

# Vehicular Cloud Networking: Architecture and Design Principles

Euisin Lee, Eun-Kyu Lee, Soon Y. Oh, and Mario Gerla

---

◆

---

## Abstract

Over the past several decades, VANET has been a core networking technology to provide safety and comfort to drivers in vehicular environments. Emerging applications and services, however, require major changes on its underlying computing and networking models, which demands new network planning for VANET. This article especially examines how VANET evolves with two emerging paradigms - Vehicular Cloud Computing and Information Centric Networking. VCC brings the mobile cloud model to vehicular networks and thus changes the way of network service provisioning, whereas ICN changes the notion of data routing and dissemination. We envision a new vehicular networking system, Vehicular Cloud Networking, on top of them. This article scrutinizes its architecture and operations and discusses its design principles.

## 1 INTRODUCTION

Over the past few decades, lots of research efforts have been devoted to Vehicular Ad-hoc Networks (VANETs), ranging from physical layer communications to networking challenges in Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I). With technological advancement, VANETs provide drivers and passengers with safety and convenience, and furthermore introduce new applications for entertainment and environment monitoring. The vehicular communication is about to evolve with emerging paradigms. The inherent properties of VANET contents and the way that VANET applications consume the contents are leading the evolution. More specifically, (1) vehicles produce and consume a great amount of contents having the property of local relevance (time, space, and consumer); (2) vehicles simply seek contents regardless of their providers; and (3) vehicles collaborate using their resources to create value-added services with minimum help from the Internet infrastructure. These characteristics, started from applications, impact on various aspects in VANET. In particular, they will significantly change the computing (content generation-processing-consumption) and networking (content distribution) models, which leads to a future vehicular networking system - *Vehicular Cloud Networking (VCN)*.

### 1.1 Contribution

This article examines a VCN system that is built on top of Vehicular Cloud Computing and Information Centric Networking and discusses its design principles. More specifically, our contributions include:

- Reviewing emerging VANET applications and existing VANET networking
- Investigating state-of-the-art computing and networking models for vehicular communication
- Introducing VCN and its architecture
- Discussing VCN design principles

## 2 VANET

### 2.1 Emerging Applications on Wheels

Applications in vehicle communications have ranged from safety and convenience to entertainment and commercial services. This subsection discusses three noticeable characteristics observed in emerging VANET applications.

**Application content time-space validity.** Vehicles produce a great amount of content, while at the same time consuming the content. That is, they become rich data “prosumers”. Such contents show several common properties

- 
- E. Lee, E.-K. Lee, and M. Gerla are with Computer Science Department, University of California, Los Angeles, Los Angeles, CA 90095.  
E-mail: euisinlee@ucla.edu and {eklee, gerla}@cs.ucla.edu
  - S. Oh is with UtopiaCompression Corporation, 11150 Olympic Blvd. Suite 820, Los Angeles, CA 90064.  
E-mail: soon@utopiacompression.com
  - E.-K. Lee is the corresponding author.

TABLE 1  
VANET applications and their content properties: local interest, local validity, explicit lifetime.

Application type	Content type	Local interest	Local validity	Lifetime
Active safety	Dangerous road warning	All the vehicles	100 m	10 s
	Car accident warning	All the vehicles	500 m	30 s
	Work zone warning	All the vehicles	1 Km	Construction duration
Public service	Emergency vehicle warning	All the vehicles	500 m	10 min
	Highway Information	All the vehicles	5 Km	All days
Improved driving	Road congestion region information	All the vehicles	5 Km	30 min
	Traffic navigation map	Subscribers only	10 Km	30 min
Business and entertainment	Commercial advertisement	Subscribers only	1-5 Km	1-10 days
	Multimedia file sharing	Subscribers only	1 Km	10 min

of local relevance [1], [2] - local validity, explicit lifetime, and local interest. *Local validity* indicates that vehicle-generated content has its own spatial scope of utility to consumers. In safety applications, for instance, a speed-warning message near a sharp corner is only valid to vehicles approaching to the corner, say within 100 m. *Explicit lifetime* reflects the fact that vehicle content has its own temporal scope of validity. This also implies that the content must be available during its entire lifetime. For instance, the road congestion information may be valid for 30 min, while the validity of roadwork warning must last as long as work is finished. *Local interest* indicates that nearby vehicles represent the bulk of potential content consumers. This concept is further extended so as to distinguish the scope of consumers. For instance, all the vehicles in the vicinity want to receive safety messages, while only a fraction of vehicles are interested in commercial advertisements. Table 1 shows the overview of vehicle applications and their content properties.

