave. Another important component of ForCES is the logical function block (LFB).

FIGURE 1: Logical representation of software defined network [7].

This functional block depends on the FE and is controlled by the CE using protocols. The LFB enables the CE to configure the FE's functionality, including its processing of the data packet. In the OpenFlow architecture, forwarding devices such as switches contain flow tables, which may include more than one in each device. These flow tables contain flow entries that contain information about how the packets are forwarded [11, 12]. Flow entries contain information that is used to match the incoming packet. This information is usually in the packet header. Counters are also used to collect information about the number of packets received. The instructions in the OpenFlow system help in deciding which action is to be performed if a match is found. When a packet is received by a node, the node first searches the type of action in the packet in the matching field of each entry in the flow tables. If a match is found, the packet is then processed to the controller by the OpenFlow system, which is used to control the flow of a packet in the entire network [13]. Whenever a traffic flow matches the flow table, the OpenFlow system knows the way in which to deal with the traffic flow. This facility of the OpenFlow allows the system to use its resources efficiently. With the help of the network controller, the configuration of the system is simplified because, by modifying an individual device, all other devices in the entire network are also modified. All network devices simply update the policies as defined by the controller without the need to test them in different network protocols. This activity helps to save the time and resources required to test and process the receiving data. The communication in an SDN-based network is shown in Figure 2.

2.2. Vehicular Ad Hoc Network. The VANET is a field of MANET that is used for vehicular communication to ensure the safety of the driver. VANET enables decision making.

FIGURE 2: Communication pattern of the SDN-based VANET system.

FIGURE 3: Architecture of simple VANET.

The transmission among different vehicles and the road side unit (RSU) is attained by a wireless medium. This strategy is used for the communication of wide range of data that provide information to drivers and enhance road safety. The fundamental components of this system are the RSU, the on-board unit (OBU), and the application unit (AU), which are shown in Figure 3. Different kinds of air interfaces and communication protocols are proposed for VANETs. These services are provided by communication access for land mobiles (CALM) architecture. It provides V2I (vehicle to infrastructure), V2V (vehicle to vehicle), and I2I (infrastructure to infrastructure) air interfaces paradigms. Various kinds of communication technologies such as GSM, WiMax, RFID, and DSRC can be utilized in it for transmission purpose.

2.2.1. On-Board Unit. The on-board unit is attached to the vehicle and is used to exchange information with other OBUs and RSUs. It contains a resource command processor and a memory resource, which are used to read and write information that is then sent to RSUs or other vehicles [1]. The OBU is connected to a user interface and a network device, which provides short-range wireless communication based on a radio technology.

2.2.2. Application Unit. The AU is placed inside a vehicle containing an application or user interface that uses the communication abilities of the OBU. The AU is linked with the OBU through a wireless or wired connection, which are sometimes placed in the same physical unit. The AU can be used as a personal device or a personal digital assistant. AU communication is carried out with the help of OBU devices [2].

2.2.3. Roadside Unit. RSU is a physical device that is permanently attached at the side of the road or at a parking stand. RSU devices are connected to an internet source to provide communication between vehicles. The RSU is used to provide services to host an application, and the OBU uses these services to run the application [4].

3. Software Defined VANET

The following section describes the basic architecture of the SDN-based VANET system.

3.1. Software Defined VANET Architecture. The transmission among different vehicles and the RSU is attained by a wireless medium. This strategy is used for the communication of a wide range of data that provide information to drivers and enhances road safety [8]. In the SDN-based VANET system, communication is carried out between the data plane and the control plane. The data plane contains the traffic forwarding devices while the control plane controls the flow of communication. The control plane has different layers, each of which collaborate to provide the functionality of the controller to the network as shown in Figure 4.

3.1.1. Data Plane. This layer contains forwarding and data processing devices such as the RSU. The devices attached to the RSU or the vehicle are used to perform different actions based on the policies defined on each device by the controller. The communication between the SDN controller and the devices is carried out through the standard protocol, OpenFlow [14], which is the protocol used for communication between the controller and the RSU or nodes that are directly connected to the controller. The OpenFlow protocol is divided into four fields as shown in Figure 5. The first one is the priority field, which is used to define the order in which the data are matched with the defined rules. The rules assign a priority level. The data that are matched to the higher priority rule are processed first. The second field is the matching condition, which is defined according to the IP perspective. When data are received from a different

FIGURE 4: Layered architecture of an SDN network (VANET).

FIGURE 5: Structure of an open flow rule.

IP node, this field matches the conditions according to the IP, which is usually the source and the destination IP. The third field is the action field, which is used to define the action that should be taken when the data packet is received. The fourth field is the counter, which is used to count the number of data packets received about a specific task. It helps the system to verify whether the data are completely received or not. Over time, many versions of the OpenFlow system have been released to meet the new terms and conditions of data forwarding. By 2014, five versions of the OpenFlow system had been released. In the OpenFlow system, software that runs on Linux OS is used to switch the network devices. This system is mainly used in cloud computing to connect multiple devices to a single host in the same external network. The well-known OpenFlow software Open vSwitch [15] is used to switch software in a virtual environment. Its performance with respect to the flow rate is 25% greater compared to a Linux kernel-based software switch. OpenFlow is the southbound interface is the most widely used protocol. Several other protocols exist, such as the Open vSwitch database management protocol (OVSDb) [16], which is used to provide a programmable interface to the Open vSwitch to support many advanced features.

