Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 5603518, 15 pages
https://doi.org/10.1155/2018/5603518

Review Article

Challenges of Future VANET and Cloud-Based Approaches

Rakesh Shrestha, Rojeena Bajracharya, and Seung Yeob Nam

Department of Information and Communication Engineering, Yeungnam University, 280 Daehak-Ro, Gyeongsan-si, Gyeongsangbuk-do 712-749, Republic of Korea

Correspondence should be addressed to Seung Yeob Nam; synam@ynu.ac.kr

Received 26 September 2017; Revised 25 February 2018; Accepted 25 March 2018; Published 2 May 2018

Academic Editor: Orhan Gazi

Vehicular ad hoc networks (VANETs) have been studied intensively due to their wide variety of applications and services, such as passenger safety, enhanced traffic efficiency, and infotainment. With the evolution of technology and sudden growth in the number of smart vehicles, traditional VANETs face several technical challenges in deployment and management due to less flexibility, scalability, poor connectivity, and inadequate intelligence. Cloud computing is considered a way to satisfy these requirements in VANETs. However, next-generation VANETs will have special requirements of autonomous vehicles with high mobility, low latency, real-time applications, and connectivity, which may not be resolved by conventional cloud computing. Hence, merging of fog computing with the conventional cloud for VANETs is discussed as a potential solution for several issues in current and future VANETs. In addition, fog computing can be enhanced by integrating Software-Defined Network (SDN), which provides flexibility, programmability, and global knowledge of the network. We present two example scenarios for timely dissemination of safety messages in future VANETs based on fog and a combination of fog and SDN. We also explained the issues that need to be resolved for the deployment of three different cloud-based approaches.

1. Introduction

Vehicular ad hoc networks (VANETs) have gained popularity in recent years. Traffic accidents, road congestion, fuel consumption, and environmental pollution due to the large number of vehicles have become serious global issues. Traffic incidents are persistent problems in both developed and developing countries, which result in huge loss of life and property. In order to overcome these issues and make the journey safer, efficient, hassle-free, and entertaining, Intelligent Transportation Systems (ITS) introduced VANETs to create a safer infrastructure for road transportation [1, 2]. VANETs focus on road safety and efficient traffic management for public roads, while offering comfort and entertainment for drivers and passengers throughout their journeys [3]. Vehicular communication in VANETs can be achieved by exchanging information using Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications. Vehicles communicate with other vehicles through On-Board Units (OBUs) forming Mobile Ad Hoc Networks (MANETs) that allow wireless communication in a completely distributed manner while they can communicate with Roadside Units (RSUs) in an infrastructure mode. The Wireless Access in Vehicular Environment (WAVE) protocol is based on IEEE 802.11p standard and provides basic radio standard for Dedicated Short-Range Communications (DSRC) operating in 5.9 GHz frequency band [4].

There are mainly two types of applications used in VANETs, safety, and nonsafety applications. Safety applications in VANETs are used to send safety messages, for example, various warning messages that assist vehicles on the road so that proper actions can be taken to prevent accidents and save people from hazardous situations. Safety messages include events such as road accident reports, traffic jam notifications, road construction reports, and emergency vehicle warnings [5]. These types of safety applications require low latency and high reliability. On the other hand, nonsafety applications provide an efficient and comfortable driving experience. Nonsafety applications are classified into two categories: traffic management and infotainment. Traffic management applications are used to improve traffic flow and resolve congestion on the road. Infotainment applications are used for information and entertainment purposes, providing Internet access to passengers, such as data storage, video streaming, and video calling. These types of applications do not require high reliability and low latency, compared to the safety applications. In VANETs, some critical event information (e.g., accident reports) must be disseminated quickly and reliably. Even though VANETs are feasible for disseminating event information, it is still a challenge to disseminate critical messages timely and reliably to a targeted area under a dynamic vehicular environment. This is due to the limited transmission range of DSRC, as well as the Contention-based Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) scheme in the IEEE 802.11p protocol [6]. The critical messages face delays due to MAC contention, which is not suitable for VANETs. Failure in timely and accurate dissemination of such time-critical information might lead to collateral damage to neighboring vehicles. Recently, researchers have studied the feasibility of cellular networks for supporting VANET services while providing wide coverage and high data rate services to vehicular users [7]. However, a single wireless communications technology, such as either DSRC or a cellular network, may not fully support ITS services in an environment with heavy loads, high mobility, and a dynamic network topology. Thus, by integrating cellular networks and IEEE 802.11p, a heterogeneous vehicular ad hoc network could be a possible platform that can meet several demanding communication requirements. It can deal with seamless data connectivity between spatially separated vehicles in VANETs [8].

The traditional centralized VANET technology may not be efficient in handling massive amounts of traffic data generated by smart vehicles such as video and sensor data. In order to collect and process large amount of instantaneous traffic information, additional servers are required in distributed areas. VANETs using cloud computing might be an appropriate solution for these types of situations. Recently, cloud computing has been adopted by various mobile devices to handle complex computations that can be hardly accomplished locally [9]. A lot of research has been done in VANETs integrated with cloud computing to extend the capabilities of VANETs even further [10–12]. In general, uploading to and downloading from the cloud by the vehicles consume time and energy [10]. As the density of the vehicles increases in urban areas, the existing cloud computing paradigm can barely satisfy the requirements of location awareness, mobility support, and latency.

A solution based on edge cloud computing has been proposed with the concept of fog computing to overcome issues between the vehicular nodes and the main cloud [13]. The transmission delays in messages can be satisfied by fog computing. In addition, the safety messages can be sent to desired locations that are in different geographic areas with the help of geo-distributed fog servers with low latency. Fog computing extends the cloud computing paradigm to the edge of the network [14]. Fog computing is a lightweight version of cloud computing at the proximity of the vehicular nodes, with functions similar to the cloud. Fog computing has a multitiered architecture, with vehicular nodes at the edge of the network. The fog platform is located between vehicular nodes and the datacenters of the conventional cloud environments. The main motivation for implementing fog computing in VANETs is to leverage the advantages of fog computing in the distributed cloud environment. Fog computing provides very low latency between the vehicles and the cloud. In addition, fog computing can support vehicles with high mobility. Along with the evolution of 5G technology, fog computing integrated with Software-Defined Network (SDN) plays an important role in enhancing the performance of the VANETs [15]. The centralized and systematic paradigm of SDN provides scalability, flexibility, and programmability to VANET. The basic feature of SDN is separation of the control plane for network control and the data plane for the forwarding function [16]. By introducing the SDN controller along with fog computing, the system can manage geographically distributed fog devices and can manage heterogeneous networks by providing programmability, flexibility, and global information about the network. Due to the open, reconfigurable interface and flexibility of SDN, it provides intelligence that can be applied in VANETs efficiently. On the other hand, the decentralized paradigm of fog computing reduces latency and optimizes resource utilization in future VANETs.

The rest of this paper is organized as follows. In Section 2, we investigate several key challenges and requirements of future VANETs. In Section 3, we introduce a brief overview of emerging technologies that act as a key enabler for future VANETs. We present a comparative study on edge based cloud computing and SDN in VANETs. Section 4 presents two examples of safety message dissemination in fog integrated with SDN in future VANETs. Section 5 discusses the deployment issues of cloud-based approaches in terms of networking and computing infrastructure followed by conclusion in Section 6. We summarize the definitions of acronyms used in this paper in "Summary of acronyms" for reference.

2. Key Challenges and Requirements of Future VANETs

VANETs are one of the fundamental technologies for efficient, safe, informative, and entertaining transportation system [17]. These days, people spend a significant amount of their time in vehicles. In order to make that time more secure, efficient, pleasant, pollution-free, and economical, smart vehicles based on VANETs have emerged. In this context, higher safety can be provided by exchanging critical events reliably. Improved efficiency is achieved by reducing traffic jams and pollution and making travel time more predictable. Furthermore, VANETs can be connected to the Internet to make the journey more entertaining by offering files to download and access to social networks [18]. VANETs use two types of messages: beacon messages and safety messages. Vehicles use beacons to periodically broadcast and advertise status information to neighboring vehicles at intervals of 100 ms. The sender reports its velocity, location, and pseudo-ID to neighboring vehicles via beacon messages [19]. On the other hand, safety messages assist vehicles on the road by delivering emergency information so that proper action can be taken to prevent accidents and to save people from life-threatening situations. Vehicles broadcast a WAVE Short Message (WSM) to neighboring vehicles when they encounter events on the road [5]. The message payload includes information about the vehicle's position, the messages sending time, the vehicle's location, and nearby road events, and so forth [20, 21]. Each vehicle gathers information about the neighboring vehicles within communication range. We consider heterogeneous VANETs where the OBUs of the vehicles are equipped with IEEE 802.11p and Long-Term Evolution (LTE) cellular technologies.

