Toward Cloud-Based Vehicular Networks with Efficient Resource Management

Rong Yu, Guangdong University of Technology
Yan Zhang and Stein Gjessing, University of Oslo
Wenlong Xia, Guangdong University of Technology
Kun Yang, University of Essex

Abstract

In the era of the Internet of Things, all components in intelligent transportation systems will be connected to improve transport safety, relieve traffic congestion, reduce air pollution, and enhance the comfort of driving. The vision of *all vehicles connected* poses a significant challenge to the collection and storage of large amounts of traffic-related data. In this article, we propose to integrate cloud computing into vehicular networks such that the vehicles can share computation resources, storage resources, and bandwidth resources. The proposed architecture includes a vehicular cloud, a roadside cloud, and a central cloud. Then we study cloud resource allocation and virtual machine migration for effective resource management in this cloud-based vehicular network. A game-theoretical approach is presented to optimally allocate cloud resources. Virtual machine migration due to vehicle mobility is solved based on a resource reservation scheme.

Vehicular networks are in the progress of merging with the Internet to constitute a fundamental information platform, which is an indispensable part of an intelligent transport system (ITS) [1]. This will eventually evolve into all vehicles connected in the era of the Internet of Things (IoT) [2]. By supporting traffic-related data gathering and processing, vehicular networks are able to notably improve transport safety, relieve traffic congestion, reduce air pollution, and enhance driving comfortability [3]. It has been reported that, in Western Europe, deaths due to car accidents could be reduced 25 percent by deploying safety warning systems at highway intersections [4]. Another example is that real-time traffic information could be collected and transmitted to data centers for processing, and in return, information could be broadcast to drivers for route planning. City traffic congestion would be alleviated and travel time reduced, leading to greener cities.

A variety of information technologies have been developed for intelligent vehicles, roads, and traffic infrastructures such that all vehicles are connected. Smart sensors and actuators are deployed in vehicles and roadside infrastructures for data acquisition and decisions. Advanced communication technologies are used to interconnect vehicles and roadside infrastructures, and eventually access to the Internet. For instance, dedicated short-range communications (DSRC) is specifically designed for vehicle-to-vehicle (V2V) and vehicle-to-roadside (V2R) communications. IEEE 802.11p, called Wireless Access in Vehicular Environments (WAVE) [5], is currently a popular standard for DSRC. Besides, the Long Term Evolution (LTE), LTE-Advanced, and cognitive radio (CR) [6, 7] are all fairly competitive technologies for vehicular networking [8, 9].

Despite the well developed information technologies, there is a significant challenge that hinders the rapid development of vehicular networks. Vehicles are normally constrained by resources, including computation, storage, and radio spectrum bandwidth. Due to the requirements of small-size and low-cost hardware systems, a single vehicle has limited computation and storage resources, which may result in low data processing capability. On the other hand, many emerging applications demand complex computation and large storage, including in-vehicle multimedia entertainment, vehicular social networking, and location-based services. It is becoming increasingly difficult for an individual vehicle to efficiently support these applications. A very promising solution is to share the computation and storage resources among all vehicles or physically nearby vehicles. This motivates us to study the new paradigm of cloud-based vehicular networks.

Recently, a few research projects have been reported that study the combination of cloud computing and vehicular networks. In [10], the concept of autonomous vehicular clouds (AVCs) is proposed to exploit the underutilized resources in vehicular ad hoc networks (VANETs). A platform as a service (PaaS) model is designed in [11] to support cloud services for mobile vehicles. The work in [12] proposes architectures of vehicular clouds (VCs), vehicles using clouds (VuCs), and hybrid clouds (HCs). Vehicles act as cloud service providers and clients in VCs and VuCs, respectively, and as both in HCs.

In this article, we propose a hierarchical cloud architecture for vehicular networks. Our work is different from previous research in three main aspects. First, we aim to create a pervasive cloud environment for mobile vehicles by integrating redundant physical resources in ITS infrastructures, including data centers, roadside units, and vehicles. The aggregation of these sporadic physical resources potentially compose massive and powerful cloud resources for vehicles. Second, we propose a three-layered architecture to organize the cloud resources. The layered structure allows vehicles to select their

0890-8044/13/$25.00 © 2013 IEEE

cloud services resiliently. Central clouds have sufficient cloud resources but large end-to-end communications delay. On the contrary, roadside and vehicular clouds have limited cloud resources but satisfy communications quality. Third, we emphasize the efficiency, continuity, and reliability of cloud services for mobile vehicles. As a consequence, efficient cloud resource management strategies are elaborately proposed. Countermeasures to deal with vehicle mobility are devised.

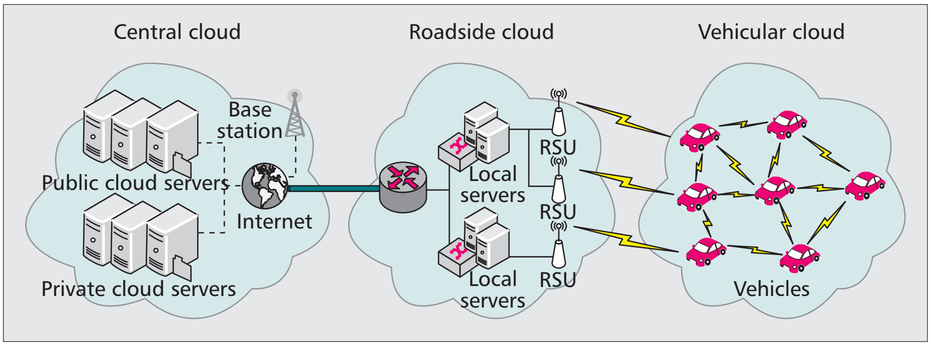

Figure 1. *Proposed cloud-based vehicular network architecture.*

The remainder of the article is organized as follows. We illustrate the proposed architecture that includes vehicular, roadside, and central clouds. Cloud deployment strategies are discussed for these three layers. We envision several promising applications for sharing different resources in cloud-based vehicular networks. We focus on cloud resource allocation problems, and a game-theoretical approach is presented to optimally allocate cloud resources. We study virtual machine migration due to vehicle mobility. Illustrative results indicate optimized resource allocation and virtual machine migration performance. Finally, our conclusion is presented.