3.1.2. Control Plane. In a traditional computer system, it is the responsibility of the OS to manage all the resources of the system. However, in the SDN-based VANET, this role is carried out by the SDN controller. A network controller is in the second layer of the SDN system, which is also defined as the brain of the network. This controller is used to install, update, and delete the rules on each device that is directly connected to the network controller. There is no standard rule for the SDN controller to define the services definition. The services are defined as the facilities that are provided by the SDN controller. This property of the SDN controller is difficult to implement, and many issues are encountered in implementing new policies and techniques to make the system work more efficiently [17]. Furthermore, the centralization of the system may include issues such as security threats, unreliability, unstable performance, and the instability of the system. There are many research challenges in this layer with regard to placing the controller properly.

The vehicles in the SDN-based VANET move with high mobility. Each vehicle has policies for communicating with the received data. In the case of any new flow of data, the nodes, or the vehicles, are helped by the RSU or another device that is directly connected to the SDN controller. To avoid latency or waiting for the response, it is compulsory to place the controller near a dense traffic area to facilitate every node in contacting the controller. In addition, the RSU also should be placed near dense traffic, thereby reducing the overall cost of communication. Two options are available for controlling the functionality of the network: centralized control and distributed control. In centralized control, the functionality of the system can be controlled by a central controller, but if it malfunctions or fails, it is not possible to control the system's functionality until the control gain is in its original state. A solution to this problem is the distributed controller in which more than one controller is used to control the functioning of the system. These controllers are interconnected through a wireless medium, which decreases the complexity of the system. By using a single controller as the parent controller, the functions of the other controllers can also be controlled. A new, logical centralized system was introduced by ONIX [18], which was used to handle the different distributed states between individual controllers. The HyperFlow [19] system helps by providing advanced features for the distributed controller, which include localizing the flow decision to the individual controller and minimizing latency during a communication. It also provides a resilience pattern to the network in the case of a failure caused by a fault in any controller. In SDN-based VANET, the network OS is used to control the overall functionality of the network. For example, in traditional computer systems, the OS provides abstraction in the form of APIs to provide access to shareable resources. SDN programming languages are used by the network OS to obtain the desired functionality of the system.

3.1.3. Network Compiler. Network programs are designed using the SDN-based programming languages and compilers associated with them. These programs are used to control the functionality of the controller. Applications are built directly in the controller API, but this layer facilitates the application's

portability, code reusability, and network abstraction. First, the high-level network abstraction is discussed. By using the high-level language for the development, we neglect the low-level operation of the programming language. The developer should develop the program with respect to high-level policies. The compiler will automatically generate rules related to the OpenFlow, which are then installed in each device that is connected directly to the controller. The network compiler also facilitates the reusability of the code. To obtain the proper functionality of the system, two separate programs are required to be carried out by the system. One program forwards the packet from one interface to another interface, and the other program measures the web traffic entering the system. Because the network compiler facilitates the reusability of resources, a single program that carries out both functionalities is created.

3.1.4. Challenges Related to the Cross Layer. The cross layer of the control plane is used to resolve challenges that are related to this plane, such as testing, troubleshooting, and SDN software debugging. Several tools are available for testing the functionality of OpenFlow system. For example, the Mininet [20] system is used for the prototyping of the SDN-based VANET using a single computer system. The system's virtualization capability simplifies the detection of problems in the system. The fs-sdn [21] is also used to perform simulations using the flow let instead of a packet. It can simulate a large network with results that are more accurate compared to the Mininet. For the troubleshooting of the network, ping or traceroute are used to discover faults in the network. However, it is sometimes very difficult to identify the problem if it is generated by the configuration settings. The software debugging program NICE is used to debug the application of SDN system, which helps in resolving errors related to debugging.

3.2. Fundamental Components of SDN Based VANETs. The fundamental components of this system are the SDN controllers, SDN wireless node, and the SDN RSU, which are shown in Figure 6.

3.2.1. SDN Controller. The SDN controller is used to control the network behavior of the complete system. It is usually connected to the RSU.

3.2.2. SDN Nodes. The vehicles in the SDN-based VANET are known as nodes. They are equipped with OBU and AU. They collectively participate during the communication. The AU can be used as a personal device or a personal digital assistant. The AU communicates through the network using the OBU device and is responsible for all kinds of communication.

3.2.3. SDN Roadside Unit. The RSU is a physical device that is permanently attached at the side of the road or at a parking stand as shown in Figure 6. The RSU device is connected to a network to provide communication between vehicles and with the SDN controller. The RSU is used to provide the services required to host an application, and the OBU uses these services. The RSU has the ability to connect to the

FIGURE 6: Core concept of SDN-based VANET.

internet, which allows the AUs in multiple vehicles to connect to the internet.

3.2.4. SDN Operation Overview. The main functionality of the SDN system is carried out by the separation of the control and data components. The SDN-based network can operate in different modes. Currently, the SDN system can operate in three modes: central control mode, distributed control mode, and hybrid control mode [8]:

(a) *Central Control Mode.* In this mode, the SDN controller is used to control the functionality of the wireless nodes and the RSUs. The SDN controller predefines the functionality of the SDN data element. The SDN controller defines all the rules of the data flow in the network. The central control mode of an SDN-based VANET system is shown in Figure 7, in which the two vehicles and a road side tower take their respective parameters from controller. The central controller basically disseminates the flow rules which further populate the forwarding tables in vehicles. Moreover, the yellow arrows and red arrows show the control and data plane communication, respectively.

(b) *Distributed Control Mode.* In this mode, the communication between the vehicles and the RSU is not under the guidance of the SDN controller. This mode is similar to a self-organized network without any features of SDN, such as running under the GPSR routing protocol, as shown in Figure 8, in which the two vehicles and a road side tower exchange their communication parameters by themselves without involving the controller.

FIGURE 7: Central control mode [8].

FIGURE 8: Distributed control mode [8].

