Convergence of Software-Defined Vehicular Cloud

Subjects: Computer Science, Interdisciplinary Applications

Contributor: Lionel Nkenyereye , Lewis Nkenyereye , Jong-Wook Jang

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, VCC architectures face challenges that hinder the vision of providing reliable services to connected vehicles. As a result, deploying cloud computing (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.

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

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][9][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][15][16][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.

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 [18]. 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 [19][20].

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.

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.

2. Fundamentals of Software-Defined Vehicular Cloud Computing

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 [21][22] and mobile edge computing cloud (MEC cloud) [23].

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 [22]. 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 [21]. Thus, RSU-based VCC opens up a plethora of new possibilities for executing situation-aware applications without relying on standard cloud infrastructure resources [24].

2.1.2. Mobile Edge Vehicular Cloud

The European Telecommunications Standards Institute (ETSI) has proposed MEC [22]. 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 [25].

Figure 1 shows the use of dynamic VM synthesis to deliver a VM state to infrastructure [26][27]. 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.

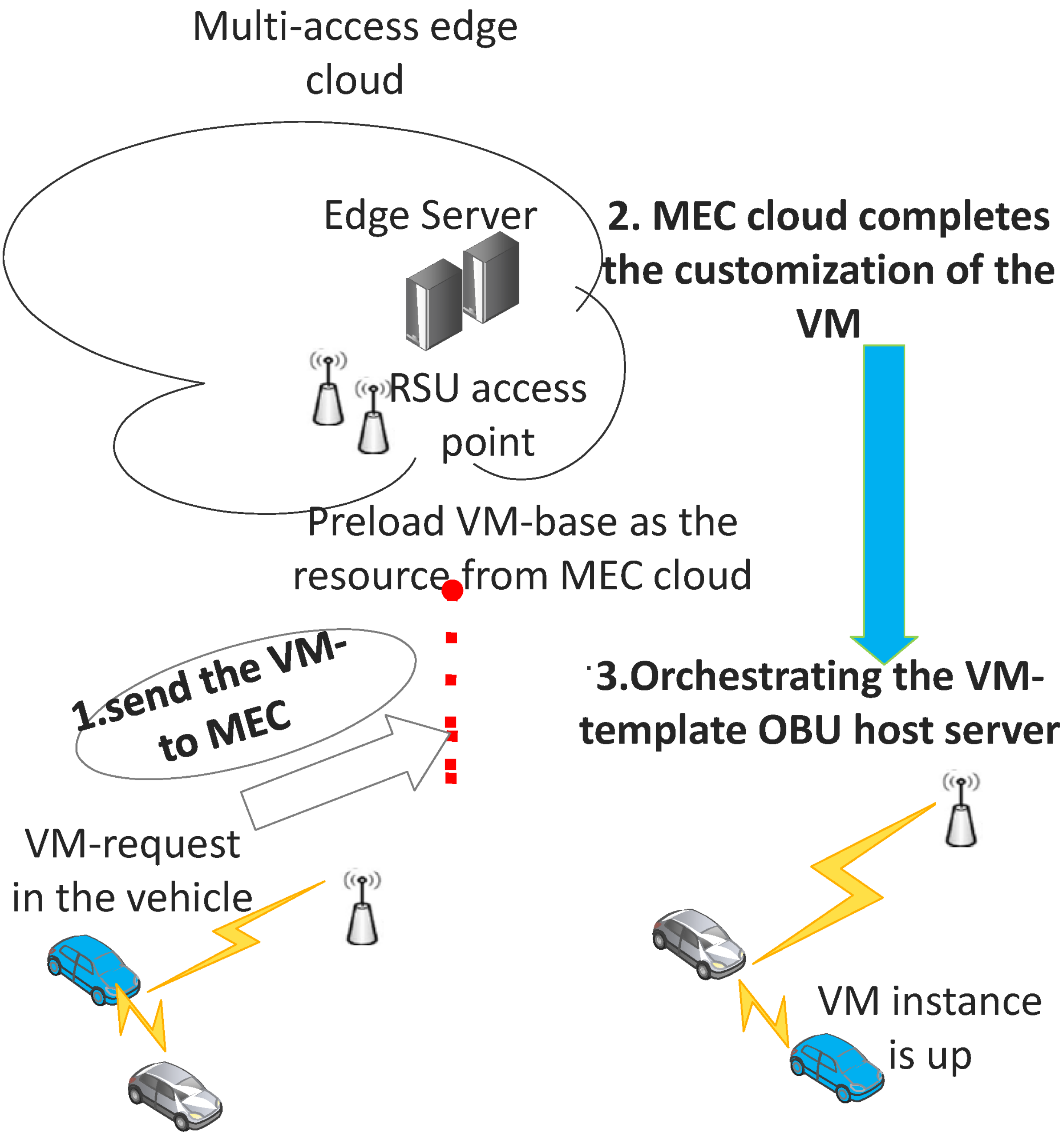
Multi-access edge cloud
Edge Server
RSU access point
2. MEC cloud completes the customization of the VM
Preload VM-base as the resource from MEC cloud
1.send the VM-to MEC
3.Orchestrating the VM-template OBU host server
VM-request in the vehicle
VM instance is up

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

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.

