



Review

Convergence of Software-Defined Vehicular Cloud and 5G Enabling Technologies: A Survey

Lionel Nkenyereye ^{1,†} , Lewis Nkenyereye ^{2,*,†} and Jong-Wook Jang ^{3,†} ¹ Research & Education Group of AI Convergence, Pukyong National University, Busan 48513, Republic of Korea; nklionel@pknu.ac.kr² Department of Computer and Information Security, Sejong University, Seoul 05006, Republic of Korea³ Department of Computer Engineering, Dong-Eui University, Busan 47340, Republic of Korea; jwjang@deu.ac.kr

* Correspondence: nkenyele@sejong.ac.kr

† These authors contributed equally to this work.

Abstract: Vehicular cloud computing (VCC) and connected vehicles have prompted the intensive investigation of communication and computing solutions. As an important enabler, software-defined network (SDN) broadly changes the design of vehicle services, from resource allocation to ambitious autonomous cars. However, current VCC architectures face challenges that hinder the vision of providing reliable services to connected vehicles. As a result, deploying VC services using SDN network has emerged as a viable option. Therefore, software-defined VC architecture (SDVC) dynamically manages the control and resource utilization of VC by centralizing the overall knowledge. In addition, SDN stands as the representative technique of virtual resources and network function virtualization (NFV). NFV is integrated into SDVC frameworks to design extended SDVC (ESDVC) for dynamic, adaptive VC maintenance, VC network slicing management, and to meet constraint requirements such as network latency and reliable connectivity. This paper presents and discusses: (1) the architecture scenario of both SDVC and ESDVC; (2) the effective deployment methods enabling NFV and network slicing (NS) frameworks to customize VC frameworks; (3) challenges and future concepts of more VC services based on ESDVC architecture. From this survey, we believe readers would find relevant methods for realigning information dispersed across the SDVC, fifth generation (5G)-based VC, and NS domains and comprehending the relationships between these technologies while encouraging further debate on the fusion of 5G enabling technologies over SDVC to enable VC network slicing.

Keywords: vehicular network; network function virtualization; software-defined vehicular cloud network; network slicing; vehicular cloud; multi-access edge computing



Citation: Nkenyereye, L.; Nkenyereye, L.; Jang, J.-W. Convergence of Software-Defined Vehicular Cloud and 5G Enabling Technologies: A Survey. *Electronics* **2023**, *12*, 2066. <https://doi.org/10.3390/electronics12092066>

Academic Editor: Martin Reisslein

Received: 10 March 2023

Revised: 21 April 2023

Accepted: 27 April 2023

Published: 29 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Vehicular cloud computing (VCC) has revolutionized intelligent transportation systems (ITSs) and become a primitive platform for connected vehicles [1]. In VC, a cluster of vehicles shares resources and completes tasks in a mobile ad hoc cloud. Users in VC have access to a variety of cloud-based applications. For instance, “geo-advertising” provides a group of entertainment applications in a particular zone and direction. Moreover, these applications deliver content on comfort, safety, and enjoyment by using technological advancements in the Internet of vehicles (IoVs) [2]. VC applications are offered to the public during event evacuation (or emergency rescue) for instance. The VC cluster nearby the evacuation scene constitutes an important pool of information for the emergency management team. The team receives real-time information on traffic conditions, and travel time calculations to the hospitals or sites of rescue where the first necessity such as food, water, and shelter are offered. Road safety messages are constantly disseminated to VC vehicles. This application in VC requires real-time data from the VC cluster. By leveraging

the IoV processing server provided by the VC, each vehicle shares its surrounding data (e.g., potential road hazard ahead, speed breakers, holes, black ice, and road conditions) with other vehicles [3]. Research on IoVs has focused on network connectivity, computing power, and data storage [4]. The formation of a VC cluster features an important pool of computing resources, but the lack of advanced VC services results in failure to manage a pool of underutilized VC resources. Therefore, the integration of VC and emerging network enables VC clusters to cooperatively compute resource-intensive tasks generally scheduled to run in the cloud platforms. The computation of cloud-based applications using VC resources reduces the computation delay [5]. VCC improves the performance of resource-intensive tasks traditionally scheduled to run in the cloud.

Traditional challenges inherited from mobile computing environments are reported to VC. VC, as opposed to mobile computing end devices, is made up of vehicles that can move at high speeds, change location, switch to unstable wireless links, and have time-varying resources. As a result, it is difficult for transportation authorities to rely on VC for administering or controlling VC-provided resources. This, in turn, prompts VC administrators to wait for resource re-unification before resuming the computation of the running VC service. Furthermore, it makes it difficult to monitor billing cost for the usage of computing resources. It also challenges the optimization of VC operations. To address these issues, research in the formation and operation of the VC has led to proposal of a new VC architecture based on software-defined network (SDN) called software-defined vehicular cloud (SDVC) [2,4]. By leveraging software modules and virtualization of infrastructure and network functions, the SDVC architecture enables the centralized controller to flexibly and efficiently manage resource utilization. Compared with VC, the main advantages of SDVC are threefold: (1) reliable view of the mobility prediction and resource pool virtualization—improves location awareness service via a cloud-radio access network (C-RAN) and multi-access edge computing (MEC) [6,7]; (2) selection of dedicated mobile communication virtualization functions—increase bandwidth use and optimize routing protocols by leveraging fifth generation (5G) solutions (e.g., network slicing (NS)) and artificial neural networks (ANNs) [8–10]; (3) powerful VC network programming—allows the controller server to create and maintain a database containing wide information on the VC such as network topology [11,12]. In [13], SDVC employs SDN and cloud computing to enable the distribution of software updates on SDN-enabled vehicles. The works in [14–17] enhance packet delivery, latency, and overhead communication. VC system has a broader range of application scenarios in smart cities that were successfully designed in line with SDVC architecture. A summary of advantages of SDVC compared with VCC is listed in Table 1. Table 2 summarizes some of the challenges addressed through SDVC systems.

SDVC has addressed some of the following challenges.

- Vehicle mobility: Replication and relocation of virtual machines (VMs) are combined to increase the quality of services (QoS) of cloud-based vehicular applications. The high mobility in VC influences the relocation of virtualized resources. In this situation, the SDN controller is in charge of VM migration and replication.
- Provisioning of computing resource: Information from VC providers and users is composed mainly of resource management, connectivity loss, and the decision rules from SDN application to SDN control layer. Based on those information, the management of VC resources are efficiently allocated to vehicles forming a cluster to compute intensive tasks [12].
- Heterogeneous wireless communication: The assistance of SDN applications in SDVC enables the control layer to generate forwarding rules on data plane switches regardless of the wireless communication modules deployed among vehicles of the VC cluster. Therefore, VC improved its ability to share computation tasks, cache (store) data, and provide reliable bandwidth.
- Privacy: Privacy modules running in SDN controllers would predict the privacy issue and then protect forwarding devices.

Table 1. Advantages of SDVC compared to VCC. Most features of VCC are discussed by Waheed et al. in [18] and Jang et al. in [4].

SN	Parameters	VCC	SDVC
1	Computational Resource.	<ul style="list-style-type: none"> Leveraging nearby vehicles' resources. Collaborative and autonomous resource formation. 	<ul style="list-style-type: none"> SDN application abstracts resource management. SD networking maintains RP status.
2	Architecture model [4].	<ul style="list-style-type: none"> Users. VC resource pool. VC administrator. 	<ul style="list-style-type: none"> Application plane. Control plane (virtualized resources, VC formation, and mobility status modules). Data plane (in-vehicle resources, wired/wireless communications).
3	Monitoring actual resources.	<ul style="list-style-type: none"> VC stores up-to-date resource information. Administrators update vehicles' resources and location. 	VC controller collects: <ul style="list-style-type: none"> VC relevant information. Consistent view of predicted mobility and resource pool virtualization.
4	Optimization of VC operations.	<ul style="list-style-type: none"> Virtualization of resources. VM migration. Self-organizing of VC formation. Forms infrastructure, network, storage as a service. 	<ul style="list-style-type: none"> VC controller maintains a consistent view of VC location via a cloud radio access network (C-RAN). Selection of dedicated mobile communication virtualization functions. Virtual routing protocols and function solutions. SDN application for mobility and VM migration decisions.

Advanced 5G technologies are gradually being combined with SDVC to meet customized requirements for VC services. The integration of 5G enabling solutions (e.g., NS, NFV) along with SDVC designs extended SDVC (ESDVC) architecture. SDVC and ESDVC benefit each other in providing architecture customized to fit the requirements of future VC services leveraging 5G and beyond technology. ESDVC and SDVC are not independent of each other. SDVC is the main target, and vehicular cloud services based on 5G technology are also a part of the SDVC. In turn, ESDVC can provide higher bandwidth slicing as a service, ultra-reliable low-latency communications (URLLC), and efficient resource utilization in VC systems. A issue with reference to the VC scalability has been addressed by accepting multiple SDN controllers to deal with the huge number of flow requests of vehicles [19]. These controllers are, in general, deployed on remote clouds data centers, edge of network core network, or road side units (RSUs), LTE eNBs (long-term evolution evolved node B) or 5G gNBs, for instance. However, even with the possibility of positioning the controllers at access points closer to vehicles, RSU-assisted MEC controllers allow achieving reduced communication delay [6], integrating the needed functionalities near the user for fast decision making. In ESDVC, the VC system faces critical requirements, which should be designed using VC network slicing that would host MEC controller modules deployed at the edge of the network. Therefore, ESDVC should meet the requirements by adopting dynamic deployment of SDN controllers and NFV service chaining [20,21]. A summary of advantages from comparison between SDVC and ESDVC is given in Table 3.

SDVC is expected to push knowledge of VC communication, VC formation, and VC resource organization (exploitation) to a remote controller infrastructure-based software virtualization, thereby enabling various distributed, low-latency, and reliable future VC applications or services. SDVC has the following advantages:

- Communication decisions are made closer to the VC with the help of the SDN control, which computes the network topology of the vehicle cluster. The cloud participates only when additional processing is required, hence significantly reducing latency and cost of (billing) of consumed resources;
- Vehicle privacy is enhanced because information about inter-vehicular communication plans and network topology required for running cooperative tasks by a cluster of vehicles are stored locally on the vehicles themselves rather than in the cloud;
- Self-organizing architecture of SD controllers provides more reliable computation of VC tasks;

- SDVC can promote the ubiquity of VC services and acknowledge the goal of “delivering massive computation resource to every vehicle user and every IoV systems everywhere” with the ability to process tasks of different sizes;
- Diverse and valuable VC services would increase the commercial value of SDVC infrastructure as a service and speed up its deployment and growth.

ESDVC, on the other hand, aims to integrate 5G and beyond technologies based on NFV (e.g., network slicing, virtual/container network functions (VNF/CNF)) into the SDVC for interactive, responsive SDVC management, implementation, and maintenance. It has the following advantages:

- The advancement of heterogeneous wireless communication in vehicles and the availability of diverse virtual network functions generate massive traffic data.
- At the same time, the VCC infrastructures are implementing cloud-native applications using the serverless and micro-services approach. The virtualization of physical networks creates logical instances tailored to handle customized VC services that require the use of management and orchestration platforms.
- Improving the reliability, the persistence of the connection between vehicle users and the cloud, and shared resources will require virtual network instances of SDVC that implement VC network slicing.
- NFV service chaining and decision learning based on ANN algorithms accurately optimize virtual network instances of SDVC inside NS systems. NFV service chaining is helpful for maintaining the QoS of instances when unexpected failures occur in a single VC network slicing.
- When confronted with complex and time-consuming vehicular cloud network information, VC can rely on ESDVC to create a platform as a service over VC network slicing to powerfully learn how to extract valuable information on the management and maintenance of SDVC.

Table 2. Summary of VC challenges addressed by SDVC.

SN	Description	Causes	Consequences	Techniques
1	Vehicle mobility	Drive at high speeds	<ul style="list-style-type: none"> • Replication and relocation of virtual machines (VMs). • Influences the relocation of virtualized resources. • Time-varying resource. • Change location and switch to unstable wireless links. • Difficult to administer or control VC-provided resources. • Wait for resource re-unification before resuming the computation. • Difficult to monitor billing cost on the usage of computing resources. • Optimization of the VC's operations. 	<ul style="list-style-type: none"> • SDN controller is in charge of VM migration and replication. • Centralized controller to flexibly and efficiently manage resource utilization.
2	Provisioning of computing resource	VC lacks functions for: <ul style="list-style-type: none"> • Resource management. • Connectivity loss. • Decision resources sharing rules. 	<ul style="list-style-type: none"> • Waste of computation resource. • High latency by accessing the cloud platform. 	<ul style="list-style-type: none"> • SDN application assists computation of intensive tasks. • SDN applications set rules for resource allocation. • SDN controller manages VC formation.
3	Privacy	Malicious nodes among VC cluster	Protect forwarding devices	Privacy modules running in SDN controllers would predict the privacy issue.

Therefore, considering that SDVC and ESDVC, i.e., VC network slicing services, together face multiple aspects of some of the same challenges and practical issues, we identify the following three technologies that are essential for VC network slicing:

Table 3. Comparison between SDVC and ESDVC.

SN	Parameters	SDVC	ESDVC
1	Goal	Push knowledge of VC communication, VC formation, and VC resources to a remote controller server.	Integrate 5G and beyond technologies based on NFV (e.g., NS, VNF/CNF) into the SDVC.
2	Data plane	<ul style="list-style-type: none"> • RSU-based VC. • MEC cluster-based VC. 	<ul style="list-style-type: none"> • RSU-based VC. • MEC cluster-based VC. • VC data plane slicing.
3	Control plane	<ul style="list-style-type: none"> • RSU-MEC controller. • VC controller. • SDVC controller. 	<ul style="list-style-type: none"> • VC network slicing controller. • VC controller.
4	Features and advantages	<ul style="list-style-type: none"> • Communication decisions are made closer to the VC. • Vehicle privacy is enhanced. • Self-organizing architecture of SD controllers. • SDVC can promote the ubiquity of VC services. 	<ul style="list-style-type: none"> • Implementing cloud-native applications using serverless and micro-services approach. • Create logical instances tailored to handle customized VC services. • Management and orchestration platform to provision VNFs. • NFV service chaining and decision learning based on ANN algorithms. • MEC application as a service over VC network slicing.