**Content-centric distribution.** Vehicle applications are mainly interested in content itself, not its provenance. This memoryless property is characteristic of VANETs. In the fixed Internet, when one wants to check traffic congestion, she visits a favorite service site. Namely, the explicit site's URL guarantees access to ample, reliable information. In contrast, vehicle applications flood query messages to a local area, not to a specific vehicle, accepting responses regardless of the identity of the content providers. In fact, the response may come from a vehicle in the vicinity that has in turn received such traffic information indirectly through neighboring vehicles. In this case, the vehicle does not care who started the broadcast. This characteristic is mainly due to the fact that the sources of information (vehicles) are mobile and geographically scattered.

**Vehicle collaboration sharing sensory data.** Emerging vehicle applications consume a huge amount of sensor data in a collaborative manner. That is, multiple sensors, installed on vehicles, record a myriad of physical phenomena. Vehicle applications collect such sensor records, even from neighboring vehicles, to produce value-added services. In MobEyes [3], for example, vehicles use a few sensors (including a video camera) to record all surrounding events such as car accidents while driving. Thereafter, Internet agents and/or mobile agents (e.g., police) search the vehicular network for witnesses as part of their investigations. The CarSpeak application [4] enables a vehicle to access sensors on neighboring vehicles, in the same manner as it can access its own. The vehicle then runs an autonomous driving application using the sensor collection, without knowing who produced what.

## 2.2 Networking

The existing VANET networking model has been derived primarily from traditional wired networking protocols as illustrated on the left-hand side in Fig. 1. However, due to the huge difference between the Internet and the infrastructureless ad hoc condition, the model shows several intrinsic limitations.

First, the VANET protocol still assumes using IP address to represent a host. Assigning IP addresses to moving nodes is not trivial in ad-hoc environments. The assignment task often requires infrastructure support such as a central DHCP server, which directly conflicts against the philosophy of ad-hoc networks that operate in a self-organized manner without any infrastructure. Second, it is not easy to discover the IP address of the publisher of a specific content in an ad-hoc network. Nodes join and leave the network frequently, and any node can become a new publisher of the content. Thus, the content of interest cannot be consistently bound to a unique IP address. Last, the VANET protocol simply performs IP-based end-to-end communications. During routing procedure, a router simply relays and then deletes content. Although the content is so popular that many nodes also want it, the router cannot directly send it to them because the router does not save it.

## 3 NEW MODELS FOR VANET APPLICATION AND NETWORKING

Our observation of the existing VANET is summarized as the VANET applications evolve from simple data consumers to ones that enable local collaborations with ample contents for richer user experience (UX). But,