(c) *Hybrid Control Mode.* This mode includes the functionality when the SDN-based controller has full control over the entire network. As shown in Figure 9, the SDN controller relinquishes control to the local nodes for packet processing. Therefore, the control is switched among all SDN elements. The SDN controller sends out a policy that contains the rules for the functionality of the network. However, the RSU and SDN nodes use their own intelligence to forward the packets. In specific cases, the RSU and the nodes are used to run specific protocols with predefined parameters.

The SDN wireless network has features that are similar to wired SDN architecture. The SDN controller is used to specify all the rules for the entire network. There is always an issue of availability in the wireless channel because it causes the loss of communication between the nodes and the controller. To overcome the problem of failure, the software-defined VANET includes a recovery setup that restores the system's functionality. Moreover, each SDN node contains an intelligent system that deals with such problems. For example, in the case of the loss of communication, the system will start

TABLE 1: Services provided by SDN-based VANETs.

Service	Mechanism	Advantages
Intrusion detection	(i) Intrusion detection system run by SDN controller (ii) Analyze the traffic behavior	Enhance detection capability as compared to the traditional network
Detection of malicious behavior	(i) Detection performed by RSU or vehicle node (ii) Use probing technique	(i) Easily implemented (ii) Reduce load on controller
Network forensics	Monitor flow of data and detect the modification in device settings	Identify the attacker
Self-healing	Rules are defined by the controller that provides detection, diagnoses, and recovery	Automatic recovery against any attack
Path selection	SDN controller performs rerouting in order to obtain balanced routing	Avoid congestion
Channel/frequency selection	SDN controller selects the appropriate channel for communication	Efficient resource utilization

FIGURE 9: Hybrid control mode [8].

running under the traditional routing protocols. To increase the system's intelligence, it is necessary to learn the network topology first. A beacon message is sent to all the neighbor nodes to obtain the details of all the nearby nodes. This information is then sent to the SDN controller to construct a graph of the connected nodes, which is used to make decisions about the communication, such as selecting a path on which to route the data packet through the entire network.

4. Services of Software-Defined VANETs

The SDN-based VANET system is of great help in controlling the network and the SDN model because it allows the management of the transmission of the entire network without the need to configure each network device manually. This is the main advantage of the SDN-based VANET over the traditional VANET system. In the next step, the security of the entire system is considered. The SDN-based system provides a global overview of the entire network. The reason is that the SDN controller provides a global overview of the system, thus providing a security mechanism that is better

than in the traditional network. The centralized controller is used to collect information about the traffic flow in the entire network. In the traditional VANET system, devices are used to exchange large amounts of information in order to obtain network security [22]. Several security advantages are provided by the SDN-based VANET system, such as the detection of network-wide intrusion, the detection of malicious behavior of a RSU or vehicle, and the provision of network forensics as shown in Table 1.

4.1. Network-Wide Intrusion Detection. In network-wide intrusion detection, the SDN controller is used to run a fault detection system known as the intrusion detection system (IDS), which analyzes the traffic behavior that is collected by the RSU or the vehicles in the system to detect any malicious traffic. The SDN controller is connected to the RSU, and the SDN controller generates the policies and rules that are then applied to the RSU. The RSU is used to send these policies to all vehicles that are directly connected to it [23]. In the case of malicious activity, the information is first sent to the RSU and then to the controller, which responds to the threat. In some cases, the solution to these threats is defined on each node of the system. In contrast, the traditional network contains an IDS device that is usually installed in a certain part of the network, so its detection capacity is limited malicious activity within a small network region. Two operations are performed when the IDS on the SDN controller receives information from the entire network. The first operation is the misuse detection, in which a profile is built against various kinds of attacks. In the case of malicious activity, an interrupt is generated to stop this activity. The second operation is the detection of anomalies in which the network traffic flow is summarized based on common characteristics in the data packet that is moving from one node to another. In the case of a change in the network's behavior, a message is generated to provide detailed information about the change. Both mechanisms have advantages and disadvantages. For example, the anomaly detection strategy has the ability to detect a new attack, and it requires less knowledge about the malicious behavior. However, this strategy produces a false result when there is a minor change in the flow of traffic,

causing erroneous reports of errors or malicious behavior. Researchers are still working on the IDS framework to reduce the overhead caused by processing, increase the system's accuracy, and update the framework to support new routing protocols and policies for data transmission.

4.2. Detection of Malicious Behavior by RSU or Vehicle. The SDN controller not only helps in the detection of malicious data traffic but also helps in detecting malicious activities by the RSU or vehicle node. For example, the RSU and the vehicles sometimes cause the dropping of incoming packets, which creates a black hole. In a traditional network, the routing protocols are supposed to be responsible for this task, which is not a good strategy. To solve this issue in a traditional network, a probing technique is used. For example, the tracer route is used to display the available routes. However, detecting malicious behavior is easy in the SDN-based system because the RSU periodically sends detailed information about the number of packets received, forwarded, and dropped. This information is then used to detect the node or RSU that is causing the problem throughout the network. The SDN controller facilitates the detection of the faulty behavior of any RSU when they communicate with each other. The results of the forwarded and received packets from the RSU are sent to the controller, which easily detects the issue. When more than one device is involved in malicious activity, it is difficult to manage the functionality of the network because the controller has to detect a large amount of simultaneously, which causes loading on the processor of the SDN controller [24].

4.3. Network Forensics. Network forensics, which is a domain of digital forensics, is used to monitor the flow of data in the network. Information about the flow of traffic or data is gathered to detect any error. It is used to monitor the network traffic flow. If an attacker deletes or alters the configuration of any device, network-based evidence is used in the forensic analysis. This analysis determines the effects of an undetected attack on the system's settings. An effective mechanism was introduced to eliminate such attacks. It also identifies the host that is used in them.