VANETs use conventional cloud for information storage, retrieval, management, complicated computation and global networking [22]. The cloud service can be classified into three basic distribution models in the form of layers: software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS). Hence, consumers can rent processing, storage, and network resources on which they can deploy and run required application software and system software. In this case, customers have virtually unlimited resources based only on their budget [23]. Examples of cloud computing are EC2 provided by Amazon, Azure from Microsoft, and Google Apps by Google.

VANETs use RSUs as gateways that provide a virtualization layer to connect to cloud services while on the move. The vehicular nodes use cloud services to obtain traffic and multimedia information. This is a two-tiered architecture where vehicles are at the edge of the network, and the cloud is at the center [23]. A VANET using the cloud can have macrocontrol of the geographic position of all vehicles, obtains instantaneous traffic flows, and determines the targeted area and desired recipients of safety messages [24, 25].

Even though VANETs use the cloud to solve existing problems, there are still some fundamental challenges that need to be resolved along with the growth in the number of smart vehicles. Based on the rapid progress and development trend of VANETs, some key challenges and requirements of future VANETs are identified. Future VANETs and its applications will go beyond the current trend and integrate with new emerging technologies, which introduce new functionalities. Some of the key challenges of the future VANETs are as follows:

(i) Intermittent connectivity: the control and management of network connection among vehicles and infrastructure is a key challenge. The intermittent connections due to the high mobility of vehicles or high packet loss in vehicular networks must be avoided.

(ii) High mobility and location awareness: future VANETs require high mobility and location awareness of the vehicles participating in communication. Each vehicle should have the correct position of other vehicles in the network to cope with an emergency situation.

(iii) Heterogeneous vehicle management: in the future, there will be a large number of heterogeneous smart vehicles. The management of heterogeneous vehicles and their sporadic connections is another challenge of future VANETs.

(iv) Security: there is always a risk to the privacy of user's data content and location. The vehicles communicating within the infrastructure should allow users to decide what information should be exchanged and what information should be kept private. Privacy can be assured by examining sensitive data locally, instead of sending it to the cloud for analysis.

(v) Support of network intelligence: one of the challenges of future VANETs is that it needs to support network intelligence. In future VANETs, there will be a large number of sensors installed in vehicles, and the edge cloud collects and preprocesses the collected data before sharing them with other parts of the network, for example, conventional cloud servers.

Considering those challenges, the major requirements of future VANETs can be summarized as follows:

(i) Low latency and real-time application: low latency is the fundamental requirement in future VANETs regarding real-time applications. Future VANETs should support real-time applications, like safety messages with very low latency.

(ii) High bandwidth: in future, infotainment and comfort applications such as high quality video streaming will be in high demand. In addition, traffic applications such as 3D maps and navigation systems require frequent automatic updates.

(iii) Connectivity: to meet the high communication requirements, future VANETs necessitate seamless connectivity between connected vehicles. Connected and driverless vehicles should maintain continuous and highly reliable communication between vehicles and fog devices. It should be able to avoid transmission failures in the communication system.

3. Emerging Technologies for Future VANET

3.1. Edge Cloud Computing. In future, there will be a massive number of smart devices that will be a part of the Internet of Things (IoT) and future VANETs. Those devices need to communicate with each other, for example, intelligent traffic light systems for autonomous vehicles and Vehicles-to-Everything (V2X), including other home-and-office smart appliances. The edge cloud computing (ECC) framework interconnects edge devices of the IoT and the conventional cloud computing system. The edge cloud consists of virtualized micro datacenters distributed in different geographic locations. The actual distribution of edge clouds depends on the customers' needs and preferences, economic circumstances, latency requirements of the network, and so on. The edge cloud provides typical services in close proximity (within a single hop) to end users, such as vehicles, mobile devices, and IoT devices. It has flexible interface technology providing high bandwidth and low latency. The IaaS in ECC may satisfy on-demand migration of application code between edge clouds during runtime to cope with mobility of vehicles and devices. By using an edge cloud, the computation, storage, and processing loads are handled in a distributed manner, instead of backhauling every bit of traffic into main datacenter. The IaaS in the edge cloud can host third-party application that helps offload computations from vehicles to the network. In

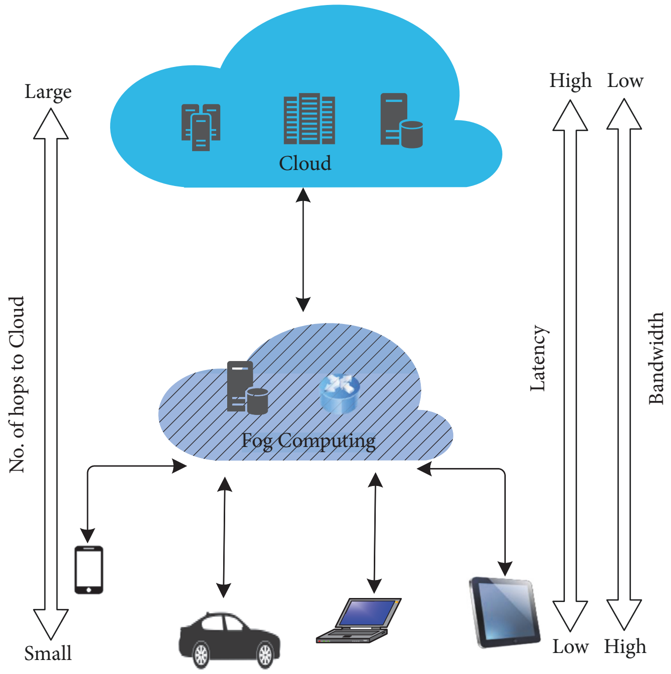


Figure 1: Comparison of number of hops, latency, and bandwidth from edge nodes to the cloud.

addition, vehicles can benefit from computation and storage, as well as very low delay in communications between the vehicles and the edge cloud.

Figure 1 shows latency, bandwidth, and number of hops comparison from edge nodes to the conventional cloud. The basic ECC can have three-vertical-tier hierarchy. As we can see from the hierarchy, the number of hops from the edge devices to the cloud increases as we go up from the bottom tier to the top tier. The edge devices have to go through several intermediate connecting devices, such as switches, routers, controllers, and servers using different communication technologies. The increased number of hops degrades the system performance due to delays and decrease throughput in end-to-end communication. The edge cloud can bring cloud-like services with better performance. In ECC, there are some similar concepts or flavors of edge cloud that fall under the same umbrella. In next subsection, we discuss a brief comparative study between three different edge cloud computing.

3.1.1. Comparative Study of Edge Cloud Computing. The three-edge cloud computing with similar concepts are fog computing, the mobile edge computing (MEC), and the cloudlets, which aims to bridge the gap between edge devices (such as VANETs and IoT devices) and conventional cloud computing as shown in Figure 2. We present a comparison between each of those concepts and the conventional cloud.

(A) Fog Computing. A fog computing was first introduced by Cisco Systems, Inc., engineers to extend the cloud computing paradigm to the edge of wireless networks for IoT devices [16]. OpenFog Consortium drives the fog computing architecture. Its objective is to influence standard bodies to create standards so that the edge devices can interoperate securely with other edge devices such as IoT and cloud services efficiently [26]. The implementation of IoT applications in the two-tiered architecture of the conventional cloud does not meet requirements related to mobility, low latency, and location awareness of smart devices. Thus, the evolution of the multitiered fog computing architecture is investigated. In general, users must download their data (multimedia files, documents, etc.) from the cloud. In fog, the data will be stored in fog servers close to the users, decreasing latency and increasing throughput. In the first tier, ITS application is deployed in vehicles. The second tier is the fog platform, including fog devices such as RSUs and wireless access networks. The third tier is the hyperscale datacenter of the conventional cloud. Fog computing provides high bandwidth, low latency, location awareness, and Quality of Service (QoS) for streaming and real-time applications in vehicles [27]. Fog computing relies on resource virtualization by using hypervisors for input/output resources and computing, virtual file systems for storage, and an SDN for network virtualization infrastructure [28].