Proposed Cloud-Based Vehicular Networks Architecture

Figure 1 shows the proposed cloud architecture for vehicular networks. It is a hierarchical architecture that consists of three interacting layers: the vehicular cloud, roadside cloud, and central cloud. Vehicles are mobile nodes that exploit cloud resources and services:

- *Vehicular cloud*: A local cloud established among a group of cooperative vehicles. An intervehicle network (i.e., a VANET) is formed by V2V communications. The vehicles in a group are viewed as mobile cloud sites and cooperatively create a vehicular cloud.
- *Roadside cloud*: A local cloud established among a set of adjacent roadside units. In a roadside cloud, there are dedicated local cloud servers attached to roadside units (RSUs). A vehicle accesses a roadside cloud by V2R communications.
- *Central cloud:* A cloud established among a group of dedicated servers in the Internet. A vehicle accesses a central cloud by V2R or cellular communications.

This architecture has several essential advantages. First, the architecture fully utilizes the physical resources in an entire network. From vehicles to roadside infrastructures and data centers, the computation and storage resources are all merged into the cloud. All clouds are accessible to all vehicles. Second, the hierarchical nature of the architecture allows vehicles using different communication technologies to access different layers of clouds accordingly. Hence, the architecture is flexible and compatible with heterogeneous wireless communication technologies such as DSRC, LTE/LTE-Advanced, and CR technologies. Third, the vehicular and roadside clouds are small-scale localized clouds. Such distributed clouds can be rapidly deployed and provide services quickly.

Vehicular Cloud

In a vehicular cloud, a group of vehicles share their computation resources, storage resources, and spectrum resources. Each vehicle can access the cloud and utilize services for its own purpose. Through cooperation in the group, the physical resources of vehicles are dynamically scheduled on demand. The overall resource utilization is significantly enhanced. Compared to an individual vehicle, a vehicular cloud has much more resources.

Due to vehicle mobility, vehicular cloud implementation is very different from a cloud in a traditional computer network. We propose two customization strategies for vehicular clouds: generalized vehicular cloud customization (GVCC) and specified vehicular cloud customization (SVCC).

In GVCC, a *cloud controller* is introduced in a vehicular cloud. A cloud controller is responsible for the creation, maintenance, and deletion of a vehicular cloud. All vehicles will virtualize their physical resources and register the virtual resources in the cloud controller. All virtual resources of the vehicular cloud are scheduled by the cloud controller. If a vehicle needs some resources of the vehicular cloud, it should apply to the cloud controller. In contrast to GVCC, SVCC has no cloud controller. A vehicle will specify some vehicles as candidate cloud sites, and directly apply for resources from these vehicles. If the application is approved, the corresponding vehicles become cloud sites, which will customize virtual machines (VMs) according to the vehicle demand.

These two strategies, GVCC and SVCC, are quite different. With respect to resource management, GVCC is similar to a conventional cloud deployment strategy in which cloud resources are scheduled by a controller. A vehicle is not aware of the cloud sites where the VMs are built up. The cloud controller should maintain the cloud resources. During a cloud service, if a cloud site is not available due to vehicle mobility, the controller should schedule a new site to replace it. In SVCC, since there is no cloud controller, a vehicle has to select other vehicles as cloud sites and maintain the cloud resources itself. In terms of resource utilization, GVCC is able to globally schedule and allocate all resources of a vehicular cloud. GVCC has higher resource utilization than SVCC. However, the operation of the cloud controller will need extra computation. Therefore, SVCC may be more efficient than GVCC in terms of lower system overhead.

Roadside Cloud

A roadside cloud is composed of two main parts: dedicated local servers and RSUs. The dedicated local servers virtualize physical resources and act as a potential cloud site. RSUs provide radio interfaces for vehicles to access the cloud. A roadside cloud is accessible only to nearby vehicles (i.e., those located within the radio coverage area of the cloud site's RSU). This fact helps us recall the concept of a *cloudlet*. A cloudlet is a trusted resource-rich computer or cluster of computers connected to the Internet and available for use by nearby mobile devices [13]. In this article, we propose the concept of a roadside cloudlet. A roadside cloudlet

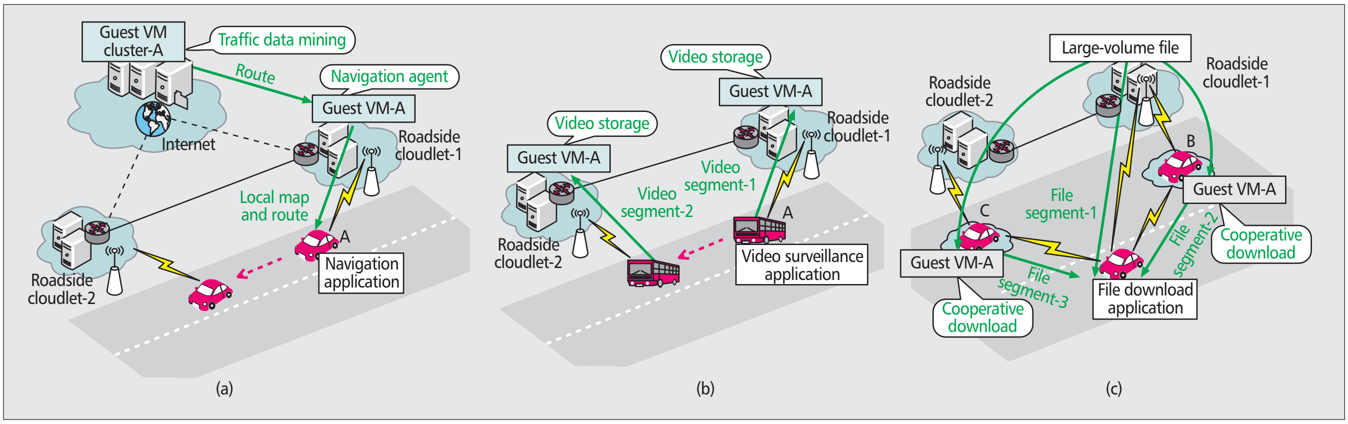

Figure 2. *Applications of cloud-based vehicular network: a) traffic data mining for real-time navigation; b) distributed storage in video surveillance; c) cooperative download of a large file.*