4.4. Self-Healing Mechanisms. The SDN-based VANET also differs from the simple VANET system because it includes a self-healing mechanism. Rules are defined on every node or device involved in the entire network. These rules are developed by the control plane in the controller, which takes into account the limitations of the nodes. When a certain condition of the rule is satisfied, the rule provides a solution to the node for reacting under a certain condition. In the case of an attack, these rules are used to provide the node with a way to prevent the attack, such as dropping the packet that is defined as malicious by the rules [25]. Self-healing is the process of automatic recovery in the case of an attack on the system. The self-healing mechanism detects malicious activity in the network, diagnoses it, and provides the recovery mechanism to regain the original form of the system. Firewalls are used to detect faults in the system and provide an appropriate recovery mechanism. There is still

a large amount of work to be done, such as a swapping mechanism in the controller and an alternative forwarding technique in the OpenFlow system in the case of a fault in the system [26].

4.5. Path Selection. The SDN-based system facilitates decision making about optimal routing. In VANET, data traffic sometimes becomes unbalanced because of the shortest routing path. In some cases of video streaming, the node already consumes a large amount of bandwidth. If this situation occurs in the SDN-based VANET system, the SDN controller reroutes the traffic process to increase the network utility and to decrease the possibility of congestion.

4.6. Selection of Channel and Frequency. Wireless channels on different frequencies are available for data communication. The SDN system allows the SDN controller to select a channel that is appropriate for data transmission. The SDN controller dynamically selects the frequency that is the most beneficial for data transmission at a specific time. Emergency messages are sent by using a specific frequency channel.

5. Security Threats and Challenges Related to SDN Based VANET

In the previous section, the advantages of using the SDN-based VANET system were discussed. In this section, the challenges and security threats that are involved in the field of SDN-based VANETS are described. The challenges to security must be considered during the design of the system. First, the diverse types of attackers are described.

5.1. Attackers in SDN Based VANET. Three types of attackers may attack the system: (1) inside and outside attackers; (2) rational or malicious attackers; (3) active and passive attackers. Inside attackers are authenticated members of a network, whereas outsider attackers are intruders in the network and hence have a limited capacity to attack. Rational attackers attack the system for personal benefit. Malicious attackers aim to destroy the network without the goal of any personal benefit. Active attackers generate a false signal to make a cluster of data at any node, whereas passive attackers only sense the presence of the network.

When the communication system is independent of the SDN controller, many challenges must be considered in designing the system [27].

5.2. Attacks on Security Requirements

5.2.1. Hijacking of Session. The authentication process is conducted when the session starts. After the connection has been established, it is easy to hijack the session. In this type of attack, the attackers obtain detailed information about the session and then become the central node between the nodes.

5.2.2. Identity Revealing. In most cases, the driver is the owner of the vehicle and may use personal information in the authentication process. Therefore, it is easy for attackers to enter the system.

5.2.3. Location Tracking. The location of the vehicle may be used to track it and obtain information about the driver and the passengers.

5.2.4. Eavesdropping. This type of attack affects the network layer, which then allows access to confidential data.

5.2.5. Denial of Service. This is the most prominent attack. The attackers prevent a specific node from accessing services. This attack is carried out in two ways:

- (a) Jamming: in this technique, the attacker accesses information about the frequency at which signals are received by the receiver and then transmits a signal at the same frequency with the intention to block the legitimate signals.
- (b) Distributed DOS attack: in this type of attack, multiple attackers attack a specific node to prevent it from accessing services.

5.3. Challenges Related to SDN-Based VANET. These challenges include real-time constraints. First, messages related to safety are delivered within the maximum transmission delay of 100 ms. A fast cryptographic algorithm must be used to achieve real-time constraints. Authentication must be done in real time. The second challenge concerns data consistency liability, in which the authentication node may perform malicious activities that may cause accidents. Hence, a mechanism must be developed to achieve consistency. To remove such inconsistencies, the correlation among the received data is determined. The third challenge is the low tolerance for error. Many protocols for SDN-based VANET have been developed based on possibilities. Action must be taken by the algorithm used in the protocols within a brief period of time. A slight problem or delay in the algorithm may cause harm. The fourth challenge is key distribution. The security mechanisms used by SDN-based VANET depend upon the key. The message is encrypted and decrypted on the receiver side by using the key. The key must be protected to ensure the security of the data. The distribution of the key is the main challenge in designing security protocols.

The SDN-based system helps to provide a solution to the issues in the traditional VANET system. However, many problems occur in implementing the SDN-based VANET, such as dynamic traffic control and high bandwidth for communication. In the following section, the most prominent challenges that are faced in implementing the SDN-based VANET system are described [28].

5.3.1. SDN Communication with Traditional Networks. The SDN-based system has policies for data communication, but the traditional network is completely different. An issue arises in deploying the SDN-based system in the presence of an existing network. Many traditional networks are used to support complex applications. Thus, the SDN-based VANET should be installed carefully on a traditional network in order that all operations will be performed accurately. To ensure the operation of the SDN-based system, it is compulsory for it to interact with a traditional network.

The potential solution to this problem is to develop a new routing protocol that is used to specify the functionality of the SDN-based system. A uniform acceptance level is made based on the result. An internet engineering task force is currently working to develop a standard for the different mechanisms in the SDN-based system [29].

5.3.2. Security Issues. Most research has focused on the development of the SDN-based VANET system, and little attention has been given to its security. Without ensuring the security of the system, it is impossible to obtain its full functionality. The system is considered open to attackers. It is easy for them to attack the system if the security of the SDN-based VANET system is neglected. An effective security mechanism is required to run the system efficiently and to protect it from external threats [30].