- Cloud frameworks on SDVC, technical frameworks for systematically organizing 5G enabling technologies and SDN to provide efficient VC applications;
- NFV technology in SDVC, concentrating on the practical deployment, programmability, and management of SDVC to meet various requirements;
- Data plane for SDVC, in terms of virtual network functions, and hardware to assist 5G vehicle-to-everything (V2X) communication modes.

Previous surveys on SDN in VCC (SDVC), vehicular ad hoc networks (VANETs) called SDVN, and 5G-based SDN (5G-SDVN) are summarized in Table 4. We focus on traditional VN challenges and related SDN-based VN architecture.

Wang et al. [22] present a comprehensive survey on SDVC architecture for VCC systems. They classify SDVN components to address issues of network topology and reliability on resource formation. Nkenyereye et al. [2] address the issue of VM migration in SDVC. They classify techniques for managing several VM migration scenarios. The MEC system at the network edge reduces the computation decision on VM migration process.

The SDVN architectures have been surveyed in [23–27]. Bhatia et al. [23] present a comprehensive explanation of SDVN components, VANETs access technologies, and protocols. The presented SDVN components are designed to improve issues related to VANET communication protocols. Bhatia et al. [24,25] focus on SDVN architecture proposed to mitigate security and privacy concerns in VANETs. They study the impact of SDN paradigm and privacy frameworks deployed in SDVN controller layer to design countermeasures. By classifying SDVN architecture to address the issue of energy and resources utilization for green IoVs, Chen et al. [28] propose a 5G-SDVN architecture for rational use of resources tailored to offer techniques that reduce energy consumption and resource utilization of IoVs' data plane. A unified SDN architecture for classic vehicular networks, 5G-VANETs, VC, and Fog vehicular networks are discussed in the survey presented by Mekki et al. [29]. The comparison between our work and existing surveys in term of focus of discussion is summarized in Table 5.

In contrast to the existing surveys depicted in Table 5, this survey focused on SDVN techniques toward extended SDVC by leveraging 5G enabling technologies such as NS, NFV, and MEC. ESDVC is referred to support advanced 5G-based V2X applications. A use case of ESDVC we discuss in the article is VC network slicing service.

To the best of our knowledge, existing articles that are relevant to our work include [2,22]. In contrast to our more extensive coverage of SDVC, Ref. [22] focused on the technical advantages of cloud computing and SDN that are applied to vehicular networks. Moreover,

discussions about the ability of SDVC for managing the migration scenarios of several virtual machines (VMs) are the main contributions of [2]. Different from these works, this survey focuses on these respects: (1) the architecture scenario of both SDVC and ESDVC architectures for enabling VC network slicing; (2) in terms of the three enablers, glancing into the entire technical spectrum of ESDVC architecture toward VC network slicing; (3) emphasizing that 5G solutions and SDVC are mutually beneficial for next VCC systems. The key contributions of this article are as follows:

- We provide the architecture of vehicular-cloud and mobile edge vehicular cloud-based architecture and their architecture for VC systems.
- We summarize techniques or methods-based SDN to address VC challenges through the SDVC architecture.
- We discuss the idea behind the extended software-defined vehicular technologies. An architecture, paradigm-based 5G enabling technologies toward the ESDVC, is provided. We present a summary of comparison between SDVC and ESDVC.
- We discuss the architecture of a VC network slicing service as a use case of ESDVC.
- We discuss techniques and methods based on SDVC from the literature. These techniques are classified into three layers. The application layer concerns the cloud framework services on SDVC. The control layer concerns software-defined control systems. The data plane includes the data plane for IoV systems.
- We discuss lessons learned and open challenges for VC network slicing that leverages ESDVC.

The paper is organized as follows: We have provided the background and motivations of this survey in this section. Next, we present some fundamentals related to vehicular VCC, and SDVC in Section 2. The architecture of extended SDVC (ESDVC) and its use case of VC network slicing are presented in Section 3. The next section introduces the three supporting technologies (cloud frameworks on SDVC, software-defined control techniques in SDVC, and data plane for SDVC) in Section 4. Finally, we present lessons learned and discuss open challenges in Section 5 and conclude this paper in Section 6.

Table 4. Summary of existing surveys on software-defined network-based vehicular networks.

Ref	VN Challenges				SDN-Based VN				Focus of Discussion
	Data Analytic	Energy and Resource Utilization	Network and Routing	Security	VM Migration	SDVN	5G-SDVN	SDVC	
[22]			{√}					{√}	SDVN components
[2]					{√}			{√}	MEC and virtualization
[23]			{√}			{√}			SDVN components
[24]				{√}		{√}			SDN applications
[25]				{√}		{√}			SDVN architecture
[26]			{√}			{√}			Advanced routing
[28]		{√}					{√}		Rational use of resources
[27]			{√}			{√}			SDN techniques
[29]			{√}				{√}		SDVN deployment
[30]	{√}					{√}	{√}		Big data pipelines
[31]				{√}		{√}	{√}		End-to-end security
[32]				{√}		{√}			Power model on IaaS

Table 5. Summary of existing surveys and our survey: techniques in SDVC and 5G enabling techniques toward extended SDVC (ESDVC).

Ref	Focus of Survey Discussion
[22]	Technical advantages of cloud computing and SDN applied to vehicular networks
[2]	Ability of SDVC for managing migration scenarios of several virtual machines (VMs)
[23]	Comprehensive explanation of SDVN components, access technologies, and protocols
[24]	Impact of SDN paradigm along with implementation SDN applications in vehicular communication
[25]	SDVN architectures against major security threats and future 5G information-centric networking
[26]	Advanced routing schemes based on SDN and named data networks for VANETs and ITSs
[28]	Rational use of resources to enable the sustainable development of IoVs
[27]	SDN techniques tailored for VANET domain in terms of architecture and communications requirements
[29]	SDVN deployment in classic vehicular networks, 5G-VANETs, vehicular cloud, and vehicular fog
[30]	Comprehensive overview of big data and data analysis pipelines for smart 5G
[31]	Comprehensive framework for end-to-end security mechanisms in 5G-SDVN
[32]	Power models on IaaS for enabling VM migration
our survey	Techniques toward extended SDVC by leveraging 5G enabling technologies such as network slicing, MEC

2. Fundamentals of Software-Defined Vehicular Cloud Computing

Because of its advantages in reducing data transmission and massive computation resources, SDVC improves VC service latency and eases cloud computing constraints. VCC has emerged as an important solution to break the bottleneck of emerging technologies. VCC architecture will become a more important complement to SDN technology, even allowing SDN to fulfill the architecture of advanced VC services such as autonomous cars. More detailed information can be found in [33,34].

2.1. Models of Vehicular Cloud Computing

In the development of VCC, there have been various models aimed at working at the edge of vehicles, with same principles but different focuses, such as RSU cloud [35,36] and mobile edge computing cloud (MEC cloud) [37]. In this section, various VCC concepts are introduced and distinguished. A summary of VCC models is presented in Table 6.

Table 6. Summary of vehicular and mobile edge cloud computing architecture: proposed layers and advantages.

Works	Proposed Layers	Criteria	Advantages
2019, [36]	<ul style="list-style-type: none"> RSU cloud. 	<ul style="list-style-type: none"> Local RSU cluster. V2X communication. 	<ul style="list-style-type: none"> Collaborative task scheduling. Resource sharing.
2021, [35]	<ul style="list-style-type: none"> RSU-based VC. Mobile cluster-based VC. 	<ul style="list-style-type: none"> RSU-aided cluster. Mobile computing. 	<ul style="list-style-type: none"> Gradually exposes RP vehicles. Situation awareness applications.
2019, [38]	<ul style="list-style-type: none"> RSU cloud. Central cloud. 	<ul style="list-style-type: none"> Computation on VE. Computation on the cloud. 	<ul style="list-style-type: none"> Offload tasks to the cloud.
2022, [39]	<ul style="list-style-type: none"> Vehicular layer. Edge layer. Core layer. 	<ul style="list-style-type: none"> Link vehicle to the core layer. Distributed edge servers. 	<ul style="list-style-type: none"> Bring cloud resources closer to road users. Reduce latency. Task offloading to edge servers.
2013, [33,40]	<ul style="list-style-type: none"> Vehicular cloud. RSU cloudlet layer. Central cloud. 	<ul style="list-style-type: none"> Virtual resources. Migration of VMs. QoS. 	<ul style="list-style-type: none"> Load balancing. Maximize resource utilization. Minimizes migration cost.

2.1.1. Road Side Unit and Road Side Unit-Aided Cluster Vehicular Cloud

A local RSU cloud groups together a set number of neighboring vehicles. V2X communications are used to create an ad hoc inter-vehicle network. RSU cloud sites include vehicles and anything in direct communication. They collaborate to build a VC service

in which computing and networking resources are shared among partaking vehicles [36]. RSU-aided cluster VC constructs a topology in which the RSU directory and cluster head vehicle directory are merged to gradually expose the resources of providers vehicle [35]. Thus, RSU-based VCC opens up a plethora of new possibilities for executing situation-aware applications without relying on standard cloud infrastructure resources [38].

2.1.2. Mobile Edge Vehicular Cloud

The European Telecommunications Standards Institute (ETSI) has proposed MEC [36]. MEC is an advanced technology that includes distributed servers at the edge of the network to support multiple applications and services that require ultra-low latency. In MEC-assisted vehicular systems, a computation mobile application can be executed on the vehicle itself (local execution) or offloaded to an MEC server (edge execution) for computation. The MEC-assisted vehicular systems provide a supply of various V2X-based cloud services through the distribution and integration of distributed MEC-assisted vehicular servers and cloud radio access networks according to the location of V2X users [41].

A distributed MEC cloud that provides a variety of V2X-based cloud services has been proposed in [39]. The architecture integrates distributed edge cloud servers and cloud radio access networks based on the location of V2X users. A V2X user connects to a MEC via vehicle-to-network (V2N) or cellular network using cellular V2X (C-V2X) communication. Vehicles can offload computation-intensive tasks to the MEC by sharing computing resources as a service to nearby consumer vehicles. The MEC server instantiates, customizes, and delivers dedicated resources to the nearby vehicle joining the VC cluster. Figure 1 shows the use of dynamic VM synthesis to deliver a VM state to infrastructure [33,40]. A vehicle chooses a MEC server in the vicinity of the radio coverage area. To use MEC server resources, a vehicle customizes transient resources on it. It establishes a transient VM when it moves into the range of MEC servers. The latter customizes a transitory cloud service that the vehicle will use for a limited time.

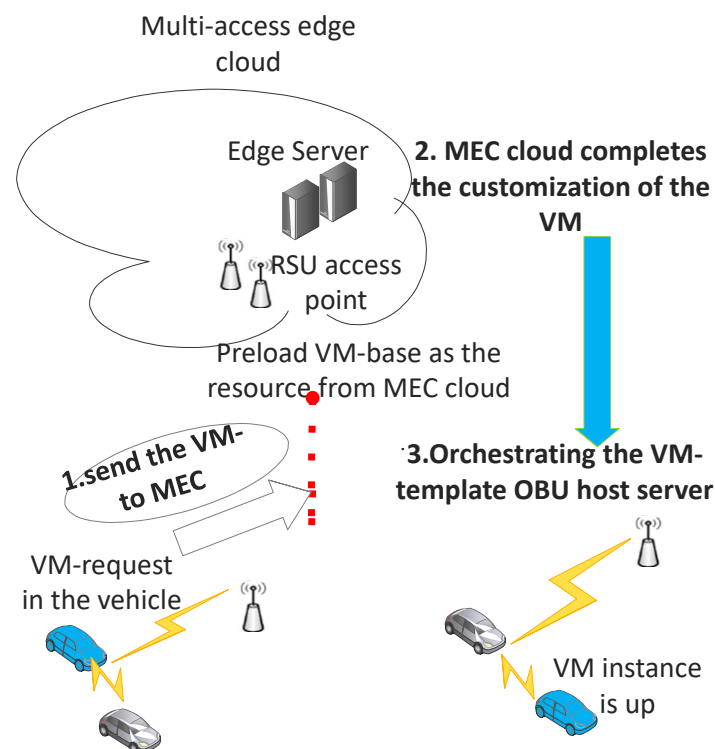


Figure 1. Dynamic VM orchestration approach to instantiate a VM for a VC application at MEC VC.

A VC forms a local cloud that clusters a certain number of neighboring vehicles. The vehicles and everything in direct communication are considered mobile cloud sites. They

offload computation-intensive mobile applications to the remote cloud for computation. However, the transmission delay to and from the cloud may not meet the ultra-low-delay requirement for advanced vehicular applications.

2.2. Software-Defined Network

The objective of an SDN is to decouple the control plane from the forwarding plane. SDN defines a methodology where the control plane can be centralized, moving its functionality from the network device(s) to a central device or cluster. This decouples the forwarding and control planes. The network devices receive, from the control plane, relevant routing rules from the application plane. The control plane allows the network device to perform pure forwarding plane functions. In an SDN-based implementation, the control plane may be managed through an application, which can interact with the control plane. This application plane abstracts device information and configuration as well as network topology and traffic path information from the control plane. The application can therefore maintain a complete consolidated view of the network and use these pieces of information to make decisions that are passed down to the control plane, then to the data plane.

Figure 2 depicts the fundamental logical architecture of an SDN. It consists of three layers, namely the application layer, the control layer, and the network layer. A standalone device known as the SDN controller cluster serves as the SDN control layer and transmits control choices to networking devices. In order to make appropriate control decisions, the controller also receives information from various devices. It employs protocols known as the SDN control protocol to communicate with the devices. Northbound and southbound protocols make up the SDN control protocol. The northbound interface is used by the SDN controller to communicate with the application layer. The northbound protocols provide programmatic network management for application developers. The communication from the control layer to the lower layer (network layer) is referred to as southbound communication. Moreover, OpenFlow-based network devices [42] accept user-defined policies that can be used to preset the routing algorithms. The application layer notifies the control layer of new paths and requests that the control layer begin using them. Programs and scripts that automatically respond to expected and unexpected events can be directly integrated into the controller since the SDN centralizes control and intelligence in the SDN controller.