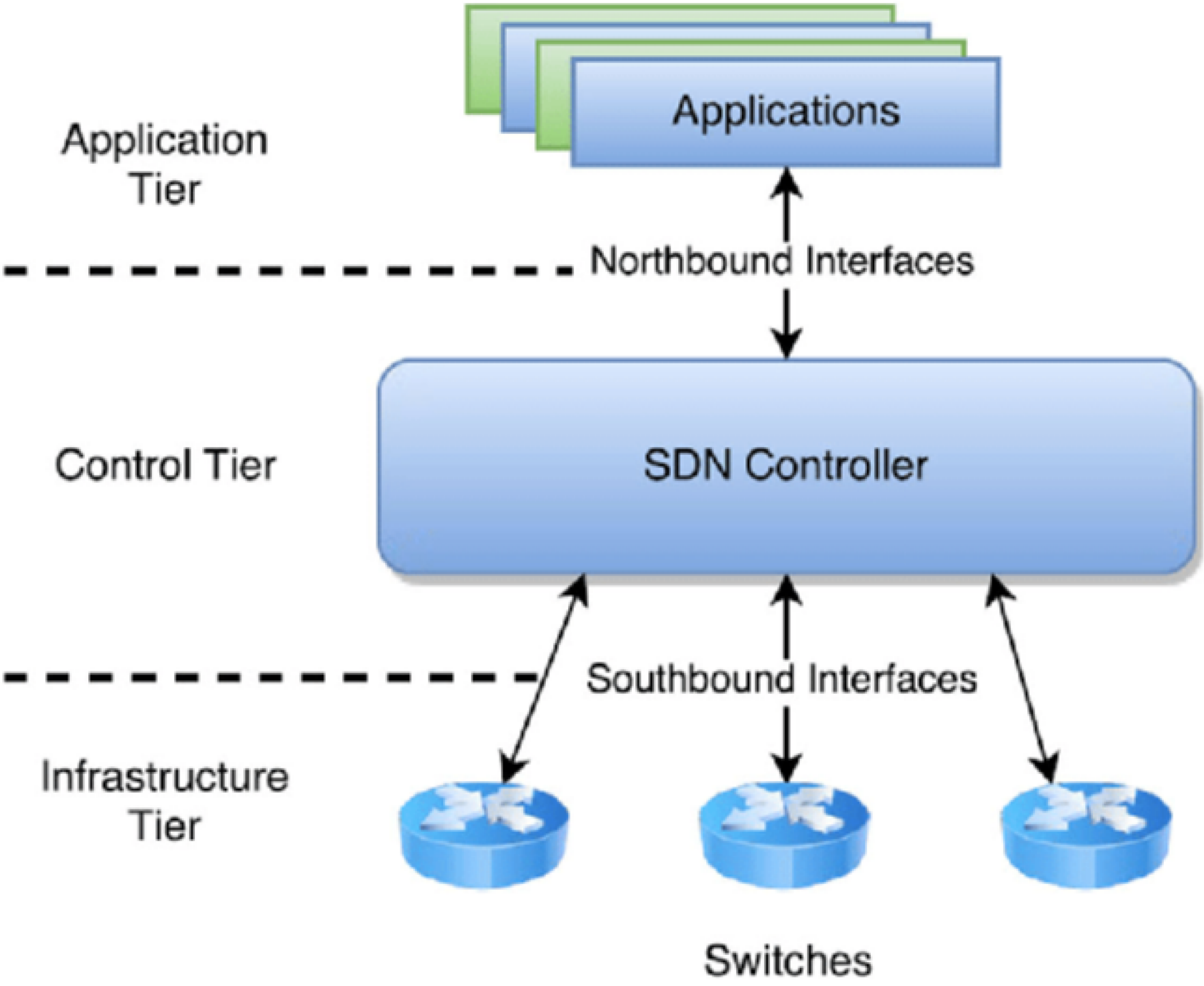

Figure 2. Logical view of a software-defined network.

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.

