Ahmed *et al. EURASIP Journal on Wireless Communications and Networking* (2019) 2019:4
https://doi.org/10.1186/s13638-018-1315-y

EURASIP Journal on Wireless Communications and Networking

REVIEW Open Access

Services and simulation frameworks for vehicular cloud computing: a contemporary survey

Bilal Ahmed, Asad Waqar Malik*, Taimur Hafeez and Nadeem Ahmed

Abstract

Vehicular cloud is getting significant research attention due to the technological advancements in smart vehicles. In near future, vehicles are envisioned to become part of a grid network providing cloud services, such as computing, storage, network, and application as a service. Vehicular cloud computing is an emerging area, designed to support delay-sensitive applications. However, this integration of vehicular network and cloud computing introduces new challenges for the research community. New frameworks have been proposed to assimilate and efficiently manage this merger. In this survey paper, we present the recent advancements in vehicular cloud computing domain. The review is primarily focused on two areas. First, we discuss the frameworks designed to utilize the vehicles' onboard resources to provide cloud services and highlight the design issues and research challenges. Secondly, we focus on a detailed study of mobility generators, network, and vehicular ad hoc network simulators, as well as the available vehicular data sets. We thus provide an overarching view of the complete domain of vehicular cloud computing and identify areas for future research directions.

Keywords: VCC, V2X, Vehicular network simulations

1 Introduction

Recently Vehicular Ad Hoc Networks (VANETs) have received significant research attention based on the growing interest in smart vehicles and cities. A smart vehicle is typically equipped with storage and computing/computable resources. These vehicles are equipped with built-in sensors to gather a myriad of useful information. A network of these smart vehicles, where vehicles communicate and share information with each other in real-time, can help realize the vision of Intelligent Transportation Systems (ITS) supporting numerous safety-related applications, such as traffic management, dissemination of crucial emergency alerts etc. VANETs can also provide infotainment to the passengers on the move. Furthermore, the advancement of mobile communications into the 5G era has further strengthened the VANETs paradigm, as it improves the communication [1]. Therefore, in future, every vehicle will be able to connect, communicate and share context-aware information in real-time.

Traditionally, the smart vehicles in VANETs use Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) communications to access the public Internet resources in order to download content or to store content in the Internet cloud infrastructure. It is costly to upload content or to search and pull content to and from the Internet cloud. Moreover, many vehicles in the vicinity would also be searching for contents that have relevance in terms of spatial and temporal scope, and local interest. For example, an accident warning message is only of relevance to vehicles in a specific region (spatial scope) while a roadwork information dissemination message must remain valid till the roadwork is in progress (temporal scope). Similarly, the neighboring vehicles constitute the majority of consumers that have an interest in the information produced by a vehicle. These unique characteristics of VANETs have thus evolved in the concept of Vehicular Cloud Computing (VCC) where vehicles effectively form a cloud within which content is produced, maintained, and consumed

*Correspondence: asad.malik@seecs.edu.pk
Department of Computing, School of Electrical Engineering and Computer Science (SEECS) National University of Sciences and Technology (NUST), Islamabad, Pakistan

[2]. This cloud is formed by leveraging the computational, storage, and sensing capabilities of multiple vehicles to provide cost effective, contextual real-time services.

VCC thus opens up vistas of new opportunities to run context-aware applications without using the traditional cloud infrastructure. It is aimed at providing cloud services through a dynamic, self-organized cloud formation based on the availability of neighboring vehicles. It eliminates the need of backend servers to compute basic traffic management tasks such as traffic safety analysis, route calculation, and congestion prediction and provides enhanced computing and storage facilities. It removes the network delays involved in accessing the Internet clouds, thus supporting latency sensitive applications.

Similar to the traditional clouds, VCC can support services in the context of computation, storage, network, and applications. In *Computing as a Service*, mobile phone users and vehicles can offload compute intensive tasks to the cloud consisting of other willing vehicles. Similarly, for *Storage as a Service*, the VCC provides a virtual storage abstraction to implement a distributed data storage facility using the available group of vehicles. A virtual network platform is provided by VCC in *Network as a Service* by employing vehicles as mobile gateways offering connectivity to other vehicles that are without internet connectivity, and as data mules for information forwarding and providing quality of service. For *Application as a Service*, vehicular cloud network can be used to support various ITS-related applications such as road safety, traffic density, navigation guidance, and parking lot availability.

In this paper, our main contributions are as follows:

- Firstly, we present a taxonomy of existing research with a major focus on services in computing, storage, network and application in VCC. There are surveys papers on VCC available in literature; however, most of these are exploring the VCC applications space [3–5]. In this work, we explore the frameworks designed to provide various services through VCC, where applications are only a sub-part of the domain.
- Secondly, to provide a comprehensive overview of the research area, we have also reviewed existing simulators and tools required to evaluate the performance of a proposed framework in VANETs/VCC. To the best of our knowledge, VCC has not been fully explored in a review paper from its service frameworks and simulation tool perspective.
- The VCC framework design is very complex due to its ad hoc nature, wireless communications, and inherent dynamism associated with the vehicles. Thus, this survey on frameworks highlights existing challenges and identifies new research directions for the research community. Moreover, a better understanding of the available simulation tools can help researchers in the selection of an appropriate simulation platform for performance testing of any new proposed framework.

The rest of the paper is organized as follows. The concept of vehicular networks is explained in Section 2. Section 3 covers the VCC in detail. The existing vehicular simulation and related tools are presented in Section 4. Finally, Section 5 concludes the paper.

2 Vehicular networks

In this modern age, users are highly mobile and demand services during their mobility. For instance, a user having smart devices (phone, tablets etc.) wants to use uninterrupted services while on the move. Though vehicles of the next generation will be equipped with more intelligent sensors, storage, and computation capabilities along with wireless communication modules [6], a standalone vehicle cannot entertain huge computation and storage demands of rich applications. The enormous amount of vehicles on the road present an opportunity of vehicles forming a network called VANET. Thus, in the years ahead, VANET will be exploited to improve the experience of not only drivers but also the passengers who use interactive, storage, and computation-intensive applications. The VANETs can provide entertainment services including internet browsing, file transfer, mobile-e-commerce, and video-on-demand [7]. Similarly, any surrounding event captured by a vehicle can be of interest to other vehicles in its vicinity. The sharing of this information through inter-vehicle communication can be vital for traffic management and safety purposes. Therefore, most well-known applications of VANETs have been developed to support ITS.

The VANETs are not only limited to V2V communication. Many other possibilities are evolved from the domain of vehicular networks such as V2I, Vehicle to Grid (V2G), Vehicle to Home (V2H), V2P (Vehicle to Pedestrian) and V2D (Vehicle to Device). They are collectively referred as Vehicle to Everything (V2X). Typically, a VANET system consists of various components including smart vehicles, Road Side Units (RSUs), portable smart devices, and, in some cases, traditional deployed infrastructure as well. Vehicles can communicate with other vehicles, RSUs, infrastructure, and handheld devices. They communicate either to share their resources or to utilize shared resources offered by others in its vicinity. Figure 1 captures the domain of V2X.

The vehicular network is an emerging research area. The challenges of vehicular networks include the dynamicity of vehicles, short-range wireless communication, bandwidth limitation, signal fading, frequent disconnectivity, routing in a mobile environment, security, and privacy [8]. In [9], authors provide an insight

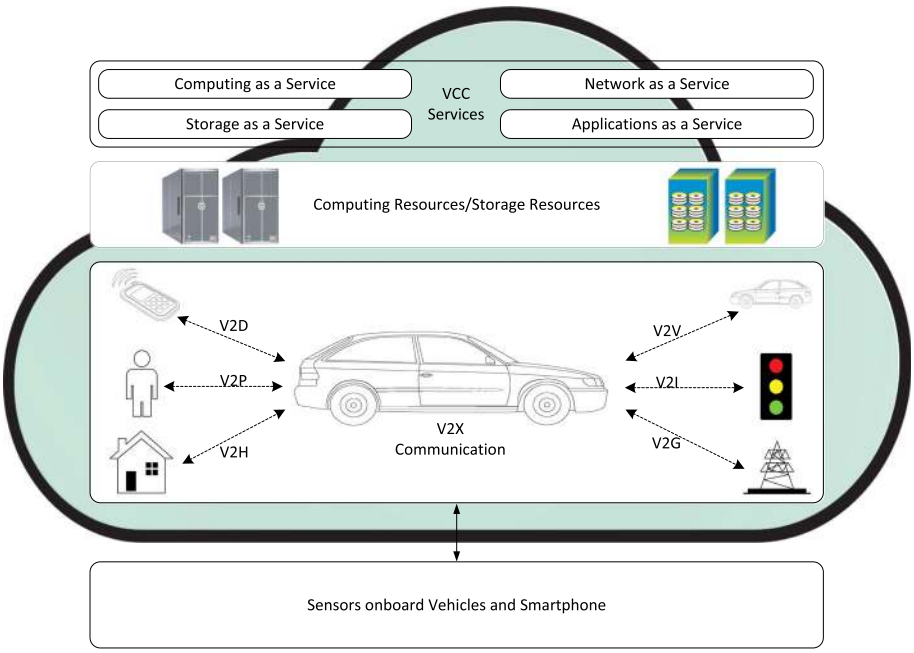

Fig. 1 V2X communication. Graphical representation of V2X communication

of research efforts that address the routing issue in VANETs. Similarly, the work in [10] reviews state-of-the-art research work that aims to address security issues. It is pertinent to mention that unlike other wireless networks, VANETs are not energy constrained. Mostly, there is enough sustained power available for vehicles to communicate either with other vehicles or RSUs.

3 Vehicular cloud computing

The ever increasing demand for storage, computation, and communication has vexed researchers, especially with the inception of Internet of Things (IoT)-based cyber-physical systems. In the past few years, the computational power required by different applications has significantly increased. For example, applications supporting medical sciences, finance, and artificial intelligence are increasingly dependent upon high computation systems. Both parallel computing and distributed computing have presented different solutions accordingly to overcome the high demand of computational power. Cloud computing is another solution that has evolved from Grid Computing. Basically, cloud computing is enabled by easily accessible data centers providing very high computation and storage capabilities. However, the initial setup cost for software and hardware, as well as running cost of infrastructure which includes the maintenance cost, cooling power consumption, and direct equipment power consumption are of major concern. Vehicular cloud computing provides a cost-effective alternative. According to this concept, a cloud can be formed anywhere on the roads using the computational capabilities of the onboard equipment in the vehicles or using the smartphone in those vehicles. VCC has extended the concept of mobile cloud computing whereby mobile phones offload all computation and storage related tasks to the Internet clouds. In VCC, mobile phones as well as vehicles can offload tasks to another vehicle or the infrastructure.

The architecture of VCC consists of mainly three layers, vehicle equipment, communication, and the cloud [11]. Vehicle equipment includes sensors to monitor driver, on board processors, storage, and OBU (On Board Unit). OBUs, at the communication layer, connect vehicle equipment to deployed infrastructure (3G, 4G/long-term evolution (LTE), Direct Short-Range Communication (DSRC)) to exchange information. This communication is named as V2I. Similarly, V2V makes inter-vehicle communication possible using DSRC. The last layer, the cloud, spans further three layers, including cloud infrastructure, cloud applications, and the cloud platform. The cloud infrastructure has mainly two modules. One for computation and the other for storage. For instance, some computation can be performed locally on the sensed data before storing it in the cloud. Architecture of VCC is shown in Fig. 1.

With the emergence of VCC paradigm, different services have been proposed that use under-utilized vehicle resources to improve performance. We categorize all vehicular cloud-related services in four categories: (i) Computing as a Service, covering different proposed

works that use vehicles computing resources to meet computation demand; (ii) Storage as a Service, describing frameworks proposed to store data in VCC resources; (iii) Network as a Service, frameworks that use VCC as a network; and (iv) Application as a Service, covering various utility applications that have been proposed for the VCC architecture. Figure 2 shows the taxonomy of all VCC-related services discussed in this paper. We discuss all these services in detail in subsequent sections.