(B) Mobile Edge Computing. A new Industry Specification Group (ISG) within ETSI launched mobile edge computing: ETSI is developing a system architecture and is working towards standardization of several APIs essentially for MEC. The objective of the ISG is to create a standardized, open environment that will allow efficient and seamless integration of applications from service providers, vendors, and third parties across multivendor MEC platforms. The main purpose of MEC is to unite the telecommunications sectors and IT cloud worlds, providing IT and cloud computing capabilities

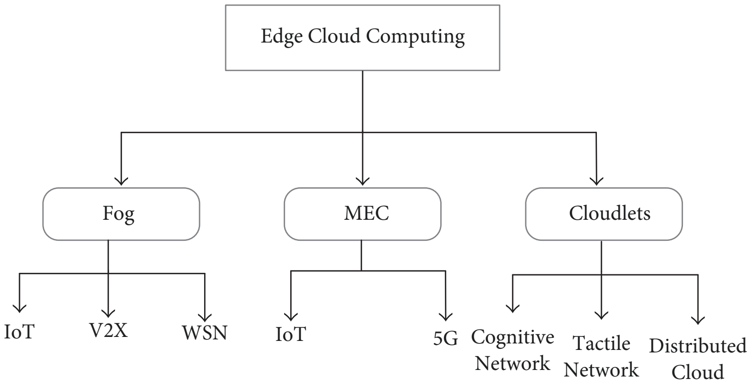


Figure 2: Various types of edge cloud computing.

within the RAN. MEC servers can be deployed at multiple sites, for example, at a Radio Network Controller (RNC), at a multi-Radio Access Technology (RAT) cell aggregation site, and at LTE macro base station (eNodeB) sites [29]. The MEC server offers computing resources, storage capacity, connectivity, radio, and network information focusing on the cellular network. MEC allows services and applications to be hosted above the network layer, that is, on top of the mobile network elements. These services and applications can benefit from close proximity to mobile devices, as well as from receiving local radio network contextual information [30]. Network operators focus on MEC to deliver a standardized edge computing architecture and industry-standardized APIs for third-party applications.

(C) Cloudlets. Satyanarayanan and his team at Carnegie Mellon University (CMU) developed cloudlets. The Open Edge Computing (OEC) initiative drives the development of software ecosystems surrounding cloudlets [31]. A cloudlet can be regarded as a datacenter in a box designed with the goal of bringing the cloud closer to mobile devices. According to Satyanarayanan, cloudlets are motivated with consideration of end-to-end latency as well as by the role of mobile computing in hostile environments [31]. A cloudlet can be viewed as a proxy of the real cloud and is located in the middle tier of the three-tier hierarchy. One of the applications of the cloudlet is to cover the absence of the cloud by performing its essential services under hostile environments, such as a failure of wireless backhaul due to Denial of Service (DoS) attacks. The cloudlets that are one wireless hop away from their associated mobile devices serve as physically nearby representatives of the cloud. This approach preserves the benefits of cloud mobile convergence while improving service availability [31]. CMU implemented various mechanisms as open source code for cloudlets. A thorough technical introduction and more information on recent usage of cloudlets are presented elsewhere in [32, 33]. Cloudlets are more useful in hostile scenarios, where the connections between edge devices and cloudlets are intermittent.

(D) Comparison of Edge Cloud Computing Approaches. A brief comparison between edge cloud computing approaches (i.e., fog, MEC, and cloudlets) and the conventional cloud is given in Table 1. Based on the comparison in Table 1, all three-edge cloud computing approaches share a similar vision, although the motivations are different. Among them, fog computing focuses on defining the virtual network topology for each fog application. The inspiration for fog comes from the IoT, Wireless Sensor Networks (WSNs), and VANETs [34]. MEC attempts to provide a highly distributed computing environment by combining cloud computing with cellular networks. The inspiration for cloudlets comes from research into distributed clouds, tactile, and cognitive networks. Cloudlets offer open source code to target any industry that can benefit from low-latency edge cloud computing, including the IoT, tactile Internet, 5G, web content delivery, or online gaming.

Fog computing and MEC support mobility applications on devices, whereas these applications have not been considered in cloudlets yet. Fog and MEC emphasize online analytics and big data analytics, as well as interaction with the cloud, whereas cloudlets do not support big data analytics. Fog computing collects and secures data from vehicles travelling in a wide geographic area under different environmental conditions supporting geomobility while it is not inherent to cloudlets. Fog computing analyzes extremely time-critical data and makes a decision near the vehicles to reduce latency. Prompt action and decision on the data can make a big difference in avoiding disaster in hazardous locations. In case of fog, the location of the fog server is flexible; that is, the fog server can be placed in between the end devices and the cloud, while, in case of MEC, MEC servers are usually collocated with the base station (BS). In case of cloudlets, the edge servers are located at the network edge. In the VANET combined with fog, which will be referred to as fog VANET hereafter, RSUs are used to broadcast the time-critical messages to the vehicles around the event spot. In case of MEC, the interworking between MEC servers and RSUs has not been investigated in detail yet. MEC is based on cellular networks where the BS supports cell based broadcasting, that is, point-to-multipoint communication, such as evolved Multimedia Broadcast Multicast Service (eMBMS) [35] in a single cell. Thus, the emergency messages will be broadcast to a wide range beyond the accident area, even to the vehicles who do not need that information due to wide coverage of

Table 1: Comparative study of edge clouds (i.e., fog computing, MEC, and cloudlets) and the conventional cloud.

Key features	Edge cloud computing			Conventional cloud
	Fog	Mobile edge computing (MEC)	Cloudlets	
Proposed/introduced by	Cisco in Aug. 2012	ETSI in Sep. 2014	CMU team in Oct. 2013	Amazon
Promoted by	Cisco, Princeton, Microsoft, Intel, Dell	ETSI ISG MEC: Nokia, Huawei, IBM, Intel, NTT DoCoMo, Vodafone	Prof. Mahadev at CMU, later supported by Intel, Huawei, Vodafone, Deutsche Telekom, Nokia and NTT	Microsoft, Google, Amazon
Industry organizations	OpenFog Consortium	ETSI MEC ISG	Open Edge Computing initiative	Open Cloud Consortium
Inspiration	IoT, WSN, VANETs	MEC as stepping stone towards 5G mobile networks	Distributed cloud, cognitive and tactical network	Remote storage, computing, and applications
Architecture framework	Yes	Yes	Not necessary, it can be an enabler for fog and MEC	Yes
Geo-distributed	Yes	Yes	Yes	No
Network Architecture	Decentralized	Decentralized	Decentralized	Centralized
Low latency/jitter	Yes	Yes	Yes	No
Geo-mobility support for applications on device	Yes	Yes	Not inherent, but can be added	No
Near real-time interaction	Yes	Partial	Partial	No
Emphasis on online data analytics & interaction with cloud	Yes	Yes	N/A	Yes
Edge server location	Flexible b/w end device and cloud	Collocated with BS	Network Edge	N/A
Dynamic broadcasting range depending on the message type	Yes	Partial, not dynamic	Yes	N/A
Location awareness	Yes	Yes	Yes	No
Mobility support	Yes	Yes	N/A	Limited
Channel Bandwidth	75 MHz for 802.11p	20 MHz for LTE	20 MHz for 802.11n	---
Capacity	6 Mbps for 802.11p	150 Mbps downlink & 50 Mbps uplink for LTE	50 Mbps for 802.11n	---
Scalability	High	High	High	Average
Run experimental testbed	Yes, at Princeton	No	Yes, at CMU	Yes

a BS. If the BS wants to broadcast the emergency messages only to the vehicles near the event spot within a limited area, it needs to run searching operation to know the vehicles in that area, which can be complex and time-consuming. The situation might get worse if the vehicles near the event spot are served by different mobile network operators.

3.1.2. Fog Computing for Future VANETs. Fog computing stands as a good candidate among other edge clouds for future VANETs based on Table 1. It complements and extends the cloud to the edge of the network. Fog computing devices at the edge collect data generated by vehicles. Fog computing supports big data analytics by examining the raw data to draw conclusions about a specific information based on inference and enhances data processing at the edge in real time. It also filters the data collected locally and sends the filtered data to a higher tier, that is, the cloud, so it can retrieve the data in the future. For example, the mining of real-time data through edge analytics can produce more accurate reports of traffic congestion on the road or the behavior of a crowd in a festival area. When a vehicle moves far from a fog server, service latency increases drastically. Then, the fog can redirect the application in the vehicle to associate it with a new application instance on a fog server that is now closer to that vehicle. The distributed fog servers can cooperate with each other and share local information with vehicles in a VANET. In addition, fog can work autonomously, creating a local loop by sharing local traffic information, even when disconnected from the main cloud. In fog, data privacy can be enhanced to allow users to spread personal information to different physical locations. Privacy can be assured by examining sensitive data locally, instead of sending it to the cloud for analysis. In fog computing, distributed infrastructure consists of heterogeneous resources that are managed efficiently in a distributed manner. SDN can be used along with fog, which provides flexibility and programmability to heterogeneous vehicular networks.