refers to a small-scale roadside cloud site that offers cloud services to bypassing vehicles. A vehicle can select a nearby roadside cloudlet and customize a transient cloud for use. Here, we call the customized cloud a *transient cloud* because the cloud can only serve the vehicle for a while. After the vehicle moves out of the radio range of the current serving RSU, the cloud will be deleted, and the vehicle will customize a new cloud from the next roadside cloudlet in its moving direction.

When a vehicle customizes a transient cloud from a roadside cloudlet, it is offered by virtual resources in terms of a VM. This VM consists of two interacting components: the VM-base in the roadside cloudlet and the VM-overlay in the vehicle. A VM-base is a resource template recording the basic structure of a VM, while a VM-overlay mainly contains the specific resource requirements of the customized VM. Before a cloud service starts, the vehicle will send the VM-overlay to the roadside cloudlet. After combining the VM-overlay with the VM-base, the roadside cloudlet completes the customization of a dedicated VM. During a cloud service, as the vehicle moves along the roadside, it will switch between different RSUs. For the continuity of cloud service, the customized VM should be synchronously transferred between the respective roadside cloudlets. This process is referred to as VM migration. VM migration scenarios will be further elaborated.

Central Cloud

Compared to a vehicular cloud and a roadside cloud, a central cloud has much more resources. The central cloud can be driven by either dedicated servers in vehicular networks data center or servers in the Internet. A central cloud is mainly used for complicated computation, massive data storage, and global decisions. There are already mature open source or commercial software platforms that could be employed for the deployment of a central cloud. Openstack is an open source cloud platform using the infrastructure as a service (IaaS) model. Other potential commercial platforms are Amazon Web Services, Microsoft Azure, and Google App Engine.

Promising Applications of Cloud-Based Vehicular Networks

With powerful cloud computing, cloud-based vehicular networks can support many unprecedented applications. In this section, we illustrate potential applications and explain the exploitation of a vehicular cloud, a roadside cloud, and a central cloud to facilitate new applications.

Real-Time Navigation with Computation Resource Sharing

In a real-time navigation application, the computation resources in the central cloud are utilized for traffic data mining. Vehicles may offer services that use resources beyond their own computing ability. Different from traditional navigation, which can only provide static geographic maps, real-time navigation is able to offer dynamic three-dimensional maps and adaptively optimize routes based on traffic data mining.

In Fig. 2a, vehicle A is using real-time navigation during its travel. It first requests cloud service from the central cloud and roadside cloud. Then a VM cluster and a VM are established in the central cloud and roadside cloud, respectively. VM cluster-A in the central cloud is in charge of traffic data mining and suggests several routes based on the current traffic conditions. Once a route is selected by A, real-time navigation starts. VM-A in the roadside cloud acts as an agent to push messages to vehicle A, updating the driver with traffic conditions on the road. As vehicle A moves on, VM-A will migrate to different roadside cloud sites. During the entire trip, VM cluster-A in the central cloud keeps updating the route information based on real-time traffic conditions. Once there is an unexpected event (e.g., traffic congestion), VM cluster-A will report the situation quickly and compute a new route.

Video Surveillance with Storage Resource Sharing

Video surveillance is an important application that utilizes shared storage resources. Currently, many buses in cities have installed high-definition (HD) camera systems to monitor in-bus conditions. A very large-volume hard drive is needed to store video content for a couple of days. This video storage scheme has several disadvantages. First, to save HD video content for days, the hard drive should have very large storage, which leads to high cost and big size. Second, video content can only be checked in an offline manner, so the Department of Transportation is not able to make timely and proper decisions immediately after an accident. In cloud-based vehicular networks, a new distributed storage paradigm can address this problem. The storage capability of in-bus video camera systems is significantly extended.

In Fig. 2b, bus A exploits the roadside cloud to facilitate storage of in-bus video surveillance content. Specifically, the bus applies for cloud services and receives a VM in the roadside cloud. The video content is uploaded to guest VM-A in roadside cloudlet-1 in a real-time manner. When the bus moves along the road and is located in the coverage area of roadside cloudlet-2, VM-A will be migrated accordingly. As a result, the video content is divided into several segments and separately stored in different roadside cloudlets along the

Potential applications	Relevant cloud assistance			Resource sharing		
	Central cloud	Roadside cloud	Vehicular cloud	Computation	Storage	Bandwidth
Real-time traffic condition analysis and broadcast	✓	✓		✓	✓	
Real-time car navigation	✓	✓		✓		
Video surveillance		✓			✓	
LBS commercial advertisement		✓	✓		✓	✓
Mobile social networking	✓		✓	✓		✓
In-vehicle multimedia entertainment		✓	✓	✓	✓	✓
Intervehicle video and audio communications			✓			✓
Remote vehicle diagnosis	✓	✓		✓		

Table 1. *Applications of cloud-based vehicular networks.*

road. The video segments in the roadside cloudlets will be transmitted to a data center on demand. When an accident is reported, the Department of Transportation can request roadside cloudlets to send back video to the data center.