3.1 Computing as a service

In this service, vehicles and mobile phone users can request the vehicular cloud to assist in performing compute intensive tasks *locally* to support delay-sensitive applications (Fig. 3). The vehicular cloud consists of dynamic membership of willing vehicles whereby vehicles are frequently moving in and out of the region and may not be able to fully support the requested tasks. Integration of traditional cloud with the vehicular mobile cloud termed as VANET-Cloud is proposed by Bitam et al. [12]. The proposed framework form a temporary cloud using moving or parked vehicles. Consequently, the request from end user can either be entertained with temporary mobile cloud or fixed cloud. The proposed scheme is able to achieve a considerable improvement in performance even in a high load environment. The service providers are responsible for managing the complexity of combining vehicular cloud with a traditional cloud.

Zhang et al. [13] proposed a VCC framework that combines vehicular cloud and cloudlets[1] to provide computing capabilities to smartphones. The framework requires two conditions to initiate computational tasks offloading from smartphones. The first condition is the availability of a reliable connection between the smartphone and VCC. Secondly, the availability of appropriate resources in the vehicular cloud. If the vehicle moves away from the vicinity of the smartphone during computation, cloudlet serves as multi-hop between the smartphone and the vehicular cloud node. The authors performed theoretical analysis, and it is shown that the technique has improved the performance and saves energy of smartphones from depletion.

Besides numerous benefits of cloud computing, however, its performance is constrained by IP core network's limitations (latency, bandwidth). In this context, fog computing [14, 15] is a new computing paradigm that brings part of cloud computing or cloud storage (dedicated cloud resources) in the vicinity of the end user. Consequently, a large number of requests for cloud services is entertained through local Fog resources rather than traversing the entire Wide Area Network (WAN). However, during the peak hours, Fog may be overwhelmed by a large number of simultaneous requests, resulting in degraded performance faced by the end user. Thus, the concept of Vehicular fog computing (VFC) [16] is based on providing computing capacity through underutilized infrastructure available in slow-moving and parked vehicles. The VFC utilizes the infrastructure available close to end-user or near-user edge devices to provide support for delay-sensitive applications.

A similar concept is discussed in [17] whereby the authors proposed the integration of Fog networks with vehicular cloud network based on parked vehicles in a shopping center. The policy management layer in the proposed framework decides which Fog node can fulfill the request in a timely manner. As a first step, the framework computes completion time and complexity of the task, then delegates the request either to the fog network or the vehicular network. Moreover, there is a local scheduler for both resources. The Fog-based vehicular cloud cultivates best features from both Fog and vehicular such as

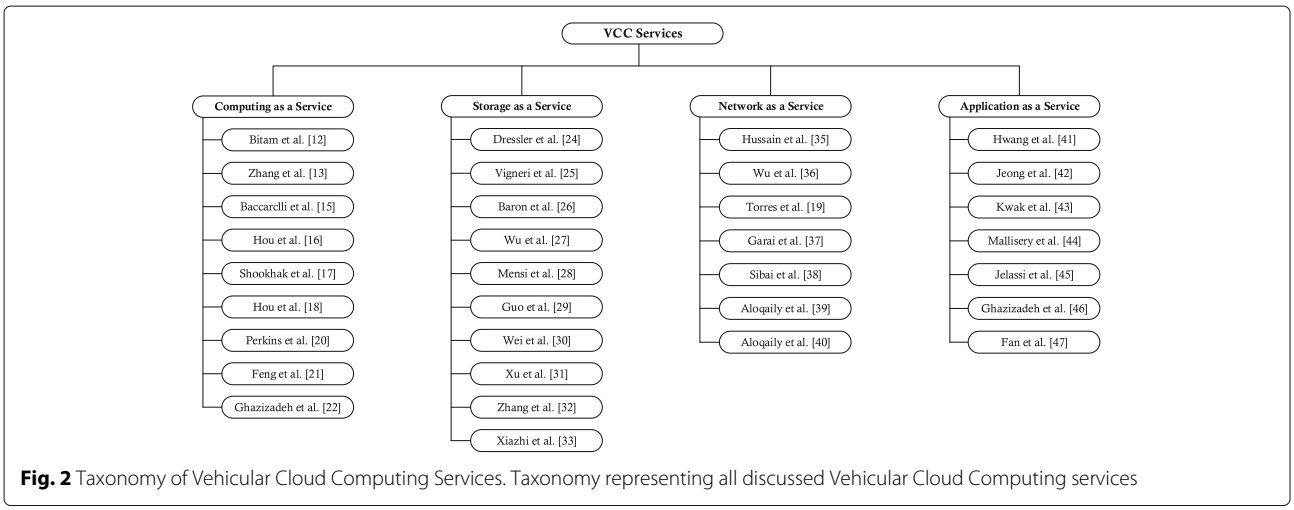

Fig. 2 Taxonomy of Vehicular Cloud Computing Services. Taxonomy representing all discussed Vehicular Cloud Computing services

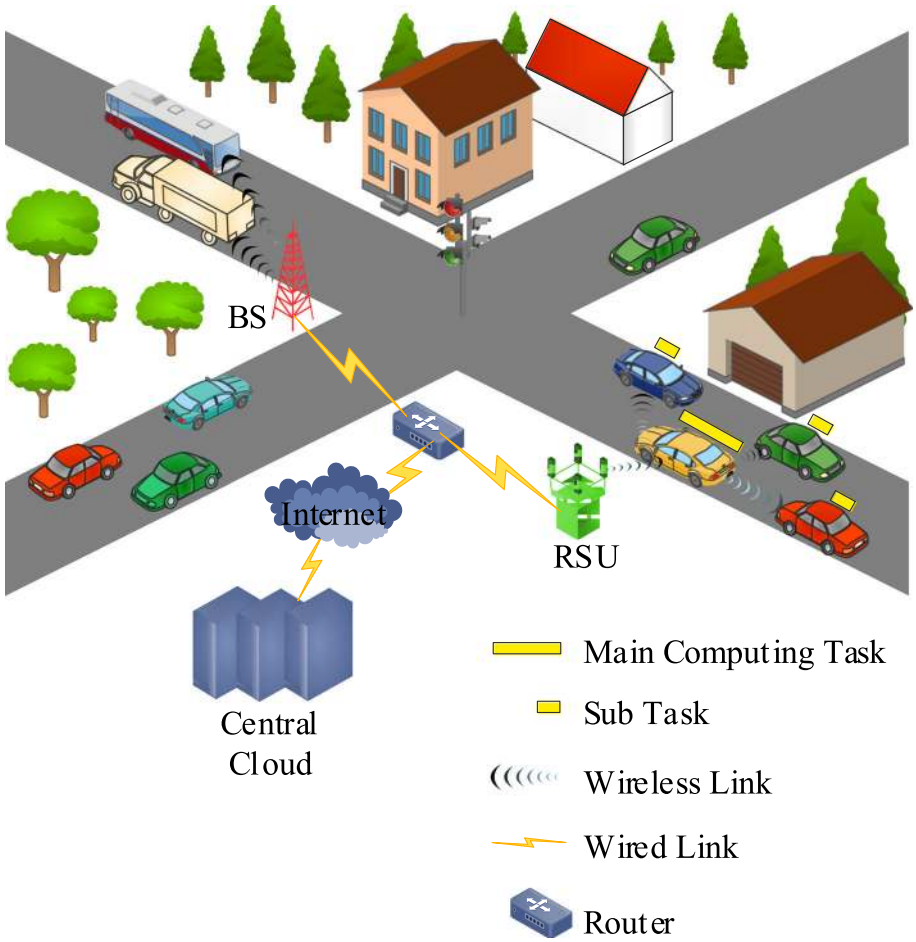

Fig. 3 VCC–Computing as a Service. Usage of vehicular network to support computational services

high computation, moderate cost, and low latency among others.

A similar computing framework is proposed by Hou et al. in [18]. It forms a cloud based on moving and parked vehicles. The vehicles can also serve as relay node for communication between vehicles and the RSUs. Consequently, RSUs with the help of vehicular computing resources form a Fog. It is worth mentioning that the major benefit of this framework lies in utilization of vehicular mobile cloud for supporting remote cloud services. However, the computing tasks from the traditional cloud can also be migrated to the mobile vehicular cloud. The performance of the proposed scheme has been investigated using a real data sets of different cities of China.

Torres et al. [19] proposed a framework that creates virtual regions of vehicles. The vehicles inside a region can only communicate with vehicles in the same region. In each region, a vehicle is elected to serve as a leader based on priority, time spent in the region etc. The leader is responsible to receive, buffer, and forward packets to the leaders of the other regions. The non-leader vehicles provide the backup to the leader node. The approach is adapted to handle the leader failure; in that case, another vehicle is selected as a leader that takes the state of the network from backup nodes. Thus, reducing the overhead involved in gathering information significantly. A modified version of Ad hoc On-Demand Distance Vector (AODV) [20] is used to support the routing within a region. The vehicle that requests the task execution and the leader vehicle that has assigned the tasks maintains a table which includes a list of members and assigned tasks. The proposed work achieves the benefits of VCC at the cost of communication overhead.

Feng et al. [21] presented a framework for distributed Autonomous Vehicular Edge (AVE). In the proposed framework, the vehicles can offload computing task to other vehicles. However, the vehicles are assigned on requester priority. Besides the role as requester and entertainer, the vehicle can serve as a relay node to enable multi-hop communication. The proposed AVE mainly consists of two modules i.e., flow and beaconing. The former one includes caching requests, discovery of available

resources, scheduling of jobs, and data transmission. The latter is responsible for periodic beaconing to maintain a list of vehicles within the vicinity. At scheduler, jobs are served according to ant colony optimization algorithm. The applications are installed on native operating systems of vehicles and the available idle resources are managed in a virtualized manner.

Vehicular clouds are enriched with dynamicity as vehicles can move unexpectedly, which can affect the overall performance of the vehicular cloud. Ghazizadeh et al. [22] proposed a fault tolerance model for task distribution in a vehicular cloud, which is validated through simulations. Traditionally, fault tolerance is achieved through saving the states of the computing resources periodically but in the proposed strategy, the state of the computing resources is saved only when the vehicle leaves the cloud. Thus, high bandwidth connectivity and extra storage in the vehicular cloud is not required to save the state at regular intervals. The proposed prediction model considers the environment where vehicle departure and arrival time is highly probable, parking space remains nearly occupied, each vehicle is equipped with a virtual machine capable of computing the assigned task, and vehicular cloud is capable of providing the infrastructure as a service. For redundancy purpose, each computing task is assigned to a group of multiple vehicles. When a vehicle leaves a group, its state is recorded. In case of successful state recording, results are saved and remaining task is reassigned to a new group of vehicles. However, if state recording fails, the task is forced to be restarted from its last correctly saved state.

3.1.1 Research challenges and future directions

The entire fabric of VCC is based on volunteer computing, where vehicles share their computing power with other vehicles [23]. However, mobility of the vehicular network presents a challenging environment for task distribution and results collection. To simplify, researchers have proposed frameworks that utilize the parked vehicles and static RSUs. The computation tasks are distributed by a RSU at one intersection, and results are retrieved by another RSU on other intersection. Similarly, tasks are distributed to the parked vehicles and results gathered back before their departure from the parking lot. Moreover, to reduce the internet traffic, a group head is defined who is responsible for task distribution and result gathering. However, in all such frameworks, the role of the group leader or task distributor is very critical; this can easily lead to a single point of failure. The computing as a service in VCC is still an emerging area that needs further exploration to provide fault tolerance and reliable services. Moreover, an incentive-based mechanism is required that encourage volunteers to become the part of the vehicular computing grid. Areas that require further attention are the security of distributed tasks, the integrity of obtained results, and authorization/authentication mechanism for identification of legitimate users.