5.3.3. Availability of Services. When a device fails in the traditional network, the traffic is routed through an alternative path and processed by another network device to maintain the availability of the service. In the centralized system, if it fails while communicating with another device, the overall functionality of the entire network will be affected. To resolve this issue, a standby controller may also be used to provide a backup for the system. Another solution to this problem is the use of a distributed controller, which helps to control the functionality of the network by implementing a load balancing process. The failure of a data traffic path may also affect the system's output, so the SDN must support multipaths to redirect the traffic in case of a failure in the network [31].

5.3.4. Issues Related to Scalability. In a traditional network, configuration is simple because the same settings are required by all the routers and switches. However, the configuration of a dynamic network is much more difficult because it requires the reconfiguration of the existing network. In the case of an SDN-based VANET system, this process is carried out with the help of the control plane. The data traffic is directly proportional to the traffic on the road. That is, as the traffic on the road increases, the rate of communication between the vehicles and the RSU increases. When there is highly dense traffic on the road, it is difficult for the controller to control the overall flow of the network. The issue of scalability arises because of the increased traffic flow. This process requires the addition of new entries to the flow tables, which helps to maintain the flow of the data traffic. In this issue, communication takes place between all the devices directly connected to the controller, such as the RSU and the vehicle nodes, which places an extra processing load on the controller. Although different solutions to this problem have been proposed to decrease the extra processing load on the controller, the best solution is to use a multicore system within the SDN controller, which balances the overall load [32].

5.3.5. Challenges Related to Performance. The use of a centralized architecture results in the slowdown of communication throughout the network. To maximize the output of the system, a hybrid controller may be used. An optimal

algorithm should be designed to assist in carrying out the responsibility of the controller. It should also help the system use its resources efficiently.

5.4. Attacks on Different Planes. In this section, the attacks related to the SDN controller are described, and the attacks and threats are divided into distinct categories based on which part of SDN controller is affected by the attacks.

5.4.1. Attacks on the Forwarding Plane. In this section, we will discuss the threats that are faced by the forwarding plane in the SDN system.

DOS in Node. The nodes, which are the vehicles in the network, have limited storage capacity. However, the rules defined by the controller are in the storage area of the vehicle nodes. Another mechanism, reactive caching, is used in the modern SDN-based system. When a node does not find a path for the flow of its incoming packets, the packet is temporarily stored in the node buffer, and a query is sent to the RSU to ask the controller about the missing rule. When the node receives the rule, it decides accordingly. The mechanism of reactive caching also creates a suspect for the DOS attack in which the attacker sends a large amount of data that belong to different rules for the vehicle. In response, the vehicle node is unable to decide, and it requires help from the RSU. The node then starts storing the packets in the cache, but when the cache storage limit is reached, some packets may be dropped. There are several solutions to this problem. The main solution is that the node's memory contains as many rules as possible. This solution is effective because it reduces the delay in decision making, and it helps increase the overall efficiency of the system [33].

5.4.2. Attacks on the Control Plane. The attacks on the control plane of SDN-based VANET system are elaborated in this section.

DDOS Attack. The SDN-based system is controlled by the control plane. However, a distributed denial of service (DDOS) may attack the control plane, in which multiple vehicles in the network simultaneously send hundreds or thousands of packets to one or more vehicles. Because all rules are not available on the switch, multiple queries are generated and sent to the controller, which then must utilize immense processing power, thus causing a delay in the result or the dropping of queries. The best solution to this type of attack is the use of multiple controllers in the area where the traffic density is high. Instead of using a single controller, this solution can be used to manage the overall functionality of the system. Nevertheless, the programming of the entire network is done with the help of the single controller system, which may be defined as the main SDN controller. All controllers are logically linked with a single link.

5.4.3. Fake Controller Attacks. Attackers are able to access the SDN controller by hosting a fake controller in the network, which is used to run policies for the attackers' benefit. Fake controllers sometimes force the RSU to stop communication

by dropping data packets or injecting the RSU and using it as a base node to attack the entire system. Controller replicas are sometimes used to address this problem, but this solution is not effective. In gaining access to a single controller, it is easy to access another controller because the same security policies are defined in a range of controllers. To solve this problem, every controller must have rules and policies that differ from those in other controllers.

5.4.4. OpenFlow's Robustness to Security Threats. The OpenFlow mechanism is the first data communication mechanism released for the SDN system. After multiple revisions in the definition of OpenFlow, it is currently the most widely used mechanism in the SDN system. OpenFlow is not concerned with the rules that are applied to the nodes. The OpenFlow system cannot handle malicious activity in an RSU that is directly connected to the controller. It simply redirects the activity to the control layer, where the issues are resolved. Moreover, because OpenFlow does not encrypt its communication, the security of the system is lessened [25].

6. Evolution of SDN Controller Architecture for VANETs

The SDN-based VANET architecture supports both the centralized and distributed control models. Both models have specific infrastructure and requirements. In this section, the SDN-based model is discussed with respect to its advantages and disadvantages. A new hybrid approach that contains benefits of both central and distributed control model is described.

6.1. Centralized Controller Model. The centralized model consists of a single controller that is used to manage the entire network. This centralized SDN is supported by the OpenFlow protocol, which is used by the centralized controller to control and manage the entire network. Hence, because a single controller is used it must have a global vision of the entire network. The controller communicates with the device that is directly connected to the network to detect the fault and attacks that occur on the entire network. These directly connected devices forward information to the controller. By using the single controller, it is easy to manage the functionality of the entire network. However, a single controller has several limitations. It needs to update the whole network more frequently than the traditional network does because when the flow changes, the flow table must be updated in order to maintain efficiency. This process produces a high overload because the producers need to be carried out by the controller, and this producer increases the processing cost of the controller. An example is classifying flows with different priorities into multiple categories. Increasing the functionality in a single node requires higher amounts of power in processing, data storage capacity, and throughput to deliver the data. Over time, new rules and flows are introduced to facilitate communication, but the memory of the controller must be expanded repeatedly [34].