Cooperative Download/Upload with Bandwidth Sharing

Cooperative downloading and uploading services are interesting applications that share bandwidth resources. Many new applications involve large-volume data upload or download. Typical examples include in-vehicle multimedia entertainment, location-based rich media advertisements, and big-size email services. Due to limited wireless bandwidth and vehicle movement, it is very difficult to download an entire large file from a specific RSU. While the vehicle drives by, there is not enough time to complete the download of large amounts of data. Here, we illustrate that usage of a vehicular cloud will make such applications feasible.

In Fig. 2c, vehicle A is going to download a large file from a roadside infrastructure. The cooperative downloading has two phases. In the first phase, vehicle A observes neighboring vehicles B and C, and then sets up a vehicular cloud for cooperative downloading. Then a guest VM is constructed in both B and C. File downloading is carried out by the vehicular cloud that consists of vehicle A and the two VMs on B and C. Since the file is downloaded by three vehicles in parallel, the total transmission rate becomes much faster. In this way, vehicle A has a high possibility to finish downloading before moving out of the range of the roadside infrastructure. In the second phase, the VMs in B and C further cooperatively transmit two separate segments of the file to A. Since only V2V communications is involved, the second phase can be performed without the roadside infrastructure. After that, A will reassemble the file segments into an entire file.

Table 1 summarizes potential applications in cloud-based vehicular networks. We also show the relevant cloud resource sharing in each application.

A Game-Theoretical Approach to Resource Allocation

Vehicle and roadside clouds are both resource-intensive components. Resource management is very crucial for these two types of clouds. Resources in vehicle and roadside clouds are represented in the form of VMs. In the literature, VM research has been studied mainly in computer networks. In a recent study [14], VM migration is considered for dynamic resource management in cloud environments. In [15], VM replication and scheduling are intelligently combined for VM migration across wide area network environments. However, there are few studies on VM resource management in mobile cloud environments. In [13], the cloudlet is discussed and customized in the mobile computing environments.

In this section, we mainly focus on VM resource allocation in vehicular and roadside clouds. In a roadside cloud, there are multiple VMs since a cloud site provides services to several vehicles simultaneously. In this case, the resources in a cloud site should be appropriately allocated. VM resource allocation should consider several aspects:

- Efficiency: VM resource allocation strategy should be efficient such that the limited resources are fully utilized.
- Quality of service (QoS): The resources allocated to a specific VM should be sufficient for the accomplishment of the VM's tasks to achieve its QoS requirements.
- Fairness: VMs with the same workload should be offered statistically equal resources. Here, we formulate the competition among VMs for cloud resources as a non-cooperative game.

Game-Theoretical Model

Consider a roadside cloudlet with N VMs (i.e., the *players* of the game). The VMs will apply to the cloud site and compete for resources. These VMs are selfish in the sense that they aim to obtain as many resources as possible for their own usage. The cloud will allocate the total available resources to the VMs in proportion to the number of requested resources.

Let C and M represent the total available computation and storage resources of the cloud site, respectively. Let c_i ($0 < c_i \le C$) and m_i ($0 < m_i \le M$) denote the number of requested resources from the ith VM in computation and storage, respectively. Define $c_{-i} = \Sigma_{n=1,n\neq i}^{N} c_n$ and $m_{-i} = \Sigma_{n=1,n\neq i}^{N} m_n$. The ith VM will be allocated computation and storage resources

$$\frac{c_i C}{c_i + c_{-i}} \text{ and } \frac{m_i M}{m_i + m_{-i}},$$

respectively. For the sake of fairness, the cloud site sets up two virtual resource counters (VRCs) for each VM. These two VRCs are used to record the accumulative number of applied resources, one for computation and the other for stor-

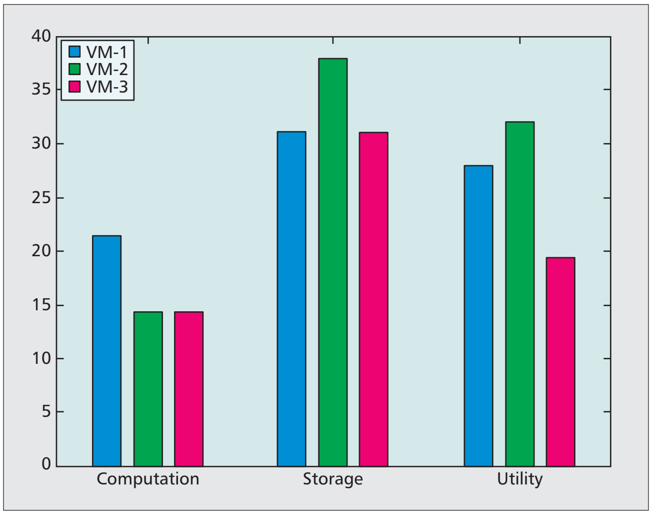

Figure 3. *Resource allocation result in roadside cloud.*

age. When a VRC reaches its maximal value, the VM is not allowed to apply for that type of resource. By using VRCs, the total amount of allocated resources are equal for all VMs from a long-term perspective. Let α_i and β_i ($\alpha_i > 0$, $\beta_i > 0$), respectively, denote the predefined resource weights that indicate the importance of computation and storage resources in the workloads of the *i*th VM, and let λ_i and γ_i ($\lambda_i > 0$, $\gamma_i > 0$) denote the pricing factors associated with applied computation and storage resources, respectively, of the *i*th VM. The utility function, or *payoff*, for the *i*th VM is given by

$$U(c_i, m_i) = \frac{\alpha_i c_i C}{c_i + c_{-i}} + \frac{\beta_i m_i M}{m_i + m_{-i}} - (\lambda_i c_i + \gamma_i m_i). \tag{1}$$

The proposed game-theoretical model is specially devised for mobile cloud applications in vehicular networks. In particular, the resource weights α_i and β_i in the utility function make the game model adaptable to resources preference in different applications. The pricing factors λ_i and γ_i are set to prevent resource waste imposed by excessive competition, and thus potentially enhance resource utilization. These key parameters α_i, β_i, λ_i, and γ_i are elaborately selected regarding the mobile environment of the cloud-assisted vehicles. For example, vehicles may have different quality of radio links to the cloud site. Their VMs should be provided with different α_i, β_i, λ_i, and γ_i according to the link quality. Typically, in a mobile multimedia application where scalable video coding (SVC) technique is involved, the VM is responsible for adaptive video decoding in the cloud site. The required VM resource mostly depends on the link quality. Because the link rate restricts the affordable quality of a video stream, it consequently determines the amount of VM resources for video processing.