3.2 Storage as a service

In VCC, the storage service is inherently different from traditional cloud storage service. In the cloud, unlimited storage is available at a very economical rate, managed by cloud management software. For example, Amazon provides Simple Storage Service (S3)[2] that can be integrated with applications through Application Programming Interface (APIs). In VCC, it is difficult to provide unlimited storage—the storage relies on multiple vehicles (Fig. 4). In order to provide virtual storage abstractions for VCC, an extensive framework is required. In this section, we explore the existing storage frameworks, techniques, and state-of-the-art mechanisms proposed under the VCC paradigm.

Dressler et al. [24] presented a data storage and retrieval framework based on Virtual Cord Protocol (VCP). The framework is designed to work on built-in storage of parked vehicles, that dynamically creates a vehicular network, eliminating the need for RSUs. The data management is performed through an overlay network. The framework utilizes the concept of a distributed hash table to store and retrieve chunks through hashing. The framework works on the principle of clusters and virtual cord. On receiving a hello message, the new vehicle can connect the existing cord or create a new virtual cord. In case, the hello message is received from multiple cords, the new vehicle can prioritize cord based on maximum neighbors. After joining the virtual cord network, the vehicle adjusts its routing table. However, in case no control message is received within the fixed time interval, the vehicle can start a new cord. All nodes in a cord form a cluster, also called a domain. The proposed framework can also support inter-domain routing. In every cord, there is a gateway node that connects with other domains to perform data/query routing.

In order to evaluate the proposed framework, authors have developed an application that can store and retrieve data through publish and lookup function calls. The vehicle departure is considered as a node failure, and the replica of every data chunk is stored on different vehicles. In case of a vehicle departure, the affected data block is replicated on another vehicle. However, there are a few drawbacks of the proposed framework. The broadcast of hello packets and acknowledgments incur a considerable amount of network overhead. Due to dynamic cord creation in the proposed framework, it is also possible that every cord has one vehicle in it, thus increasing the overhead significantly. Lastly, the framework simply drops the request in case adequate storage is not available forcing the user to regenerate request after some time. This can further increase network traffic.

Fig. 4 VCC—Storage as a Service. Usage of vehicular network to support data storage services in off-road and on-road scenarios

Vigneri et al. [25] proposed a framework to utilize the vehicles as data caches to reduce the load on cellular infrastructure. In a typical system, edge nodes are used to cache data which a user can directly access without involving the source node through cellular network. In the proposed framework, base station pushes data to available vehicles in its vicinity. The user request for contents is routed to vehicles enabling the user to directly fetch the data from a nearby vehicle. However, if data is not available, the users wait for the Time To Live (TTL) value before forwarding the request to the cellular network. An optimum scheme for management of cached contents and its refresh technique can improve the performance of proposed framework significantly.

Baron et al. [26] presented a Software Defined Network (SDN)-based data carrier framework over vehicular networks. The central controller manages the vehicles to optimally route data to the destination node. The network consists of offloading spots at different locations. The offloading spots serve as temporary storage points where moving vehicles drop data before deviating from the destination node route. Other vehicles can again load data and offload to the next offload point, until the data reaches the destination node. The SDN controller is connected with the offloading spots and can plan the data movement considering direction of vehicles, data request, and data transfer from offloading spots to the destination. Similarly, the controller also ensures the reliability of data transfer using redundancy and Automatic Repeat reQuest (ARQ)-based techniques.

Another data forwarding framework in VANET is proposed by Wu et al. [27]. The framework uses low cost and bandwidth efficient unicast communications for handover of data between vehicles. The vehicles share their location with other vehicles through beacon messages. Before leaving a specified region, a vehicle first offloads to a new vehicle using a fuzzy logic algorithm based on throughput, stability, and bandwidth. To reduce contention at the Media Access Control (MAC) layer, a clustering mechanism is used to limit the number of senders in dense network scenarios. For the formation of the cluster, the algorithm considers channel conditions, number of neighbors heading in the same direction, and velocity of the vehicle. Aforementioned parameters are shared with other vehicles using hello messages. Each node computes competency value for itself and neighbor vehicles. If competency value of the vehicle is larger than all nodes in its R/2 range (R being the transmission range), the node declares itself as a cluster head. Each node maintains a local table for next hop either for the destination or cluster head. If the destination is not within the same vicinity of a vehicle, data is forwarded to the cluster head of the destination. The proposed framework is verified through theoretical analysis and simulations.

Vehicles data storage services are not only limited to the data offloading mechanism, but the combined storage

capabilities can also be used to create a vehicular data center. Mensi et al. [28] proposed a framework to utilize the storage capacity of volunteer vehicles parked in a parking lot or in a traffic jam. The vehicular data center is also connected to the traditional data center. In the proposed framework, a vehicle transfers its stored data to a newly arrived vehicle before moving away. In case, if there is no new volunteer vehicle available, the data is transferred to the coordinator vehicle that holds the data item temporarily until a new vehicle is found. In this model, each vehicle contains a unique data item with no replica being maintained to handle data loss. Data has to be fetched from the traditional data center in case of data loss. The proposed model has been validated through simulations.

Content/data caching is an important technique for the optimization of vehicular networks as it reduces the backhaul cost, transmission load as well as improves the user-perceived experience. The basic concept of caching is to distribute popular content locally. In general, multimedia users are more concerned about the quality of the content rather than its caching location. A niche application of caching is video streaming in vehicular networks. Guo et al. [29] proposed a time-scaled caching scheme for Adaptive Bit Rate (ABR) video streaming in vehicular networks, in which caching is performed at base station (BS). This scheme manages the video quality, cache placement, and video transmission. The caching in vehicular networks is different from the traditional caching in wireless networks due to high-speed mobility. Wei et al. [30] proposed a cache management technique to improve the quality of experience (QoE) for adaptive scale video streaming at an appropriate bit rate. The authors presented the concept of pushing data hop-by-hop to the adjacent nodes. Furthermore, the caching model is presented for the nodes. However, the mobility of the vehicular network makes the selection of potential nodes a complex task. Moreover, average freeze time, freeze ratio, and bit rate were used to evaluate the QoE from the user's perspective.

Caching applications are not limited to multimedia only, and Su et al. [31] presented the concept of caching content in parked vehicles. The parked vehicles have the potential to distribute the large-sized content efficiently as compare to the nodes along the roads. In such frameworks, one of the main challenges is the selection of vehicles for content caching as vehicles can join and leave at any time. To overcome churn behavior of vehicles, Zhang et al. [32] proposed the renewable energy-based green RSUs for caching the contents for vehicular networks.

Lai et al. [33] investigate the effects of cache under modifying channel state information where multiple nodes participate in a decode-and-forward relay mechanism. The authors also presented the selection of secondary nodes to maximize the channel gain. To evaluate the proposed system, authors derived analytical expressions while considering a system with or without cache. Further, the authors conclude that cache-based systems reduce the transmission time.

3.2.1 Research challenges and future directions

The distributed storage service in VCC is helpful for applications that require a significant amount of storage. However, due to inherent ad hoc nature of VCC, it is difficult to provide a reliable storage service. The service assumes the presence of a large number of vehicles that can stay within the region for a relatively long time, e.g., in a parking lot. The research contributions discussed in this section are mainly designed to handle this particular challenge. Moreover, data transfer depends on data rate available on wireless links. Therefore, keeping multiple copies of data within the VCC and continuously synchronizing these is very expensive. To avoid such issues, VCC storage is mostly used for immutable objects sharing.

There exist many research challenges in caching content for vehicular networks, such as cache invalidation and placement [34]. Machine learning and other optimization techniques can be adapted to select suitable vehicles to cache the contents. Some other challenges that need further exploration in this context are data security, user privacy, and resiliency.

3.3 Network as a service

Installing proper infrastructure for a vehicular network is an expensive process because of additional hardware cost. An interesting use of existing vehicular infrastructure has been highlighted in [35] where public busses are employed as mobile gateways. In the proposed framework, busses collect the requests for services and beacons in the form of aggregated messages from their respective communication range and transmit them to the cloud for further processing (Fig. 5). The busses are suitable for this function due to their unique features such as the height of bus means better line-of-sight increasing the overall communication range, predefined time schedule, predefined local routes, and availability of multiple busses on the same segment. Inter-vehicular communication is based on DSRC while the mobile gateway uses the cellular network to communicate with the cloud. However, busses are unable to provide mobile gateway services at late night and during a strike or any other time period during which public busses are not available, which is a major limitation. The proposed model is validated on real data collected from public busses in Seoul, South Korea.

Wu et al. [36] proposed a VANETs based content access model. The cost of accessing the contents through cellular networks are not feasible for everyone. Therefore, the authors have presented a scheme where only cluster head connects to the LTE data network and spreads the content using the V2V communication. The most common

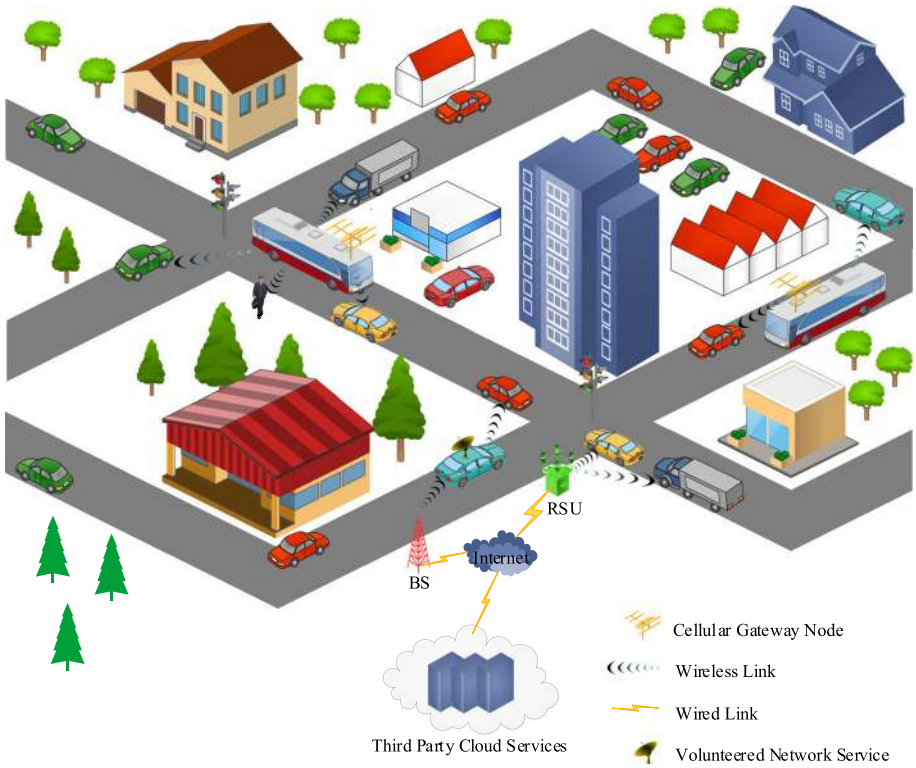

Fig. 5 VCC—Network as a Service Usage of vehicular network to support network services

use case scenarios is the game streaming or sports score updates. An incentive-based mechanism is also introduced to encourage users to become the cluster head for content distribution. Moreover, the proposed model also uses scheduling in order to balance the load among multiple cluster head nodes.