The third issue in the centralized model is that regardless of where the first pack of the new flow is introduced, it is first

forwarded to the centralized controller for inspection. The controller determines the path for the flow and extends this information to all the devices for the entire network. The ram used by the controller to define the rules can be overloaded by large amount of data required to be processed. In that situation, any failure in the controller would cause the failure of the entire network. Over time, the SDN network system becomes increasingly complex because of new requirements of the role of traffic. They are designed to support variable communication with the addition of security, load balancing, and firewalls [35]. Different services are designed to coordinate with the control plane to achieve their goal. In using this inconsistent routing topology, it is difficult for the controller to achieve optimal performance.

6.2. Distributed Controller Model. The distributed SDN model was introduced to eliminate the failures and limitations of the single SDN controller. The distributed SDN model is used to balance the load among different controllers and multicore systems in the controller. This model used to handle the entire network effectively. The distributed controller is used to share large amounts of data, thereby ensuring the consistency of communication. The comparison of the distributed SDN model with the central control model shows that the distributed systems are more responsive, faster, and more efficient with respect to large global network areas. However, despite the benefits of the distributed SDN model, many challenges must be overcome before the distributed controller can be implemented. The mapping of the control plane and forwarding plane must be configured automatically instead of manually. The controller must have a wide view of the entire network to support the system. In most cases, it is difficult for every controller to have wide access to networks. These controllers use local algorithms to develop coordination between different controllers. Therefore, an algorithm or procedure is required to synchronize the entire network and provide a global picture of it.

6.3. Hybrid Controller Architecture. The hybrid control architecture is a new approach to addressing the limitations of central and distributed SDN controller systems. This hybrid system was designed to support the architectures of both central and distributed systems. The logical system used in the hybrid model is the same as that used in the central controller. However, it uses the data communication pattern that is used in the distributed controller. The hybrid SDN model supports the simple control management pattern of a single controller, and it has the flexibility of the distributed model. The hybrid SDN model helps the controller use resources efficiently to increase the performance of the network. The hybrid model also allows policies to heighten the security of the entire network. Furthermore, it allows the updating of the system without the need to change the current network settings [10].

7. Application of SDN-Based VANET System

The communication between vehicles provides various types of information that is used to provide a safe journey facility to the travelers. It helps the driver to take a better decision

FIGURE 10: Applications of SDN-based VANETs.

before any kind of mishap happens. By integrating, different kinds of sensors provide the facility to the vehicle to collect and process information that is further used to increase the safety of the vehicle and the passengers inside [36]. Application services of VANETs are categorized into safety and efficiency as shown in Figure 10. In designing and implementing VANET systems, there are many difficulties including connectivity, privacy, security, routing, and quality of services.

7.1. Comfort/Entertainment Applications. The aim of this category of application is to improve the the comfort level of driver and passengers. They are used to provide the information about the weather, traffic, and the location of the nearest petrol station, hotel, and restaurants. Passengers can assess the internet and are able to send and receive messages.

7.2. Safety Applications. As the name indicates, these types of applications are used to improve the safety of vehicles and passengers. The main purpose of the application is to save the travelers from any harm and to provide a clear environment for traveling purpose. This application gathers information from the sensors or from other vehicles traveling around or maybe both. The main functionality of the safety applications depends upon the numbers of sensors it uses to collect information and the software that is used to process the information and broadcast a safety message to other vehicles. For the broadcast purpose, it uses a wireless communication medium in vehicles [1].

7.2.1. Intersection Collision Avoidance. It is used to help the vehicle or driver for the decision making while crossing an intersection; this system uses the communication medium of the V2I vehicle to infrastructure communication. A RU gathers data from vehicles moving near to it and processes that data if there is any probability of warning or any kind of accident; a warning message is sent to the vehicles that are

near to the warning area so that they can take right decision to avoid it. It has numerous applications in collision avoidance.

7.2.2. Warning in Case of Violating Stop Signal. These applications are used to send alert messages to the drivers in case of violating the limit of a stop sign. The system will send a maximum speed required alert to prevent the need of emergency breaking. This will lead to prevention of any hazardous situation.

7.2.3. Stop Movement Sign Assistant. These applications are used to warn the drivers not to pass the intersection or otherwise any dangerous situation may occur. It involves the communication between the vehicle sensors and the roadside unit. This application informs the driver that other vehicles are nearer to the intersection, so he has to wait for some time. As the other vehicle crossed the intersection the vehicle receives a green signal to cross the intersection.

7.2.4. Left Turn Assistant. Left turn assistant is used to help the driver to make the the decision to turn left in case of safe situation. It gathers the information about the left side of the road by sending a request message to the roadside unit; the RSU will reply back a message when it is safe to cross the road.

7.2.5. Warning in Case of Blind Merge. This feature of the application is used to warn the drivers at the merge point of the road where visibility is not good. It is used to collect the data at the intersection point and generate a result if it is unsafe and warns the driver.

7.2.6. Pedestrian Intersection Information. This feature of the application is used to send a stop sign signal to the driver if a pedestrian is crossing the road meanwhile. The RSU collects the information from the vehicle which is near to the pedestrian crossing line and checks if the signal is of a stop sign and the vehicle is still moving; it generates a warning message and alert the driver to stop the car because a pedestrian is crossing the road.

7.3. Public Safety Application. In the case of any accident, this feature of application sends a message to the emergency teams about the accident and provide them the details of the vehicle and the location. As well-known, the emergency vehicles waste their time while reaching to the destination point because they did not know the shortcut path or the emergency team choose a way already congested by other vehicles. Such a delay may cause the death of the people. This application is essentially useful at that time. There are many other similar applications in this category as follows.