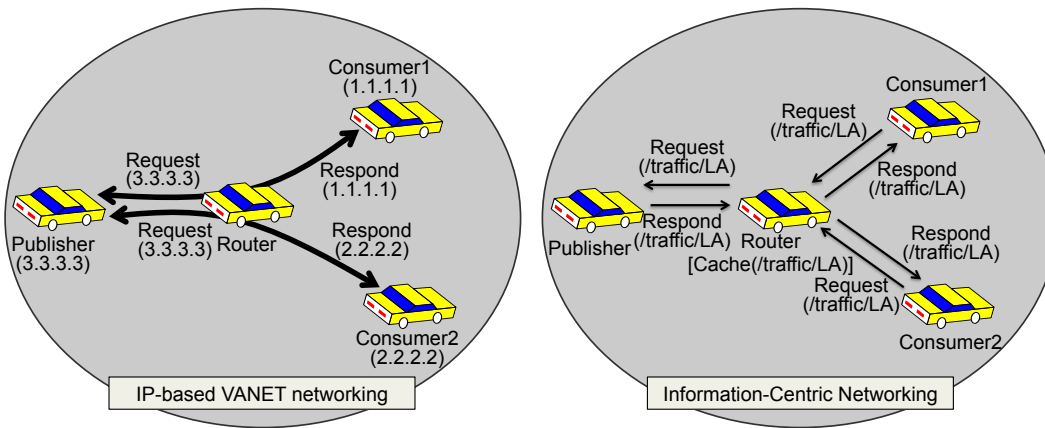


Fig. 1. The existing VANET networking vs. Information Centric Networking

the underlying networking does not seem to efficiently support the core functions that the evolving applications demand. This section introduces recent research efforts that address the issues under two categories of computing and networking.

### 3.1 Vehicular Cloud Computing

Vehicles and sensors within a local area generate vehicle contents. These contents are stored and searched in the vicinity; and processed and consumed within their lifetime period by neighbors. Recently, Gerla [5] introduced a new computing model, Vehicular Cloud Computing (VCC), to account for these characteristics. VCC is a variant of Mobile Cloud Computing (MCC) [6], that begins from a conventional cloud computing model. To mobile nodes with limited resources, the Internet cloud offers network access both for using unlimited computing resources on the Internet and for storing/downloading contents to/from the Internet. However, it is too costly to upload every content to the Internet cloud, and too time-consuming to search and download interesting contents from the Internet cloud. Besides, most of the contents picked up by vehicles have local relevance only and could be best stored locally.

In VCC, most of queries from drivers are about the world surrounding us (i.e., local relevance), and vehicles are the best probes of this environment. VCC resolves the queries, using a self-organized model of the local environment. That is, vehicles effectively form a cloud within which services are produced, maintained, and consumed. To realize the model, VCC leverages the increasing processing and storage capacity of vehicles: it constructs a cloud by using the collection of vehicles' computing resources, which primarily aim at extending the capability of interactions amongst vehicles.

### 3.2 Information Centric Networking

Information Centric Networking (ICN) is initially conceptualized as a general form of communication architecture to achieve efficient content distribution on the Internet. ICN focuses on *what (content)* instead of *where (host)*. This is to fulfill the primary demands that consumers are only interested in content itself, not its provenance, and publishers strive to efficiently distribute contents to consumers. To this end, ICN uses node or data attributes (e.g., content name, geo-location, or context) for routing rather than a specific node address (i.e., IP address). This decouples the content from the publishers. In this sense, content-based routing (see the right-hand side in Fig. 1), geo-routing and context-based routing can be classified into types of ICN. Some of the recently proposed architecture for ICN in the Internet context [7] include DONA (Data-Oriented Network Architecture), NDN (Named Data Networking), PSIRP (Publish-Subscribe Internet Routing Paradigm), and NetInf (Network of Information).

Of these architectures, NDN [8] has been recently extended to VANET [9], [10]. NDN has two types of packets: Interest from consumers and Data (i.e., content) from publishers. Content name in these packets is used for routing. A consumer requests content by broadcasting an Interest with its name toward potential publishers. When a publisher receives the Interest and has data matching the Interest, it replies with the data back to consumer using path of the Interest in reverse. NDN allows routers on the path to cache the content so that they can reply the cached content to consumers once they receive the matching Interest. This way, NDN achieves an effective content distribution that VCC critically requires to support its content oriented applications.