In typical VCC, the requested contents are fetched through the Internet and served to other vehicles through the vehicular network. In [19], authors have presented a technique to access the required contents. The proposed technique is based on a virtualized version of AODV protocol. The content is accessed by the requesting node using the traditional Hyper Text Transfer Protocol (HTTP), while the physical node which has access to the Internet can work as the proxies on HTTP. In each region, a leader and a backup node are selected who has access to the Internet. A leader backup node keeps the record of every message transmitted to the leader. The purpose of having a dedicated backup node is to reduce the failure recovery time. Moreover, snapshots of adjacent regions are recorded, and whenever a leader of a region leaves it transmits its state information to the leader of the adjacent leader.

In the vehicular cloud, the Quality of Service (QoS) is difficult to maintain. Garai et al. in [37] proposed a three-layered hierarchical vehicular cloud architecture to achieve better QoS. The first layer deals with the organization of vehicles that connects them in a tree topology. The second layer deals with the cloudlets formed through the roadside units, whereas the third layer deals with the cloud formation over the Internet. The proposed framework enables QoS aware creation and migration of Virtual Machines (VMs) in RSUs to cope with the high mobility of the vehicles. Techniques have also been proposed to handle the handover management, bandwidth aggregation, management, and estimation of de-jitter buffer to improve the QoS.

Sibai et al. in [38] have also presented a technique for maintaining connectivity while the service provisioning is in process. As the service provider and requestor both are the vehicles, the duration of their communication is an important parameter to consider. The requested service is assumed to be spanned over the sensing data, storage of data, computing capabilities, or networking parameters such as customized routing and on-demand bandwidth. The service type and service duration are specified at the time of vehicular cloud formation. In the proposed model, a requesting vehicle forms a cloud. The cloud formation includes a process to find volunteer vehicles which can be part of the cloud to provide the services. The vehicle responsible for formation is also responsible for destructing the cloud upon successful completion

of requested service. The service requestor broadcasts the service request for cloud formation, top three service providers are shortlisted, and the requestor can choose any of them according to the pre-defined criteria such as computing power, storage, or bandwidth offered. The authors have assumed that pre-defined routes of vehicles can be accessed through cloud leader using vehicle location.

In [39, 40], Aloqaily et al. proposed a framework that caters for the privacy, latency, and cost to provide the optimal Quality of Experience (QoE). The main objective of the proposed models is to address the privacy and latency issues. Moreover, the framework also ensures safe and secure migration from one service provider to the other. Thus, the user can access services from multiple providers and gets charged only once. On the successful completion of service, user rate the third party for its services keeping in mind the cost, latency, and privacy factors. Such models are suitable for pay as you go scenarios.

3.3.1 Research challenges and future directions

The QoE and QoS are difficult to maintain in VCC, as the complete system is based on a number of vehicles available in the vicinity. Moreover, all neighboring vehicles may not be willing to become a part of the communication grid. A more robust mechanism is therefore required to support quality of services throughout the region of interest. A machine learning-based scheme to manage the existing resources and to efficiently utilize the connectivity to the cellular networks can be adapted to support dynamic provisioning of QoS.

3.4 Application as a service

Vehicular cloud network can be used to support various ITS-related applications such as road safety, traffic density, navigation guidance, and parking lot availability (Fig. 6). Moreover, road safety is not only limited to the vehicles but also involves the pedestrians on the roads. Therefore, applications also exist that facilitate the pedestrians and commuters. This section covers applications designed to cover all these areas.

Hwang et al. [41] presented a Safety Aware Navigation Application (SANA) to ensure the safety of pedestrians/strollers along the roads. In the proposed framework, it is assumed that all vehicles and the strollers connected with SANA have a smart cell phone or onboard computing and communication resources. The vehicular cloud is managed by a trusted traffic control center that keeps the log of current positions of vehicles as well as strollers. If a collision is predicted by SANA, it generates a pre-warning message to both vehicle and stroller. In case the probability of collision increases, warning messages are generated to both of them. The decision of sending a pre-warning message or a warning message is based on the distance between the stroller and vehicle. In order to make the application energy efficient, only vehicles near the specific stroller are considered for collision detection. However, it is worth mentioning that false positive and false negative predictions can cause unnecessary distractions. Moreover, when a collision between vehicle and pedestrian is imminent, then looking at a warning message might just distract the pedestrian or vehicle's driver even more and make the situation even more critical.

Jeong et al. [42] presented a navigation-based vehicular cloud framework called Self-Adaptive Interactive Navigation Tool (SAINT). A traffic control center is the backbone of the navigation application, and each requestor sends the navigation query to the traffic control center. The main objective of this application is to provide the details of the least congested route between a source and destination. The proposed framework is self-adaptive. It maintains the link congestion matrix of diverted traffic while considering the overall capacity of the route so that no more vehicles are diverted on that particular route to avoid congestion. Traffic light schedule is also considered while suggesting the routes to vehicles increasing the efficiency of application.

Integration of social media platforms with vehicular clouds is an interesting area of research. Kwak et al. [43] presented a social vehicle navigation system whereby users share the geo-tagged traffic images, videos, or messages referred as Navigation Tweets (NaviTweets). After processing, the system builds an online visual traffic information system referred as traffic digest. Smartphone users can employ this navigation application to query the system to check the traffic intensity and expected traveling duration on their desired routes. However, system performance is dependent upon the traffic intensity on that particular route and how many users are willing to share the traffic information with the system.

Services provided through VCC are not only limited to the road users. The law enforcement agencies can use VCC for surveillance and rule violation monitoring. Mallissery et al. [44] presented a framework that allows traffic police to monitor violations remotely. The traffic police manages the vehicular cloud, and all moving vehicles become the members of the cloud upon issue of registration by the registration authority. The cloud system collects speed readings from the onboard sensors which compares it with pre-defined applicable speed limits on that particular section of the road to detect the violation. All the communication is encrypted to preserve the user privacy.

Entertainment is another interesting area of research under the vehicular domain. The entertainment services include but are not limited to video streaming and interactive gaming for passengers. Jelassi et al. [45]

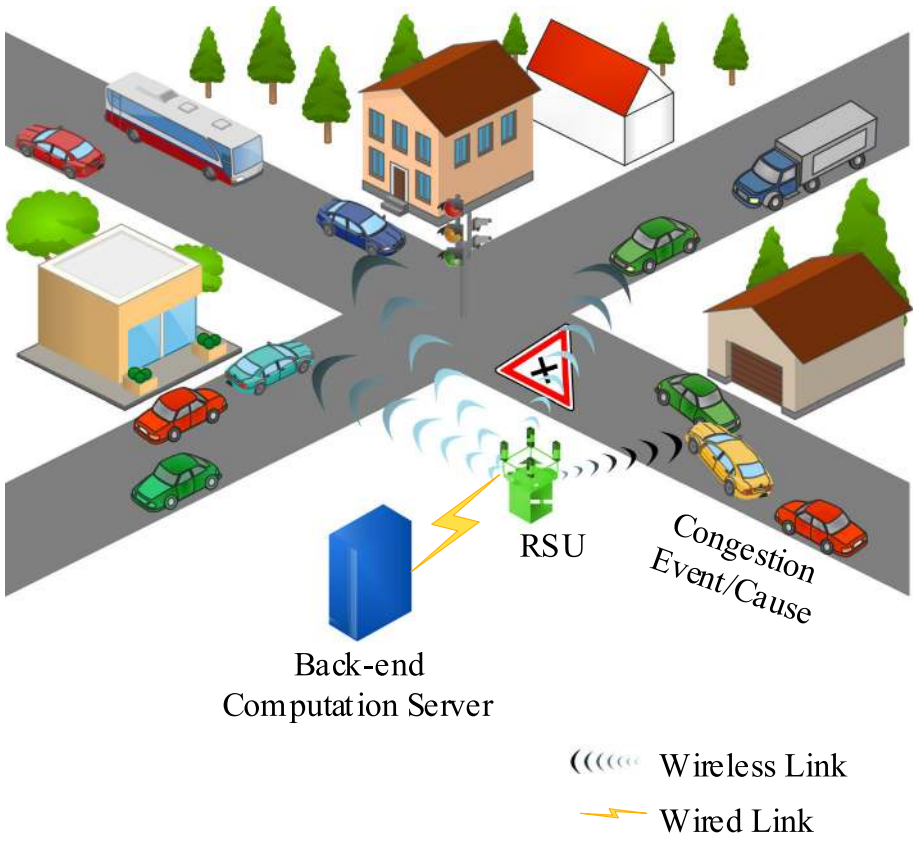

Fig. 6 VCC—Application as a Service Usage of vehicular cloud network to support applications such as road safety, traffic density, and navigation

presented a framework that exploits traditional cloud, RSUs, and vehicular cloud to provide video streaming to the mobile end users. The proposed framework integrates the video player installed on vehicles with VM placed at RSUs/cloudlets, and VM residing at traditional cloud data center. The VMs placed at cloudlets are responsible for video fetching, streaming, caching, and maintaining QoE. On the other hand, client software installed on vehicles supports fetching and streaming. For streaming, vehicles request to VMs that are installed on roadside cloudlets acting as the servers. If the video manager of the cloudlets is unable to find a video, it forwards requests to the traditional cloud. Uninterrupted video streaming is thus provided with the help of vehicles and roadside cloudlets.

The application design based on the vehicular network must handle the connection issues due to the inherent mobility. Authors in [46] proposed a distributed and fault tolerant job assignment mechanism for vehicular nodes. It does not require saving state to any centralized server. The mechanism works by assigning each job to two vehicles in the parking lot. As a vehicle can leave and join the network at any time, the second vehicle performs fresh recruitment. It first stops and saves the state of Virtual Machine (VM) image and then copies it to any other available vehicle. Once this is done, VM is again started on both of vehicles. The overall job is terminated when one of the vehicles finishes execution. The checkpoint is an effective approach in providing fault tolerant computing resources. In [47], authors presented the two-level checkpoint strategy to improve job completion rate for different job sizes. The proposed mechanism reduces the risk for missing checkpoint while executing computation-intensive jobs. The node which lies close to the executing node and in between initiating and executing node is used as a first level server for the checkpoint, while job initiating node serves as a second level checkpoint server. When executing node leaves the network, middle node reports checkpoint to initiate new node selection with the previously saved states.

3.4.1 Research challenges and future directions

Highly dynamic nature of VANETs and the use of wireless medium, where interference-related problems are common, restrict the performance of VCC to provide entertainment applications. One approach is to reduce the number of transmissions required for communications [48]. Though this approach promises minimum interference; however, it also restricts nodes from broadcasting,

resulting in minimum delivery of the content. This trade-off demands an effective broadcast approach to share the contents with maximum nodes in less number of transmissions.

4 Vehicular simulation tools

Simulation is a popular choice for modeling of real-life activities/objects because it offers a cost-effective and scalable mechanism to analyze the behavior of the model under different conditions/parameters. For VANETs, as practical experiments are not feasible, one has to rely upon simulation of network protocols for evaluating their performance. Different types of simulators, both commercial and open source, are available that can be employed to run simulation modeling for VANETs.

Traffic simulations are classified in three different categories based on the levels of detail. Microscopic simulations provide the most detail to the level of individual vehicles in the system. Mesoscopic simulations treat the traffic as platoons of homogeneous vehicles using, e.g., aggregated speed-density functions to model their behavior. Macroscopic simulations model traffic at a large scale focusing on aggregated status of traffic. Clearly, microscopic simulations provide the finest level of detail enabling accurate modeling of traffic behavior but are the slowest to run.

Generally for vehicular environment, one option available is to use mobility generators to model the vehicle characteristics such as traffic, spatial, and temporal mobility and to generate mobility traces. These traces are then fed to a network simulator that models the communication between the vehicles. Note that mobility traces can also be generated by observing real vehicles in a city or highway environment and processing these mobility observations for use in network simulations. One inherent problem with these trace-driven network simulations is that they cannot study the effect of changing network parameters on the traffic mobility. Also, it is restricted to using the mobility pattern fed through the traces. An alternative is to utilize a VANET simulator, a class of simulator that has integrated the VANET mobility support within the simulator itself.