7.3.1. Emergency Vehicle Is Approaching Warning. The aim for this application is to warn a vehicle in the case if any emergency vehicle is approaching towards itself. This system involves V2V communication. It warns the vehicle on the same way as the emergency vehicles drive to clear that road.

7.3.2. Emergency Vehicle Signal Preemption. This application is used intentionally to turn the signal light of the specific

roadside to green light and to turn the other traffic signal to red. This feature of application helps to decrease the waiting time of the emergency vehicle at the red light signal.

7.3.3. Post-Crash Warning. This application feature is used to send a warning message to all the vehicles in the foggy weather because there may be a chance of an accident due to heavy fog. This application will send a message containing the maximum speed limit and the lane suitable for driving the car. This system uses both V2I and V2V communications.

7.4. Sign Extension. This application is used to make the driver to drive attentively and to check the signs that are placed on the roadside. These applications are used to provide a message with the range of 100 m–500 m; these applications are further classified into different categories.

7.4.1. In-Vehicle Signage. This application is used to send an alert message to the vehicle that is approaching towards the schools and hospital zone. This application involves the communication between the vehicle and the RSU unit.

7.4.2. Curve Road Warning. This application is used to warn the vehicle that is approaching towards the curve path above the speed limit. It involves the communication between a RSU and a vehicle.

7.4.3. Warning of Wrong-Way Driving. This application is used to alert the vehicle if it is trying to enter the wrong line or the wrong direction. It sends an alert message to a vehicle moving in a wrong direction. This system also warns the other vehicles that are near the wrong direction, so that the vehicle can prevent the vehicles from any kind of accident.

7.4.4. Work Zone Warning. This system will provide an alert message to the vehicles that are near the working zone to limit their vehicle speed. This the system uses V2I communication.

7.4.5. In-Vehicle Amber Alert. Police delivers this alert message announcing that a specific car is involved in some crime scene. This amber alert message is sent to the entire vehicles except for the vehicle that is involved in the crime scene.

7.5. Getting Useful Data from Other Vehicles. This type of application depends on V2V communication or V2I communication or may be both to form the functionality of the applications.

7.5.1. Cooperative forward Collision Warning. This application feature is used to assist the vehicles in avoiding the accident with the other vehicles that are traveling ahead. This application feature involves the V2V type of communication. In the result form, it provides the danger level ahead.

7.5.2. Emergency Brake Lights. This application feature is used to alert the vehicles within a specific area if they may require some sudden breaks due to poor visibility and the breaks lights of other vehicles are not such visible. It uses V2V communications.

7.5.3. Road Condition Warning. While the sensors are used to collect the information about the road situation, OBUs process the information from the sensors and send analysis results to RSU. RSUs send alert messages to the entire vehicles that are moving towards a poor-conditioned area, to make vehicles slow down. This application protects the vehicles from emergency breaks from accidents that are being caused by applying the emergency breaks of the vehicle.

7.5.4. Lane Change Warning. This application feature is used to alert the driver that it is unsafe to change the lane as the distance between the current vehicle and the vehicle moving on the other lane is so narrow. This type of system uses data of the vehicles that are moving around. V2V communication is involved in this type of warning application.

7.5.5. Highway Merge Assistant. This application works when a vehicle is trying to enter the highway. When there is a vehicle near the blind spot, then the system sends a message containing the direction of other vehicles to prevent the accidents.

7.5.6. Visibility Enhancer. This feature of the application is used to warn the drivers about the bad weather condition such as fog, rain, and snow. As a result, it leads to helping him in safe driving.

7.5.7. Train on Railway Track Ahead. This feature of application is used to inform the driver to stop the car due to the movement of the train ahead. It involves the communication between the vehicle and a RSU. The RSU is used to send an alert message to all the vehicles near its specific region.

8. Conclusion

The main objective of this paper is to provide a comprehensive survey on SDN-based VANET systems to help researchers to efficiently understand this area. In this paper, we elaborate not only the architecture, components, and operations of the SDN-based VANETs, but also how the SDN-based VANETs enable providing better communications than the simple conventional VANETs. Furthermore, SDN controller can control the security threats. Throughout this survey, it is ensured that this modern vehicle system helps a lot in controlling and managing the entire vehicular networks that were not possible before.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (no. 2018R1A2B6002399) and in part by Basic Science Research Program through the National Research

Foundation of Korea (NRF) funded by the Ministry of Education (2015R1D1A1A01059186).