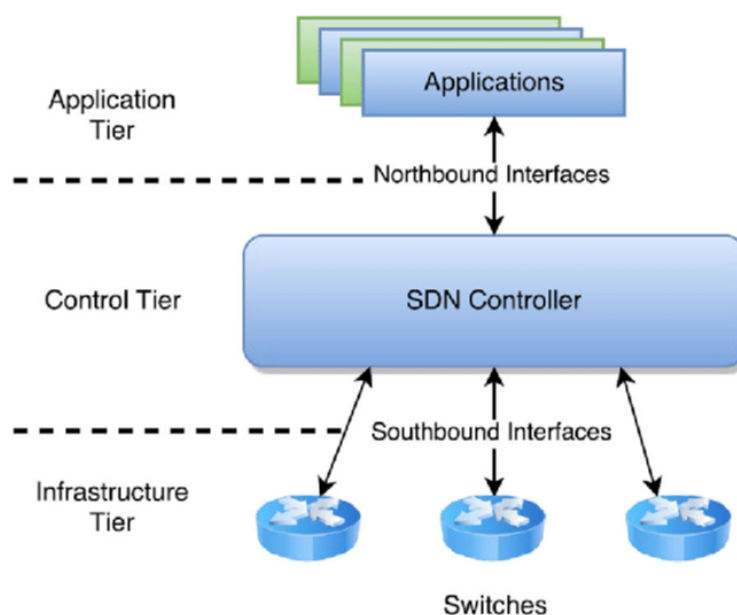


Figure 2. . Logical view of a software-defined network.

2.3. Software-Defined Vehicular Cloud Network

Figure 3 presents the logical structure of an SDVC network. It is composed of the data plane, the control plane, and the application plane.

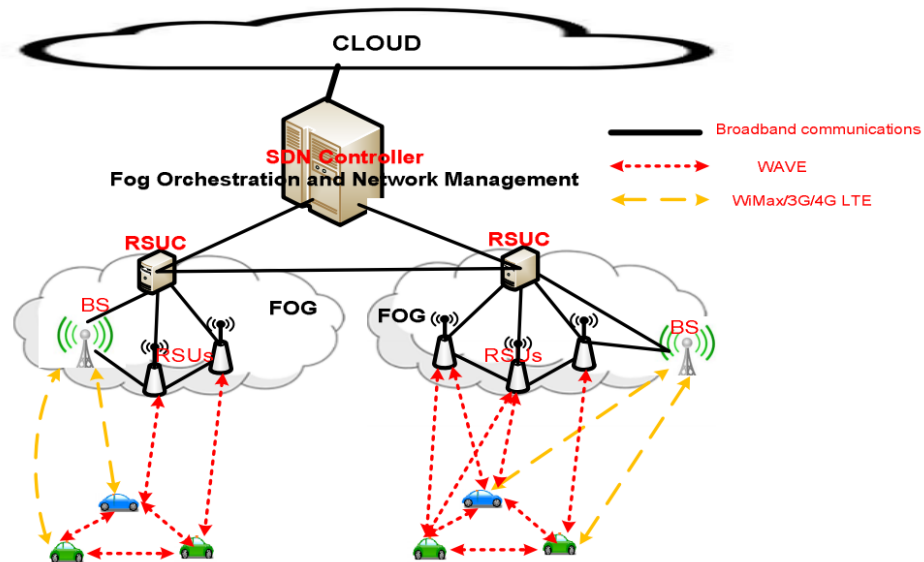


Figure 3. Logical view of software-defined vehicular cloud network [43].

The data plane includes vehicles and an OpenFlow-based onboard Unit. They are considered forwarding devices and have the ability to exchange data with the control plane. Some of the function modules in the data plane allow:

- Collecting in-vehicle sensor information such as about speed, the surrounding nodes (e.g., adjacent vehicles, pedestrians on the roads, weather conditions), and the current position information.
- Exchanging vehicle supporting V2V application messages.
- Communication between the OpenFlow-based onboard unit and the SDN controller located at the edge of the network by leveraging a distributed MEC server placed to reduce network delay in processing sensed data, then improve routing protocols.

The control plane includes an OpenFlow SDN controller. The SDN controller is configured to dynamically allocate V2V resources and improve data transmission. Therefore, the control plane takes charge to control data flow as well as to alter the characteristics of the routing devices by drawing the global network topology. Based on data information forwarded from the data plane, the SDN controller implements programs and scripts that automatically react to expected and unexpected events based on rules defined through software modules provided by the application plane without the need to deal with proprietary interfaces between the data plane and control plane. To support the above functions of the control plane, the SDN controller in the SDVC is configured with the following function modules:

- Information collection module—provides global knowledge of the RSU or fog-assisted VC based on the data information from the data plane.
- Communication decision module—monitors the link status of the decision of using unicast or broadcast communication to vehicles clustered to one RSU or fog-assisted VC.
- Caching V2X messages module—saves the V2X application data at RSU or MEC-assisted VCs to reduce the transmission delay for V2X services.
- Network status module—monitors and queries the network state of the SDVC.

The application plane directly faces different VC service requirements for VC applications. Based on the VC service requirements, rules and instructions of SDVC are generated and forwarded to the control plane.

Three (3) classification criteria were used to perform the investigation of existing studies on SDN-based vehicular networks. First, we highlighted the specific problem that the researchers tried to solve. The second factor is the classification of the SDVN or VANET systems currently in use that are proposed to address VN issues. The third is system analysis. This helps to pinpoint SDVN systems that are pertinent to the issue. The summary of existing works on SDN-based vehicular networks is described in Table 7.

Table 7. Summary of SDN-based RSU, fog/edge, and 5G-based vehicular networks grouped according to vehicular network challenges and SDN methods.

Challenge	Ref and Year	Architecture	System Analysis
Latency	[44], 2019	Vehicular networking heterogeneity of radio access technologies	<ul style="list-style-type: none"> Redesign of the present vehicular network, SDN controller, and vehicle network architecture for resource management. Model SDHVNet architecture.
	[45], 2017	Software-defined mobile edge computing	<ul style="list-style-type: none"> Latency-controlling techniques: base station radio access steering (BSs). Vehicular cloud/edge networking using software. Local awareness of nearby nodes, an SDN controller, and a broadcast beacon message.
	[11], 2017	Software-defined VANET with 5G	<ul style="list-style-type: none"> Network model integration with cellular networks, SDN control of eNB infrastructure, and RSU controller controls RSU. The optimal decision between the vehicle and the controller is achieved by optimizing southbound communication using a rebating mechanism, game equilibrium, and a two-stage leader-follower game.
	[46], 2016	Software-defined VANET with 5G	<ul style="list-style-type: none"> VANET-based, cellular network-based, and hybrid-based control communication.
Resource utilization	[43], 2015	Software-defined cloud/fog network	<ul style="list-style-type: none"> To deliver FSDN, the fog computing idea is still being included. The control plane is shared by the SDN controller, BS, and RSU in the hybrid mode that SDN enables.
	[47], 2017	Software-defined VANETs	<ul style="list-style-type: none"> SDN controller topology, Mininet-WiFi node model. Expand the vehicle node's modeling in Mininet-WiFi.
Loss of connectivity	[19], 2017	SDVN	<ul style="list-style-type: none"> SDN controllers are arranged hierarchically to reduce connectivity latency between them. Clustering for local SDN controller domains.
Heterogeneity of wireless infrastructures	[8], 2016	SDVANETs	<ul style="list-style-type: none"> Provide an adaptive protocol for heterogeneous multihop routing, reduce the administration burden of SDN by monitoring the status of SDN switches, and enable V2V, V2I, and V2N with SDN. As SDN switches, abstract heterogeneous wireless nodes are enabled. OpenFlow uses SDN controller to allocate network resources. Monitor message overhead between the controller and the vehicles.
	[48], 2014	SDN-based routing	<ul style="list-style-type: none"> Regulate the SDN controller's overhead and the packet delivery ratio.

In order to control the overhead message on the SDN controller, the model proposed by Ku et al. in [48] includes a new routing protocol that enhances the packet delivery ratio by choosing stable routes with the lowest latency. Zongjian et al. in [8] provide system analysis that focuses on abstracting heterogeneous wireless nodes as SDN switches enabled OpenFlow and designing an SDN controller to dynamically manage network resources due to the inability to change protocol deployment due to the heterogeneity of wireless infrastructures. Correia et al. in [19] proposed system analysis based on choosing local SDN controller domains through clustering principles to increase the performance in communication by reducing the connectivity loss between vehicles and central SDN controller.

Truong et al. in [43] researched the issue related to fog (edge) computing technology; as a result, a fog network based on SDN is examined to offer location-aware services with lower communication latency. The deployment of SDN to support hybrid mode (central-based and distributed-based configuration of the SDN controller) and on the configuration of the control plane (SDN controller) in distributed mode with both the base station (BS)

and RSU are the main focuses of the system analysis in order to achieve this. The system study presented by the Fontes et al. in [47] focus on the deployment of SDN controllers and the potential to model communication nodes (vehicles) as an SDN switch using the free and open source simulation program known as Mininet-WiFi.

Deng et al. in [45] look into the prospect of using software-defined mobile edge computing to deliver a VANET application with low latency and high reliability communication delay. The edge vehicular network design is taken into account in the system analysis. They provide a modeling solution on how to reduce communication latency using radio access steering at the BS. A systematic analysis based on the management of latency at the VANET position, the cellular network, or hybrid basis is provided by the modeling and implementation of VANET and cellular position [46]. The results of the system analysis prompt Mianxiong et al. in [46] to model their approach to reduce communication latency by optimizing southbound communication using a rebating mechanism and the game equilibrium connected to the two-stage leader-follower game. The two proposed algorithms allow choosing the best routing paths between the vehicles and SDN controller. The modeling ideas by Dhawankar et al. in [11] center on the system analysis based on broadcasting V2V beaconing messages so that the SDN controller can access local information about the network topology of nearby nodes. Following system analysis, the suggested solution offers a paradigm that combines a 5G-based SDN concept with an eNB infrastructure, an RSU controller, and an SDN controller.

3. The Architecture of Extended Software-Defined Vehicular Cloud

In this section, we provide an extensive description of ESDVC and its related functional layers. The architecture of ESDVC is presented. A model of vehicular network architecture based on SDN, MEC cloud, and central cloud computing is described in [4,13]. The architecture enables the deployment of a cloud-based vehicular network. Its SDN controller provides several critical networking solutions. SDN also enables on-demand network connectivity and fully programmable capability in deploying meaningful cloud-based applications. SDVC facilitates V2V communication. According to resource management strategies, computing resources are available on both the cloud and the MEC. Three types of resources are shared: computing resources, storage resources, and bandwidth resources.

3.1. 5G Enabling Technology for the Extended SDVC Framework

By reviewing and analyzing literature on 5G, we found that the 5G ecosystem is a key driver toward the adoption of (edge) cloudification solutions in V2X. V2X requires a very low-latency environment for a important part of the correlated use cases. Autonomous driving is the most critical example in this respect. A list of vehicular cloud applications with their latency requirements over SDVC from previous work [2] is presented in Table 8.

NFV uses softwarization techniques to implement network elements that are deployed on general-purpose servers (i.e., industry-standard servers, switches, and gateways) [49]. Consequently, service providers reduce cost on capital and operational expenses (CAPEX/OPEX) because they do not require to launch network services that require new network infrastructures. Network functions are virtualized instead of running on their own physical network infrastructure. Furthermore, NFV abstracts physical network resources and decouples virtual network functions (VNFs) from the underlying infrastructure for ensuring a hardware-independent life cycle [50].

NS is the process of dividing a physical network to create a logically divided network instance that can provide users with a dedicated network specialized for the service. Network resources such as virtualized server resources and virtualized network resources are guaranteed for each network slice. Therefore, separate network slices are sequestered from other such that inconsistencies or failures in one network slice do not destabilize communication between them. NFV and SDN technologies support the practicability of NS service.

As depicted in Figure 4, the existing techniques are classified according to their contributions for the adoption of (edge) cloudification toward the construction of ESDC. Vehicular cloud network slicing service is the use case in this respect. These techniques are identified as an aggregation of two paradigms: SDVC and 5G enabling technologies (NFV, MEC). SDVC is mostly developed from SDN and VCC, in which the communication among networking nodes is managed through an SDN controller located at hop or is multi-hop-based. Besides that, NS and NFV have emerged from SDN, where the primary concern consists of flexibly supporting the diverse and even conflicting demands of 5G use case services. To meet the latency requirements of V2X applications and facilitate the exchange of V2X-related information between (mobile and fixed) nodes via the underlying communication network, MEC provides fast access and distribution of information. The paradigm evolution of VCC toward ESDVC is summarized in Figure 5.

Table 8. Application-based vehicular network over SDVC.

Potential Application	Relevant SDN Controllers Assistance			Resource Sharing			
	SDN OpenFlow Controller	SDN Communication Infrastructure Controller	Active VM Controller	Computation	Storage	Bandwidth	Latency (ms)
Real-time traffic condition analysis and broadcast	✓	✓	✓	✓	✓		10–100 [34]
Video surveillance		✓	✓		✓		120–200 [51]
Mobile social networking	✓	✓	✓	✓		✓	10–100 [52]
In-vehicle multimedia entertainment		✓	✓	✓	✓	✓	10–100 [34]
Remote vehicle diagnostics	✓	✓	✓	✓			10–100 [34]
Location-based services recommendations based on personalized and precise service [53]	✓	✓	✓	✓	✓		10–65 [53]

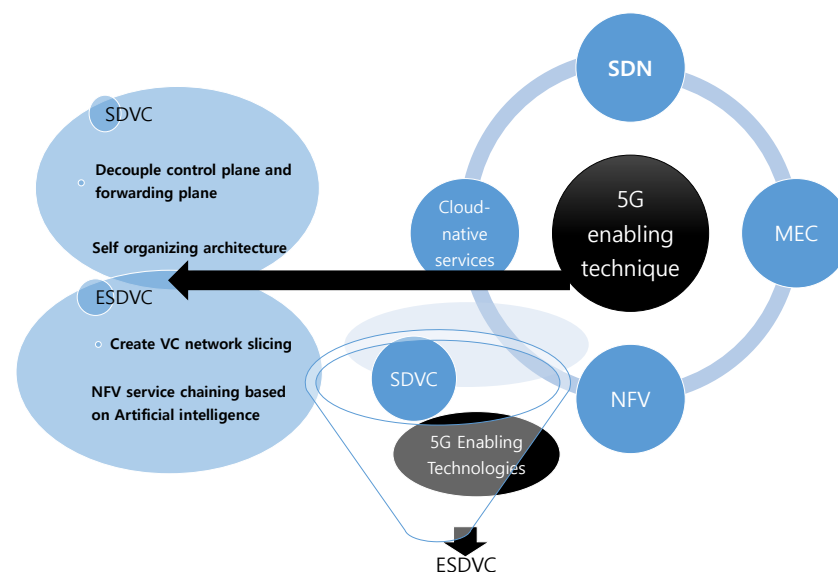


Figure 4. The 5G enabling technologies (e.g., NFV, MEC, and NS) integrated with SDVC for constructing ESDVC.