In this section, we provide an overview of the most commonly used vehicular simulation tools to assist the interested researchers in selecting the most suitable simulator based on their specific requirements. We detail the characteristics of mobility generators, network simulators and VANET simulators separately. Figure 7 shows the taxonomy of simulation tools discussed in these three categories. We also list publicly available vehicular mobility data sets that can be utilized for running network simulations.

Mobility generator is a simulation application, that models the vehicular traffic in order to study the mobility aspects of the vehicles. Mobility generation is not a trivial task, it is affected by many factors such as road length, speed limits, vehicle type, vehicle density, number of lanes, deployment of RSUs on roads, and traffic events such as accidents. Driver's behavior that influences the mobility pattern also varies among individuals' characteristics such as age, gender, acceleration, deceleration, and overtaking criteria [49]. Different mobility generators differ in their support for available network simulators.

4.1 Mobility generators

There are a number of well-known traffic simulators used by the transportation engineering researchers to aid in decision making in traffic engineering of large-scale cities. However, the design space of using the output of these simulators for VANET communications remains largely unexplored. Their compatibility with existing network simulators is thus an open research issue. We include some of these traffic engineering simulators for completeness purpose. Table 1 lists some of major characteristics of the commonly adopted traffic generators.

- SUMO (Simulation of Urban Mobility) [50] is an open-source traffic simulator. The main features of SUMO include the traffic portability, handling of large road networks, multiple vehicles categories, vehicle collision avoidance, traffic junction support, and single and multiple vehicle routing with Graphical User Interface (GUI) support. It also supports the traffic simulation on real locations across the globe. Its output cannot be directly incorporated with a network simulator; however, third party open-source software is available to convert its output for import in Objective Modular Network Testbed (OMNeT++).
- MOVE (MObility model generator for VEhicular networks) [51] is an extension of SUMO for realistic traffic pattern generator in VANET simulations. It supports the traffic generation using the GUI, resulting in saving time and effort of writing traffic generation scripts. It also supports the map editing function. The output of MOVE can be directly incorporated with several network simulators such as Network Simulator 2 (NS-2) and Global Mobile Information System Simulator (GloMoSim).
- CityMob [52] is a traffic generator with multiple models. Mobility models include Downtown model, Simple model, and Manhattan model. It supports the uniform blocking size, non-uniform traffic distribution as in realistic environment, effect of traffic density on vehicle movement speed, all streets

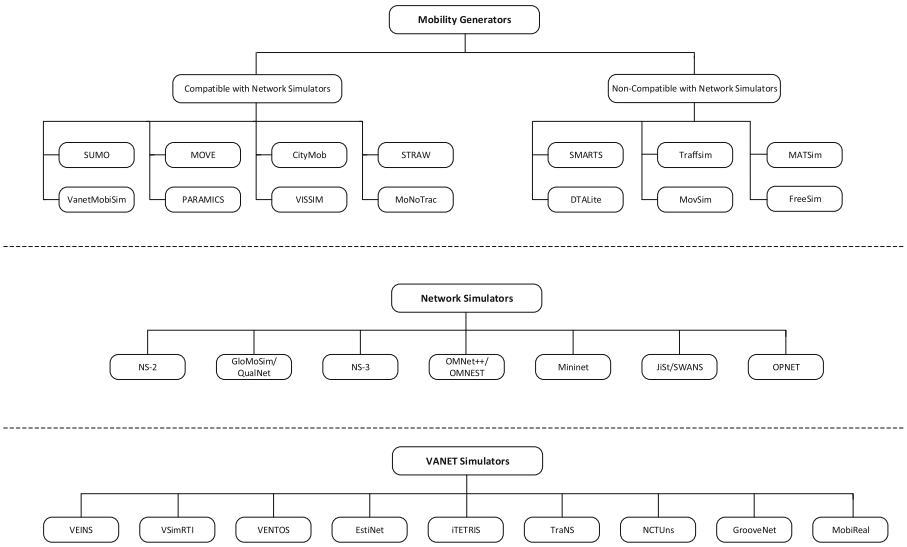

Fig. 7 Taxonomy of mobility generators, network, and VANET simulators' list of all mobility generators, network simulators, and VANET simulation tools discussed in this work

with bidirectional lanes, pre-defined range for vehicle movement speed, and vehicle queuing.

- STRAW (Street Random Waypoint) [53] is a traffic generator based on real geographical location in USA. STRAW output is directly compatible with the network simulator SWANS only. Modifications are required in order to incorporate the output of STRAW with other network simulator such as NS-2.
- VanetMobiSim [54] is a JAVA-based traffic generator that produces realistic traffic at microscopic and macroscopic level. It supports the additional feature of importing real geographical US-based maps beside generating random maps. VanetMobiSim also supports the pre-compiled intelligent driving models such as Intelligent Driving Model with Lane Changing and Intelligent Driving Model with Intersection Management. The output of VanetMobiSim is directly compatible with multiple network simulators such as NS-2, GloMoSim, and QualNet.
- PARAMICS [55] is a commercial traffic generator application. Offering more enhanced features for mobility patterns such as 3D visualization, support for economical evaluation, large-scaled traffic generation, and integration of real statistics.
- VISSIM [56] is a commercial multi-modal traffic generator developed and maintained by Planung Transport Verkehr (PTV) AG group, which specializes in transport engineering and related solutions. Basically it is a microscopic level traffic simulator but a mesoscopic module is also available. It also supports the hybrid simulation (mixture of microscopic and mesoscopic level traffic flows). It can simulate more than one kind of traffic, e.g., vehicles, public transport, pedestrians, and cycles. Moreover, the output traces are exportable into 3D graphical platforms such as AutoCAD 3D max software.
- MoNoTrac [57] (Mobile Node Trace generator) is a mobility generator developed in JAVA, which aims to ease generation of mobility data based on real geographical data. The user first defines description using frontend and selects map to simulate from Openstreet Map. Then nodes along with their configurations are defined. User can also give simulation options such as simulation time. In addition, replication and specific position of the nodes can be defined. It generates eXtensible Markup Language (XML)-based meta-format which needs to be converted to other format for use with network simulators like OMNET++, NS-2, and Network Simulator 3 (NS-3). With the help of a plug-in, it also supports custom models.
- SMARTS (Scalable Microscopic Adaptive Road Traffic Simulator) [58] is a microscopic level distributed mobility generator that supports the execution of multiple process in parallel making it capable of simulating huge number of vehicles (in millions). Due to its quick simulation time, it can be used as a traffic forecaster for the future events. Moreover, the output trace can be exported to multiple formats. A major limitation of SMARTS is the assumption of uniform rational car behavior for all the traffic.
- TraffSim [59] is a JAVA-based platform independent traffic simulator capable of executing simulation

Table 1 Mobility generators

	Active development	Map support	Traffic model	Network simulator compatibility	Major characteristics
SUMO [50]	✓	Both real and user defined	Microscopic	NS-2, NS-3, OMNeT++	Lane change, collision avoidance, multiple vehicles types, traffic intersections, speed control
MOVE [51]	×	Both real and user defined	Microscopic	NS-2, QualNet, GloMoSim	Lane change, collision avoidance multiple vehicles types, traffic intersections, speed control
STRAW [53]	×	Built-in models only	Microscopic	NS-2	Lane change, collision avoidance, multiple vehicles types, speed control
STRAW [53]	×	Real and maps only	Microscopic	SWANS	Lane change, speed control
Vanet-MobiSim [54]	×	Both real	Microscopic	NS-2, QualNet, OMNeT++ GloMoSim	Lane change, speed control traffic intersections
PARAMICS [55]	✓	Both real and user defined	Microscopic	OMNeT++, NS-2	Lane change, collision avoidance multiple vehicles types, traffic intersections, speed control
VISSIM [56]	✓	Both real and user defined	Microscopic and mesoscopic	NS-2, QualNet	Lane change, collision avoidance, multiple vehicles types, traffic intersections, speed control
MoNoTrac [57]	×	Real maps only	Microscopic	NS-2, NS-3, OMNeT++	Speed control, collision avoidance
SMARTS [58]	✓	Both real and user defined	Microscopic	N/A	Lane change, collision avoidance multiple vehicles types, traffic intersections, Speed control
Traffsim [59]	✓	Both real and user defined	Microscopic	N/A	Lane change, collision avoidance multiple vehicles types, traffic intersections, Speed control
MATSim [60]	✓	Both real and user defined	Microscopic	N/A	Lane change, collision avoidance multiple vehicles types, traffic intersections, Speed control
DTALite [61]	✓	Real maps only	Mesoscopic	N/A	Lane change, speed control, traffic intersections
MovSim [62]	✓	Built-in maps only	Microscopic	N/A	Lane change, traffic intersections, multiple vehicles types, speed control
FreeSim [63]	×	Real maps only	Microscopic and macroscopic	N/A	Speed control

scenarios in parallel. Basically, it is a microscopic level traffic generator with the support of speed calculations, distance traveled, fuel consumption, carbon dioxide emissions, traveling time, and other such parameters for each vehicle individually. It also supports the occasional events such as traffic jams and congestion in peak hours. XML is used for the scripting of each simulation. It can support various third party maps such as Google Maps or even hybrid satellite road layouts.

- MATSim (Multi Agent Transport Simulation) [60] is a microscopic traffic simulator, originally designed to study the traffic movement of Swiss daily traffic spanning over more than 7 million trips each day. It

supports XML-based input and output for setting parameters of nodes/persons. By default, it uses second as the smallest time entity and the overall simulation time depends upon the computing speed of the machine. It also supports variation in quality and quantity in data as input.

- DTALite [61] is a mesoscopic level open-source DTA (Dynamic Traffic Assignment) traffic generator, which is used with NeXTA (Network eXplorer for Traffic Analysis) for the simulations. Output traces are supported in the 3D environment. DTALite can be used for large-scale transportation modeling applications because of its support for multi-threaded processing, which significantly reduces model run-times. Model includes the impacts of work zones, proposed freeways/ highways, and tolling facilities on the traffic flows.
- MovSim (Multi-model open-source vehicular-traffic Simulator) [62] is another JAVA-based platform independent microscopic simulator. It supports XML-based configuration and produces csv text output. It has implemented several different models for car following and uses its own physics-based fuel-consumption model and lane change models. The simulator does not support multiple lane junctions.
- FreeSim [63] is an open-source customizable traffic generator for microscopic and macroscopic level traffic. It supports the traffic algorithm for single and multiple vehicles on multiple lanes and data collected on real time can also be used for traffic generation. The major advantage of FreeSim is its support for ITS as it enables the communication of vehicles with the monitoring system on the highways.

4.2 Network simulators

The basic function of a network simulator is to analyze the performance of network protocols under different network topologies and settings. Although there are many network simulators available, we are particularly interested in network simulators that are suitable to use with the vehicular network simulations. We highlight the simulators that support the new IEEE 802.11p standard for data exchange in high-speed V2V and V2I scenarios without involving association and authentication mechanism of vanilla 802.11 standard. Table 2 highlights the features' comparison of the discussed network simulators.