References

- [1] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular Ad Hoc network," *Journal of Network and Computer Applications*, vol. 37, no. 1, pp. 380–392, 2014.
- [2] B. T. Sharef, R. A. Alsaqour, and M. Ismail, "Vehicular communication ad hoc routing protocols: a survey," *Journal of Network and Computer Applications*, vol. 40, no. 1, pp. 363–396, 2014.
- [3] K. Tanuja, T. M. Sushma, M. MBharathi, and K. H. Arun, "A survey on vanet technologies," *International Journal of Computer Applications*, vol. 121, no. 18, pp. 1–9, 2015.
- [4] M. Ealias and R. N. Gaur, "A survey on different routing models in cognitive radio ad-hoc network," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 03, no. 12, pp. 13741–13748, 2014.
- [5] H. Hartenstein and K. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Communications Magazine*, vol. 46, no. 6, pp. 164–171, 2008.
- [6] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (VANETS): status, results, and challenges," *Telecommunication Systems*, vol. 50, no. 4, pp. 217–241, 2012.
- [7] B. A. A. Nunes, M. Mendonca, X. N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: past, present, and future of programmable networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, 2014.
- [8] I. Ku, Y. Lu, M. Gerla, F. Ongaro, R. L. Gomes, and E. Cerqueira, "Towards software-defined VANET: architecture and services," in *Proceedings of the 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET '14)*, pp. 103–110, IEEE, Piran, Slovenia, June 2014.
- [9] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: a comprehensive survey," *Networking and Internet Architecture*, vol. 103, no. 1, pp. 14–76, 2014.
- [10] H. Farhady, H. Lee, and A. Nakao, "Software-Defined Networking: a survey," *Computer Networks*, vol. 81, pp. 79–95, 2015.
- [11] M. Mousa, A. M. Bahaa-Eldin, and M. Sobh, "Software Defined Networking concepts and challenges," in *Proceedings of the 2016 11th International Conference on Computer Engineering & Systems (ICCES '16)*, pp. 79–90, Cairo, Egypt, December 2016.
- [12] N. Feamster, J. Rexford, and E. Zegura, "The road to SDN: an intellectual history of programmable networks," *Computer Communication Review*, vol. 44, no. 2, pp. 87–98, 2014.
- [13] A. Hakiri, A. Gokhale, P. Berthou, D. C. Schmidt, and T. Gayraud, "Software-Defined Networking: Challenges and research opportunities for Future Internet," *Computer Networks*, vol. 75, pp. 453–471, 2014.
- [14] F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and openflow: from concept to implementation," *Abstract—Software-Defined IEEE Communication Surveys & Tutorials*, vol. 16, no. 4, 2014.
- [15] B. Pfaff, J. Pettit, T. Koponen et al., "The design and implementation of open vSwitch," in *Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI '15)*, pp. 117–130, May 2015.

- [16] B. Pfaff and B. Davie, "The Open vSwitch Database Management Protocol," RFC Editor RFC7047, 2013.
- [17] M. Mousa, A. M. Bahaa-ELDin, and M. Sobh, "Software Defined Networking concepts and challenges," in *Proceedings of the 11th International Conference on Computer Engineering and Systems (ICCES '16)*, pp. 79–90, December 2016.
- [18] T. Koponen, M. Casado, N. Gude et al., "Onix: a distributed control platform for large-scale production networks," *The Ocular Surface Disease Index*, vol. 10, 2010.
- [19] A. Tootoonchian and Y. Ganjali, "HyperFlow: a distributed control plane for OpenFlow," in *Proceedings of the 2010 internet network management conference on Research on enterprise networking*, 2010.
- [20] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: rapid prototyping for software-defined networks," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks (Hotnets '10)*, 19:6, 19:1 pages, ACM, October 2010.
- [21] M. Gupta, J. Sommers, and P. Barford, "Fast, accurate simulation for SDN prototyping," in *Proceedings of the 2013 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN '13)*, pp. 31–36, August 2013.
- [22] P. Baskett, Y. Shang, W. Zeng, and B. Guttersohn, "SDNAN: software-defined networking in ad hoc networks of smart-phones," in *Proceedings of the 2013 IEEE 10th Consumer Communications and Networking Conference (CCNC '13)*, pp. 861–862, January 2013.
- [23] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, "Software-defined networking security: pros and cons," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 73–79, 2015.
- [24] M. A. Sayeed, M. A. Sayeed, and S. Saxena, "Intrusion detection system based on Software Defined Network firewall," in *Proceedings of the 1st International Conference on Next Generation Computing Technologies (NGCT '15)*, pp. 379–382, September 2015.
- [25] P. Zhang, H. Wang, C. Hu, and C. Lin, "On denial of service attacks in software defined networks," *IEEE Network*, vol. 30, no. 6, pp. 28–33, 2016.
- [26] J. Vilchez, I. Yahia, and N. Crespi, "Self-healing Mechanisms for Software Defined Networks," 2014.
- [27] S. Tomovic, M. Radonjic, M. Pejanovic-Djurisic, and I. Radusinovic, "Software-defined wireless sensor networks: opportunities and challenges".
- [28] S. Sezer, S. Scott-Hayward, P. Chouhan et al., "Are we ready for SDN? Implementation challenges for software-defined networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 36–43, 2013.
- [29] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114–119, 2013.
- [30] D. Kreutz, F. Ramos, and P. Verissimo, "Towards secure and dependable software-defined networks," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN '13)*, pp. 55–60, Hong Kong, China, August 2013.
- [31] A. Tootoonchian and Y. Ganjali, "HyperFlow: a distributed control plane for openflow," in *Proceedings of the 2010 Internet Network Management Conference on Research on Enterprise Networking*, p. 3, 2010.
- [32] A. Voellmy and J. Wang, "Scalable software defined network controllers," in *Proceedings of the ACM SIGCOMM 2012 Conference Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '12)*, pp. 289–290, August 2012.
- [33] Q. Yan and F. R. Yu, "Distributed denial of service attacks in software-defined networking with cloud computing," *IEEE Communications Magazine*, vol. 53, no. 4, pp. 52–59, 2015.
- [34] A. Hakiri, A. Gokhale, P. Berthou, D. C. Schmidt, and T. Gayraud, "Software-defined networking: challenges and research opportunities for future internet," *Computer Networks*, pp. 453–471, 2014.
- [35] M. Dixit, R. Kumar, and A. K. Sagar, "VANET: Architectures, research issues, routing protocols, and its applications," in *Proceedings of the 2016 IEEE International Conference on Computing, Communication and Automation (ICCCA '16)*, pp. 555–561, April 2016.
- [36] A. Ghosh, V. V. Paranthaman, G. Mapp, O. Gemikonakli, and J. Loo, "Enabling seamless V2I communications: toward developing cooperative automotive applications in VANET systems," *IEEE Communications Magazine*, vol. 530, no. 12, pp. 80–86, 2015.