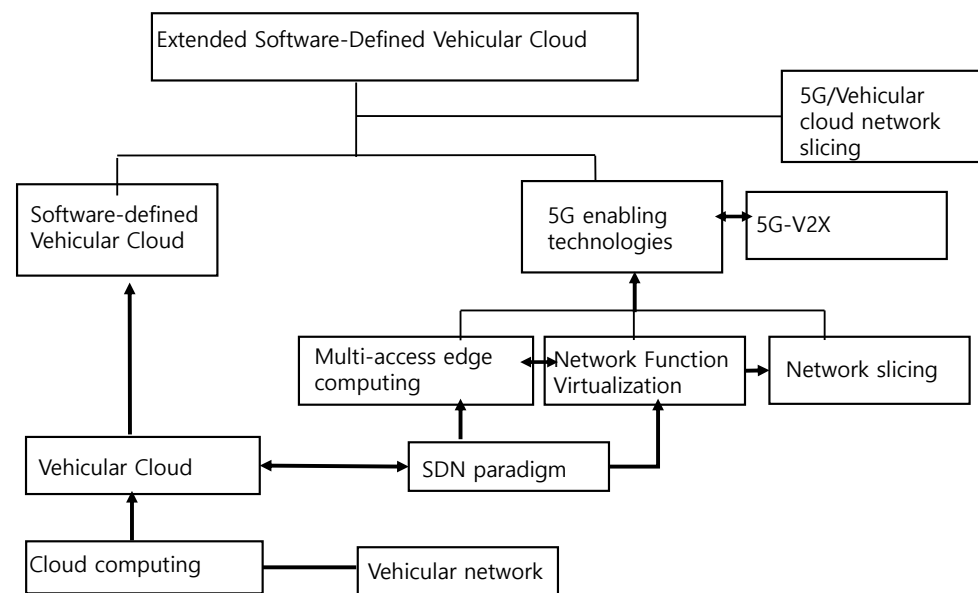


Figure 5. Paradigm evolution toward extended software-defined vehicular cloud networks.

3.2. Extended SDVC for VC Network Slicing System

Figure 6 shows the NFV technique in ESDVC architecture. The vehicles are considered as SDN devices with OpenFlow-enabled Open vSwitch [13]. This architecture follows the 5G enabling techniques such as NS as discussed in [2,21]. The ESDVC architecture grants to SDN devices the ability to support virtualization technology (e.g., hypervisor, docker container) such that VNFs are dynamically created and removed through an orchestration platform to ensure their life-cycle management. On the vehicle, virtualization software installs cloud and edge micro-service-based VNFs. The architecture involves the presence of MEC infrastructure such that the computation of data would be processed at the network edge instead of the remote cloud infrastructure. A MEC micro-data center is a collection of MEC servers equipped with virtualization infrastructure that provide computation, storage, and network resources for cloud/edge-based vehicular applications [13]. The MEC servers implement the features adopted by NFV infrastructure compliant with ETSI [54]. The dedicated VNFs support cloud-based vehicular service computation resources. For instance, in case the vehicle moves too far away from the current MEC micro-data center, the ESDVC controller collects the location of the next MEC micro-data center to ensure the relocation (migration) of VNFs. The continuity of computation is such a solution to vehicle mobility in VC network slicing because the life cycle of VNFs is efficiently supported through logical functions customized to a specific challenge in the ESDVC system. MEC computing is used in ESDVC-assisted decision making process [55].

3.3. Architecture of ESDVC Compared to SDVC

The ESDVC architecture is shown in Figure 7 (right side). Figure 7 (on left side of Figure 7) shows the SDVC architecture. The architecture includes vehicles considered as SDN devices. It is also composed of the OpenFlow-based virtual switch [13]. The SDVC architecture in [13] integrates SDN devices that host virtualization infrastructures (VIs). Furthermore, RSUs co-located with evolved node base (eNB) and MEC host expose third party applications at the network edge [55]. The SDN infrastructure communication controller maintains various VC networks that have been established to allow their management [4]. The design based on distributed SDN controllers improves the allocation of virtualized resources pool and the fault tolerance of the entire SDVC system. The VC controller is in charge of managing VC formation while the RSU/base station controller is in charge of managing the traffic and control rules about the 5G infrastructure, RSU units, and MEC platform enabled for 5G vehicular cloud [56].

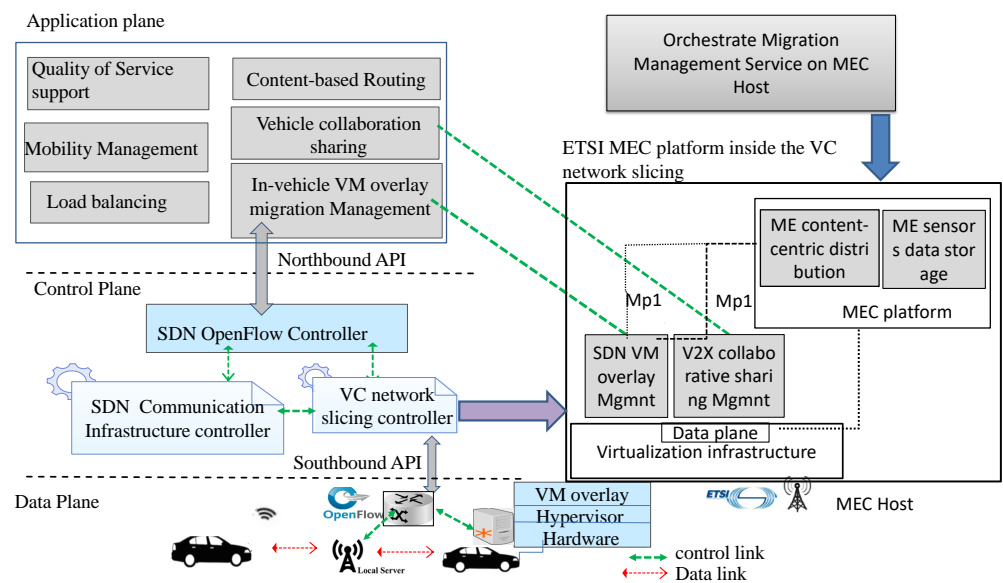


Figure 6. The 5G vehicular cloud-enabled SDN (left) and MEC server (right) for vehicular cloud network slicing.

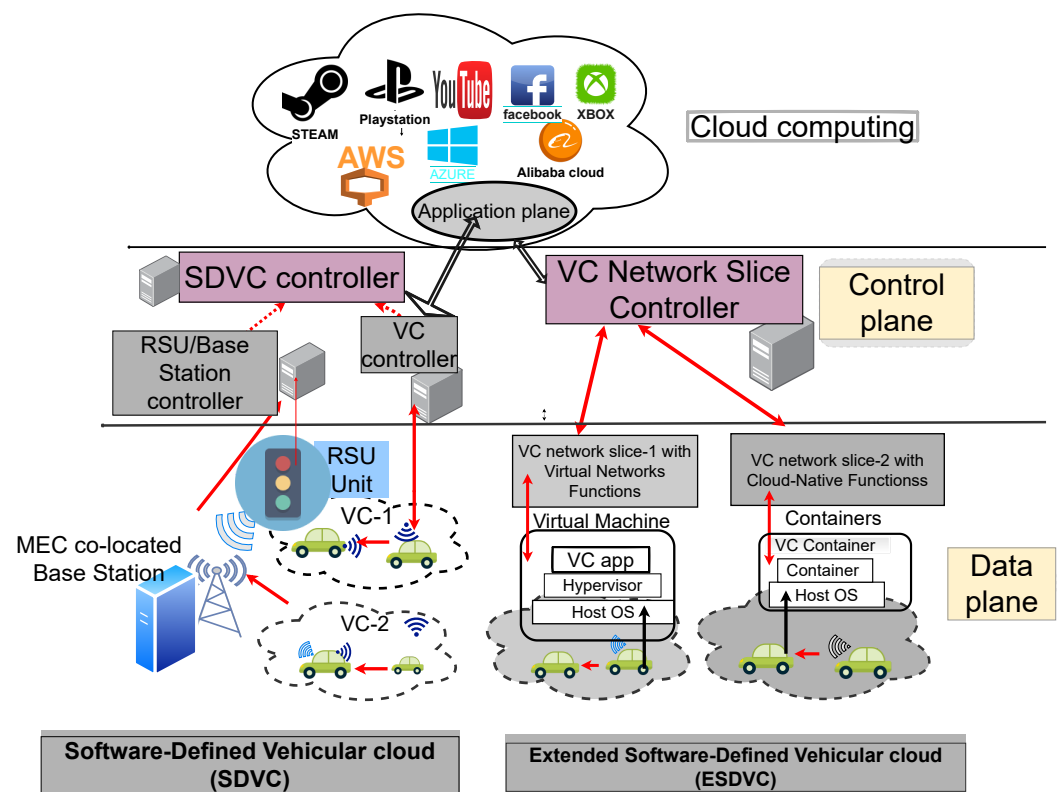


Figure 7. High-level architecture for software-defined vehicular cloud and extended software-defined vehicular cloud.

Based on NFV, the VC network architecture is customized to meet the URLLC service. For instance, by processing real-time traffic conditions with a latency of 100 ms, the advanced driver assistance system outperforms the best human driver in terms of safety [57]. The in-vehicle host server is part of the VC network slicing. The cloud-native [58] applications deployed on top of hardware inside the vehicle abstract resource utilization information required by the VC slicing controller to deliver requirements features associated with the current provisioned service. SDN VC slicing controller is in charge of

managing the communication of vehicle cloud site subscribed to a service with constraints requirements, then isolated from other VC network system supporting cooperative tasks. The MEC server inside the VC network slicing manages the orchestration of cloud-native applications services (e.g., replication and migration). During VM migration and resume procedures, the wireless access communication radio is chosen by the controller to connect the central cloud or MEC network resources. The SDN OpenFlow controller is referred to the primary SDN controller.

4. Technologies in SDVC and ESDVC

Through an extensive review of the literature, we design an architecture for both SDVC and ESDVC. The SDN architecture has three functional layers: the application layer, control plane layer, and data plane layer. The design architecture follows the three functional layers. We classify three categories of technology, and each category matches each of the functional layer.

As illustrated in Figure 8, “cloud frameworks on SDVC” on top as the application layer corresponds to the SDVC and ESDVC theoretical goals. To support them, various cloud frameworks should be designed using intensive cloud computation or pushed to run on the VC site. The intelligence, control, and management of policies from cloud framework services are handled by the “software-defined control techniques” that match with the control plane layer. It includes a various number of controllers located either in the cloud or at network edge. We can list the SDVC controller that manages both the VC controller and VC slice controller. By leveraging NFV in SDVC, “VC slice controllers” support virtual VC network slicing orchestrated to meet customized QoS for specific VC services. The “software-defined control techniques” focuses on a variety of techniques that support the efficient formation of VC network slicing in VC computing via VC controllers. Finally, the data plane layer that is labeled “data plane for SDVC” includes all techniques that modify vehicular computing networks and communication access to better serve 5G-V2X communications. Upon a comprehensive overview of the most relevant works in SDVC and ESDVC, we summarize all these techniques in the taxonomy presented in Figure 9.

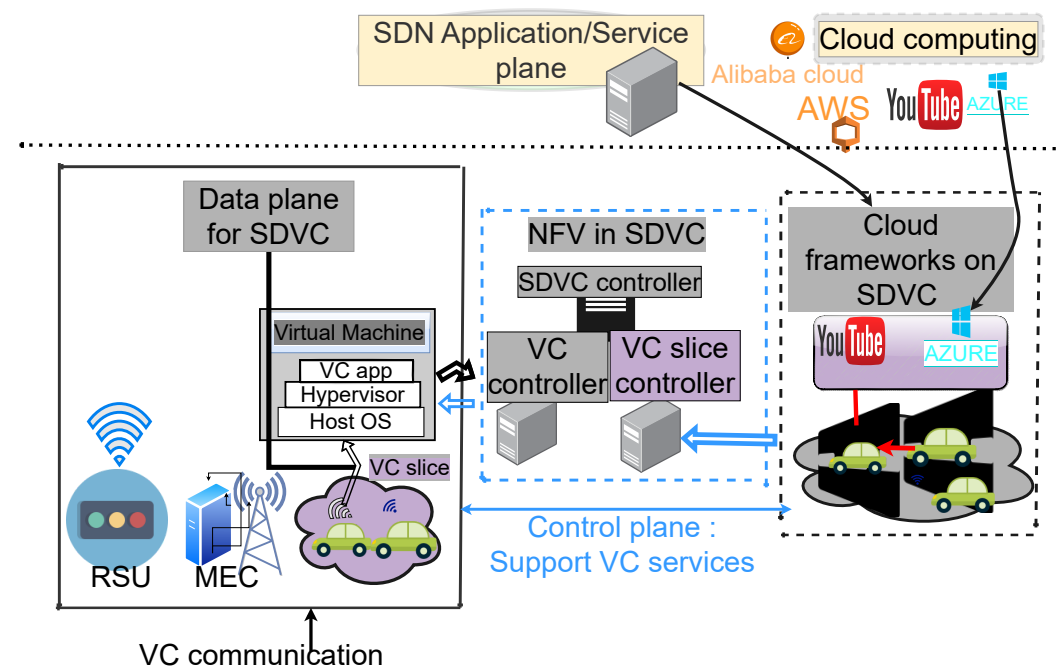


Figure 8. Technologies that are essential for the convergence of SDVC and 5G enabling technologies.