3.2. Software-Defined Network in VANETs. SDN is an emerging technology that can be used in coordination with edge cloud computing, especially fog computing, to provide centralized control, flexibility, and programmability to networks [36, 37]. The most common protocol used for communication between the SDN control plane and data plane is OpenFlow [38–40]. Thus, the use of OpenFlow will improve the management of resources and vehicles by providing an opportunity for new services and control functions. It separates the control plane and the data plane with global information about the network state. The data plane is used for data forwarding, and the control plane is used for network traffic control.

The basic SDN architecture is shown in Figure 3. It consists of three layers, that is, the application layer, the control layer, and the network layer. The SDN controller interacts with the application layer using northbound interfaces. The northbound interface supports application developers to manage the network through the program. The southbound interface interacts with the SDN controller and network devices of the network layer. Lin et al. [41] proposed an east-west bridge mechanism for intradomain communications within an SDN. At the SDN controller, the network devices based on OpenFlow accept policies from the controller without implementing various network protocol standards, resulting in direct control, programming, and managing of resources in the network [42]. OpenFlow programmatically controls flow and defines path from source to destination. It increases network efficiency and decreases the router packet processing overhead when defining the path. It also minimizes network management costs [40, 43, 44]. OpenFlow devices use flow tables that consists of flow entries, and it decides on how an incoming packet will be processed or forwarded. The switch makes a forwarding decision for each incoming packet by looking into the flow table entries and simultaneously matching the header field of the incoming packets. If it finds a match, then a corresponding decision will follow. Otherwise, the packets are forwarded to the SDN controller for additional processing. Furthermore, the control layer consists of a mobility and routing manager. The mobility manager in the control layer is responsible for collecting and maintaining the status of all the SDN switches, RSUs, and vehicles. The vehicles may be temporarily disconnected from the control plane due to the high speed of the vehicles. Route prediction can be included to estimate the possible location when vehicles are disconnected. A vehicle's future route can be obtained from the GPS or navigation system used by the vehicle [45]. It can also manage event detection and the switch-status update policy. The routing manager maintains network topology information using SDN switch status. The network topology for stationary data plane devices rarely changes, whereas, for the mobile data plane, the network topology is constructed using collected neighbor information. Because of these characteristics of the SDN, it can be used to satisfy the requirements of future VANET systems and improve VANET services in heterogeneous environments [39].

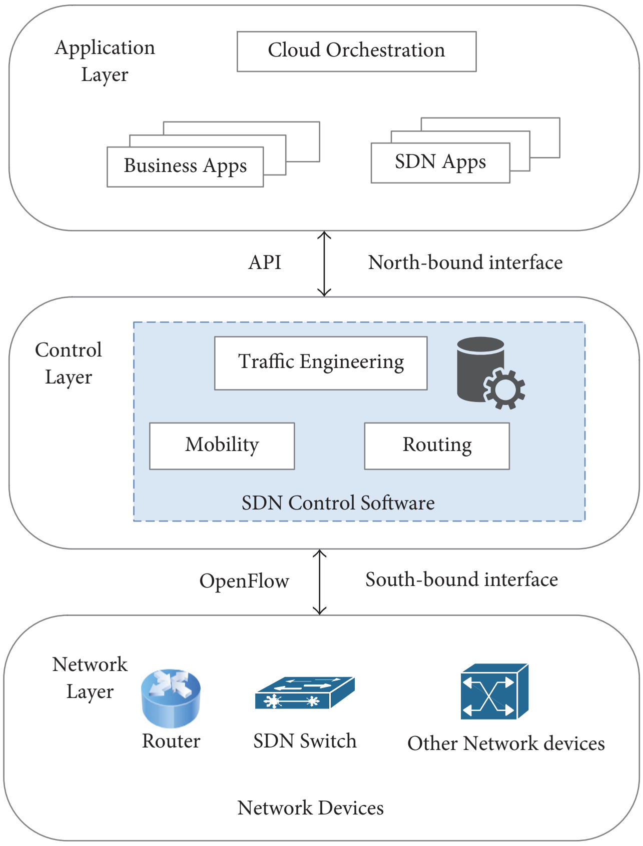


Figure 3: Basic SDN controller.

3.2.1. A Comparative Study of Software-Defined Network in VANETs. In VANETs, SDN provides synergies to solve several challenges and issues by improving the performance of the vehicular networks. There are several research work based on SDN in VANETs. We investigate different protocols that use SDN in VANETs and provide a brief comparative study of those protocols, which is given in Table 2.

The authors in [45] proposed Software-Defined Vehicular Ad Hoc Networks (SDVN), which integrates heterogeneous networks such as V2V, V2I, and vehicle-to-cloud. It provides solution for constructing a logically centralized but physically

Table 2: A comparative study of software defined network in VANETs.

References	Protocol	Controller	Simulator	Remarks
[45]	Software Defined Vehicular Network (SDVN)	Open source SDN controller POX	NS3 + SUMO	Discuss solutions to improve and mitigate heterogeneity of VANETs and provide solutions for constructing a logically centralized but physically distributed SDN controller to improve scalability.
[46]	Optimized Software-Defined Vehicular Ad Hoc Networks (OSDVN)	----	Omnet + Veins + Sumo	Control plane optimization Software-Defined Vehicular Ad Hoc Networks to balance the latency requirement and the cost on 5G networks.
[47]	Software-Defined Internet Of Vehicles (SDIV)	Floodlight	Mininet	Proposed a novel rule installation mechanism aimed at reducing flow table size for real-time query services in SDIV
[48]	SDN based Vehicle Ad-Hoc On-Demand Routing (SVAO)	------	NS3 + SUMO	Separates network control layer and data transfer layer of SDN to enhance the data transmission efficiency within VANETs.
[49]	Software Defined Internet of Vehicles (SD-IoV)	OpenFlow controllers	-----------	Proposed a layered SD-IoV architecture, which is capable of improving resource utilization, QoS, and network optimization.
[50]	RSU Cloud Resource Management (RSU-CRM)	OpenFlow controllers	Mininet	A road side Cloud resource management (CRM) model provides multi-objective optimization for minimizing control plane overhead, VM migration, number of service hosts and reducing infrastructure delays.
[51]	Software-defined Trust based Ad-Hoc On-demand Distance Vector routing (SD-TAODV)	NOS	OPNET	SDN based VANET focused on on-demand distance vector routing protocol with trust management scheme.
[52]	Fog Computing based on SDN (FSDN)	OpenFlow controllers	-----------	By leveraging the SDN basic functionality into fog computing, SDN can support services like a resource manager and fog orchestration models.

distributed SDN controller to improve scalability. The authors employed a trajectory-prediction-based vehicle status update policy to minimize SDN management overhead. They evaluated their proposed protocol using the network simulator NS-3 integrated with traffic simulator SUMO and open source SDN controller POX.

The authors in [46] proposed an Optimized Software-Defined Vehicular Ad Hoc Networks (OSDVN) that used hybrid mode for the southbound communication in the control plane of SDVN with low latency in 5G network. The authors optimized control plane of SDVN to balance the latency requirement and the cost on 5G networks. They formulated the bandwidth-rebating problem as a two-stage Stackelberg game and analyzed the game equilibrium to optimize southbound communication between vehicles and controllers. They evaluated their scheme using OMNeT++ and compared their scheme with other existing southbound communication modes. The results of their scheme showed improved latency over other existing southbound communication modes. However, the basic rule installation approach of the SDN control plane is not efficient for dynamic vehicle due to real-time requirements by VANET. In order to overcome this issue, [45] proposed a novel rule installation mechanism, which aimed at reducing flow table size for real-time query services in Software-Defined Internet of Vehicles (SDIV). They used compact flow table with improved proactive rule installation to reduce the number of rules. They substituted multicast addresses with destination address and some modification in packet header without network performance degradation. They used Mininet for their simulation and used flood light as SDN controller.