- NS-2 (Network Simulator 2) [64] is a commonly used discrete event network simulator. It uses C++ simulation kernel and Object-oriented Tool Command Language (OTCL) for simulation modeling. It is open source; thus, new modules are easy to implement. Wireless support includes node mobility, propagation modeling for radio, and 802.11p protocol. NS-2 also supports the event scheduler, which maintains the event log and executes the simulation accordingly.
- GloMoSim [65] is an open-source multi-layered parallel discrete event simulator. Due to its multi-layered architecture, it is very convenient to incorporate different models at different layers using the standard APIs. It supports the simulation scripts written in C programming language. QualNet is the commercial version of the GloMoSim written in C++ that offers very high scalability factor as compared to GloMoSim.
- NS-3 (Network Simulator 3) [66] is the replacement of NS-2 to cater for the modern network research requirements. TCL is no longer needed for simulation modeling, and Python script support is enabled. NS-3 does not support backward compatibility with NS-2 and has the extended software integration support with the other open-source network models.
- OMNeT++ [67] (Objective Modular Network Test bed in C++) is an open-source component-based C++ discrete event simulator with a GUI support. Several modules are glued together to form a simulation setup. The output from OMNeT++ is in the form of text files that can also be processed by other tools such as MATLAB or R. It is free only for academic and non-profit use; a commercial version OMNEST is also available.
- Mininet [68] is an open-source network emulator which leverages virtualization to run networking scenarios where each host, networking switch or router act like a real machine. It provides an extensible Python API for network creation and experimentation. Simulation of wireless networks is enabled with the help of Mininet-WiFi add-on. Simulations of SDN in wireless networks as well as 802.11p protocol is supported.
- JiST (JAVA in Simulation Time) [69] is a JAVA-based open-source discrete event simulator. Whereas SWANS is a wireless network simulation platform build on the JiST platform to meet the modern requirements of the networking simulations. SWANS is capable of simulating large-scaled wireless sensor networks by consuming fewer resources as compare to the GloMoSim and NS-2.
- OPNET [70] (OPtimized Network Engineering Tools) is a discrete event commercial network simulator. It provides an interactive GUI to define topologies and to configure parameters from application layer to the physical layer. It uses object-oriented programming paradigm to map GUI-based input to simulation of the real systems. Users can also define custom packet format using OPNET. Its open interface supports the use of external libraries with OPNET.

Table 2 Network simulators

	Active development	802.11p support	Architecture language	Simulation language
NS-2 [64]	×	NS-2.33	C++	C++ & OTCL
GloMoSim [65]	×	×	C	C
NS-3 [66]	✓	✓	C++ & Python	C++ & Python
OMNet++ [67]	✓	✓	C++	C++
Mininet [68]	✓	✓	Python	Python
JiST/SWANS [69]	×	×	JAVA	JAVA
OPNET [70]	✓	✓	C++	C++ & OTCL

4.3 VANET simulators and frameworks

In this category, we discuss simulators and frameworks that integrate the mobility patterns in the VANET environment with the network simulators (Table 3). These VANET simulators and frameworks thus provide an integrated framework for executing the simulations without the need to run different softwares and resolving their inter-dependencies.

- VEINS (Vehicles in Network Simulation) [71] is an open-source VANET framework that uses SUMO as the mobility generator and OMNeT++ as the network simulator. VEINS is bidirectionally coupled between network and traffic simulators enabling it to model the influence of road traffic on network traffic and also vice versa. For example, vehicles need to change their route or slow down (change in mobility pattern) on reception of a warning message generated by the network simulator. Message exchange between mobility generator and network simulator takes place using TCP. Real maps can be used for road layout plans. Moreover, an environmental application has also been integrated that monitors the carbon dioxide gas released by the vehicles.
- VSIMRTI (V2X Simulation Runtime Infrastructure) [72] is a VANET simulator developed by DCAITI (Daimler Center for Automotive Information Technology Innovations), Germany, that provides a flexible framework for adaptive evaluations of ITS protocols. The framework can be compiled using any of the available traffic generators SUMO, VISSIM, etc. Similarly, for network simulator, it can use NS-3, OMNeT++, JiST/SWANS, and VSimRTI_Cell (cellular network simulator). It thus allows easy integration and exchange of simulators. Since it adopts a layered architecture, it can be used for the simulation of latest ITS applications using VSIMRTI_AppNT module.
- VENTOS [73] (VEhicular NeTwork Open Simulator) is a cooperative platoon management simulator that uses SUMO as mobility generator and OMNeT++ as network simulator. This advanced simulator also enables the V2I communication using DSRC enabled wireless communications. Major features consist of bi-directional message exchange using Simple Network Management Protocol (SNMP) protocol, adversary module for security attacks, and dynamic routing of traffic.
- EstiNet [74] is a commercial network emulator and simulator maintained by EstiNet Technolgies Inc. The VANET add-on enables the V2V and V2I simulation environment for high-speed environments with minimal latency. For the mobility generator functionality, it provides built in support for customized maps or importing roadmap traces and defining mobility pattern/parameters (lane changing, overtaking, traffic signals, trip type, etc.). It also provides support for IEEE802.11p for fast transmission of data.
- iTETRIS [75] (an Integrated) is a part of FP7 (Framework program 7) for the ITS applications. It couples the SUMO as the mobility generator and NS-3 as the network simulator with the help of iCS (iTETRIS Control System). The major objective of iTETRIS is to design an open-source VANET simulation platform for the large-scaled V2V and V2I communication using the state-of-the-art simulators. Due to advanced capabilities of NS-3 and architecture of iTETRIS, it can handle multiple transmissions between vehicles at the same time using multiple wireless communication medium, which is close to realistic scenarios.
- TraNS (Traffic and Network Simulation Environment) [76] is an open-source integration of SUMO and NS-2, with an objective of simulating close to real VANET environment. It is not under active development and the last release (Trans V1.2) was in 2009; hence, it is not compatible with later versions of both SUMO and NS-2. It includes the generation of random routes for vehicles, simulating

Table 3 VANET simulators

	Active development	Mobility generator	Network simulator	Traffic model	Major characteristics
VEINS [71]	✓	SUMO	OMNeT++	Microscopic	Car following, traffic light, multiple flow models, environmental model, speed model
VSimRTI [72]	✓	SUMO, VISSIM PHABMACS	OMNeT++, SNS NS-3, SWANS	Microscopic	Cellular network simulation support, car following, traffic light, speed model
VENTOS [73]	✓	SUMO	OMNeT++	Microscopic & Mesoscopic	MATLAB compatibility for plots, dynamic traffic routing, car following & traffic light models
EstiNet [74]	✓	Built-in	Estinet	Microscopic	Car following, traffic light, multiple traffic flow models, speed model
iTETRIS [75]	×	SUMO	NS-3	Microscopic	Emission and noise models, traffic rerouting, car following & speed models
TraNS [76]	×	SUMO	NS-2	Microscopic	Danger warning and accident simulation support, random & flow-based vehicles, car following, dynamic rerouting, manual traffic lights
NCTUns [77]	×	Built-in	Built-in	Microscopic	Speed models, random flow-based vehicles, car following, automatic traffic lights
GrooveNet [78]	×	Built-in	Built-in	Microscopic	Speed models, random flow-based vehicles, car following, manual traffic lights
MobiReal [79]	×	Built-in	GTNetS	Microscopic	Speed model, car following, manual traffic light, Probablistic & Rule based flows

traffic events, and visualization using the Google Earth API. It also contains pre-modeled applications for road safety and traffic efficiency.

- NCTUns (National Chiao Tung University network simulator) [77] is an integrated network simulator and emulator that utilizes Linux TCP/IP stack to provide simulation results with high fidelity. Moreover, on a multicore computing machine, NCTUns is capable of executing parallel simulations. With the built-in GUI interface, users can have customized network topologies, modification of modules inside the nodes, plotting graphs, and animation for packet transmission.
- GrooveNet [78] is a hybrid VANET simulator as it enables the communication between real vehicles and simulated vehicles. Any new designed protocol can be analyzed on the simulated roads. As it supports visualization of real road network, any geographical location of USA can be used for modeling, even at street level. This simulator is also Linux based.
- MobiReal [79] is a network simulator capable of simulating the mobility of vehicles as well as mobility of humans. It supports algorithms for traffic congestion, pedestrian's collision avoidance, and simulating multiple mobility modules simultaneously. Basically, it utilizes GTNetS (Georgia Tech Network simulator) to incorporate the mobility in network simulator. For vehicular mobility, it uses the NETSREAM, which is a non-open-source release from TOYOTA Motors limiting the use of MobiReal. However, other traffic simulators can be incorporated with MobiREAL to provide the vehicular mobility functionality.

4.4 Vehicular datasets

Accurate modeling of vehicular networks is essential to get acceptable results in the simulations. Because the vehicular model for mobility analysis has a great influence on the VANET simulations, one viable option is to re-use the available vehicular data sets for research purposes. The performance of a new protocol can thus be compared with existing solutions utilizing the same data set. Although there are many vehicular data sets available, we only focus on the data sets generated in the last decade as these represent updated road layouts and capture the modern vehicle characteristics. A list is compiled in Table 4 showing the data set geographical location and details along with the references.

4.4.1 Research challenges and future directions

We foresee a substantial increase in V2X applications utilizing VCC for diversified purposes. These applications require support from appropriate VANET simulator for

Table 4 Available mobility datasets

Origon	Detail
Rome (Italy)	Mobility traces of Taxi Cabs [81]
Shanghai (China)	Mobility traces of Taxi Cabs [82]
San Francisco (USA)	Mobility traces of Taxi Cabs [83]
New York (USA)	Trace set of vehicle with steerable antenna [84]
Washington (USA)	WiFi-based connectivity between vehicles and base station [85]
Asturias (Spain)	Mobility traces based on GPS [86]
Stockholm (Sweden)	Pedestrian mobility (Micro-simulation) [87]
Multiple locations (USA)	Human mobility [88]
Milano (Italy)	Traces of mobile phone based mobility [89]
Aachen (Germany)	Traces of 802.11p packets under RF jamming [90]
Maryland (USA)	Data set for vehicle collision (2012) [91]
Washington (USA)	Data set of new vehicles assessment (safety tests) [92]
Aarhus (Denmark)	Road Traffic Data [93]
Aarhus (Denmark)	Parking Data [94]
Europe	Time stamped real traffic data set [95]
–	Data set for short-ranged communication of V2V and V2I [96]
Cologne (Germany)	Vehicular Mobility Trace [97]
Creteil (France)	Vehicular Mobility Trace (Microscopic Level) [98]

demonstrating their effectiveness. For example, integration of multiple simulators is required for simulation of electric vehicles using power grid [80]. A simulator with modular design is thus desired, where new modules can be developed independently and seamlessly glued together to provide an appropriate platform for performance testing of futuristic applications.

An important consideration, while selecting tools for providing simulation support, is whether the platform uses mobility traces for use in network simulations. One inherent problem with these trace-driven network simulations is that they cannot study the effect of changing network condition/parameters on the traffic mobility. For example, any congestion event detected by a vehicle will result in initiation of a warning message for interested vehicles to avoid the area. The mobility pattern of vehicles in vicinity should change in response of receiving this alert message. However, trace -driven simulation is only confined to studying the effect of mobility on the network performance, hence cannot react to any event not already captured in the mobility traces. A fully coupled simulator can have bi-directional information exchange enabling simulation of realistic events. Further research is required in this direction to develop an efficient simulation framework.

5 Conclusion

In this study, we have reviewed the recent research contributions in the emerging domain of Vehicular Cloud Computing. The VCC is a combination of smart vehicles, ad hoc networks, and ubiquitous sensing. The concept is to employ the under-utilized computing and storage resources available in vehicles by offering these resources to other vehicles or customers. These resources can be efficiently provisioned if managed through a well-defined comprehensive framework developed through standards.

We have explored the recent frameworks to provide computing, storage, and network services through VCC. Moreover, the issues and challenges of existing work have also been discussed at the end of every section. Most of the existing frameworks are designed to support a particular scenario with their own abstractions; therefore, due to lack of standards, multiple VCC federates cannot communicate with others. The other areas that need further explorations are security, privacy, quality of services, and user experience. The vehicular networks are well suited for the content delivery system. However, maintaining multiple copies and keeping these consistent is still an area that needs further exploration. Moreover, integration of machine learning can help in predicting the vehicles that can be used to host services.

This survey can help researchers to understand open areas of research under the VCC domain. The VCC is the emerging paradigm that provides a platform to develop viable solutions such as intelligent transportation system.