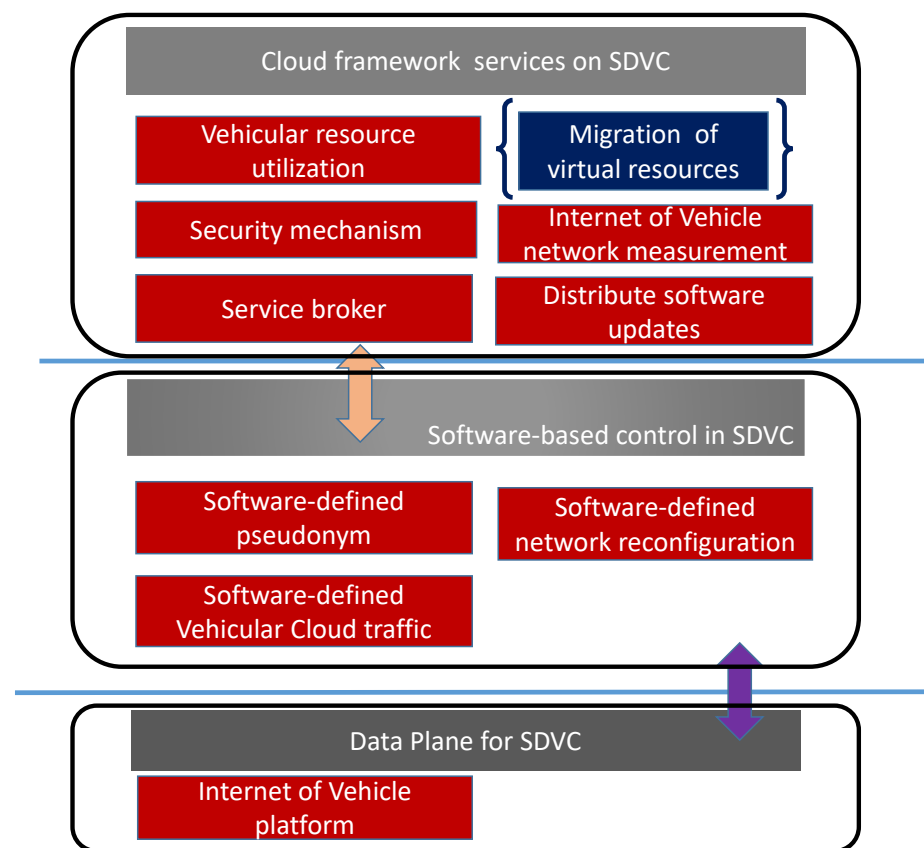


Figure 9. Taxonomy of SDVC methods/framework from the literature.

4.1. Cloud Framework Services on SDVC

In general, cloud framework services on SDVC are deployed in edge computing or in cloud data centers for providing vehicular applications in SDVC. As a result, SDVC-based cloud frameworks and existing security mechanisms would support the involvement of SDVC services. In the following, we present cloud framework services and emphasize their advantages over competing architectures without SDN technologies in VC. The summary of works related to cloud framework services on SDVC is presented in Table 9.

4.1.1. Vehicular Resource Utilization Services

Vehicular resource utilization in SDVC is important in various VC applications. VCC system uses computing resources of RSUs to fulfill service requests from vehicles. It focuses on allocating limited resources to improve VC applications for road safety. In general, applying vehicle equipment (VE) of vehicles with cloud for resource sharing enables the execution of intensive-load tasks. Unfortunately, sharing resources with cloud computing incurs high network consumption when the execution requires the download of some input data from the cloud to be completed on RSU nodes. Moreover, unexpected latency and reliability issues while exchanging commands and instantiating on virtual instance have prompted the SDVC to deliver potential solutions. Jang et al. in [4] proposed SDVC architecture for centralized flexible VC control and efficient resource utilization. Besides that, the architecture targets the deployment of flexible and programmable infrastructure as a service (IaaS) which aims to allow the VC controller to select reliable set of vehicle providers that will expose a pool of computing required to instantiate the VC-based IaaS. On the other hand, users obtain and use multiple VMs running in the chosen vehicle providers for their own workload. IaaS examples include collaborative content download and public traffic data collection. Moreover, a programmable architecture based on connected dominating sets (CDS) and SDVC approaches support the selection of dominant resource provider (RP) vehicles [59]. The vehicles inside CDS are selected to form a virtual backbone

(VB). The VB is composed of a group of parked vehicles. They connect other vehicles to the Internet. The pool of their computing resource reduces end-to-end delay. Authorized vehicles in this backbone are connected to the cellular link to act not only as relayed nodes but also as fog devices. They act as intermediaries between the cloud server and end-users. The SDVC manages the radio resources shared with user equipment (UE) and other machine-type-communication (MTC). Moreover, the SDVC controller hosts protocols rules for efficiently organize the communication between VBs and the fog computing (FC). The FC services are hosted on VBs as well. The primary goal of hosting FC on a virtual backbone is to bring services closer to mobile devices in order to ensure network scalability, reduce data processing and transmission delay, distribute data storage, and save cellular network resources. Wang et al. in [22] presented an architecture, applications, and challenges of a cloud-enabled software-defined vehicular networks (cloud-enabled SDVNs). Theoretical methods of both cloud computing and SDN are applied to vehicular networks, particularly the management of resource utilization by the SDN controller. The framework exploits the sharing of resources among nearby cloudlets. The VC controller allocates resources based on decision making from the application plane service.

4.1.2. Migration of Cloud Vehicular Resources

RP vehicles are potential candidates for hosting VMs. In some VCC scenarios, VMs are provisioned using vehicular resources. Their resources are transferred from one vehicle to another to continue their execution. The SDVC presented in [2] provides use cases of VM migration by leveraging the presence of SDN controller and active VM migration application delivered by the MEC platform. The SDN control exposes modules that dynamically update the VC network topology and wireless communication interface equipped in the vehicle node. Such vehicle hosts various dedicated VMs that continually offload information on vehicle position and resource consumption to the SDN controller. The latter uses this information to efficiently migrate those dedicated VMs from one distributed MEC micro-data center to the next in order to resume the current computation tasks. The VM migration of the active VM controller is decided based on data collected from the SDN communication infrastructure controller. The decision is whether or not an active VM controller should be migrated with the vehicle as shown in Figure 10. The control plane includes the MEC server. In the context of SDN-assisted VM migration, imperative application modules such as V2X collaborative sharing and SDN overlay VM migration management are in charge of resource sharing and VM migration, respectively.

When two V2X-MEC applications are served in different radio access network coverage areas, the inter-MEC migration service carries out VM migration based on the short distance separating the current V2X-MEC micro-data center to the next one. In fact, the exchange of network topology between adjacent active VM controller facilitates the migration of the dedicated VM. As shown in Figure 10, the active VM controller is still running on its original MEC micro-data center-1. Consequently, the MEC micro-active datacenter's VM controller updates routing paths to decide rules in the forwarding switches for new packets coming from the central cloud service. The radio access forwarding device would then take actions based on the type of VM migration specified in the incoming packets field of flow entries. The communication process between different actors while migrating VMs from one MEC to another is shown in Figure 11. A micro-MEC service provider must decide whether to migrate a vehicle's corresponding virtual machine when it moves into another RSU. An active VC slice controller has its memory image transported from the source MEC to the destination, incurring migration costs, while execution costs are unavoidable and vary depending on the communication distance between the MEC server and its VM. Yao et al. [60] take into account a discrete time slot model where VM migration only occurs at the decision point, or the end of one time slot, and where the location of all VMs in the final time slot is known. In addition, the remaining time for each VC cluster to arrive at the endpoint will be estimated by the SDN controller in order to choose the best MEC server for the migration process. As the MEC server might not have sufficient computing resources

to resume the active VC slicing controller, it is assumed that the SDN controller selects the best MEC server according to the placement decision of all RSUs and the locations of all VC cluster (by mobility tracing of vehicles). This is because a target high-efficiency node selection strategy is required to migrate the VM to the proper host [61].

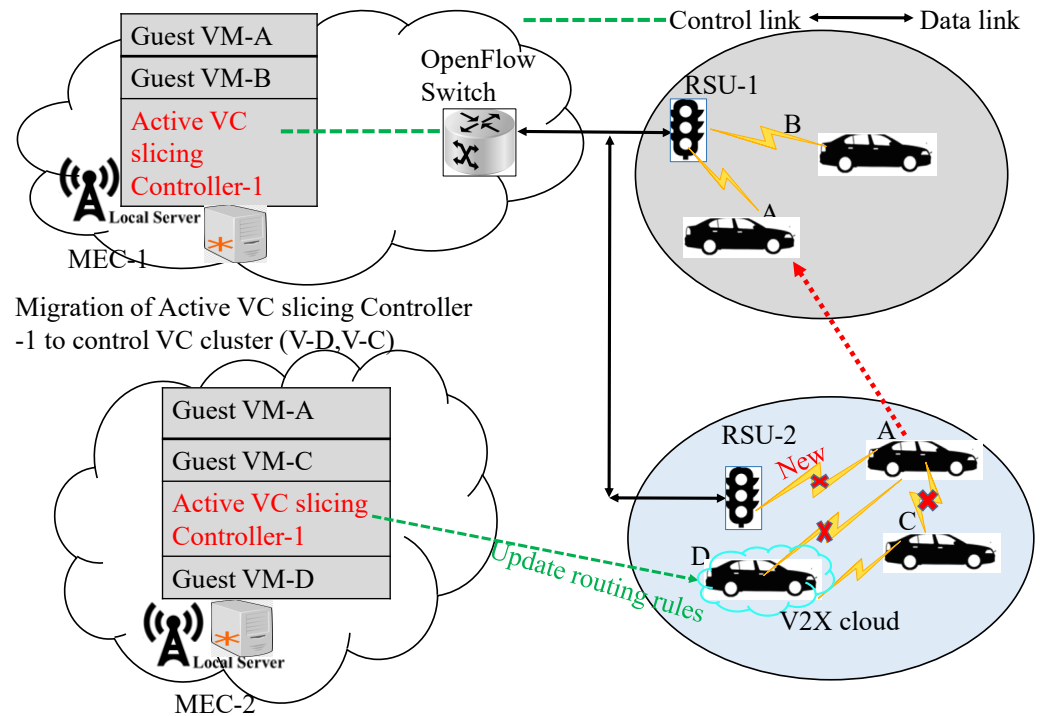


Figure 10. Active controller migration scenario in the software cloud vehicular network: intra-MEC VM migration case.

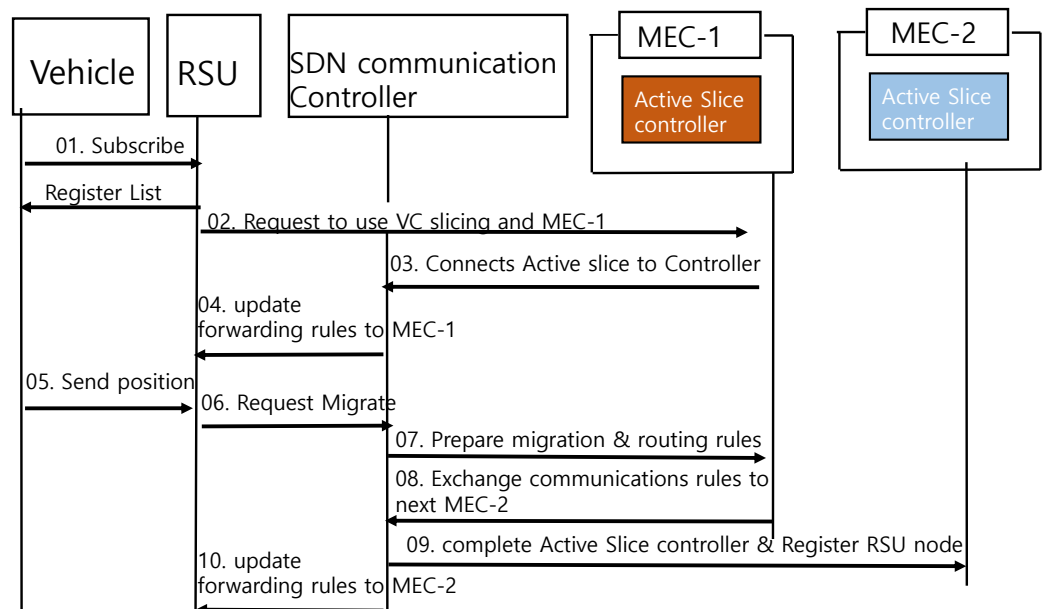


Figure 11. The interaction between different actors while migrating VMs from one MEC to another.

4.1.3. Security Mechanism

SDVC security and privacy concerns have yet to be addressed. IEEE 1609.2 specifies the use of public key infrastructure (PKI) in traditional VANETs. The PKI employs public key and private key pairs. The PKI, on the other hand, cannot guarantee privacy because SDVC uses a different network architecture than VANETs. In order to improve security, Kim et al.

in [62] proposed a security mechanism for SDVC that would satisfy fundamental security requirements such as authentication, message integrity, non-repudiation, confidentiality, and privacy. The SDVC proposed for security mechanism includes the control plane. A trusted third party (TTP) is in charge of assigning the public and private key pairs as well as the vehicle's certificate in the control plane. The centralized SDVC controller performs tasks such as vehicle information collection, VC formation, and key pair generation.