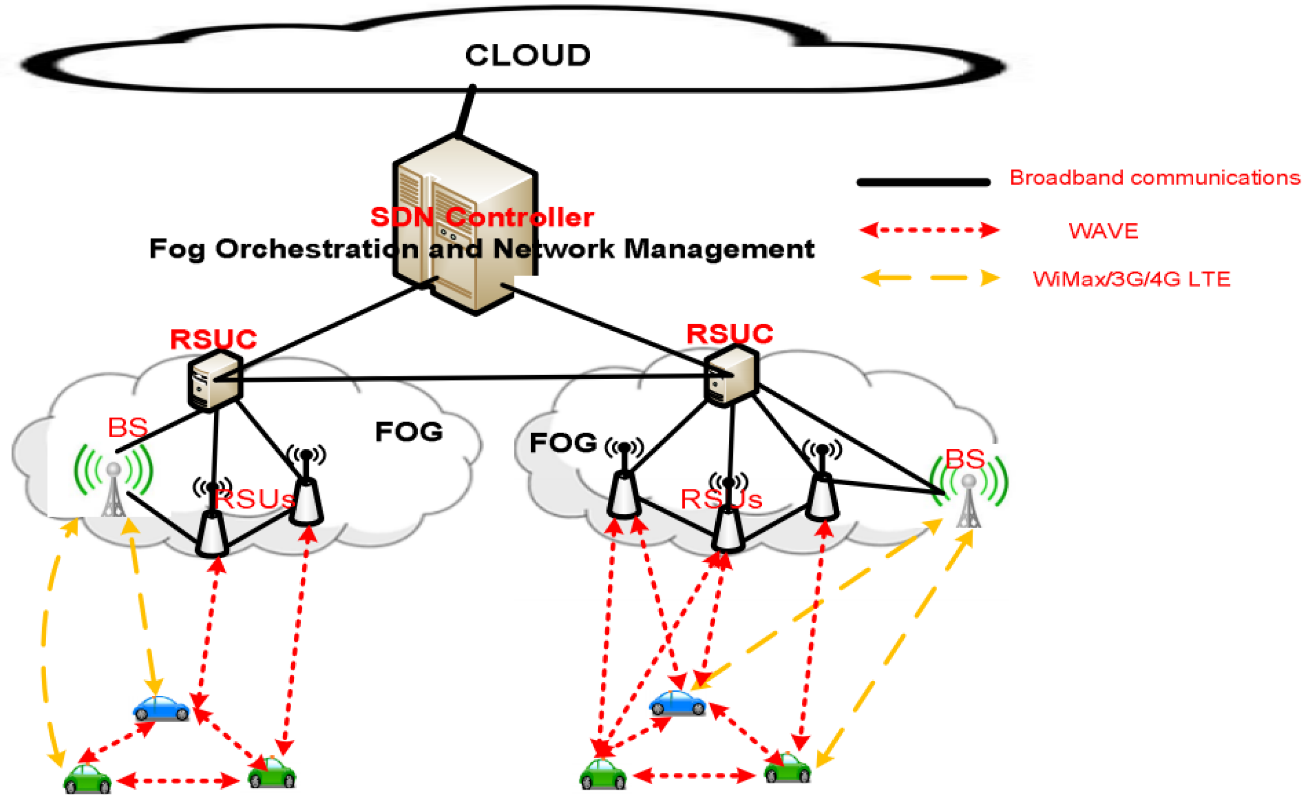

Figure 3. Logical view of software-defined vehicular cloud network [28].

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.

3. The Architecture of Extended Software-Defined Vehicular Cloud

3.1. 5G Enabling Technology for the Extended SDVC Framework

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.

NFV uses softwarization techniques to implement network elements that are deployed on general-purpose servers (i.e., industry-standard servers, switches, and gateways) [29]. 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.

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.

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.