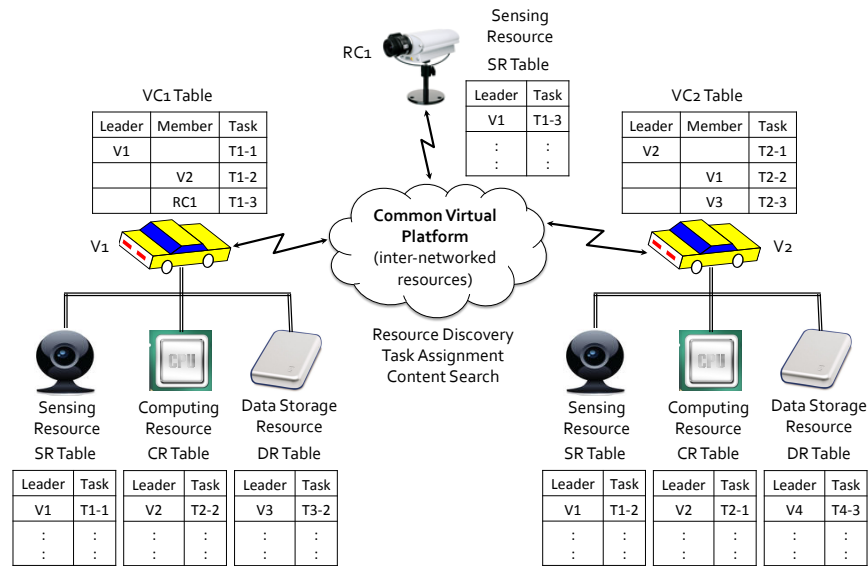


Fig. 2. Cloud resources - data storage, sensors, and computing - are shared to create a common virtual platform.

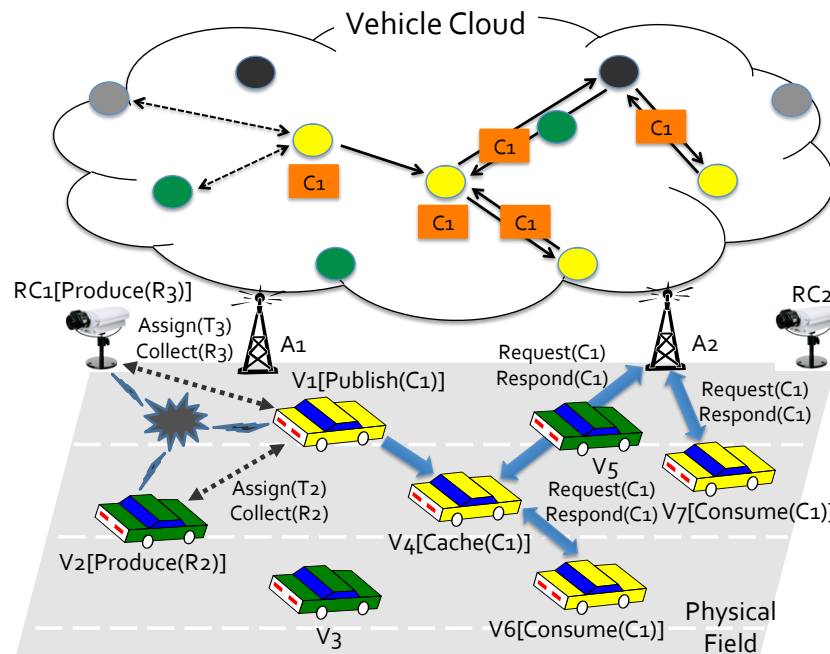


Fig. 3. Resources in VCN are inter-networked in a purely decentralized manner. We borrow the V2I communication architecture from VANET.

## 4 VEHICULAR CLOUD NETWORKING

Computing and networking models are two integral parts to support emerging VANET applications and services, which are efficiently supported by VCC and ICN, respectively. In this sense, integrating VCC and ICN will be an ideal choice, and we name the concept *Vehicle Cloud Networking (VCN)*. This section discusses VCN fundamentals, system operations, and a service scenario.

### 4.1 VCN Fundamentals

The eventual goal of VCN is to create a vehicle cloud and to encourage collaborations amongst cloud members to produce advanced vehicular services that individual alone cannot make. Unlike the Internet cloud that is created and maintained by a cloud provider, the vehicle cloud is temporarily created by inter-connecting resources available in the vehicles and RoadSide Units (RSUs). Such networked resources operate as a common virtual platform on

which the efficiency of collaboration is maximized. VCC and ICN together contribute to creating the cloud and to running the virtual platform efficiently.

Resources in the vehicle cloud are distinguished from the ones in the conventional cloud. Each vehicle has three categories of resources - data storage, sensors, and computing as shown in Fig. 2. The data storage stores vehicle contents generated from applications and sensors as well as traditional multimedia files. It supports data sharing between cloud participants by accepting an external search query and by replying with matched contents. The sensor is able to self-actuate as well as to detect events in a physical world. With technological advancement, each sensor is directly connected so that external systems can read the sensor data and/or control the sensor. The computing resource is similar to that of the Internet cloud except that its capacity is limited because it is a collection of mobile resources.