Endnotes

[1] An architectural element of fog computing (defined later) that sits between the mobile users and the traditional cloud

[2] https://aws.amazon.com/s3/

Abbreviations

AODV: Ad hoc on-demand distance vector; ARQ: Automatic repeat reQuest; AVE: Autonomous vehicular edge; ITS: Intelligent transportation systems; LTE: Long-term evolution; MAC: Media access control; OBU: On board unit; QoE: Quality of experience; QoS: vuality of service; RSU: Road side unit; V2D: Vehicle to device; V2G: Vehicle to grid; V2H: Vehicle to home; V2I: Vehicle to infrastructure; V2P: Vehicle to pedestrian; V2V: Vehicle to vehicle; V2X: Vehicle to everything; VANET: Vehicular ad hoc networks; VCC: Vehicular cloud computing; VFC: Vehicular fog computing; VM: Virtual machine; WAN: Wide area network

Funding

Not applicable.

Authors' contributions

BA contributed towards Section 1 to 5. A/Prof. AM contributed to Section 2 to 3. TH contributed to Sections 3. A/Prof. NA contributed to Sections 3 and 4. All authors read and approved the final manuscript.

Authors' information

All authors are affiliated with the Department of Computing, School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad, Pakistan.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Received: 31 May 2018 Accepted: 30 November 2018
Published online: 07 January 2019

References

1. A. Falchetti, C. Azurdia Meza, S. Cespedes, in *Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), 2015 CHILEAN Conference On*. Vehicular cloud computing in the dawn of 5g (IEEE, 2015), pp. 301–305
2. E. Lee, E. K. Lee, M. Gerla, S. Y. Oh, Vehicular cloud networking: architecture and design principles. IEEE Commun. Mag. **52**(2), 148–155 (2014)
3. L. Gu, D. Zeng, S. Guo, in *Globecom Workshops (GC Wkshps)*. Vehicular cloud computing: a survey (IEEE, 2013), pp. 403–407
4. M. Whaiduzzaman, M. Sookhak, A. Gani, R. Buyya, A survey on vehicular cloud computing. Netw. J. Comput. Appl. **40**, 325–344 (2014)
5. M. Amadeo, C. Campolo, A. Molinaro, Information-centric networking for connected vehicles: a survey and future perspectives. IEEE Commun. Mag. **54**(2), 98–104 (2016)
6. G. Grassi, D. Pesavento, G. Pau, R. Vuyyuru, R. Wakikawa, L. Zhang, in *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference On*. VANET via named data networking (IEEE, 2014), pp. 410–415
7. A. Rasheed, S. Gillani, S. Ajmal, A. Qayyum, in *Vehicular Ad-Hoc Networks for Smart Cities*. Vehicular ad hoc network (VANET): a survey, challenges, and applications (IEEE, 2017), pp. 39–51
8. S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, H. Zedan, A comprehensive survey on vehicular ad hoc network. Netw. J. Comput. Appl. **37**, 380–392 (2014)
9. J. Kakarla, S. S. Sathya, B. G. Laxmi, *et al*, A survey on routing protocols and its issues in VANET. Int. Comput. J. Appl. **28**, 38–44 (2011)
10. M. N. Mejri, J. Ben-Othman, M. Hamdi, Survey on vanet security challenges and possible cryptographic solutions. Veh. Commun. **1**(2), 53–66 (2014)
11. E. Lee, E.-K. Lee, M. Gerla, S. Y. Oh, Vehicular cloud networking: architecture and design principles. IEEE Commun. Mag. **52**(2), 148–155 (2014)
12. S. Bitam, A. Mellouk, S. Zeadally, Vanet-cloud: a generic cloud computing model for vehicular ad hoc networks. IEEE Wirel. Commun. **22**(1), 96–102 (2015)
13. H. Zhang, Q. Zhang, X. Du, Toward vehicle-assisted cloud computing for smartphones. IEEE Trans. Veh. Technol. **64**(12), 5610–5618 (2015)
14. Open Fog Consortium. https://www.openfogconsortium.org. Accessed 07 May 2018
15. E. Baccarelli, P. G. V. Naranjo, M. Scarpiniti, M. Shojafar, J. H. Abawajy, Fog of everything: energy-efficient networked computing architectures, research challenges, and a case study. IEEE Access. **5**, 9882–9910 (2017). https://doi.org/10.1109/ACCESS.2017.2702013
16. X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, S. Chen, Vehicular fog computing: a viewpoint of vehicles as the infrastructures. IEEE Trans. Veh. Technol. **65**(6), 3860–3873 (2016). https://doi.org/10.1109/TVT.2016.2532863
17. M. Sookhak, F. R. Yu, Y. He, H. Talebian, N. S. Safa, N. Zhao, M. K. Khan, N. Kumar, Fog vehicular computing: augmentation of fog computing using vehicular cloud computing. IEEE Veh. Technol. Mag. **12**(3), 55–64 (2017)
18. X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, S. Chen, Vehicular fog computing: a viewpoint of vehicles as the infrastructures. IEEE Trans. Veh. Technol. **65**(6), 3860–3873 (2016)
19. J. F. Bravo-Torres, M. López-Nores, Y. Blanco-Fernández, J. J. Pazos-Arias, E.F. Ordóñez-Morales, Vanetlayer: a virtualization layer supporting access to web contents from within vehicular networks. Comput, J. Sci. **11**, 185–195 (2015)
20. C. Perkins, E. Belding-Royer, S. Das, Ad hoc on-demand distance vector (aodv) routing. RFC 3561, RFC Editor (2003). http://www.rfc-editor.org/rfc/rfc3561.txt. Accessed 15 May 2018
21. J. Feng, Z. Liu, C. Wu, Y. Ji, Ave: Autonomous vehicular edge computing framework with aco-based scheduling. IEEE Trans. Veh. Technol. **66**, 10660–10675 (2017)
22. P. Ghazizadeh, S. Olariu, A. G. Zadeh, S. El-Tawab, in *Services Computing (SCC), 2015 IEEE International Conference On*. Towards fault-tolerant job assignment in vehicular cloud (IEEE, 2015), pp. 17–24
23. C. Funai, C. Tapparello, H. Ba, B. Karaoglu, W. Heinzelman, in *Global Communications Conference (GLOBECOM), 2014 IEEE*. Extending volunteer computing through mobile ad hoc networking (IEEE, 2014), pp. 32–38
24. F. Dressler, P. Handle, C. Sommer, in *Proceedings of the 2014 ACM International Workshop on Wireless and Mobile Technologies for Smart Cities*. Towards a vehicular cloud-using parked vehicles as a temporary network and storage infrastructure (ACM, 2014), pp. 11–18
25. L. Vigneri, T. Spyropoulos, C. Barakat, in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2016 IEEE 17th International Symposium on A*. Storage on wheels: offloading popular contents through a vehicular cloud (IEEE, 2016), pp. 1–9
26. B. Baron, P. Spathis, H. Rivano, M. D. de Amorim, Y. Viniotis, M. H. Ammar, Centrally controlled mass data offloading using vehicular traffic. IEEE Trans. Netw. Serv. Manag. **14**(2), 401–415 (2017)
27. C. Wu, T. Yoshinaga, Y. Ji, T. Murase, Y. Zhang, A reinforcement learning-based data storage scheme for vehicular ad hoc networks. IEEE Trans. Veh. Technol. **66**(7), 6336–6348 (2017)
28. N. Mensi, M. Guizani, A. Makhlouf, in *Control Engineering & Information Technology (CEIT), 2016 IEEE 4th International Conference On*. Study of vehicular cloud during traffic congestion (IEEE, 2016), pp. 1–6
29. Y. Guo, Q. Yang, F. R. Yu, V. C. Leung, Cache-enabled adaptive video streaming over vehicular networks: a dynamic approach. IEEE Trans. Veh. Technol. **67**, 5445–5459 (2018)
30. Y. Wei, C. Xu, M. Wang, J. Guan, in *Networking and Network Applications (NaNA), 2016 International Conference On*. Cache management for adaptive scalable video streaming in vehicular content-centric network (IEEE, 2016), pp. 410–414
31. Z. Su, P. Ren, X. Gan, in *Communications (ICC), 2014 IEEE International Conference On*. A novel algorithm to cache vehicular content with parked vehicles applications (IEEE, 2014), pp. 5665–5669
32. S. Zhang, N. Zhang, X. Fang, P. Yang, X. S. Shen, in *Communications (ICC), 2017 IEEE International Conference On*. Cost-effective vehicular network planning with cache-enabled green roadside units (IEEE, 2017), pp. 1–6
33. X. Lai, J. Xia, M. Tang, H. Zhang, J. Zhao, Cache-aided multiuser cognitive relay networks with outdated channel state information. IEEE Access. **6**, 21879–21887 (2018)
34. S. Lim, C. Yu, C. R. Das, Cache invalidation strategies for internet-based vehicular ad hoc networks. Comput. Commun. **35**(3), 380–391 (2012)
35. R. Hussain, F. Abbas, J. Son, S. Kim, H. Oh, in *Consumer Electronics (ICCE), 2014 IEEE International Conference On*. Using public buses as mobile gateways in vehicular clouds (IEEE, 2014), pp. 175–176
36. C. Wu, M. Gerla, N. Mastronarde, in *Ad Hoc Networking Workshop (MED-HOC-NET), 2015 IEEE 14th Annual Mediterranean*. Incentive driven LTE content distribution in VANETs (IEEE, 2015), pp. 1–8
37. M. Garai, S. Rekhis, N. Boudriga, in *Computers and Communication (ISCC), 2015 IEEE Symposium On*. Communication as a service for cloud vanets (IEEE, 2015), pp. 371–377
38. R. El Sibaï, T. Atéchian, J. B. Abdo, R. Tawil, J. Demerjian, in *Cloud Technologies and Applications (CloudTech), 2015 IEEE International Conference On*. Connectivity-aware service provision in vehicular cloud (IEEE, 2015), pp. 1–5
39. M. Aloqaily, B. Kantarci, H. T. Mouftah, in *High-capacity Optical Networks and Emerging/Enabling Technologies (HONET), 2014 IEEE 11th Annual*. On the impact of quality of experience (QOE) in a vehicular cloud with various providers (IEEE, 2014), pp. 94–98
40. M. Aloqaily, B. Kantarci, H. T. Mouftah, in *Ubiquitous Wireless Broadband (ICUWB), 2015 IEEE International Conference On*. A generalized framework for quality of experience (QOE)-based provisioning in a vehicular cloud (IEEE, 2015), pp. 1–5
41. T. Hwang, J. P. Jeong, E. Lee, in *Information and Communication Technology Convergence (ICTC), 2014 IEEE International Conference On*. Sana: Safety-aware navigation app for pedestrian protection in vehicular networks (IEEE, 2014), pp. 947–953
42. J. Jeong, H. Jeong, E. Lee, T. Oh, D. H. Du, Saint: self-adaptive interactive navigation tool for cloud-based vehicular traffic optimization. IEEE Trans. Veh. Technol. **65**(6), 4053–4067 (2016)