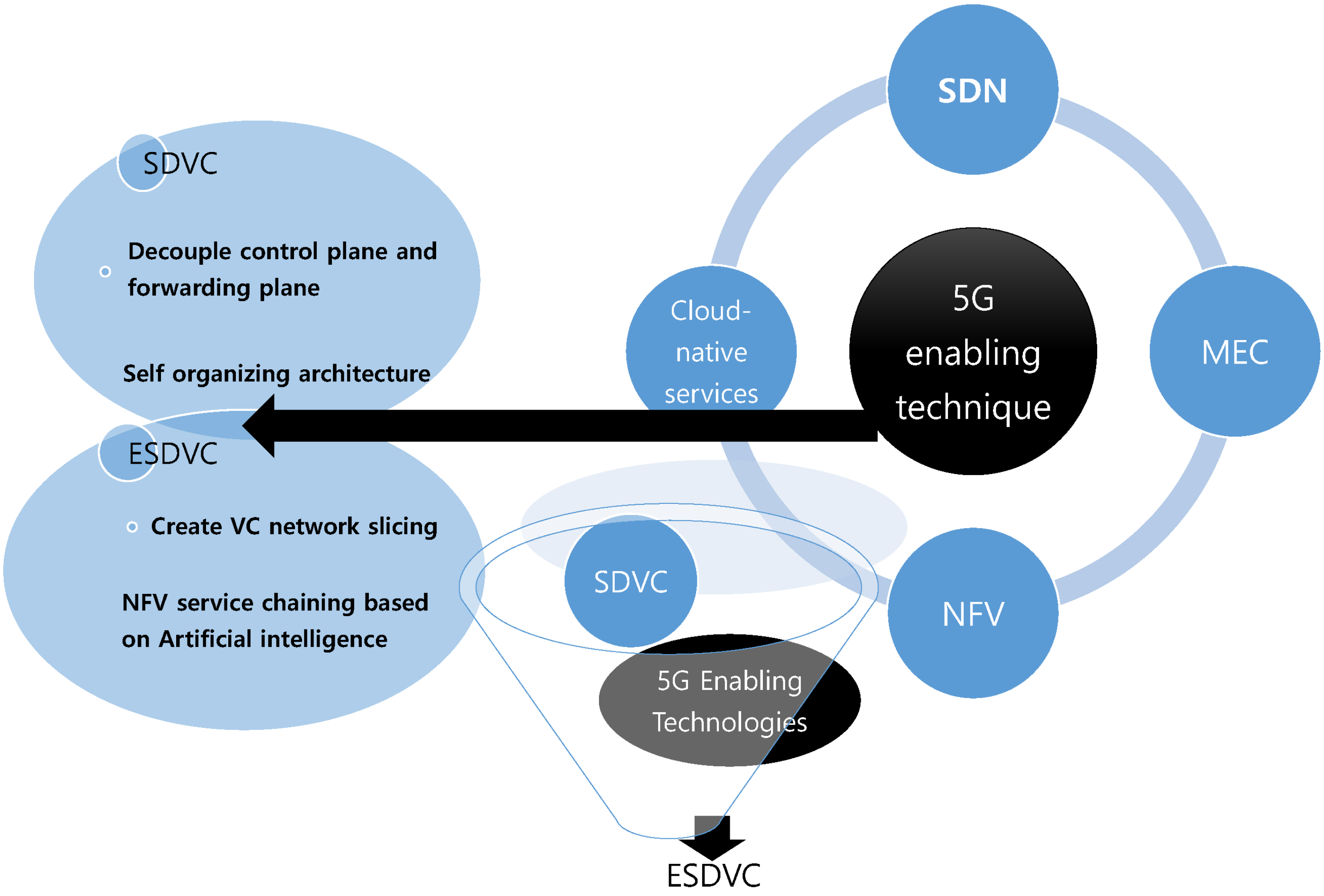
SDVC
Decouple control plane and forwarding plane
Self organizing architecture
ESDVC
Create VC network slicing
NFV service chaining based on Artificial intelligence
SDN
Cloud-native services
5G enabling technique
MEC
NFV
SDVC
5G Enabling Technologies
ESDVC

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

3.2. Extended SDVC for VC Network Slicing System

Figure 5 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][20]. 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].