In VCN, the resources are inter-networked via purely peer-to-peer connections. That is, each vehicle negotiates the level of resource sharing directly with each other. For efficiency, one vehicle in a cloud can be elected as a broker based on some selected metrics (e.g., connectivity to vehicles). Then, it mediates the process of resource sharing as well as other cloud operations. A RSU, joining the cloud as a stationary member in Fig. 3, can be a good candidate for the negotiator role. We also envision the deployment of resource-constrained RSUs such as cameras. They may not have enough storage and computing power, but still have reliable connections to vehicles. If this is the case, they can store and manage data indexes for effective content discovery.

## 4.2 VCN - System Operations

Given the collection of resources from vehicles and RSUs and their potential interconnections, we illustrate how the VCN system operates to establish a virtual computing platform and to enable cloud type collaboration in it.

- 1) *Cloud resource discovery.* When a vehicle runs an application, it becomes a cloud leader and recruits members (i.e., vehicles and RSUs in the vicinity) that can provide their resources to construct a vehicle cloud for the application. The types and search range of necessary resources depend on applications. For instance, the search range can be a predefined distance, a road section, or an intersection. Once determining the set of necessary resource types, the cloud leader broadcasts a resource request message, RREQ, to nodes within the search range. Nodes, willing to share their resources, send a resource reply message, RREP, back to the leader with information of their resource capabilities.
- 2) *Cloud formation.* Upon receiving RREPs, the cloud leader selects nodes (cloud members) with which it organizes a cloud. The selection mechanism may pursue minimizing total resource usage in the cloud, while ensuring the correct operation of the application.
- 3) *Task assignment and result collection.* The cloud leader splits the application into several tasks and distributes them to cloud members, considering the availability and accessibility of their resources. The leader also maintains a cloud table with members' ID and their task assignment information. The cloud members return the results back to the leader after completing their tasks.
- 4) *Content publishing and sharing.* Once collecting all the task results from the members, the leader processes them to obtain output content and saves/publishes it. In the case that the task results cannot be processed in a single node or the published contents cannot be stored in a single node, the leader organizes another vehicle cloud for the data processing or content storage. The published contents are shared over the entire networks. That is, any vehicle can send a content request to search them. Upon receiving the request and having the matched contents in data storage, the cloud members including the leader can reply with the contents.
- 5) *Cloud maintenance.* In a vehicle cloud, a member sends a cloud leave message to the leader on its leaving. Then, the leader selects a replacement among nodes that sent RREPs in the resource discovery phase and have resources enough to complete the assigned task of the leaving member. The leader distributes the task to the new cloud member and updates the cloud table.
- 6) *Cloud release.* When the leader no longer uses the cloud or moves out of the cloud, it releases the resources so that other clouds can reuse them. To this end, the leader sends a cloud release message to all the members and removes the cloud member list.

## 4.3 VCN Services

While VANET just provides communications and routing services, VCN offers content routing, security privacy, monitoring, and virtualization services. So, it is easier to build applications on top of VCNs. VCN integrates networking services so it can monitor and select the most appropriate schemes and features. In case of group mobility, VCN can track group mobility as part of its monitoring service. Vehicles discover that they are traveling in a group via crowd sourcing - a VCN service. VCN can then exploit group mobility and offer it as a property to applications. For instance, Intelligent Transport can exploit group mobility in the rerouting of traffic, by rerouting groups of cars going to same destination rather than rerouting individual cars. This helps scalability.