The authors in [48] proposed SDN based On-Demand Routing Protocol in Vehicle ad hoc networks (SVAO) that separates network control layer and data transfer layer of SDN to enhance the data transmission efficiency within VANETs. They proposed a two-level structure. A distributed global level collects all the vehicle information and centralized local level selects forwarding devices. They have used NS3 with SUMO to compare the performance of their proposed protocol with other routing protocols with different vehicle density, at different vehicle speeds in diverse communication ranges. On the other hand, Jiacheng et al. proposed a Software-Defined IoV (SD-IoV) architecture that is capable of improving resource utilization, QoS, and network optimization in harsh VANET environments [49]. The authors discussed SD-IoV major functions and challenges, and then they explained in detail on how those challenges can be resolved. They gave an illustration on how functions enabled by SD-IoV solved the current challenges in IoV. They also provided the benefits of SDN in SD-IoV such as vendor neutralization, network management, flexibility, and easy application deployment. They pointed out open issues of SD-IoV to shed light on future research.

Similarly, the authors in [50] designed RSU Cloud Resource Management (RSU-CRM) that contributes to the VANET cloud by increasing its advantages for service providers and provides QoS for the users in the vehicle grid. The primary demand of the vehicle grid is served by leveraging the flexibility of SDN to reconfigure the services hosted in the network dynamically and forwarding the data efficiently. The CRM minimizes the infrastructure delays in the RSU cloud and minimizes the network configurations. It also minimizes the operational costs by reducing the number of service replications. In addition to this, it provides multiobjective optimization for minimizing control plane overhead and VM migration. Zhang et al. [51] proposed a Software-Defined Trust Based Ad Hoc On-Demand Distance Vector routing (SD-TAODV) protocol in software-defined VANETs with trust management to ensure security and QoS. They used OPNET modeler to compare their protocol with AODV protocol, which showed better network performance than AODV but shows higher end-to-end delay than AODV. The authors in [52] adopted the hybrid control mode proposed by Ku et al. [42]. In their architecture, the SDN controller does not take full control of the system, but the task was shared with BS and RSUs. They assumed that SDN wireless nodes use a cellular node for the control channel and DSRC for the data channel. The authors used SDN wireless nodes as vehicles that act as a forwarding data plane that are operating on OpenFlow.

4. Example Scenario of Fog for Safety Message Dissemination in VANETs

In future VANETs, fog can play an important role in the dissemination of time-critical event messages, as it is located at the proximity of the vehicles. Let us consider a case of using fog in VANET environment. The main components of fog VANETs are fog servers, which are geographically distributed to control all the local activities in the region of interest. It can contribute to timely processing of data. A fog server interconnects with the cloud or other fog devices using wired connections on the Internet [53]. It helps to lower operating expenses through conserving network bandwidth by processing the selected data locally, instead of sending them to the central cloud for analysis. We assume that fog servers have intelligence and can find an appropriate destination of emergency messages. The fog server uses its computing resources to provide necessary location-based services and interact with vehicles and other fog devices for real-time information processing.

Table 3: A table that summarizes the propagation range of a safety message depending on the event types.

Event type	Range
E_1 (e.g., intersection accident)	d_1 (e.g., 10 m)
E_2 (e.g., fatal accident)	d_2 (e.g., 100 m)
⋮	⋮
E_n (e.g., road pile-up accident)	d_n (e.g., <10 km)

Fog VANET can be composed of conventional cloud, fog computing infrastructure, and end devices (such as smart vehicles) as shown in Figure 4. The top tier is a conventional cloud, which consists of servers or datacenters that provide computation capability, massive data storage, and global networking. Fog VANET interconnects fog devices (such as fog servers, router, switches, and RSUs) with the vehicles. Each RSU is connected to a switch, which is connected to a router. Each router is connected with fog servers, adjacent routers, and the main cloud through Internet. The vehicles are the end nodes located at the bottom tier of the framework. In fog infrastructure, the fog servers usually provide computing service, and the routers and switches are the networking devices and the RSUs are the wireless access devices. In this example, we consider two types of messages, that is, safety messages and infotainment messages. We investigate how the safety messages and infotainment messages can be disseminated from the vehicles to an appropriate destination. All the communications in fog VANET are based on IP protocol.

Let us consider an emergency situation where a vehicle broadcasts safety messages to the neighbor vehicles and nearby RSUs, when it detects an event. Depending upon the event types, the propagation range of a safety message can be different as shown in Table 3. Some event types are advertised to a limited range while other events are disseminated to a very wide area of several kilometers based on the severity of the event. Event types such as injury accidents or road pile-ups need to be disseminated to a longer distance to prevent from further causalities. However, the propagation range of a safety messages might be dependent on additional factors, for example, weather conditions, and, thus, a more complicated logic might be required to determine an appropriate propagation range. This algorithm needs to be deployed in the fog servers in this scenario. In this example, we consider an event type of road pile-up accident, which needs to be disseminated to a wider area. When a vehicle detects a pile-up accident, it broadcasts that information to nearby RSUs using broadcast IP address. Then, RSUs forward this safety message to the switch since this message was sent with a broadcast IP address, it reaches a router in Figure 4. When router receives the broadcast safety message, that message needs to be treated as an exceptional message. We can set up the routing table of the router so that every message with a broadcast IP address is sent to the fog server. When the router receives the safety message with a broadcast destination IP address, the router will forward the safety message to the fog

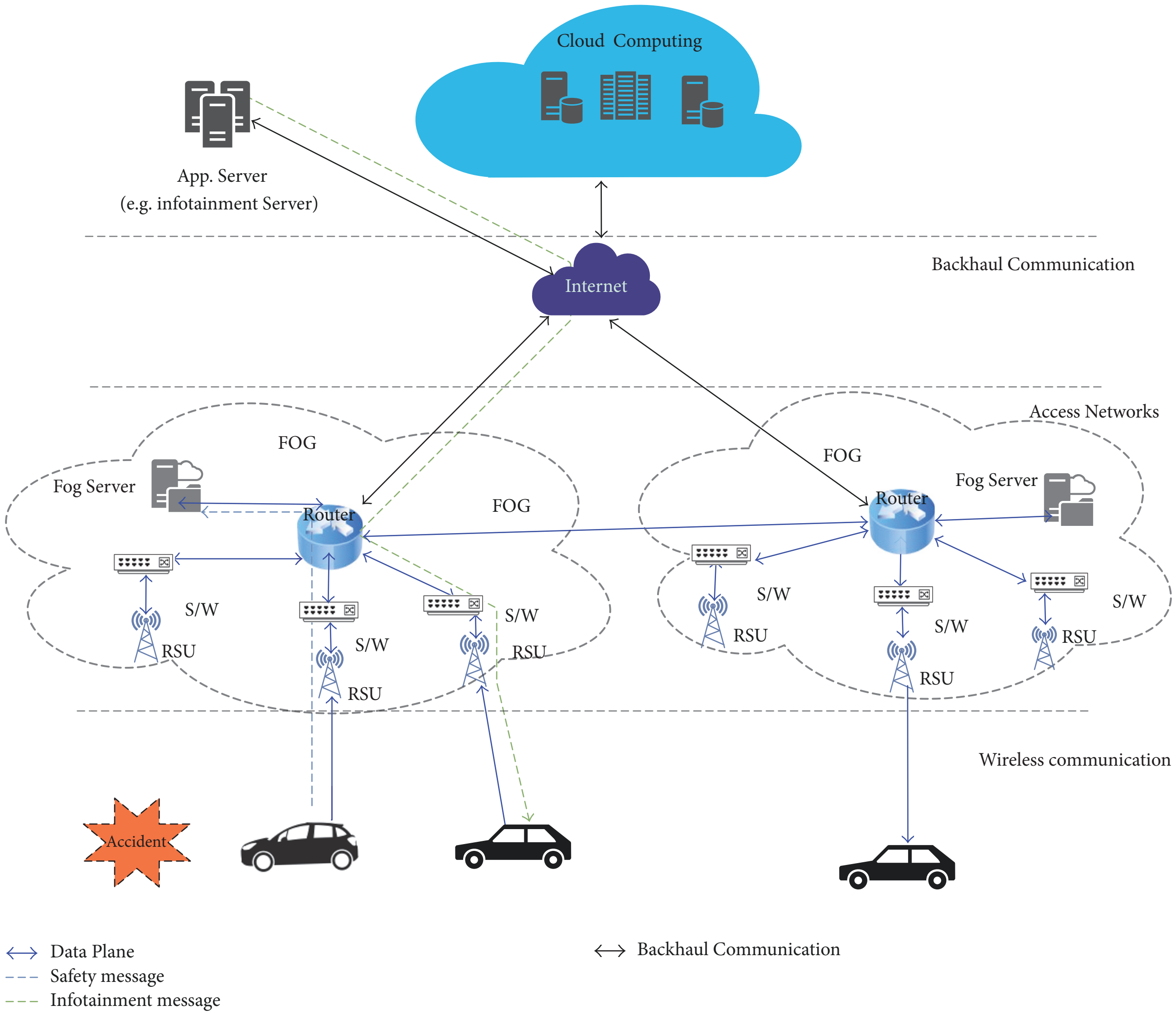


Figure 4: Safety message dissemination in fog VANET.

server for further processing. The message flow is represented by a blue dashed line in Figure 4. The fog server examines the received message focusing on the event type, contents, timeliness, and priority. We assume that the fog servers have a logic such as machine-learning algorithms to decide whether to forward the message to the central cloud or to other fog devices. The fog server transmits the message to other fog servers geographically close to the accident area. The fog server can also transmit the accumulated data to cloud for big data analysis and the results can be provided to the third parties such as insurance company, police departments, or hospital for further processing.