4.1.4. Serving Broker

VCC envisions a wide range of services. To support various QoS, responsive resource management solutions are required. In fact, dynamic resource management solutions involve RSUs. They act as event subscriber and publisher broker among the moving vehicles. The deployment of RSUs acting as broker should be maximized by effectively optimizing their placement. The physical factors to RSU placement are traffic patterns and mobility of moving vehicles. On the other hand, cyber factors are communication protocols and messaging. In order to meet these requirements, the SDVC architecture in [63] assists users in VC to receive highly efficient services. The SDNBroker application is designed in that work and superimposed on the SDN controller through northbound interface (NBI). SDNBroker uses the linear programming algorithm to schedule cloud serving resources and Dijkstra's algorithm to optimize network routing.

4.1.5. IoV Network Measurement

Intelligent traffic management, dynamic information services, and intelligent vehicle control, among others, are part of the ITS services offered in VCC. However, the high mobility of RP vehicles affects resource sharing. Network disconnection and wireless bottlenecks in various geographical locations are reasons for issues. For ITS applications involving IoV, the validation of the IoV network measurement method is important because it characterizes its measurement performance for different controllers established in the control plane layer. Jiang et al. in [64] proposed IoV-based SDN. This architecture provided reliable systematic measurement analysis obtained to dynamically update the network topology using massive traffic flows. The framework uses a couple of processes, namely measurement points, to obtain measurement data. The measurement process requires the framework to first upload measurement requests to the control plane. The latter response to application plane measurement requests updates new network rules. Following the translation of measurement requests by the control plane, the process of running the measurement method involves executing the detail measurement algorithm in the control plane and sending the corresponding measurement commands to the data plane.

4.1.6. Distributing Software Updates

Vehicle manufacturers frequently need to perform software updates on vehicles. Software updates can be pushed out by the manufacturer to fix bugs requested by vehicle owners to improve functionality [13]. On-demand network programmability based on SDN and cloud computing is deployed for distributing software updates on vehicles. This SDVC architecture increases the flexibility of deploying software updates on vehicles. The SDN architecture leverages solutions for modeling vehicular networks as connectivity graphs that can be fed into the control plane. In order to improve network performance, different frequency bands are assigned to different graph edges using information saved in the SDN controller.

Table 9. Summary of works focused on cloud framework services on SDVC.

Framework	Ref	Architecture	Techniques	Computation Layer	Key Metrics
Vehicular resource utilization	[4]	SDVC	<ul style="list-style-type: none"> Centralize VC controller. Resource utilization. Select dominant RP. 	VC-based IaaS	<ul style="list-style-type: none"> Placement of VC controller. Select reliable RP.
	[59]	SDVC	<ul style="list-style-type: none"> SDN controller hosts protocol rules. Virtual backbones with parked vehicles. 	Fog computing servers	<ul style="list-style-type: none"> Reduce delay via resource sharing. Distribution of data storage.
	[22]	Cloud-enabled SDVN	SDN controller manages resources	Cloudlet servers	Decision making for resource allocation.
Migration of cloud vehicular resources	[2]	SDVC	VM migration	MEC	Update dynamically VC topology.
Security mechanism	[62]	SDVC	Security mechanism in the control plane	Trust third party	<ul style="list-style-type: none"> Update vehicle certificate in the control plane. Key pair generation between VC cluster.
Service broker	[63]	SDVC-aided RSU	<ul style="list-style-type: none"> Linear algorithm to schedule broker resource. Dijkstra's algorithm for path selection. 	SDN broker application	<ul style="list-style-type: none"> Optimize network routing. Optimize broker resources.
IoV network measurement	[64]	IoV-based SDN	<ul style="list-style-type: none"> Analysis network measurement. Update network traffic flow in the SDN controller. 	SDN controller	<ul style="list-style-type: none"> Validate IoV measurement. Update network measurement with new routing rules.
Distributing software updates	[13]	SDVC	<ul style="list-style-type: none"> SDN controller for update in-vehicle software. Model vehicles network as connectivity graph. 	Vehicle manufacturer cloud	<ul style="list-style-type: none"> SDN controller assigns frequency bands. Minimize network latency.

4.2. Software-Defined Control Technique in SDVC

This section discusses reasonably new architectures that integrate remote central cloud server, edge computing server, SDN controller, and vehicular cloud. These architectures use inconsistent and freely available resources to handle mobile applications. Users in vehicles offload their tasks to VCC instead of cloud computing. In fact, VCC exposes a large pool of resources to handle intensive mobile tasks. In the following, we present NFV technology in SDVC. Specifically, we discuss the software-defined pseudonym system and NFs that abstract traffic and connectivity management services. The summary of works focused on software-defined control technique in SDVC is presented in Table 10.

4.2.1. Use Virtual Network Functions

Wang et al. in [65] proposed an intelligent application based on VNF technique. The design involves different modules. They proposed a traffic identification module that uses deep neural networks (DNNs) and multi-grained cascade forest (gcForest) to distinguish service behaviors. The trained model fits the QoS for various VC services. The proposed ESDVC includes VNF layers. It also implements a scheduler function that selects which layer to host the next VNF for the incoming packets in the proposed architecture.

4.2.2. Software-Defined Pseudonym System

A third-party entity is involved in SDVC management and operations. Therefore, privacy is one of the critical security issues in VCC. For instance, an anonymous vehicle-to-infrastructure (V2I) cloud management system may request vehicles' identity and location privacy data. To enhance security in SDVC, NFV technology has been investigated. NFV in SDVC provides NFs that are customized to the security management system [66]. The formation of VNFs is hierarchically managed on different entities from the vehicle to the application plane in the cloud. The distributed pseudonym pools are scheduled and

elastically managed through a software-defined pseudonym system [67]. The software-defined pseudonym system exploits SDN by significantly improving the flexibility and programmability of pseudonym functions in SDVC. The distributed pseudonym pools are NFs linked to provide end-to-end security service. They are generally scheduled and elastically delivered on demand wherever numerous anonymous V2I access breaches the privacy in SDVC. In addition, the formulation of the two-sided matching theory is used to reduce system overhead caused by the cost of inter-pool communications. It solves the resource allocation problem of constantly handling the scheduling of the software-defined pseudonym.

The generation of key pairs by centralized SDN controllers is adopted to authenticate vehicles using group signature [68]. The proposed framework provides an effective security mechanism that implements a decentralized algorithm. It creates key pairs that protect vehicles from malicious nodes by employing pseudonyms and key management. It exposes a revocation list to provide authentication, confidentiality, integrity, and availability of necessary key pairs. Moreover, the framework uses the presence of the global authority in charge of managing the digital certificates of all entities, such as resource providers and vehicle requesters. It also collects and stores all zone controllers' status information by leveraging the key management and revocation mechanisms to detect malicious nodes.

4.2.3. Traffic and Connectivity Management

The V2X traffic and connectivity services in VCC are supported through vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication. However, allocating available resources to V2X users weakens the computation of V2X collaborative tasks because vehicles often fail to fetch updated road traffic conditions in order to reduce time to consolidate the pool of resources required [69]. During heavily congested situations, both drivers and passengers may become bored and connect their devices to access entertainment programs offered by the city cloud, such as games, video, social media, and so on. While accessing those services, hazardous driving would cause road accidents. Consequently, frequent access to cloud-based services increased network traffic in the closest RSU nodes. They became overloaded. This problem may be exacerbated in cases where the road traffic congestion is intensified with rush hour traffic. To avoid nearby RSUs and the linked aggregation switches to stay overloaded, Sahoo et al. in [70] proposed a SDVC architecture that computed via RSUs road traffic data. The OpenFlow switches receive routing instructions on how the RSUs should forward to SDN controllers routing rules that were not previously acknowledged. The SDN layer covers the communication aspects that occur in the city. The wired network connection between the RSU and the very first SDN switch reduces the latency in computing routing rules. The proposed SDN management scheme uses the average time required by packets to travel from an RSU to a final SDN switch to generate routing instructions. For each new item of road traffic data, the algorithm computes the data transfer time (DTT) to determine whether a specific driving path is heavily congested. That chunk of time addressed vehicle connectivity issues and is among the traffic engineering methods to reduce V2V transmission delay and packet loss in a smart city.

4.2.4. Network Reconfiguration

Changes to VC service hosts and data flows degrade not only offered services but also increase costs to VC service providers. A highly adaptive SDN controller can address the above-mentioned challenges. Important techniques in SDVC have been proposed, such as RSU cloud resource management (CRM), which is used to address the issue of network failure and network configuration to effectively update the topology of SDVC.

The RSU cloud resource management (CRM) model reduces reconfiguration overhead, service deployment costs, and infrastructure routing delays. Inconsistent configuration updates produce incorrect network behaviors. On the other hand, long reconfiguration times between current and target configurations may degrade undesired network perfor-

mance [71]. In order to reduce network reconfiguration time, the CRM uses consistent updates between current and target configurations through SDN technology in the proposed architecture. The deep programmability of SDN [72] involves the SDN RSU cloud framework to dynamically reconfigure network hosting services and their data forwarding information in order to meet the needs of basic shared mobility applications in vehicles. RSUs in the proposed SDN RSU cloud framework consist of physical servers that can store and compute data as well as switches that can exchange non-safety app information from vehicles on road sections and nearby RSUs. Furthermore, SDN is used to achieve the shortest cloud delay with the fewest hosts. To realize the computation delay, the solution follows the mixed-integer linear programming (MILP) problem. The proposed joint optimization (JO) algorithm uses two single objective algorithms to compute the overall delay as the sum of the delays in all edges topology traversed by a single routing path from RSU.

Table 10. Summary of works focused on software-defined control technique in SDVC.

Framework Services	Ref	Architecture	Techniques	Computation Layer	Key Metrics
Use Virtual network functions	[65]	ESDVC	<ul style="list-style-type: none"> VNF for traffic identification. DNN and gcForst to distinguish service behavior. Scheduler function to host next VNF. 	Cloud-based server	<ul style="list-style-type: none"> DNN model training. QoS for VC services.
Software-defined pseudonym system	[66]	SDVC	<ul style="list-style-type: none"> NFs for security management. SDN controller to manage VC. Scale distributed security rules using SDN. 	Cloud-based server	<ul style="list-style-type: none"> Improves security. Update SDN controller's rules.
	[67]	SDVC	<ul style="list-style-type: none"> Implementation of pseudonym rules. 	NF slicing server	<ul style="list-style-type: none"> Improve programmability of pseudonym rules. VNFs for pseudonym.
	[68]	SDVC	<ul style="list-style-type: none"> Decentralized algorithm. Algorithm for pseudonym management. 	Authority server	<ul style="list-style-type: none"> Revocation mechanism. Detect malicious nodes.
Traffic and connectivity management	[69]	SDVC	V2X techniques for road traffic conditions	V2X SDN controller.	<ul style="list-style-type: none"> Reduce time to consolidate pool of resources. Minimize V2I latency. Reduce latency for routing rules.
	[70]	SDVC	<ul style="list-style-type: none"> RSU and OpenFlow switches for road traffic control. SDN management using DTT to determine congestion driving path. 	SDN controller-based RSU	<ul style="list-style-type: none"> V2V transmission delay. Reduce packet loss.
Network reconfiguration	[71]	SDVC	<ul style="list-style-type: none"> RSU cloud resource management. Update target network reconfiguration using SDN. 	SDN RSU cloud	<ul style="list-style-type: none"> Reduce reconfiguration overhead. SDN service deployment cost.
	[72]	SDVC	<ul style="list-style-type: none"> SDN RSU reconfiguration. MLIP algorithm for the overall delay. 	SDN RSU cloud	Reduce computation delay.

4.3. Data Plane for SDVC

VC architecture includes an application unit (AU) deployed inside vehicles to facilitate routing and communication functions. Therefore, the research dimension of VC communication caused research to mainly focus on designing VC architectures that handle efficiently network traffic with higher QoS. In the following, we present an SDN data plane framework to improve the communication of heterogeneous network architecture in VCC.

Data Plane for IoV PLATFORM

VC networks strive for higher data rates, seamless connectivity, scalability, security, and improved QoS. They are the key enablers for IoVs. Research in SDVC proposed various solutions on how to design a unified data plane such that communication is allowed between heterogeneous VC network architectures with multiple wireless interfaces (e.g., wireless access in vehicular environment (WAVE), long-range wireless fidelity (WiFi), and fourth generation/long-term evolution (4G/LTE)). The SDIoV architecture proposed in [73] includes a VC application and communication units to provide context-aware network

connectivity. It uses the radio over fiber approach as a medium of communication. To manage efficient allocation of transmission power, the authors formulated the joint and graph job technique. It is a structure preservation-based two-stage allocation scheme that decouples template searching from power allocation. In their design, the first stage is designed as a hierarchical tree-based random subgraph. It is based on an isomorphism mechanism to identify potential mappings (templates) between graph job components and service providers. The second stage has a simulated annealing-based power allocation algorithm to achieve a trade-off between job completion time and energy consumption.

In IoVs, RSUs, connected vehicles, pedestrian devices, utilities, and charging points generate massive amounts of data that must be efficiently transmitted from one location to another. However, the volume of data brought new computing architectures to the edge where data are generated to reduce computation decisions on network management. To achieve that, the redesign of the data plane based on SDVC plays an important role. The data plane in the architecture presented in [74] is managed through the control plane. It collects the network's dynamic nature, then updates network policies on a regular basis. In the case of electric vehicles with smart grids, the authors proposed a data plane that allows communication between charging stations and electric vehicles. Since those charging stations are geographically distributed across a large area, the SDN controller implements the recommended network policy to update the driving path.

Virtual private mobile networks (VPMN) have been proposed in SDVC. This technique enhanced stable connectivity when the vehicle moves from one RSU to another [1]. By leveraging the SDN paradigm, an SDVC architecture based on VANET, cloudlet unit, and the VPMNs dynamically creates private, resource-isolated, end-to-end mobile networks. The proposed architecture solves the patch selection and channel selection issue in delivering IoV applications. The SDN controller in that architecture allows the establishment of a global view of the network. In turn, the routing rules applied to any closest cloudlet as RSU improves the overall network management. A RSU cloudlet, for example, can switch between different VC clusters without virtual computing instances terminating or changing their Internet protocol address. The SDN controller detects the change in the driving path, then forwards incoming packets to where the virtual instances resumed across RSU cloudlet units.