Fig. 3 illustrates another example scenario. Suppose that a vehicle  $V_1$  (a cloud leader) organizes a vehicle cloud to complete an autonomous driving application. The application requires images data of next three road segments in order to improve the accuracy of context awareness, yet resources in  $V_1$  only cover one road segment. The cloud leader sends out a RREQ to recruit vehicles and RSUs in adequate positions that can provide sensing resources such as a camera. Upon receiving RREPs containing resource information from them, the leader selects two cloud members (say, a vehicle  $V_2$  and a road camera  $RC_1$ ) and forms a vehicle cloud. Then, the leader assigns the tasks of taking a picture of the next two block scenes and of returning the data back to itself. After collecting images data from the cloud members, the leader processes the collection to create new content that is published to the entire network.  $V_1$  consumes the content for its autonomous driving application. At the same time, the leader asks other vehicles ( $V_4$  in Fig. 3) to store and keep the content in their storage for the purpose of its potential reuse around the cloud. When some time later following vehicles  $V_6$  and  $V_7$  run their autonomous driving applications, they request the content by broadcasting an Interest message with the content name. Finding a match,  $V_4$  can transmit the matched content to  $V_6$  and  $V_7$  directly without contacting  $V_1$ . When the cloud leader decides not to use the cloud any more, it sends a cloud release message to all the cloud members,  $V_2$  and  $RC_1$ .

## 5 DESIGN PRINCIPLES OF VCN

Research on VCN is at a very preliminary stage. This section examines potential design principles for future development.

### 5.1 System Perspective

**Service instantiation.** A cloud platform represents a software system with which cloud services are provided and consumed<sup>1</sup>. In VCN, it must instantiate three unique resources, available at each vehicle, as services: data storage service, sensing service, and computing service. However, it is still unclear in what form each computing service is realized. For instance, the SaaS can provide a set of pre-programmed applications and thus services are lightweight, which fits to the vehicular cloud. However, the Software is not extensible enough to process the various vehicle contents generated from dynamically changing road environments.

**Decentralized operations with inelastic resources.** A vehicle cloud platform operates in a decentralized manner, and the cloud resources are no longer elastic and reliable. Recall that in the Internet cloud, a service provision is a server-client interaction between a cloud service provider having unlimited resource capacity and a lightweight user device. For instance, a user device in CloneCloud [11] has a clone copy in the cloud that elastically provides computing resources enough to fulfill the user's demand. In VCN, however, there is not a centralized cloud service provider, and each vehicle plays the provider and consumer roles simultaneously. Moreover, the vehicles' resources are constrained. That is, a consumer platform must seek cloud resources on demand and cope with such situations that the resources available in the cloud are less than what it requires.

**Management of service content.** Most vehicle contents have local relevance and time scope, and thus are likely to quickly obsolesce. Therefore, the cloud platform must manage the content through the whole lifecycles. VCN contents can be produced by collaboration among multiple vehicles and RSUs with their sensory data and resources. When their validity expires, they must be eliminated on the cloud to save the cloud's resources. A publisher may predict the contents' spatial and temporal scope at their publication, which helps the consumer estimate their residual values. However, an accurate prediction is often not trivial. As an alternative, collaborative sensing and following context awareness applications can determine the contents' time to live.

### 5.2 Networking Perspective

**Naming content and resource.** In VCN, data is delivered based on the content names. Up to date, two directions on the naming schemes have been proposed - flat and hierarchical. The flat naming scheme names each content uniquely, enabling accurate content routing. However, it generates a huge number of content names in the network, leading to inefficient content search. Hierarchical naming aggregates content names under prefixes and thus deliver "similar" contents effectively at the expense of naming structure complexity. Future research must examine which direction can best take into account the local relevance property of vehicle contents. Authors in [9] designed a naming structure for traffic data that names contents according to traffic type, geolocation, and timestamp. In addition to content, the sensing and computing resources in the cloud must be named appropriately with their types and capability information so as to facilitate collaborations.

**Content distribution and resource discovery.** Once named, content must be discovered, shared, and consumed by neighboring vehicles. To this end, we may consider three general approaches for content discovery. First, proactive

1. In the Internet cloud, the platform instantiates the resources in three service types: SaaS (Software as a Service), PaaS (Platform), and IaaS (Infrastructure).

pushing of content locations and creation of an index is used when the content is popular or requires low delay. In a brute force approach, the content itself can be broadcast. Second, pulling implies a search-response process, where a service consumer actively solicits the needed contents. Last, in a content resolution, a publisher sends the content to a storage place (say, determined by hash functions) from which a consumer discovers it. In [12], Oh *et al.* proposed a hybrid model for MANETs that performs pushing for spatiotemporal contents (e.g., exact time and location information) and pulling for topic-based contents (e.g., audio and video files). Peer-to-Peer (P2P) routing can be leveraged for content distribution since participating nodes in VCN are totally distributed and equally responsible. Beyond the mere offering of P2P routing, VCN is capable to match the network and routing protocols to the applications. For example, it may use scoped flooding to propagate emergency information and alarms (say, fire in the tunnel or mud slide on the road). Similarly, the three approaches can be exploited to discover necessary resources for applications. Since resources cannot be disseminated, the search is performed on meta data of the resources.