Let us consider the infotainment case, where the user of the vehicle tries to download some multimedia contents from the application server. The communication between the vehicle and the application server is based on IP address. The first message from the vehicle, for example, http request message, can be received by an RSU and be forwarded to the router via switch. The router forwards the infotainment message to the application server, based on the server IP address, which is represented by a green dashed line in Figure 4.

Figure 5 explains what advantage can be obtained if SDN is used along with fog in the situation similar to that of Figure 4. Combining SDN with fog might give an additional benefit in future VANETs, such as low latency. The SDN controller platform uses SDN controllers running the OpenFlow protocol and it provides fog orchestrator, network, and resource manager. In Figure 5, the red dashed line represents communication for the control plane and blue line represents communication for the data plane. Different from the case of Figure 4, SDN controller platform is used between fog and the cloud as shown in Figure 5. The fog orchestration at the SDN controller organizes fog devices such as fog servers, routers, switches, and RSU controllers [14]. The RSU controller in the SDN framework is an intelligent fog device, which runs the OpenFlow protocol. It controls and

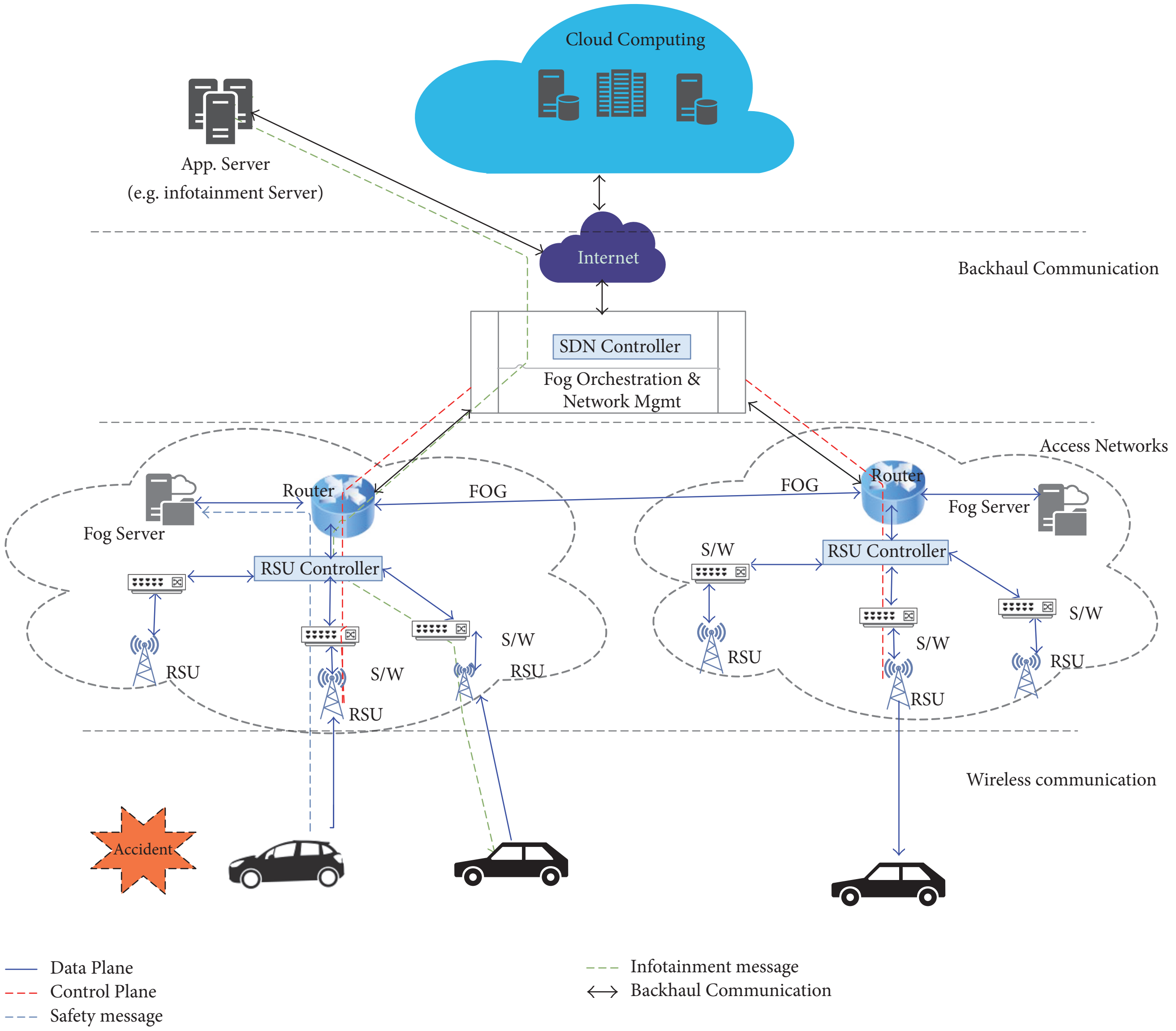


Figure 5: Safety message dissemination in fog integrated with SDN in VANETs.

manages the RSUs and connects with other distributed RSU controllers. The fog server is responsible for forwarding the data and storing local road and traffic information. We assume that the SDN controller, fog devices, and distributed RSU controllers are interconnected with each other through high-speed connections such as optical fiber.

Similar to the previous example without SDN framework, we assume that the vehicle detects a pile-up accident event on the road and broadcasts safety messages to all neighbor vehicles and nearby RSUs. Each RSU sends this event message to the fog server through a switch, RSU controller, and a router. As discussed in the previous scenario, the router sends the safety messages to the fog server based on the broadcast IP address. The fog server decides whether to forward the message to the central cloud through SDN controller or to other fog servers for further dissemination based on the event type given in Table 3. The SDN controller can establish low-latency routes between relevant components in advance, for example, between each fog server and the cloud or between nearby fog servers.

In this case, the event type needs to be included as an additional field in the forwarding table of the router that is based on SDN. If the event-to-range mapping table in Table 3 can be defined clearly for all possible cases, then the routes for each type of message can be preestablished by the SDN controller. However, if there are some exceptional cases, which cannot be covered by Table 3, then a new route needs to be established on arrival of the first packet corresponding to that exceptional case by interworking of SDN controller and fog servers.

All the packets that are covered by Table 3 can be handled by routers directly with low latency. If there is no rule

Table 4: Deployment issues for cloud-based approaches.

Deployment Issues	Fog	MEC	Cloudlets
Networking and computing infrastructure	(i) Management of fog server position and interconnection (ii) Management of neighbor fog servers (iii) Wireless access (a) Wi-Fi and mobile networks (iv) Discrimination of real-time critical message from non-real-time message	(i) Management of MEC server position and interconnection (ii) Management of neighbor MEC servers (iii) Wireless access (a) mobile networks (iv) Discrimination of real-time critical message from non-real-time message (v) Internetworking issues between existing VANET infrastructure and MEC	(i) Management of Cloudlets server position and interconnection (ii) Management of neighbor cloudlet servers (iii) Wireless access (a) Wi-Fi (iv) Discrimination of real-time critical message from non-real-time message

matching the arriving packet in the forwarding table of the router, then those packets can be sent to the fog server. The fog server will find appropriated destination of the exceptional packet and notify the SDN controller of the routing decision. Then, the SDN controller can configure the router to handle those packets with a reduced delay afterwards.