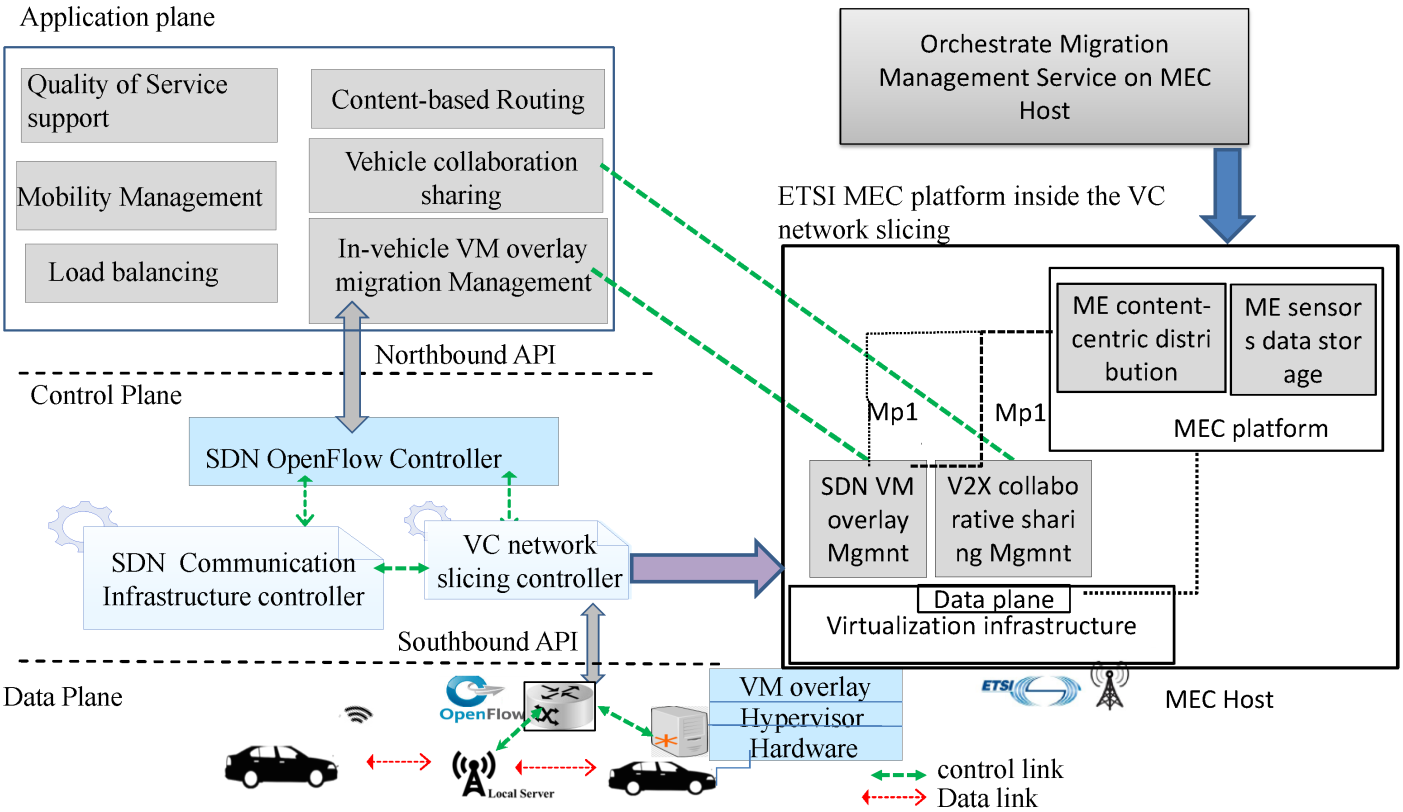
Application plane
Quality of Service support
Content-based Routing
Vehicle collaboration sharing
Mobility Management
Load balancing
In-vehicle VM overlay migration Management
Northbound API
Control Plane
SDN OpenFlow Controller
SDN Communication Infrastructure controller
VC network slicing controller
Southbound API
Data Plane
OpenFlow
Local Server
VM overlay
Hypervisor
Hardware
Orchestrate Migration Management Service on MEC Host
ETSI MEC platform inside the VC network slicing
ME content-centric distri bution
ME sensor s data stor age
Mp1
Mp1
MEC platform
SDN VM overlay Mgmnt
V2X collabo rative shari ng Mgmnt
Data plane
Virtualization infrastructure
ETSI
MEC Host
control link
Data link

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

3.3. Architecture of ESDVC Compared to SDVC

The ESDVC architecture is shown in **Figure 6** (right side). **Figure 6** (on left side of **Figure 6**) 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 [30]. 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.