**Connectivity and resource availability.** Vehicles freely join and leave the vehicular cloud as they move. This can threaten the availability of cloud resources and result in failure to collaborate. Thus, such dynamically changing connectivity must be managed properly using a cloud join/leave mechanism so as to minimize its impact on the service provision. For instance, when an autonomous vehicle leaves the cloud, another vehicle (or a set of vehicles) is selected to substitute the services it has provided. It is beneficial to exploit the property of group mobility and the support of the Internet cloud. In particular, an Internet cloud server can manage network connectivity amongst mobile members and runs the join/leave mechanism. Or, it can bridge two vehicle clouds, otherwise separated from each other.

### 5.3 Service Perspective

**Incentive.** In VCN, a cloud member must be rewarded properly for the services they provide. Otherwise, it would not share its valuable resources. This issue stands out especially in VCN since resources and services are provided in a decentralized manner<sup>2</sup>. Say, VCN services are produced by collaborations, and also benefit the vehicles that provided the data. It is unclear who charges whom for which portion of services. Instead of explicit incentive, we may leverage the notion of “tit for tat” compensation like in bit torrent. A vehicle provides as many service “units” as it receives from others. Two critical aspects in future research are how to preclude free riders and how to quantify the values for different resources. For instance, the values of sensor data gradually decrease according to their temporal and spatial scopes, and this must be carefully taken into account.

**Security and privacy.** Since VCN encourages sharing resources, the most critical security issue would be a threat targeting the cloud platform itself. An adversary may launch a DoS attack like jamming [13]. Or, it may try to inject malware into the platform to use up the platform’s resources or to opt the platform into a botnet. An intrusion detection system or a system integrity checking can help mitigate damage. Privacy is also important in VCN because the contents that each vehicle generates tend to disclose personal information. An anonymization scheme can help resolve the issue. Future research must also address the privacy concern on service consumers. They actively seek resources and contents on the cloud. Monitoring such activities can reveal the consumers’ use pattern of specific applications. A secure search scheme must make these activities invisible using proper cryptosystems.

**Fault tolerance.** VCN services heavily rely on sensory data collected from vehicles. This implies that during service generation a number of decision making processes consume the sensor collection. However, sensors can be physically broken, malfunction, or be compromised so that they may deliver erroneous data to the processes or even fail to deliver data. Similarly, a vehicle cloud with limited resource capacity may not provide enough resources and contents to satisfy the VCN services’ requirements. The VCN service must be able to tolerate such unpredictable failures.

## 6 CONCLUSION

Vehicular communication evolves with new emerging paradigms, and this article examined the very details behind the evolution. We looked into emerging VANET applications and observed three noticeable characteristics, which cannot be supported efficiently by the existing VANET technology. To accommodate such characteristics, we introduced a new VANET network planning, consisting of two recent paradigms - Vehicular Cloud Computing and Information Centric Networking. As a computing model, VCC enables vehicles to discover and share their resources so as to create a vehicle cloud on which they collaborate to produce value-added services. ICN is leveraged, as a networking model, to disseminate cloud contents efficiently among vehicles. Then, Vehicular Cloud Networking, as a proposed future vehicular networking system, is built on top of them. To help understand VCN, we went

2. A user in the Internet cloud consumes cloud services exclusively assigned to her, and a contracted service provider (e.g., Amazon) charges the user based on predetermined tariff.

through its operations, illustrated an example scenario, and discussed design principles in three perspectives - system, networking, and service.