As compared with the previous example, this example shows that fog computing integrated with SDN in VANETs can be a candidate to resolve the key challenges of future VANETs. The fog integrated with SDN provides low latency while increasing throughput by storing the data in the local fog servers that are close to the vehicular nodes. It is suitable for real-time traffic applications with low latency such as emergency messages. With the integration of cellular networks with 802.11p, it may be possible to resolve the issue of intermittent connections between V2X communication on a highway or an isolated road.

5. Deployment Issues of Cloud-Based Approaches

The three-edge cloud computing approaches, for example, fog, MEC and cloudlets, have several deployment issues in terms of computing and networking infrastructures; we summarized them in Table 4.

One of the key issues is the management of the server position and the interconnection topology between them. All the three-edge cloud computing approaches use edge servers close to the edge nodes. The position of each edge server needs to be determined considering the vehicle density in a given area, the data traffic, the communication range of each vehicle, and the processing burden on a server. In case of MEC, the MEC servers are usually collocated with the base station. The optimal position of edge servers needs to be determined carefully considering the above factors by the Internet Service Providers (ISPs) or telecom operators, and these positions need to be stored and managed systematically, for example, based on the cloud. If this information is stored in the cloud, then the cloud can have a global picture of distributed edge servers.

On the other hand, each edge server needs to know adjacent edge servers. If the cloud manages the global information about the position of edge servers, then an edge server can obtain a list of adjacent servers from the cloud by sending a request message containing its own network layer address and physical location information. Once the list is obtained, this list needs to be stored in the local memory, and the connection to those adjacent servers and the interconnection topology information need to be managed.

In VANETs, the messages generated by the vehicle can be classified into two different categories: time-critical event messages corresponding to the types defined in Table 3 and non-real-time messages corresponding to infotainment applications. As we discuss in Section, time-critical messages are usually delivered to the edge server to decide whether send it to the cloud or to adjacent edge servers. However, infotainment traffic does not need to be delivered to the edge servers since the destinations of those messages are apparent, for example, infotainment servers. In this case, some network devices such as routers or switches need to discriminate the time-critical event messages from infotainment messages to send them to appropriate destinations. This is an important issue that needs to be resolved by the network devices.

In terms of access network technology, the fog supports both Wi-Fi and mobile network access mechanism. However, MEC supports mobile network and cloudlet supports Wi-Fi only [54].

In case of MEC, there is an additional internetworking issue. Let us consider a case where we want to deploy MEC in existing VANETs. Since RSU uses 802.11p protocol and MEC uses a different protocol, for example, REPLISOM [55], they cannot communicate with each other. The messages may not be delivered from the vehicles to the MEC. We can consider two different approaches to this issue. If we upgrade the RSUs so that it can communicate with the MEC's BS, the messages from the vehicles can be delivered to the MEC by the relay of RSU. However, if the vehicle drives to new location where RSUs have not been upgraded or RSUs have not been deployed yet, then the vehicles cannot communicate with the MEC servers. As a second approach, we can consider upgrade of the hardware and software of each vehicle so that it can communicate with the MEC server directly through BS. However, if vehicles can communicate directly with BS, then the role of existing RSUs needs to be redefined.

6. Conclusion

This paper presents a review of cloud-based emerging technologies for future VANETs. We outlined the key challenges that need to be resolved in future VANETs. We present a brief overview of edge cloud computing and three different concepts of edge cloud that fall under the same umbrella. We then compared three different types of edge cloud computing-based approaches: fog, mobile edge computing (MEC), and cloudlets based on key features. Although three-edge cloud computing approaches share similar vision, their purposes are different. We discussed a possible capability of fog and a combination of fog and SDN in resolving key challenges of future VANETs through detailed examples. We also explained the issues that need to be resolved for the deployment of three different cloud-based approaches.

Acronyms

VANET:	Vehicular ad hoc networks
SDN:	Software-Defined Network
ITS:	Intelligent Transportation Systems
V2V:	Vehicle-to-Vehicle
V2I:	Vehicle-to-Infrastructure
RSU:	Roadside Unit
MANET:	Mobile Ad Hoc Networks
SDN:	Software-Defined Network
DSRC:	Dedicated Short-Range Communications
LTE:	Long-Term Evolution
CSMA/CA:	Carrier-Sense Multiple Access with Collision Avoidance
WAVE:	Wireless Access in Vehicular Environment
WSM:	WAVE Short Message
SaaS:	Software as a service
PaaS:	Platform as a service
IaaS:	Infrastructure as a service
ECC:	Edge cloud computing
IoT:	Internet of Things
BS:	Base station
MEC:	Mobile edge computing
QoS:	Quality of Service
ISG:	Industry Specification Group
RNC:	Radio Network Controller
RAT:	Radio Access Technology
V2X:	Vehicles-to-Everything
OEC:	Open Edge Computing
eMBMS:	Evolved Multimedia Broadcast Multicast Service
WSN:	Wireless Sensor Network
SDVN:	Software-Defined Vehicular Network
OSDVN:	Optimized Software-Defined Vehicular Ad Hoc Networks
SDIV:	Software-Defined Internet of Vehicles
SVAO:	SDN based Vehicle Ad Hoc
SD-IoV:	Software-Defined Internet of Vehicles
RSU-CRM:	RSU Cloud Resource Management
DoS:	Denial of Service
ISP:	Internet Service Provider.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2015R1D1A1A01058595). This research was supported in part by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) Support Program (IITP-2018-2016-0-00313) supervised by the IITP (Institute for Information Communications Technology Promotion).

References

[1] Intelligent transport systems (ITS), “Framework for public mobile networks in cooperative its (c-its)s,” Tech. Rep., European Telecommunications Standards Institute (ETSI), Palo Alto, Calif, USA, April 2012.

[2] E. Hossain, G. Chow, V. C. M. Leung et al., “Vehicular telematics over heterogeneous wireless networks: a survey,” *Computer Communications*, vol. 33, no. 7, pp. 775–793, 2010.

[3] R. I. Meneguette, L. F. Bittencourt, and E. R. M. Madeira, “A seamless flow mobility management architecture for vehicular communication networks,” *Journal of Communications and Networks*, vol. 15, no. 2, pp. 207–216, 2013.

[4] Y. L. Morgan, “Notes on DSRC & WAVE standards suite: its architecture, design, and characteristics,” *IEEE Communications Surveys & Tutorials*, vol. 12, no. 4, pp. 504–518, 2010.

[5] H. Hartenstein and K. P. Laberteaux, *VANET: Vehicular Applications and Inter-Networking Technologies*, Wiley Online Library, 2010.

[6] B. Liu, D. Jia, J. Wang, K. Lu, and L. Wu, “Cloud-assisted safety message dissemination in VANET-cellular heterogeneous wireless network,” *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–12, 2015.

[7] K. Trichias, “Modeling and evaluation of lte in intelligent transportation systems. dissertation,” *IEEE Systems Journal*, vol. PP, pp. 1–12, 2011.

[8] R. Sivaraj, A. K. Gopalakrishna, M. G. Chandra, and P. Balamuralidhar, “QoS-enabled group communication in integrated VANET-LTE heterogeneous wireless networks,” in *Proceedings of the 2011 IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob’2011*, pp. 17–24, October 2011.

[9] C.-F. Lai, H.-C. Chao, Y.-X. Lai, and J. Wan, “Cloud-assisted real-time transrating for http live streaming,” *IEEE Wireless Communications Magazine*, vol. 20, no. 3, pp. 62–70, 2013.

[10] L. Gu, D. Zeng, and S. Guo, “Vehicular cloud computing: a survey,” in *Proceedings of the IEEE Globecom Workshops (GC Wkshps ’13)*, pp. 403–407, Atlanta, Ga, USA, December 2013.

[11] J. Wan, D. Zhang, Y. Sun, K. Lin, C. Zou, and H. Cai, “VCMIA: a novel architecture for integrating vehicular cyber-physical systems and mobile cloud computing,” *Mobile Networks and Applications*, vol. 19, no. 2, pp. 153–160, 2014.

[12] J. Wan, D. Zhang, S. Zhao, L. T. Yang, and J. Lloret, “Context-aware vehicular cyber-physical systems with cloud support: architecture, challenges, and solutions,” *IEEE Communications Magazine*, vol. 52, no. 8, pp. 106–113, 2014.

[13] I. Stojmenovic, S. Wen, X. Huang, and H. Luan, "An overview of Fog computing and its security issues," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 10, pp. 2991–3005, 2016.

[14] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the 1st ACM Mobile Cloud Computing Workshop, MCC 2012*, pp. 13–15, ACM, New York, NY, USA, August 2012.