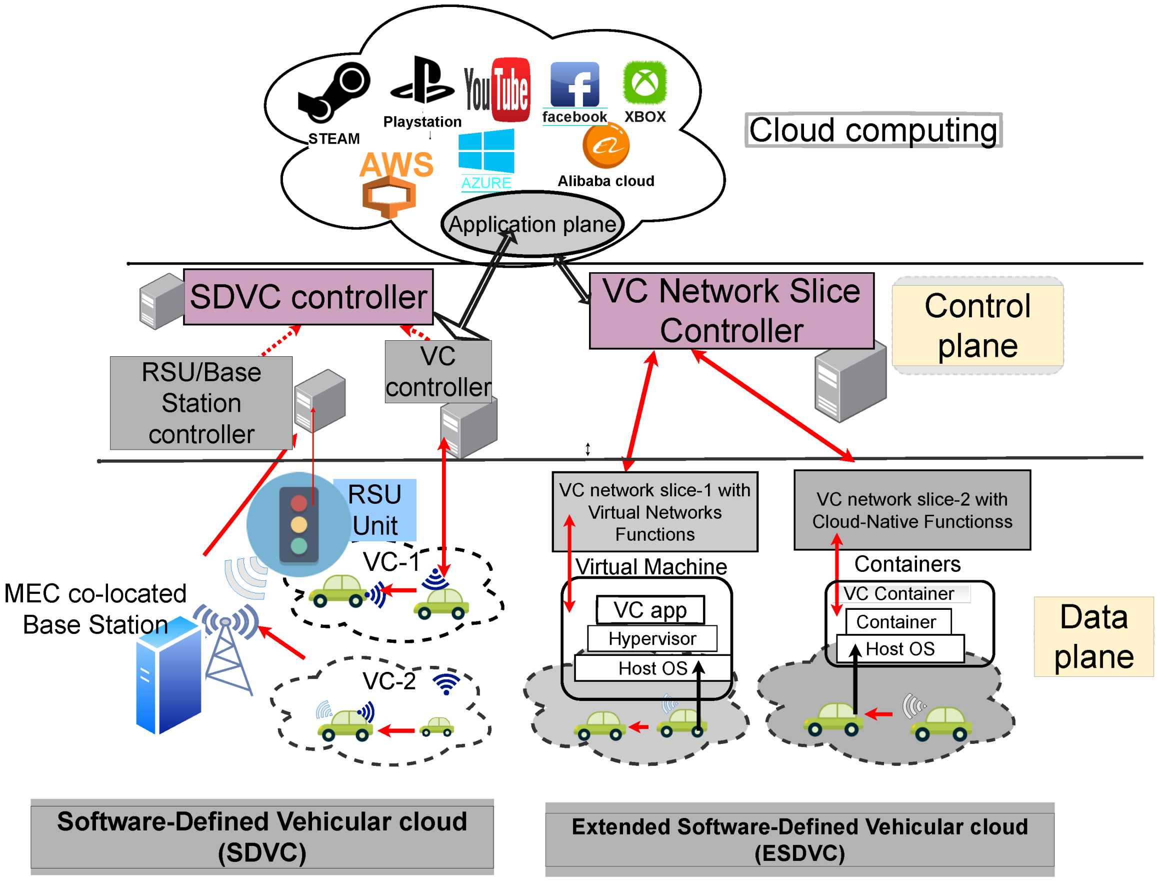
STEAM
Playstation
facebook
XBOX
AWS
AZURE
Alibaba cloud
Application plane
Cloud computing
SDVC controller
VC Network Slice Controller
Control plane
RSU/Base Station controller
VC controller
RSU Unit
VC-1
MEC co-located Base Station
VC-2
VC network slice-1 with Virtual Networks Functions
Virtual Machine
VC app
Hypervisor
Host OS
VC network slice-2 with Cloud-Native Functionss
Containers
VC Container
Container
Host OS
Data plane
Software-Defined Vehicular cloud (SDVC)
Extended Software-Defined Vehicular cloud (ESDVC)

Figure 6. High-level architecture for software-defined vehicular cloud and extended software-defined vehicular cloud.

4. Technologies in SVDC and ESDVC

As illustrated in **Figure 7**, “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. It 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, all these techniques in the taxonomy are presented in **Figure 8**.