## REFERENCES

- [1] F. Bai and B. Krishnamachari, "Exploiting the Wisdom of the Crowd: Localized, Distributed Information-Centric VANETs," *IEEE Communications Magazine*, vol. 48(5), pp. 138 – 146, May 2010.
- [2] E.-K. Lee, S. Oh, and M. Gerla, "RFID Assisted Vehicle Positioning in VANETs," *Elsevier Pervasive and Mobile Computing*, vol. 8(2), pp. 167 – 179, April 2012.
- [3] U. Lee, E. Magistretti, B. Zhou, M. Gerla, P. Bellavista, and A. Corradi, "MobEyes: Smart Mobs for Urban Monitoring with a Vehicular Sensor Network," *IEEE Communications Magazine*, vol. 13(6), pp. 52 – 57, Oct. 2006.
- [4] S. Kumar, L. Shi, N. Ahmed, S. Gil, D. Katabi, and D. Rus, "CarSpeak: A Content-Centric Network for Autonomous Driving," in *ACM SIGCOMM*, Aug. 2012.
- [5] M. Gerla, "Vehicular Cloud Computing," in *IEEE Med-Hoc-Net*, June 2012.
- [6] N. Fernando, S. Loke, and W. Rahayu, "Mobile Cloud Computing: A Survey," *Elsevier Future Generation Computer Systems*, vol. 29(1), pp. 84 – 106, July 2013.
- [7] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A Survey of Information-Centric Networking," *IEEE Communications Magazine*, vol. 50(7), pp. 26 – 36, July 2012.
- [8] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking Named Content," in *ACM CoNEXT*, Dec. 2009.
- [9] L. Wang, R. Wakikawa, R. Kuntz, R. Vuyyuru, and L. Zhang, "Data naming in Vehicle-to-Vehicle communications," in *IEEE NOMEN*, Mar. 2012.
- [10] Y.-T. Yu, T. Punihao, M. Gerla, and M. Sanadidi, "Content Routing in the Vehicle Cloud," in *IEEE MILCOM*, Oct. 2012.
- [11] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: Elastic Execution between Mobile Device and Cloud," in *ACM EuroSys*, April 2011.
- [12] S. Oh, D. Lau, and M. Gerla, "Content Centric Networking in Tactical and Emergency MANETs," in *IFIP Wireless Days*, Oct. 2010.
- [13] E.-K. Lee, S. Oh, and M. Gerla, "Randomized Channel Hopping Scheme for Anti-Jamming Communication," in *IFIP Wireless Days*, Oct. 2010.

**Dr. Euisin Lee** (euisinlee@ucla.edu) received his B.S., M.S., and Ph.D. degrees in Computer Engineering from Chungnam National University, Korea in 2005, 2007, and 2012, respectively. He is currently working as a post-doctoral researcher in the department of Computer Science at University California, Los Angeles (UCLA). His research interests include network architecture, resource management, and content dissemination in mobile cloud computing and information centric networking, and mobility, multicast, QoS, and routing in wireless sensor networks and vehicular ad hoc networks.

**Eun-Kyu Lee** (eklee@cs.ucla.edu) is pursuing a Doctorate degree in the Computer Science Department of UCLA since 2008. He worked as a researcher at Electronic and Telecommunications Research Institute, Korea, for several years. His research interests are mainly in the Internet of Things in the areas of smart grid and connected vehicles. He has been involved in various networking and systems projects with specialization in wireless networking, security, and middleware systems.

**Dr. Soon Young Oh** (soon@utopiacompression.com) received his M.S. and Ph.D. degrees in computer science from UCLA in 2005 and 2009, respectively. Currently, he is a research and development scientist in UtopiaCompression Corporation where he is conducting research and projects on wireless ad hoc networking. His main research interests include wireless mobile network routing protocol design and development, delay-tolerant networking, QoS routing, wireless network management, satellite communications, and content centric networking.

**Dr. Mario Gerla** (gerla@cs.ucla.edu) is a Professor in the Computer Science Department of UCLA. His research interests include the design, performance evaluation, and control of distributed communication systems and networks. His current research projects cover: design and performance evaluation of protocols and control schemes for Ad Hoc wireless networks; routing, congestion control and bandwidth allocation in wide area networks; and traffic measurements and characterization (see <http://www.cs.ucla.edu/NRL> for recent publications).