Nash Equilibrium

In a non-cooperative game, a Nash equilibrium is a balanced state with a strategy profile from which no game player has any incentive to deviate. In the proposed VM resource allocation game, by computing the second order derivative of $U(c_i, m_i)$ with respect to c_i and m_i, respectively, we get

$$\frac{\partial^2 U}{\partial c_i^{\,2}} = -\frac{2\alpha_i c_{-i} C}{(c_i + c_{-i})^3} < 0 \text{ and } \frac{\partial^2 U}{\partial m_i^{\,2}} = -\frac{2\beta_i m_{-i} M}{(m_i + m_{-i})^3} < 0.$$

This means that $U(c_i, m_i)$ is a concave function with respect to c_i or m_i. Therefore, the existence of a Nash equilibrium is proven in the VM resource allocation game model [16]. Given the other VMs' applications, say, c_{-i} and m_{-i}, we define $(c_i^*, m_i^*) \in \arg\max U(c_i, m_i)$ as the *best response*, or the *optimal strategy* of the *i*th VM in each iteration. We have

$$\begin{cases} c_i^* = \min\left(C, \sqrt{\dfrac{\alpha_i c_{-i} C}{\lambda_i}} - c_{-i}\right), \\ m_i^* = \min\left(M, \sqrt{\dfrac{\beta_i m_{-i} M}{\lambda_i}} - m_{-i}\right). \end{cases} \tag{2}$$

To prove the uniqueness of Nash equilibrium in the VM resource allocation game, we can validate that the best response function is a standard function, which has three features: positivity, monotonicity and scalability [16]. Following Eq. 2, it is easy to prove that the sufficient conditions for the uniqueness of Nash equilibrium are $\forall i$, $\alpha_i \geq 4(N-1)\lambda_i$ and $\beta_i \geq 4(N-1)\gamma_i$.

Figure 3 shows a numerical example of resource allocation in our game-theoretical model. In the example, there are three VMs in a roadside cloud. The total available resources in computation and storage are set to 50 and 100 units, respectively. The VMs have different resource demands. VM-1 has the highest demand on computation, while VM-2 has the highest demand on storage. We randomly select initial values of the resource applications for the three VMs, say, $c_{\{1,2,3\}} = \{10, 5, 5\}$ and $m_{\{1,2,3\}} = \{5, 15, 10\}$. In the simulation, it is observed that the game iteration converges fast. The game reaches its Nash equilibrium after nearly 10 rounds of iterations. Results indicate that the resources are appropriately allocated based on demand. In particular, VM-1, VM-2, and VM-3 are allocated 21.4, 14.3, and 14.3 units of computation, respectively. VM-1, VM-2, and VM-3 are allocated 31.1, 37.8, and 31.1 units of storage, respectively.

A Resource Reservation Scheme for Virtual Machine Migration

Virtual Machine Migration Scenarios

VM migration refers to the process through which an operating VM is transferred along with its applications across different physical machines. In VM migration, a VM image has to be copied from the source to destination roadside cloudlets. Different from traditional VM migration, VM migration in cloud-based vehicular networks has several different scenarios due to different deployments of roadside clouds and vehicle movements:

- *Inter-cloudlet case*: In Fig. 4a, when vehicle A moves from the coverage area of RSU-1 to that of RSU-2, a VM migration is needed. Since RSU-1 and RSU-2 connect to different cloudlets, guest VM-A should be transferred from roadside cloudlet-1 to roadside cloudlet-2. After that, A will access cloudlet-2 via RSU-2 to resume its service.
- *Intra-cloudlet case*: In Fig. 4b, vehicle A moves from the coverage area of RSU-1 to that of RSU-2. Since these two RSUs connect to the same roadside cloudlet, there is no need for VM migration. However, radio handoff from RSU-1 to RSU-2 may still take a short period. During handoff, the interaction between vehicle A and guest VM-A may be temporally suspended.
- *Across roadside-vehicular cloud case*: In Fig. 4c, vehicle A moves from the coverage area of RSU-2 to that of RSU-1. Before A's movement, nodes A, C, and D are connected in an ad hoc manner. Vehicle C access the roadside cloud through vehicle A. The movement of A will cause the dis-