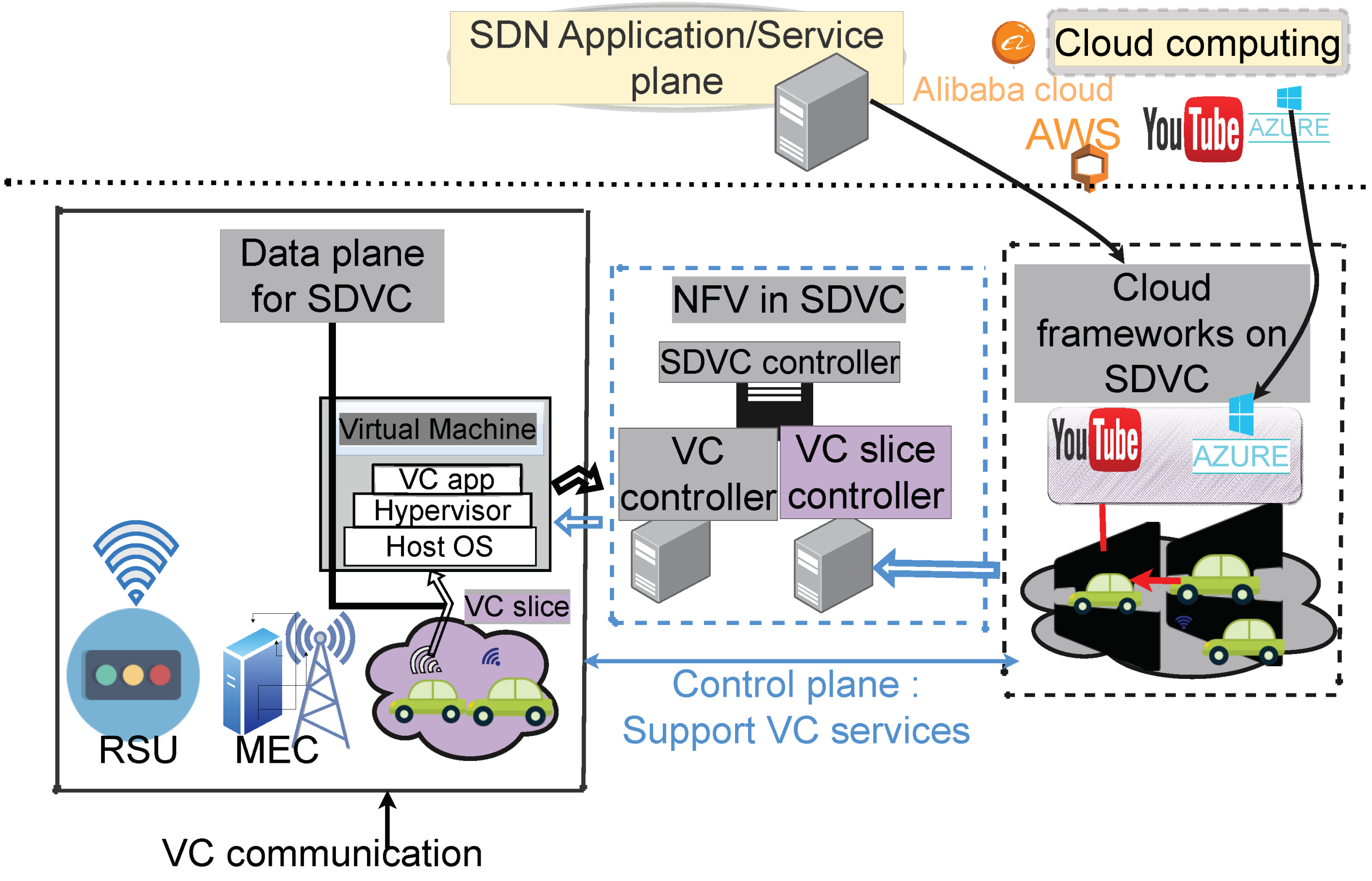

Figure 7. Technologies that are essential for the convergence of SDVC and 5G enabling technologies.

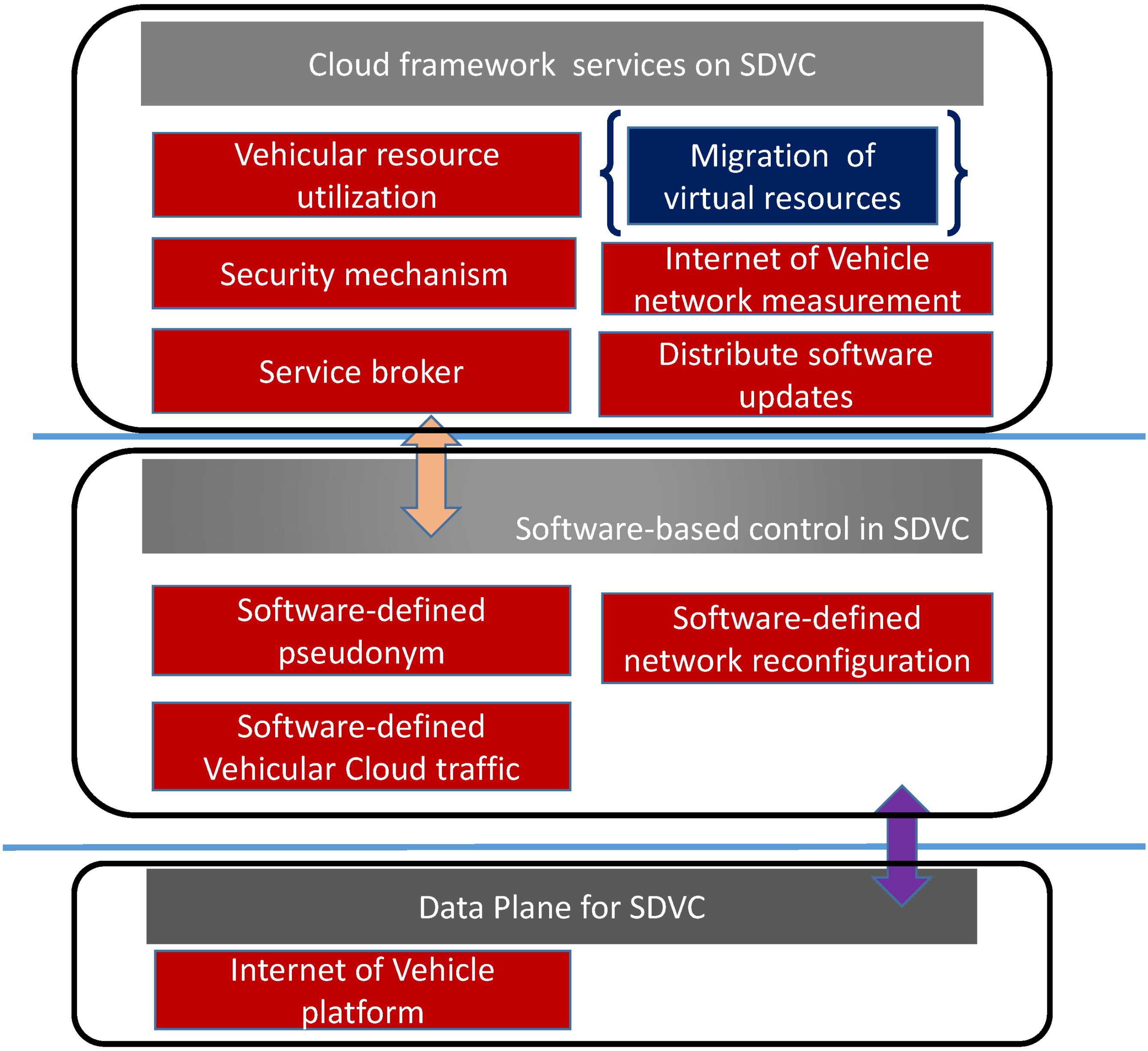
Cloud framework services on SDVC
Vehicular resource utilization
Migration of virtual resources
Security mechanism
Internet of Vehicle network measurement
Service broker
Distribute software updates
Software-based control in SDVC
Software-defined pseudonym
Software-defined network reconfiguration
Software-defined Vehicular Cloud traffic
Data Plane for SDVC
Internet of Vehicle platform