43. D. Kwak, R. Liu, D. Kim, B. Nath, L. Iftode, Seeing is believing: Sharing real-time visual traffic information via vehicular clouds. IEEE Access. **4**, 3617–3631 (2016)
44. S. Mallissery, M. M. Pai, N. Ajam, R. M. Pai, J. Mouzna, in *Consumer Communications and Networking Conference (CCNC), 2015 IEEE 12th Annual*. Transport and traffic rule violation monitoring service in its: a secured VANET cloud application (IEEE, 2015), pp. 213–218
45. S. Jelassi, A. Bouzid, H. Youssef, in *International Workshop on Communication Technologies for Vehicles*. Qoe-driven video streaming system over cloud-based VANET (Springer, 2015), pp. 84–93
46. P. Ghazizadeh, R. Mukkamala, R. Fathi, in *2015 International Conference on Computing and Network Communications (CoCoNet)*. Modeling and predicting fault tolerance in vehicular cloud computing (IEEE, 2015), pp. 395–400
47. J. Fan, R. Li, X. Zhang, in *2017 7th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC)*. Research on fault tolerance strategy based on two level checkpoint server in autonomous vehicular cloud (IEEE, 2017), pp. 381–384
48. A. Bradai, T. Ahmed, in *Vehicular Technology Conference (VTC Spring), 2014 IEEE 79th*. Reviv: Selective rebroadcast mechanism for video streaming over VANET (IEEE, 2014), pp. 1–6
49. F. J. Martinez, C. K. Toh, J. Cano, C. T. Calafate, P. Manzoni, A survey and comparative study of simulators for vehicular ad hoc networks (VANETs). Wirel. Commun. Mob. Comput. **11**, 813–828 (2011)
50. D. Krajzewicz, J. Erdmann, M. Behrisch, L. Bieker, Recent development and applications of sumo-simulation of urban mobility. Int. J Adv. Syst. Meas. **5**(3&4), 128–138 (2012)
51. K. Lan, *Move: a practical simulator for mobility model in VANET. Telematics Communication Technologies and Vehicular Networks: Wireless Architectures and Applications*. (IGI Global, 2010), p. 14
52. F. J. Martinez, J.-C. Cano, C. T. Calafate, P. Manzoni, in *Communications Workshops, 2008. ICC Workshops' 08. IEEE International Conference On*. Citymob: a mobility model pattern generator for VANETs (IEEE, 2008), pp. 370–374
53. D. R. Choffnes, F. E. Bustamante, in *ACM International Workshop on Vehicular Ad Hoc Networks (VANET), 2005*. An integrated mobility and traffic model for vehicular wireless networks (ACM, 2005)
54. J. Härri, M. Fiore, F. Filali, C. Bonnet, Vehicular mobility simulation with VanetMobiSim. Simulation. **87**(4), 275–300 (2011)
55. G. D. Cameron, G. I. Duncan, Paramics—parallel microscopic simulation of road traffic. Supercomput, J. **10**(1), 25–53 (1996)
56. Planung Transport Verkehr (PTV) AG group. http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim/. Accessed 24 Dec 2017
57. M. Güneş, F. Juraschek, B. Blywis, C. Graff, in *Mobile Adhoc and Sensor Systems (MASS), 2010 IEEE 7th International Conference On*. Monotrac: a mobility trace generator based on OpenStreetMap geo-data (IEEE, 2010), pp. 618–623
58. K. Ramamohanarao, H. Xie, L. Kulik, S. Karunasekera, R.Z.E. TANIN, E. B. Khunayn, Smarts: scalable microscopic adaptive road traffic simulator. ACM Trans. Intell. Syst. Technol. **8**(2), 26 (2016)
59. C. Backfrieder, C. F. Mecklenbrauker, G. Ostermayer, in *Modelling Symposium (EMS), 2013 European*. TraffSim–a traffic simulator for investigating benefits ensuing from intelligent traffic management (IEEE, 2013), pp. 451–456
60. A. Horni, K. Nagel, K.W.e. Axhausen, *The Multi-Agent Transport Simulation MATSim*. (Ubiquity Press, London, 2016). https://doi.org/10.5334/baw
61. X. Zhou, J. Taylor, DTALite: a queue-based mesoscopic traffic simulator for fast model evaluation and calibration. Cogent Eng. **1**, 1–19 (2014)
62. M. Treiber, A. Kesting, An open-source microscopic traffic simulator. IEEE Intell. Transp. Syst. Mag. **2**(3), 6–13 (2010)
63. J. Miller, E. Horowitz, in *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*. FreeSim-a free real-time freeway traffic simulator (IEEE, 2007), pp. 18–23
64. T. Issariyakul, E. Hossain, in *Introduction to Network Simulator NS2*. Introduction to network simulator 2 (NS2) (Springer, 2012), pp. 21–40
65. A. K. Pandey, H. Fujinoki, Study of MANET routing protocols by GloMoSim simulator. Int. Netw. J. Manag. **15**(6), 393–410 (2005)
66. T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, J. Kopena, Network simulations with the NS-3 simulator. SIGCOMM Demonstration. **14**(14), 527 (2008)
67. A. Varga, R. Hornig, in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*. An overview of the OMNeT++ simulation environment (ACM, 2008), p. 60
68. Mininet Network Emulator. http://www.mininet.org. Accessed 27 Dec 2017
69. R. Barr, Z. J. Haas, R. van Renesse, Jist/swans. Wireless Networks Laboratory, Cornell University (2005). http://jist.ece.cornell.edu. Accessed 15 May 2018
70. X. Chang, in *Simulation Conference Proceedings, 1999, vol 1*. Network simulations with OPNET (IEEE, 1999), pp. 307–314
71. C. Sommer, R. German, F. Dressler, Bidirectionally coupled network and road traffic simulation for improved IVC analysis. IEEE Trans. Mob. Comput. **10**(1), 3–15 (2011)
72. R. Protzmann, B. Schnemann, I. Radusch, in *Networking Simulation for Intelligent Transportation Systems: High Mobile Wireless Nodes*. Simulation of convergent networks for intelligent transport systems with VSimRTI (Wiley, 2017), pp. 1–28
73. M. Amoozadeh, H. Deng, C. Chuah, H. M. Zhang, D. Ghosal, Platoon management with cooperative adaptive cruise control enabled by VANET. Veh. Commun. **2**(2), 110–123 (2015)
74. EstiNet Network Simulator. http://www.estinet.com/ns/. Accessed 24 Dec 2017
75. M. Rondinone, J. Maneros, D. Krajzewicz, R. Bauza, P. Cataldi, F. Hrizi, J. Gozalvez, V. Kumar, M. Röckl, L. Lin, *et al*, iTETRIS: a modular simulation platform for the large scale evaluation of cooperative its applications. Simul. Model. Pract. Theory. **34**, 99–125 (2013)
76. M. Piorkowski, M. Raya, A. L. Lugo, P. Papadimitratos, M. Grossglauser, J.-P. Hubaux, TraNS: realistic joint traffic and network simulator for VANETs. ACM SIGMOBILE Mob. Comput. Commun. Rev. **12**(1), 31–33 (2008)
77. S.-Y. Wang, C.-L. Chou, Nctuns simulator for wireless vehicular ad hoc network research. Ad Hoc Netw: New Res. **1**(1), 97–124 (2009). Nova Science Publishers
78. R. Mangharam, D. Weller, R. Rajkumar, P. Mudalige, F. Bai, in *Mobile and Ubiquitous Systems: Networking & Services, 2006 IEEE 3rd Annual International Conference On*. Groovenet: a hybrid simulator for vehicle-to-vehicle networks (IEEE, 2006), pp. 1–8
79. K. Konishi, K. Maeda, K. Sato, A. Yamasaki, H. Yamaguchi, K. Yasumoto, T. Higashino, in *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2005. 13th IEEE International Symposium On*. Mobireal simulator-evaluating MANET applications in real environments (IEEE, 2005), pp. 499–502
80. L. Bedogni, L. Bononi, A. Borghetti, R. Bottura, A. D'Elia, M. Di Felice, F. Montori, F. Napolitano, C. Nucci, T. S. Cinotti, *et al*, An integrated traffic and power grid simulator enabling the assessment of e-mobility impact on the grid: a tool for the implementation of the smart grid/city concept. Tech. Sci. **1**(1), 73–89 (2016)
81. L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici, A. Rabuffi, CRAWDAD dataset roma/taxi (v. 2014-07-17). Downloaded from https://crawdad.org/roma/taxi/20140717 (2014). https://doi.org/10.15783/C7QC7M
82. SUVnet-Trace Data. http://wirelesslab.sjtu.edu.cn/. Accessed 07 Dec 2017
83. M. Piorkowski, N. Sarafijanovic-Djukic, M. Grossglauser, CRAWDAD dataset epfl/mobility (v. 2009-02-24). Downloaded from https://crawdad.org/epfl/mobility/20090224 (2009). https://doi.org/10.15783/C7J010
84. V. Navda, A. P. Subramanian, K. Dhanasekaran, A. Timm-Giel, S. R. Das, CRAWDAD dataset sunysb/mobisteer (v. 2007-06-30). Downloaded from https://crawdad.org/sunysb/mobisteer/20070630 (2007). https://doi.org/10.15783/C7D01R
85. R. Mahajan, CRAWDAD dataset microsoft/vanlan (v. 2007-09-14). Downloaded from https://crawdad.org/microsoft/vanlan/20070914 (2007). https://doi.org/10.15783/C7FG6S
86. S. Cabrero, R. García, X.G. García, D. Melendi, CRAWDAD dataset oviedo/asturies-er (v. 2016-08-08). Downloaded from https://crawdad.org/oviedo/asturies-er/20160808 (2016). https://doi.org/10.15783/C7302B
87. S. T. Kouyoumdjieva, R. Ó.Helgason, G. Karlsson, CRAWDAD dataset kth/walkers (v. 2014-05-05). Downloaded from https://crawdad.org/kth/walkers/20140505 (2014). https://doi.org/10.15783/C7Z30C
88. I. Rhee, M. Shin, S. Hong, K. Lee, S. Kim, S. Chong, CRAWDAD dataset ncsu/mobilitymodels (v. 2009-07-23). Downloaded from https://crawdad.org/ncsu/mobilitymodels/20090723 (2009). https://doi.org/10.15783/C7X302

89. P. Meroni, S. Gaito, E. Pagani, G. P. Rossi, CRAWDAD dataset unimi/pmtr (v. 2008-12-01). Downloaded from https://crawdad.org/unimi/pmtr/20081201 (2008). https://doi.org/10.15783/C7BS3T
90. O. Puñal, C. Pereira, A. Aguiar, J. Gross, CRAWDAD dataset uportorwthaachen/vanetjamming2014 (v. 2014-05-12). Downloaded from https://crawdad.org/uportorwthaachen/vanetjamming2014/20140512 (2014). https://doi.org/10.15783/C7Q306
91. 2012 Vehicle Collisions Investigated by State Police, Maryland. https://catalog.data.gov/dataset/2012-vehicle-collisions-investigated-by-state-police-4fcd0. Accessed 07 Dec 2017
92. New Car Assessment Program (NCAP), Safe Cars Save Lives, NHTSA, Washington. https://webapi.nhtsa.gov/Default.aspx?SafetyRatings/API/5. Accessed 07 Dec 2017
93. Real Road Traffic Data, Aarhus, Denmark. http://iot.ee.surrey.ac.uk:8080/datasets.html. Accessed 07 Dec 2017
94. Real Parking Data, Aarhus, Denmark. http://iot.ee.surrey.ac.uk:8080/datasets.html. Accessed 07 Dec 2017
95. C. Caraffi, T. Vojir, J. Trefny, J. Sochman, J. Matas, in *ITS Conference*. A system for real-time detection and tracking of vehicles from a single car-mounted camera (IEEE, 2012), pp. 975–982
96. R. M. Fujimoto, R. Guensler, M. P. Hunter, H. Wu, M. Palekar, J. Lee, J. Ko, CRAWDAD dataset gatech/vehicular (v. 2006-03-15). Downloaded from https://crawdad.org/gatech/vehicular/20060315 (2006). https://doi.org/10.15783/C74S3Z
97. Vehicular mobility dataset, TAPAS Cologne project, Germany. http://iot.ee.surrey.ac.uk:8080/datasets.html. Accessed 07 Dec 2017
98. Vehicular mobility dataset, Val de Marne, France. http://vehicular-mobility-trace.github.io/index.html#dataset. Accessed 07 Dec 2017