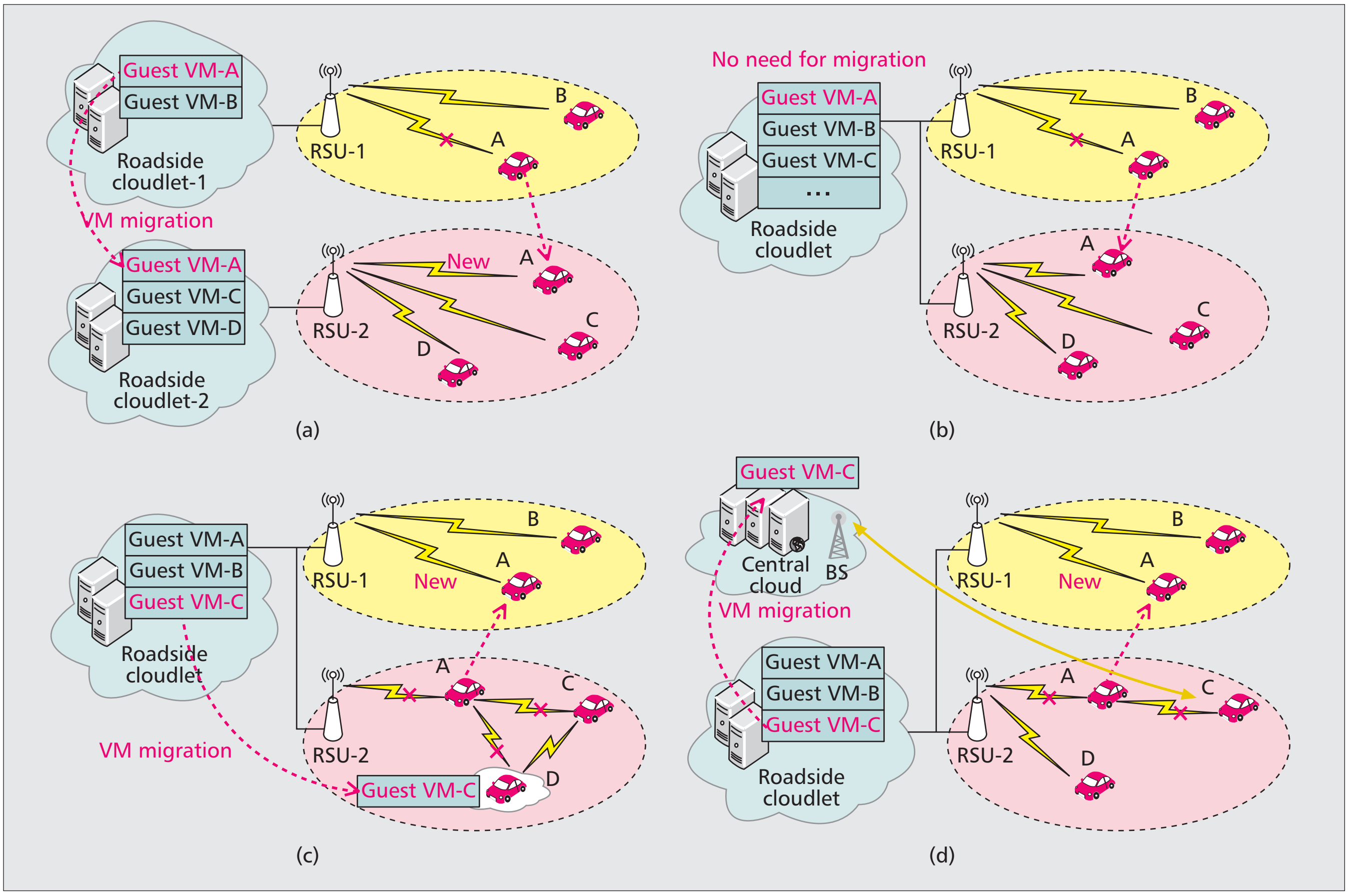

Figure 4. *Virtual machine migration scenarios: a) inter-cloudlet; b) intra-cloudlet; c) across roadside-vehicular cloud; d) across roadside-central cloud.*

connection of C from the roadside cloud. In this case, guest VM-C will be transferred from the roadside cloud to the vehicle cloud in D. Then vehicle C can continue its service through D.

- *Across roadside-central cloud case*: The scenario in Fig. 4d is similar to that in Fig. 4c, except that there is no direct link between vehicles C and D. In this case, guest VM-C has to be migrated from the roadside cloud to the central cloud. After that, C will access the central cloud to resume its service by long-distance communications (e.g., third-/fourth generation, 3G/4G, cellular).

A Resource Reservation Scheme

The discussion on VM migration indicates that the VM migration process involves resource re-allocation in the roadside cloud. If the resources of the destination cloud have been intensively occupied, after a VM migration and resource re-allocation, some VMs may not have sufficient resources and may not even resume their services. In order to avoid resource over-commitment, the target cloud site has to deny the VM migration in order to maintain the services of the existing VMs. In this case, the cloud service of a vehicle with VM migration is said to be dropped. To reduce service dropping, we propose a resource reservation scheme. In this scheme, a small portion of the cloud site resources are reserved only for migrated VMs, but not for local VMs. When there are dedicated resources for VM migration, the dropping rate of cloud services is significantly decreased.

In the proposed resource reservation scheme, resources are divided into two categories: reserved resources and common resources. Let C_r and M_r denote the reserved resources, and $C_c = C - C_r$ and M_c = M – M_r the common resources in computation and storage, respectively. In VM migration, a VM arrival refers to the event in which a VM is created for either a new local or migrated VM. A VM departure refers to a request for a VM deletion, either for an ending of VM service or VM migration to another cloud site. The resource reservation scheme operates as follows:

- *Local VM arrival*: When there is a request for creating a new local VM, resource allocation will be carried out (e.g., using the proposed game-theoretic allocation scheme). Since some of the resources are reserved, the local VMs can only share the common resources. If the resource allocation result satisfies all existing VMs, the new local VM is admitted; otherwise, it is blocked.
- *Local VM departure*: Resource allocation is also performed when the service of a local VM ends or migrates to another cloud site.
- *Migrated VM arrival*: Upon a request for a VM migration, the target cloud site will re-allocate resources. In this case, the reserved resources will be also taken into account. Specifically, the existing local VMs and migrated VM will share all available resources. After re-allocation, if all the VMs (including the migrated VM) resource requests are satisfied, VM migration is approved; otherwise, the VM migration request is rejected.
- *Migrated VM departure*: Resource allocation is also performed when the service of a migrated VM ends, or it migrates to another cloud site. It is noticeable that if there is no migrated VMs in a cloud site, the resource allocation can only use common resources. The reserved resources will be conserved for further usage upon another VM migration.