5. Lessons Learned and Open Challenges

To identify existing challenges and sidestep potential misleading directions, we briefly discuss promising applications based on “VC network slicing”, and separately discuss open issues related the two enabling techniques that we focus on, i.e., “Data plane in SDVC”, “software-defined control (e.g., NFV) in SDVC”.

5.1. More Promising Techniques in VC Network Slicing

If VC network slicing and VCC are profitably integrated, they can offer great implementation of innovative V2X applications [21,75]. There are still many areas to be explored to provide intelligent VC applications and allow access to network third party suppliers or developers with new business opportunities.

For example, with more SD techniques designed in emerged cloud-native architectures, VC network slicing would reduce the network latency computation cost observed in VCC. Storage and computing resources at the network edge still challenge VCC applications. Edge computing techniques such as edge caching and intelligent edge computing-based AI in MEC are foreseen to complement the cloud framework services in SDVC to form ESDVC. Furthermore, a cloud-native baseband base unit (BBU) server demands softwarization of its functions to run as logical VNFs such that a separate VC network slicing would fully support different requirements in implementing VCC applications at the network edge. By leveraging a cloud-native approach, the reconfiguration on demand of VC network slicing would take less effort and cost. Therefore, the VC network slicing services would involve the implementation of virtualized radio access network(RAN) (vRAN) services (with NFV)

together with MEC applications to enable advanced IoV applications (e.g., autonomous driving, mapping, high-density platooning) that require low end-to-end latency. In addition, software-defined and reconfigurable VC network slicing infrastructure on demand would improve intelligence in processing surroundings data through AI solutions embedded in vehicles. Then, a decision on driverless function would be computed over serverless computing instances deployed in RSUs or related edge servers, thus enhancing safety in autonomous driving systems.

Evolve ESDVC with 6G

AI-enabled computing components can be installed on the ESDVC controllers, which handle network control and functions, to produce intelligent policy decisions. AI algorithms, for instance, can support the detection and isolation of harmful assaults in an ESDVC. Because of this, SDN applications provide VC network slicing with a scalable platform for the deployment of sophisticated network rules, whereas AI, acting as machine intelligence can compute smart and self-adapting network policies. The 6G ESDVC platform's control layer is thought of as its machine intelligence core. Vehicles with machine intelligence built into them are integrated into the control plane to give the system the ability to make decisions. The control plane acts as an intelligent agent. Hence, V2X applications may be effectively supported and carried out such as threat identification or V2X network adaptive adjustment.

5.2. Open Challenges and Future Research Directions

When deploying SD techniques in VC, it is necessary to accelerate NFV by enabling different virtualized and independent logical networks. In this section are lessons learned and future directions for “software-defined control (NFV) in SDVC” with respect to VC network slicing control plane and VC network slicing co-location with MEC.

5.2.1. Improvement of Key Performance Metrics

For VC network slicing service for a specific VC task, there are usually a series of VC control planes that can accomplish the VC task. However, it is difficult for network service providers to choose the right NFV for each VC network slicing service. Due to the uncertain characteristics of VC networks, commonly used standard performance indicators (such as slice runtime operations as well as slice life-cycle operations) cannot reflect the runtime performance of VC network slicing in the VCC. For VC slicing services, besides network slice availability, the end-to-end latency, application-specific performance, and resource elasticity are also key indicators. Based on previously identified key performance metrics [20,76,77], extensive research has been conducted to quantitatively improve them by analyzing the factors that affect them. Exploring the trade-offs between these indicators would help improve the efficiency of deploying VC network slicing services. A summary of key performance metrics such as latency, bandwidth, and placement of SDN controllers for cloud-vehicular applications is presented in Table 8.

5.2.2. Generalization of VC Network Slicing Controller

Currently, communication between the control plane and forwarding plane in VC network slicing takes place through the SDN control protocol, but there is no generalized solution for the VC network slicing controller for a wide range of cooperative VC applications. Furthermore, in order to build intelligent VC network slicing and support VC applications, not only NFV but also the possibility of co-locating the NFV and MEC techniques (edge caching), quality of service, and VC provisioning as a result of the VC virtualization software implementations should be explored. For instance, applying deep reinforcement learning (DRL) to the VC network slicing controller would enhance real-time resource management for SDVC.

5.2.3. Hybrid NFs for VC Network Slicing Reconfiguration

Coordination issues with respect to NFs on VCC deployment, VNFs, MEC running on BBU servers at BSs or radio access points along the RSU should be thought over. These customized VC network slicing are often used independently to enable vehicular–edge–fog–cloud collaboration. Diversified service requirements will impose one network slicing infrastructure to provide low-security, low-bandwidth services, while another slice can provide high-security, high-reliability services (such as for URLLC). VC network slicing customization, such as cloud-native architecture based on 5G and MEC may impose running on the network edge sides, but because of sufficient computation resources, the cloud will store permanent information data or other metadata information related to VC network slicing policies. VC network slicing will implement the URLLC service for 5G VCC systems to run on the same physical infrastructure as other 5G services to save infrastructure investment and network operating costs. Therefore, designing a hybrid VC network slicing scheme that effectively combines the simplified VC network slicing architecture with some framework deployed in the form of a cloud-native concept in the cloud requires new advanced VCC architectures.

6. Conclusions

SDVC is a key technique of the next generation of advanced networks, and 5G enabling technologies are expected to converge together in order to establish ESDVC. One of the target services in ESDVC is VC network slicing. This survey has summarized our findings and lessons learned from our research on the state-of-the-art achievement of SDVC-related cloud-vehicular framework. Moreover, this survey introduced and discussed various cloud framework services in SDVC and fundamental enabling techniques for ESDVC, in particular VC network slicing. In summary, the key issue of extending SDVC using SDN technologies such as NFV, software virtualization, and orchestration management platform is as follows: under the multiple constraints of networking, communication, computing power, and energy consumption, how to devise and develop SDVC to achieve the best performance of VC services and VC network slicing architecture. As the RP vehicles of the VC increase, SDVC will become common, and ESDVC will play an important supporting role to improve the performance of SDVC. Future works on NFV and MEC techniques (edge caching) for orchestrating and provisioning VC virtualization software implementations should be explored. For instance, applying deep reinforcement learning (DRL) to the VC network slicing controller would enhance real-time resource management for SDVC. We hope that this survey will increase discussions and research efforts in SDVC and 5G enabling techniques that will advance future VCC communications and services.

Author Contributions: Conceptualization, L.N. (Lionel Nkenyereye); methodology, L.N. (Lionel Nkenyereye); validation, J.-W.J. and L.N. (Lionel Nkenyereye); writing—original draft preparation, L.N. (Lionel Nkenyereye) and L.N. (Lionel Nkenyereye); writing—review and editing, L.N. (Lionel Nkenyereye) and J.-W.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work is partly supported by the Pukyong National University.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

5G	fifth generation
5G URLLC	ultra-reliable low-latency communication
ANNs	artificial neural networks
BS	base station
CNFs	container network functions
C-RAN	cloud radio access network
DRL	deep reinforcement learning
eNB	evolved node base
ESDVC	extended SDVC
ETSI	European Telecommunications Standards Institute
FG	fog computing
IaaS	infrastructure-as-a-service
IoVs	Internet of vehicles
ITS	intelligent transportation system
MEC	multi-access edge computing
NFs	network functions
NFV	network function virtualization
NS	network slicing
QoS	quality of service
RP	resource providers
RSU	road side unit
SDN	software-defined network
SDVC	software-defined vehicular cloud
V2I	vehicle-to-infrastructure
V2V	vehicle-to-vehicle
V2X	vehicle-to-everything
VCC	vehicular cloud computing
VE	vehicle equipment
VMs	virtual machines
VNFs	virtual network functions
VPMN	virtual private mobile networks

References

1. Nkenyereye, L.; Jang, J.W. Lowering the Barriers of Cloud -based Vehicular Applications using Software Defined Network. In Proceedings of the 2018 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Republic of Korea, 17–19 October 2018; pp. 1119–1123. [\[CrossRef\]](#)
2. Nkenyereye, L.; Nkenyereye, L.; Adhi Tama, B.; Reddy, A.G.; Song, J. Software-Defined Vehicular Cloud Networks: Architecture, Applications and Virtual Machine Migration. *Sensors* **2020**, *20*, 1092. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Olariu, S.; Eltoweissy, M.; Younis, M. Towards Autonomous Vehicular Clouds. *ICST Trans. Mob. Commun. Appl.* **2011**, *11*, e2. [\[CrossRef\]](#)
4. Jang, I.; Choo, S.; Kim, M.; Pack, S.; Dan, G. The Software-Defined Vehicular Cloud: A New Level of Sharing the Road. *IEEE Veh. Technol. Mag.* **2017**, *12*, 78–88. [\[CrossRef\]](#)
5. Xu, S.; Guo, C. Computation Offloading in a Cognitive Vehicular Networks with Vehicular Cloud Computing and Remote Cloud Computing. *Sensors* **2020**, *20*, 6820. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Khan, A.; Abolhasan, M.; Ni, W. 5G next generation VANETs using SDN and fog computing framework. In Proceedings of the 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 12–15 January 2018; pp. 1–6. [\[CrossRef\]](#)
7. Huang, C.M.; Chiang, M.S.; Dao Duy, T.; Su, W.L.; Xu, S.; Zhou, H. V2V Data Offloading for Cellular Network based on the Software Defined Network (SDN) inside Mobile Edge Computing (MEC) Architecture. *IEEE Access* **2018**, *6*, 17741–17755. [\[CrossRef\]](#)
8. He, Z.; Cao, J.; Liu, X. SDVN: Enabling rapid network innovation for heterogeneous vehicular communication. *IEEE Netw.* **2016**, *30*, 10–15. [\[CrossRef\]](#)
9. Tang, Y.; Cheng, N.; Wu, W.; Wang, M.; Dai, Y.; Shen, X. Delay-Minimization Routing for Heterogeneous VANETs With Machine Learning Based Mobility Prediction. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3967–3979. [\[CrossRef\]](#)

10. Aujla, G.; Chaudhary, R.; Kumar, N.; Rodrigues, J.; Vinel, A. Data Offloading in 5G-Enabled Software-Defined Vehicular Networks: A Stackelberg-Game-Based Approach. *IEEE Commun. Mag.* **2017**, *55*, 100–108. [\[CrossRef\]](#)
11. Dhawankar, P.; Raza, M.; Le-Minh, H.; Aslam, N. Software-Defined Approach for Communication in Autonomous Transportation Systems. *EAI Endorsed Trans. Energy Web* **2017**, *4*, 152924. [\[CrossRef\]](#)
12. He, X.; Ren, Z.; Shi, C.; Fang, J. A novel load balancing strategy of software-defined cloud/fog networking in the Internet of Vehicles. *China Commun.* **2016**, *13*, 140–149. [\[CrossRef\]](#)
13. Azizian, M.; Cherkaoui, S.; Senhaji Hafid, A. Vehicle Software Updates Distribution with SDN and Cloud Computing. *IEEE Commun. Mag.* **2017**, *55*, 74–79. [\[CrossRef\]](#)
14. Sudheera, K.; Ma, M.; Chong, P. Link Stability Based Optimized Routing Framework for Software Defined Vehicular Networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 2934–2945. [\[CrossRef\]](#)
15. Liu, J.; Wan, J.; Zeng, B.; Wang, Q.; Song, H.; Qiu, M. A Scalable and Quick-Response Software Defined Vehicular Network Assisted by Mobile Edge Computing. *IEEE Commun. Mag.* **2017**, *55*, 94–100. [\[CrossRef\]](#)
16. Lai, C.F.; Chang, Y.C.; Chao, H.C.; Hossain, M.S.; Ghoneim, A. A Buffer-Aware QoS Streaming Approach for SDN-Enabled 5G Vehicular Networks. *IEEE Commun. Mag.* **2017**, *55*, 68–73. [\[CrossRef\]](#)
17. Qi, W.; Song, Q.; Wang, X.; Guo, L.; Ning, Z. SDN-Enabled Social-Aware Clustering in 5G-VANET Systems. *IEEE Access* **2018**, *6*, 28213–28224. [\[CrossRef\]](#)
18. Waheed, A.; Shah, M.A.; Mohsin, S.M.; Khan, A.; Maple, C.; Aslam, S.; Shamshirband, S. A Comprehensive Review of Computing Paradigms, Enabling Computation Offloading and Task Execution in Vehicular Networks. *IEEE Access* **2022**, *10*, 3580–3600. [\[CrossRef\]](#)
19. Correia, S.; Boukerche, A.; Meneguet, R. An Architecture for Hierarchical Software-Defined Vehicular Networks. *IEEE Commun. Mag.* **2017**, *55*, 80–86. [\[CrossRef\]](#)
20. Campolo, C.; Molinaro, A.; Iera, A.; Menichella, F. 5G Network Slicing for Vehicle-to-Everything Services. *IEEE Wirel. Commun.* **2017**, *24*, 38–45. [\[CrossRef\]](#)
21. Skondras, E.; Michalas, A.; Vergados, D.J.; Michailidis, E.T.; Miridakis, N.I.; Vergados, D.D. Network Slicing on 5G Vehicular Cloud Computing Systems. *Electronics* **2021**, *10*, 1474. [\[CrossRef\]](#)
22. Wang, Q.; Gao, D.; Zhu, W. Cloud-enabled Software-Defined Vehicular Networks: Architecture, Applications and Challenges. *J. Internet Technol.* **2019**, *20*, 1819–1828.
23. Bhatia, J.; Modi, Y.; Tanwar, S.; Bhavsar, M.D. Software defined vehicular networks: A comprehensive review. *Int. J. Commun. Syst.* **2019**, *32*, e4005. [\[CrossRef\]](#)
24. Chahal, M.; Harit, S.; Mishra, K.; Sangaiah, A.; Zheng, Z. A Survey on software-defined networking in vehicular ad hoc networks: Challenges, applications and use cases. *Sustain. Cities Soc.* **2017**, *35*, 830–840. [\[CrossRef\]](#)
25. Ben Jaballah, W.; Conti, M.; Lal, C. A Survey on Software-Defined VANETs: Benefits, Challenges, and Future Directions. *arXiv* **2019**, arXiv:1904.04577.
26. Khan, H.U.; Wahid, I.; Tanvir, S.; Hameed, A.; Ahmad, M. Software-Defined Networks and Named Data Networks in Vehicular Ad Hoc Network Routing: Comparative Study and Future Directions. *Secur. Commun. Netw.* **2022**, *2022*, 1270180.
27. Cardona, N.; Coronado, E.; Latré, S.; Riggio, R.; Marquez-Barja, J.M. Software-Defined Vehicular Networking: Opportunities and Challenges. *IEEE Access* **2020**, *8*, 219971–219995. [\[CrossRef\]](#)
28. Chen, H.; Zhao, T.; Li, C.; Guo, Y. Green Internet of Vehicles: Architecture, Enabling Technologies, and Applications. *IEEE Access* **2019**, *7*, 179185–179198. [\[CrossRef\]](#)
29. Mekki, T.; Jabri, I.; Rachedi, A.; Fourati, L. Software-defined networking in vehicular networks: A survey. *Trans. Emerg. Telecommun. Technol.* **2021**, *33*, e4265. [\[CrossRef\]](#)
30. Alhilal, A.Y.; Finley, B.; Braud, T.; Su, D.; Hui, P. Street Smart in 5G: Vehicular Applications, Communication, and Computing. *IEEE Access* **2022**, *10*, 105631–105656. [\[CrossRef\]](#)
31. Garg, S.; Kaur, K.; Kaddoum, G.; Ahmed, S.H.; Jayakody, D.N.K. SDN-Based Secure and Privacy-Preserving Scheme for Vehicular Networks: A 5G Perspective. *IEEE Trans. Veh. Technol.* **2019**, *68*, 8421–8434. [\[CrossRef\]](#)
32. Jena, S.; Vijayaraja, V.; Sahu, A.K. Performance Evaluation of Energy Efficient Power Models for Digital Cloud. *Indian J. Sci. Technol.* **2016**, *9*, 1–7. [\[CrossRef\]](#)
33. Zhang, Y.; Gjessing, S.; Xia, W.; Yang, K. Toward Cloud-Based Vehicular Networks with Efficient Resource Management. *Netw. IEEE* **2013**, *27*, 48–55. [\[CrossRef\]](#)
34. Sutton, G.; Zeng, J.; Liu, R.; Ni, W.; Nguyen, D.; Jayawickrama, B.; Huang, X.; Abolhasan, M.; Zhang, Z.; Dutkiewicz, E.; et al. Enabling Technologies for Ultra-Reliable and Low Latency Communications: From PHY and MAC Layer Perspectives. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2488–2524. [\[CrossRef\]](#)
35. Ben bezziane, M.; Korichi, A.; Kerrache, C.A.; Fekair, M.e.A. RCV: RSU-Aided Cluster-Based Vehicular Clouds Architecture for Urban Areas. *Electronics* **2021**, *10*, 193. [\[CrossRef\]](#)
36. Wang, J.; Shao, Y.; Ge, Y.; Yu, R. A Survey of Vehicle to Everything (V2X) Testing. *Sensors* **2019**, *19*, 334. [\[CrossRef\]](#) [\[PubMed\]](#)
37. Taleb, T.; Samdanis, K.; Mada, B.E.; Flinck, H.; Dutta, S.; Sabella, D. On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Architecture & Orchestration. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1657–1681. [\[CrossRef\]](#)
38. Bilal, A.; Asad Waqar, M.; Taimur, H.; Nadeem, A. Services and simulation frameworks for vehicular cloud computing: A contemporary survey. *J. Wirel. Commun. Netw.* **2019**, *4*. [\[CrossRef\]](#)