Figure 8. Taxonomy of SDVC methods/framework from the literature.

4.1. Cloud Framework Services on SDVC

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.

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 uses 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.

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 [31] 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.

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.

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.

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.

4.2. Software-Defined Control Technique in SDVC

4.2.1. Use Virtual Network Functions

Wang et al. in [32] 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 [33]. 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 [34]. 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.

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 [35]. 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.

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 performance [36]. 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 [37] 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.

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.

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 [38] includes a VC application and communication units to provide context-aware network connectivity.

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 [39] 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.

5. Lessons Learned and Open Challenges

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 [20][40]. 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.

5.2. Open Challenges and Future Research Directions

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.

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.

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.
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.
3. Olariu, S.; Eltoweissy, M.; Younis, M. Towards Autonomous Vehicular Clouds. ICST Trans. Mob. Commun. Appl. 2011, 11, e2.
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.
5. Xu, S.; Guo, C. Computation Offloading in a Cognitive Vehicular Networks with Vehicular Cloud Computing and Remote Cloud Computing. Sensors 2020, 20, 6820.
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.
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.
8. He, Z.; Cao, J.; Liu, X. SDVN: Enabling rapid network innovation for heterogeneous vehicular communication. IEEE Netw. 2016, 30, 10–15.
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.

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.
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.
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.
13. Azizian, M.; Cherkaoui, S.; Senhaji Hafid, A. Vehicle Software Updates Distribution with SDN and Cloud Computing. IEEE Commun. Mag. 2017, 55, 74–79.
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.
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.
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.
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.
18. Correia, S.; Boukerche, A.; Meneguette, R. An Architecture for Hierarchical Software-Defined Vehicular Networks. IEEE Commun. Mag. 2017, 55, 80–86.
19. Campolo, C.; Molinaro, A.; Iera, A.; Menichella, F. 5G Network Slicing for Vehicle-to-Everything Services. IEEE Wirel. Commun. 2017, 24, 38–45.
20. 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.
21. Ben bezziane, M.; Korichi, A.; Kerrache, C.A.; Fekair, M.e.A. RCVC: RSU-Aided Cluster-Based Vehicular Clouds Architecture for Urban Areas. Electronics 2021, 10, 193.
22. Wang, J.; Shao, Y.; Ge, Y.; Yu, R. A Survey of Vehicle to Everything (V2X) Testing. Sensors 2019, 19, 334.
23. 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.
24. 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.

25. 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.
26. Zhang, Y.; Gjessing, S.; Xia, W.; Yang, K. Toward Cloud-Based Vehicular Networks with Efficient Resource Management. Netw. IEEE 2013, 27, 48–55.
27. Satyanarayanan, M.; Bahl, P.; Cáceres, R.; Davies, N. The Case for VM-Based Cloudlets in Mobile Computing. IEEE Pervasive Comput. 2009, 8, 14–23.
28. 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.
29. Han, B.; Gopalakrishnan, V.; Ji, L.; Lee, S. Network function virtualization: Challenges and opportunities for innovations. IEEE Commun. Mag. 2015, 53, 90–97.
30. 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.
31. 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.
32. 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.
33. 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.
34. 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.
35. 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.
36. Song, H.; Guo, S.; Li, P.; Liu, G. FCNR: Fast and Consistent Network Reconfiguration with low latency for SDN. Comput. Netw. 2021, 193, 108113.
37. 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.

38. 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.

39. 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.

40. 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.

Retrieved from https://encyclopedia.pub/entry/history/show/100065