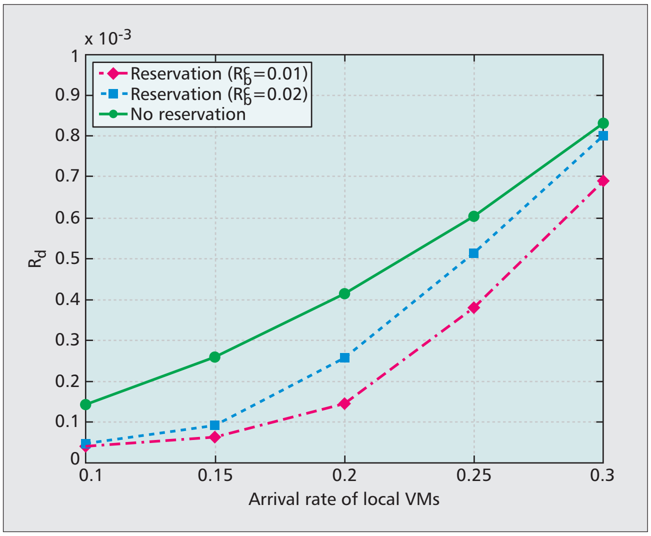

Figure 5. *Dropping rate in terms of local VM arrival rate.*

Optimal Resource Reservation

We consider K classes of VMs. Let c_k and m_k represent the amount of required resources by the kth class of VMs in computation and storage, respectively. Let n_k^l and n_k^g denote the number of local and migrated VMs of class k, respectively. Suppose that the arrivals and departures of both local and migrated VMs follow a Poisson process model. The system state transition may be formulated as continuous-time Markov process. Let $\mathbf{n}_l = (n_1^l, \cdots, n_k^l, \cdots, n_K^l)$ and $\mathbf{n}_g = (n_l^g, \cdots, n_k^g, \cdots, n_K^g)$. We represent the system state by $\mathbf{s} = (\mathbf{n}_l, \mathbf{n}_g)$ and the state space by $\mathcal{S}$. Let $\pi_{\mathbf{s}}$ denote the steady state probability of state $\mathbf{s}$. Given the arrival and departure rates of new and migrated VMs, the steady state probability matrix $\Pi = \{\pi_{\mathbf{s}} | \mathbf{s} \in \mathcal{S}\}$ will be derived by a $2K$-dimension Markov chain model.

Let R_b and R_d denote the blocking rate and dropping rate, respectively. Then a new local VM is blocked if the total amount of required resources of the local VMs (including the new one) exceeds that of the common resources, that is, $\Sigma_{k=1}^K n_k^l c_k > C_c$ or $\Sigma_{k=1}^K n_k^l m_k > M_c$. A migrated VM is dropped if the total amount of required resources of all VMs (including the migrated one) is more than that of all resources, that is, $\Sigma_{k=1}^K (n_k^l + n_k^g) c_k > C$ or $\Sigma_{k=1}^K (n_k^l + n_k^g) m_k > M$. Let λ_k^l, μ_k^l, λ_k^g, and μ_k^g denote the arrival and departure rates of local and migrated VMs; then $\mathcal{S}_b$ and $\mathcal{S}_d$ the sets of states that encounter blocking and dropping, respectively. We can derive $R_b(C_r, M_r) = \Sigma_{\mathbf{s} \in \mathcal{S}_b} \Sigma_k \pi_{\mathbf{s}} \lambda_k^l$, and $R_d(C_r, M_r) = \Sigma_{\mathbf{s} \in \mathcal{S}_d} \Sigma_k \pi_{\mathbf{s}} \lambda_k^g$. Let R_b^c denote the constraint of the blocking rate. The optimal number of reserved resources is derived by solving the following optimization problem:

$$\begin{aligned} &\min \ R_d(C_r, M_r), \\ &\text{s.t.} \ R_b(C_r, M_r) \leq R_b^c. \end{aligned} \qquad (3)$$

Figure 5 shows a performance comparison with and without resource reservation. The total resources of the roadside clouds are 50 and 100 units in computation and storage, respectively. Two classes of VMs are considered. VMs of class 1 are mainly for computation-type applications, which needs 20 units in computational resources and 15 units in storage resources. VMs of class 2 are mainly for storage-type applications, which need 10 units in computational resources and 40 units in storage resources. The two classes of VMs are assumed to have identical arrival and departure rates. We set the range of local VM arrival rate from 0.1 to 0.3, the local VM departure rate by 2.0, the arrival and departure rates of migrated VM by 0.05 and 0.1, respectively. The simulation results show that the dropping rate of migrated VMs is significantly reduced with resource reservation, which demonstrates the efficiency of our proposed mechanism.

Conclusions

In this article, we first discuss the opportunities and challenges in exploiting cloud computing in vehicular networks. Then we present a hierarchical architecture for cloud-based vehicular networks that facilitates sharing of computational resources, storage resources, and bandwidth resources among vehicles. Furthermore, we focus on efficient resource management in the proposed architecture. The resource competition among virtual machines is formulated and solved in a game-theoretical framework. Virtual resource migration due to vehicle mobility is addressed based on a resource reservation scheme. Finally, illustrative results indicate a significant reduction of the service dropping rate during virtual machine migration.

Acknowledgment

This research is partially supported by program of NSFC (grant no. 61370159 U1035001, U1201253, 61203117), the Opening Project of Key Lab. of Cognitive Radio and Information Processing (GUET), Ministry of Education (grant no. 2011KF06), the project 217006 funded by the Research Council of Norway, the European Commission FP7 Project EVANS (grant no. 2010-269323), and the European Commission COST Action IC0902, IC0905 and IC1004.

References

[1] M. Miche, and T. M. Bohnert, "The Internet of Vehicles or the Second Generation of Telematic Services," *ERCIM News*, vol. 77, 2009, pp. 43–45.