[15] X. Duan, Y. Liu, and X. Wang, "SDN enabled 5G-VANET: Adaptive vehicle clustering and beamformed transmission for aggregated traffic," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 120–127, 2017.

[16] Open Network Foundation, "White paper: Software defined networking: The new norm for networks," Tech. Rep., CISCO, Palo Alto,Calif, USA, April 2012.

[17] W. He, G. Yan, and L. D. Xu, "Developing vehicular data cloud services in the IoT environment," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1587–1595, 2014.

[18] R. I. Meneguette, "A vehicular cloud-based framework for the intelligent transport management of big cities," *International Journal of Distributed Sensor Networks*, vol. 12, no. 5, article 8198597, 2016.

[19] S. Bai, F. A. Zaid, and F. Bai, "Vehicle safety communications-applications (vsc-a) vehicle safety communications," Tech. Rep., The National Academy of Sciences, Engineering and Medicine, April 2011.

[20] X. Lin, X. Sun, P.-H. Ho, and X. Shen, "GSIS: a secure and privacy-preserving protocol for vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3442–3456, 2007.

[21] H. Sarvada, T. R. Nikhil, and H. J. Kulkarni, "Identification of black spots and improvements to junctions in Bangalore city," *International Journal of Scientific Research*, vol. 2, no. 8, pp. 136–139, 2013.

[22] M. Armbrust, A. Fox, R. Griffith et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[23] R. Hussain, J. Son, H. Eun, S. Kim, and H. Oh, "Rethinking vehicular communications: merging VANET with cloud computing," in *Proceedings of the 4th IEEE International Conference on Cloud Computing Technology and Science (CloudCom '12)*, pp. 606–609, IEEE, Taipei, Taiwan, December 2012.

[24] G. Yan, S. Olariu, and T. Hristov, "The next paradigm shift: From vehicular networks to vehicular clouds," *Mobile Ad Hoc Networking: Cutting Edge Directions*, vol. 56, no. 6, pp. 645–700, 2013.

[25] S. Islam and J.-C. Grégoire, "Giving users an edge: A flexible Cloud model and its application for multimedia," *Future Generation Computer Systems*, vol. 28, no. 6, pp. 823–832, 2012.

[26] Openfog Consortium Architecture Working Group, "OpenFog Reference Architecture for Fog Computing," February 2017, http://www.OpenFogConsortium.org/.

[27] P. Bellavista, "Mobile cloud networking: Lessons learnt, open research directions, and industrial innovation opportunities," in *Proceedings of the 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, MobileCloud 2016*, pp. 79-80, April 2016.

[28] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog Computing: A Platform for Internet of Things and Analytics," in *Studies in Computational Intelligence*, vol. 546, pp. 169–186, Springer, 2014.

[29] Y. C Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "White paper: mobile edge computing a key technology towards 5G," ETSI (European Telecommunications Standards Institute), Sophia Antipolis CEDEX, Nice, France, September 2015.

[30] Mobile Edge Computing (MEC), ETSI, and Industry Specification Group (ISG), "White paper: mobile edge computing (MEC); framework and reference architecture," ETSI (European Telecommunications Standards Institute), Sophia Antipolis CEDEX, Nice, France, 2016.

[31] M. Satyanarayanan, G. Lewis, E. Morris, S. Simanta, J. Boleng, and K. Ha, "The role of cloudlets in hostile environments," *IEEE Pervasive Computing*, vol. 12, no. 4, pp. 40–49, 2013.

[32] M. Satyanarayanan, R. Schuster, M. Ebling et al., "An open ecosystem for mobile-cloud convergence," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 63–70, 2015.

[33] M. Satyanarayanan, P. Bahl, R. Cáceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.

[34] Mobile Edge Computing (MEC), ETSI, and Industry Specification Group (ISG), *White Paper: Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are*, CISCO, San Jose, Calif, USA.

[35] D. Lecompte and F. Gabin, "Evolved multimedia broadcast/multicast service (eMBMS) in LTE-advanced: overview and Rel-11 enhancements," *IEEE Communications Magazine*, vol. 50, no. 11, pp. 68–74, 2012.

[36] H. Farhady, H. Lee, and A. Nakao, "Software-defined networking," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 81, no. C, pp. 79–95, 2015.

[37] N. McKeown, T. Anderson, H. Balakrishnan et al., "OpenFlow: enabling innovation in campus networks," *Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[38] W. Stallings, "Openflow switch specification," Tech. Rep., Open Networking Foundation, Menlo Park, Calif, USA, 2014.

[39] W. Stallings, "Software defined networks and openflow," *The Internet Protocol Journal*, vol. 16, no. 1, pp. 1–7, April 2010.

[40] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114–119, 2013.

[41] P. Lin, J. Bi, and Y. Wang, "East-West Bridge for SDN Network Peering," in *Communications in Computer and Information Science*, vol. 401, pp. 170–181, Springer, Berlin, Germany, 2013.

[42] 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.

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

[44] M.-K. Shin, K.-H. Nam, and H.-J. Kim, "Software-defined networking (SDN): A reference architecture and open APIs," in *Proceedings of the 2012 International Conference on ICT Convergence: "Global Open Innovation Summit for Smart ICT Convergence", ICTC 2012*, pp. 360-361, October 2012.

[45] Z. He, J. Cao, and X. Liu, "SDVN: Enabling rapid network innovation for heterogeneous vehicular communication," *IEEE Network*, vol. 30, no. 4, pp. 10–15, 2016.

[46] H. Li, M. Dong, and K. Ota, "Control Plane Optimization in Software-Defined Vehicular Ad Hoc Networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 7895–7904, 2016.

[47] X. Wang, C. Wang, J. Zhang, M. Zhou, and C. Jiang, "Improved rule installation for real-time query service in software-defined internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 2, pp. 225–235, 2017.

[48] B. Dong, W. Wu, Z. Yang, and J. Li, "Software Defined Networking Based On-Demand Routing Protocol in Vehicle Ad Hoc Networks," in *Proceedings of the 12th International Conference on Mobile Ad-Hoc and Sensor Networks, MSN 2016*, vol. 25, pp. 11–18, IEEE, Hefei, China, December 2016.

[49] C. Jiacheng, Z. Haibo, Z. Ning, Y. Peng, G. Lin, and S. Xuemin, "Software defined Internet of vehicles: architecture, challenges and solutions," *Journal of Communications and Information Networks*, vol. 1, no. 1, pp. 14–26, 2016.

[50] M. A. Salahuddin, A. Al-Fuqaha, and M. Guizani, "Software-defined networking for rsu clouds in support of the internet of vehicles," *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 133–144, 2015.

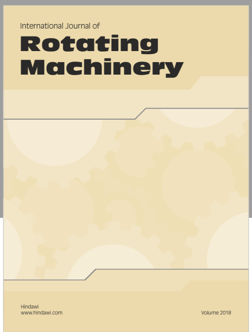
[51] D. Zhang, F. R. Yu, Z. Wei, and A. Boukerche, "Trust-based secure routing in software-defined vehicular ad hoc networks," *Networking and Internet Architecture*, 2016.

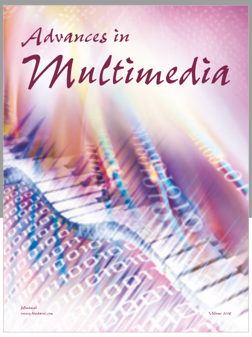
[52] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, "Software defined networking-based vehicular Adhoc Network with Fog Computing," in *Proceedings of the 14th IFIP/IEEE International Symposium on Integrated Network Management/International workshop on management of the future Internet*, pp. 1202–1207, Ottawa, Canada, May 2015.

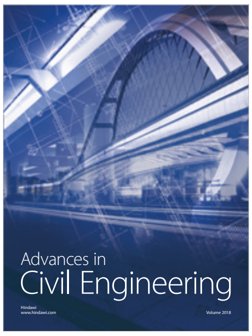
[53] T. H. Luan, L. Gao, Z. Li, W. Yang, S. Guiyi, and L. Sun, "Fog computing: focusing on mobile users at the edge, networking and internet architecture," *Networking and Internet Architecture*, 2015.

[54] K. Dolui and S. K. Datta, "Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing," in *Proceedings of the 2017 Global Internet of Things Summit, GIoTS 2017*, pp. 1–6, Geneva, Switzerland, June 2017.

[55] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, "Replisom: disciplined tiny memory replication for massive IoT devices in LTE edge cloud," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 327–338, 2016.







Hindawi

Submit your manuscripts at
www.hindawi.com