39. Bréhon–Grataloup, L.; Kacimi, R.; Beylot, A.L. Mobile edge computing for V2X architectures and applications: A survey. *Comput. Netw.* **2022**, *206*, 108797. [\[CrossRef\]](#)
40. Satyanarayanan, M.; Bahl, P.; Cáceres, R.; Davies, N. The Case for VM-Based Cloudlets in Mobile Computing. *IEEE Pervasive Comput.* **2009**, *8*, 14–23. [\[CrossRef\]](#)
41. Lien, S.Y.; Hung, S.C.; Hsu, H.; Chen, K.C. Collaborative radio access of heterogeneous cloud radio access networks and edge computing networks. In Proceedings of the 2016 IEEE International Conference on Communications Workshops (ICC), Kuala Lumpur, Malaysia, 23–27 May 2016; pp. 193–199. [\[CrossRef\]](#)
42. Smida, K.; Tounsi, H.; Frikha, M.; Song, Y.Q. Delay Study in Multi-controller Software Defined Vehicular Network Using OpenDay-light for Emergency Applications. In Proceedings of the 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, Morocco, 24–28 June 2019; pp. 615–620. [\[CrossRef\]](#)
43. Truong, N.B.; Lee, G.M.; Ghamri-Doudane, Y. Software defined networking-based vehicular Adhoc Network with Fog Computing. In Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, Canada, 11–15 May 2015; pp. 1202–1207. [\[CrossRef\]](#)
44. Mahmood, A.; Zhang, W.E.; Sheng, Q. Software-Defined Heterogeneous Vehicular Networking: The Architectural Design and Open Challenges. *Future Internet* **2019**, *11*, 70. [\[CrossRef\]](#)
45. Deng, D.J.; Lien, S.Y.; Lin, C.C.; Hung, S.C.; Chen, W.B. Latency Control in Software-Defined Mobile-Edge Vehicular Networking. *IEEE Commun. Mag.* **2017**, *55*, 87–93. [\[CrossRef\]](#)
46. Li, H.; Dong, M.; Ota, K. Control Plane Optimization in Software Defined Vehicular Ad-Hoc Networks. *IEEE Trans. Veh. Technol.* **2016**, *65*, 7895–7904. [\[CrossRef\]](#)
47. Fontes, R.; Campolo, C.; Esteve Rothenberg, C.; Molinaro, A. From Theory to Experimental Evaluation: Resource Management in Software-Defined Vehicular Networks. *IEEE Access* **2017**, *5*, 3069–3076. [\[CrossRef\]](#)
48. Ku, I.; Lu, Y.; Gerla, M.; Gomes, R.; Ongaro, F.; Cerqueira, E. Towards software-defined VANET: Architecture and services. In Proceedings of the 2014 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET), Piran, Slovenia, 2–4 June 2014; pp. 103–110. [\[CrossRef\]](#)
49. Han, B.; Gopalakrishnan, V.; Ji, L.; Lee, S. Network function virtualization: Challenges and opportunities for innovations. *IEEE Commun. Mag.* **2015**, *53*, 90–97. [\[CrossRef\]](#)
50. Hwang, J.; Nkenyereye, L.; Sung, N.; Kim, J.; Song, J. IoT Service Slicing and Task Offloading for Edge Computing. *IEEE Internet Things J.* **2021**, *8*, 11526–11547. [\[CrossRef\]](#)
51. Raza Naqvi, S.S.; Wang, S.; Ahmed, M.; Anwar, M. A Survey on Vehicular Edge Computing: Architecture, Applications, Technical Issues, and Future Directions. *Wirel. Commun. Mob. Comput.* **2019**, *2019*, 3159762. [\[CrossRef\]](#)
52. Philipp, S.; Maximilian, M.; Henrik, K.; Meryem, S.; Gerhard, F.; Junaid, A.; Shehzad-Ali, A.; Bjoern, A.; Jens, V.; Riedel, I.; et al. Latency Critical IoT Applications in 5G: Perspective on the Design of Radio Interface and Network Architecture. *IEEE Commun. Mag.* **2017**, *55*, 70–78. [\[CrossRef\]](#)
53. Ming, T.; Wenhong, W.; Shuqiang, H. Location-based trustworthy services recommendation in cooperative-communication-enabled Internet of Vehicles. *J. Netw. Comput. Appl.* **2019**, *126*, 1–11. [\[CrossRef\]](#)
54. ETSI. Mobile Edge Computing (MEC): Framework and Reference Architecture. 2019. Available online: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/01.01.01_60/gs_MEC003v010101p.pdf (accessed on 5 February 2023).
55. Soua, R.; Turcanu, I.; Adamsky, F.; Fuhrer, D.; Engel, T. Multi-Access Edge Computing for Vehicular Networks: A Position Paper. In Proceedings of the 2018 IEEE Globecom Workshops (GC Wkshps), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–6. [\[CrossRef\]](#)
56. Tao, M.; Li, J.; Zhang, J.; Hong, X.; Qu, C. Vehicular Data Cloud Platform with 5G Support: Architecture, Services, and Challenges. In Proceedings of the 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, China, 21–24 July 2017; Volume 1, pp. 32–37. [\[CrossRef\]](#)
57. Li, J.; Wu, J.; Xu, G.; Li, J.; Zheng, X.; Jolfaei, A. Integrating NFV and ICN for Advanced Driver Assistance Systems. *IEEE Internet Things J.* **2019**, *7*, 2327–4662. [\[CrossRef\]](#)
58. Wang, Y.; Bao, Q. Adapting a Container Infrastructure for Autonomous Vehicle Development. In Proceedings of the 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 6–8 January 2020; pp. 0182–0187. [\[CrossRef\]](#)
59. Rachedi, A.; Badis, H. BadZak: An Hybrid Architecture Based on Virtual Backbone and Software Defined Network for Internet of Vehicles. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–7. [\[CrossRef\]](#)
60. Yao, H.; Bai, C.; Zeng, D.; Liang, Q.; Fan, Y. Migrate or not? Exploring virtual machine migration in roadside cloudlet-based vehicular cloud. *Concurr. Comput. Pract. Exp.* **2015**, *27*, 5780–5792. [\[CrossRef\]](#)
61. Shahrah, F.; Abduljalil, F. Virtual Machine Migration in Vehicular Cloud for Service Continuity. *Int. J. Comput. Appl.* **2020**, *175*, 33–43. [\[CrossRef\]](#)
62. Kim, M.; Jang, I.; Choo, S.; Pack, S. On security in Software-defined vehicular cloud. In Proceedings of the 2016 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Republic of Korea, 19–21 October 2016; pp. 1259–1260. [\[CrossRef\]](#)

63. Chang, Y.-C.; Chen, J.-L.; Chiu, P.S. Vehicular Cloud Serving Systems with Software-Defined Networking. In *Internet of Vehicles—Safe and Intelligent Mobility*; Springer: Cham, Switzerland, 2015; Volume 9502.
64. Jiang, D.; Wang, Z.; Huo, L.; Xie, S. A Performance Measurement and Analysis Method for Software-Defined Networking of IoV. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 3707–3719. [[CrossRef](#)]
65. Wang, J.; He, B.; Wang, J.; Li, T. Intelligent VNFs Selection Based on Traffic Identification in Vehicular Cloud Networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 4140–4147. [[CrossRef](#)]
66. Park, Y.; Sur, C.; Rhee, K.H. Pseudonymous authentication for secure V2I services in cloud-based vehicular networks. *J. Ambient. Intell. Humaniz. Comput.* **2015**, *7*, 661–671. [[CrossRef](#)]
67. Huang, X.; Yu, R.; Kang, J.; Wang, N.; Maharjan, S.; Zhang, Y. Software Defined Networking With Pseudonym Systems for Secure Vehicular Clouds. *IEEE Access* **2016**, *4*, 3522–3534. [[CrossRef](#)]
68. Bousselham, M.; Abdellaoui, A.; Chaoui, H. Security against malicious node in the vehicular cloud computing using a software-defined networking architecture. In Proceedings of the 2017 International Conference on Soft Computing and its Engineering Applications (icSoftComp), Changa, India, 1–2 December 2017; pp. 1–5. [[CrossRef](#)]
69. Ahmad, I.; Noor, R.M.; Ali, I.; Qureshi, M.A. The Role of Vehicular Cloud Computing in Road Traffic Management: A Survey. In *Future Intelligent Vehicular Technologies*; Springer: Cham, Switzerland, 2017; pp. 123–131. [[CrossRef](#)]
70. Sahoo, P.K.; Yunhasnawa, Y. Ferrying vehicular data in cloud through software defined networking. In Proceedings of the 2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), New York, NY, USA, 17–19 October 2016; pp. 1–8. [[CrossRef](#)]
71. Song, H.; Guo, S.; Li, P.; Liu, G. FCNR: Fast and Consistent Network Reconfiguration with low latency for SDN. *Comput. Netw.* **2021**, *193*, 108113. [[CrossRef](#)]
72. Li, H.; Ou, D.; Rasheed, I.; Tu, M. A Software-Defined Networking Roadside Unit Cloud Resource Management Framework for Vehicle Ad Hoc Networks. *J. Adv. Transp.* **2022**, *2022*, 5918128. [[CrossRef](#)]
73. LiWang, M.; Gao, Z.; Hosseinalipour, S.; Dai, H.; Wang, X. Energy-aware Allocation of Graph Jobs in Vehicular Cloud Computing-enabled Software-defined IoV. In Proceedings of the IEEE INFOCOM 2020—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, ON, Canada, 6–9 July 2020; pp. 604–609. [[CrossRef](#)]
74. Shukla, R.M.; Sengupta, S.; Chatterjee, M. Software-Defined Network and Cloud-Edge Collaboration for Smart and Connected Vehicles. In Proceedings of the Workshop Program of the 19th International Conference on Distributed Computing and Networking, Varanasi, India, 4–7 January 2018. [[CrossRef](#)]
75. Campolo, C.; Fontes, R.; Molinaro, A.; Esteve Rothenberg, C.; Iera, A. Slicing on the Road: Enabling the Automotive Vertical through 5G Network Softwarization. *Sensors* **2018**, *18*, 4435. [[CrossRef](#)] [[PubMed](#)]
76. Mei, J.; Wang, X.; Zheng, K. Intelligent Network Slicing for V2X Services Towards 5G. *IEEE Netw.* **2019**, *33*, 196–204. [[CrossRef](#)]
77. Khan, H.; Luoto, P.; Bennis, M.; Latva-aho, M. On the Application of Network Slicing for 5G-V2X. In Proceedings of the European Wireless 2018; 24th European Wireless Conference, Catania, Italy, 2–4 May 2018; pp. 1–6.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.