[2] ITU Strategy and Policy Unit (SPU), ITU Internet Reports 2005: The Internet of Things, Geneva, 2005.

[3] J. Chen *et al.*, "Measuring the Performance of Movement — Assisted Certificate Revocation List Distribution in VANET," *Wireless Commun. and Mobile Computing*, vol. 11, no. 7, 2011, pp. 888–98.

[4] WHO, World Health Report 2002, "Reducing Risks, Promoting Healthy Life," Geneva, Switzerland, 2002.

[5] R. A. Uzcategui and G. Acosta-Marum, "WAVE: A Tutorial," *IEEE Commun. Mag.*, vol. 47, no. 5, May 2009, pp. 126–33.

[6] R. Yu *et al.*, "Secondary Users Cooperation in Cognitive Radio Networks: Balancing Sensing Accuracy and Efficiency," *IEEE Wireless Commun.*, vol. 19, no. 2, Apr. 2012, pp. 2–9.

[7] S. Xie *et al.*, "A Parallel Cooperative Spectrum Sensing in Cognitive Radio Networks," *IEEE Trans. Vehic. Tech.*, vol. 59, no. 8, 2010, pp. 4079–92.

[8] T. Wang, L. Song, and Z. Han, "Coalitional Graph Games for Popular Content Distribution in Cognitive Radio VANETs," to appear, *IEEE Trans. Vehic. Tech.*

[9] T. Wang *et al.*, "Popular Content Distribution in CR-VANETs with Joint Spectrum Sensing and Channel Access," to appear, *IEEE JSAC*.

[10] S. Olariu, M. Eltoweissy, and M. Younis, "Towards Autonomous Vehicular Clouds," I*CST Trans. Mobile Commun. and Applications*, vol. 11, no. 7-9, pp. 1-11, 2011.

[11] D. Bernstein, N. Vidovic, and S. Modi, "A Cloud PAAS for High Scale, Function, and Velocity Mobile Applications — With Reference Application as the Fully Connected Car," *Proc. 5th Int'l. Conf. Systems and Networks Communications (ICSNC)*, 2010, pp. 117–23.

[12] R. Hussain *et al.*, "Rethinking Vehicular Communications: Merging VANET with Cloud Computing," *Proc. IEEE 4th Int'l. Conf. Cloud Computing Technology and Science*, 2012, pp. 606–09.

[13] M. Satyanarayanan *et al.*, "The Case for VM-based Cloudlets in Mobile Computing," IEEE Pervasive Computing, vol. 8, no. 4, 2009.

[14] M. Mishra *et al.*, "Dynamic Resource Management Using Virtual Machine Migrations," *IEEE Commun. Mag.*, vol. 50, no. 9, 2012, pp. 34–40.

[15] S. K. Bose *et al.*, "Cloudspider: Combining Replication with Scheduling for Optimizing Live Migration of Virtual Machines Across Wide Area Networks," *Proc. Int'l. Symp. Cluster, Cloud and Grid Computing*, 2011, pp. 13–22.

[16] D. Fudenberg, *J. Tirole. Game Theory*, MIT Press, Cambridge, MA, 1991.

Biographies

RONG YU [S'05, M'08] (yurong@ieee.org) received his Ph.D. from Tsinghua University, China, in 2007. After that, he worked in the School of Electronic and Information Engineering of South China University of Technology (SCUT). In 2010, he joined the Institute of Intelligent Information Processing at Guangdong University of Technology (GDUT), where he is now an associate professor. His research interest mainly focuses on wireless communications and networking, including cognitive radio, wireless sensor networks, and home networking. He is the co-inventor of over 10 patents and author or co-author of over 50 international journal and conference papers. He is currently serving as the Deputy Secretary General of the Internet of Things (IoT) Industry Alliance, Guangdong, China, and the deputy head of the IoT Engineering Center, Guangdong, China. He is a member of the Home Networking Standard Committee in China, where he leads the standardization work of three standards.

YAN ZHANG [SM'10] (yanzhang@ieee.org) received a Ph.D. degree from Nanyang Technological University, Singapore. He is working with Simula Research Laboratory, Norway, and is an adjunct associate professor at the University of Oslo, Norway. He is an Associate Editor or Guest Editor of a number of international journals. He serves as Organizing Committee Chair for many international conferences. His research interests include resource, mobility, spectrum, energy, and data management in wireless communications and networking.

STEIN GJESSING (steing@ifi.uio.no) is a professor of computer science in the Department of Informatics, University of Oslo, and an adjunct researcher at Simula Research Laboratory. He received his Ph.D. degree from the University of Oslo in 1985. He acted as head of the Department of Informatics for four years from 1987. From February 1996 to October 2001 he was the Chairman of the national research program Distributed IT-System, founded by the Research Council of Norway. He participated in three European funded projects: Macrame, Arches, and Ascissa. His current research interests are routing, transport protocols, and wireless networks, including cognitive radio and smart grid applications.

WENLONG XIA (wenlong.xia@ieee.org) received his M.S. degree in electronics engineering from PLA Information Engineering University, China in 2011. Now he is pursuing his M.S. degree in signal and information processing from GDUT, China. His research interests include vehicular wireless networks, opportunistic networks, and cloud computing.

KUN YANG [SM] (kunyang@essex.ac.uk) received his Ph.D. from the Department of Electronic and Electrical Engineering of University College London, United Kingdom. He is currently a full professor in the School of Computer Science and Electronic Engineering, University of Essex, United Kingdom, and head of the Network Convergence Laboratory in Essex. His main research interests include wireless networks/communications, fixed mobile convergence, future Internet technology, and network virtualization. He has published over 150 papers in the above research areas. He serves on Editorial Boards of both IEEE and non-IEEE journals. He is a Fellow of IET.