Peer-to-Peer Cooperative Caching for Data Dissemination in Urban Vehicular Communications

Neeraj Kumar and Jong-Hyouk Lee, *Senior Member, IEEE*

***Abstract*—Vehicular communications are becoming an emerging technology for safety control, traffic control, urban monitoring, pollution control, and many other road safety and traffic efficiency applications. All these applications generate a lot of data which should be distributed among communication parties such as vehicles and users in an efficient manner. On the other hand, the generated data cause a significant load on a network infrastructure, which aims at providing uninterrupted services to the communication parties in an urban scenario. To make a balance of load on the network for such situations in the urban scenario, frequently accessed contents should be cached at specified locations either in the vehicles or at some other sites on the infrastructure providing connectivity to the vehicles. However, due to the high mobility and sparse distribution of the vehicles on the road, sometimes, it is not feasible to place the contents on the existing infrastructure, and useful information generated from the vehicles may not be sent to its final destination. To address this issue, in this paper, we propose a new peer-to-peer (P2P) cooperative caching scheme. To minimize the load on the infrastructure, traffic information among vehicles is shared in a P2P manner using a Markov chain model with three states. The replacement of existing data to accommodate newly arrived data is achieved in a probabilistic manner. The probability is calculated using the time to stay in a waiting state and the frequency of access of a particular data item in a given time interval. The performance of the proposed scheme is evaluated in comparison to those of existing schemes with respect to the metrics such as network congestion, query delay, and hit ratio. Analysis results show that the proposed scheme has reduced the congestion and query delay by 30% with an increase in the hit ratio by 20%.**

***Index Terms*—Cooperative caching, peer-to-peer (P2P) cooperative caching (P2PCC), vehicular ad hoc network (VANET).**

I. Introduction

RECENT ADVANCES in the high-speed Internet make it feasible to end users to access any type of services on fly [1], e.g., users sitting in vehicles may control various applications such as their household equipment while on move.

All this is feasible because of an evolution of a new emerging technology: vehicular communications in which vehicles are intelligent machines performing various types of operations such as collection of data and forwarding the data with/without using an existing infrastructure.

Vehicular communications consist of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications. In V2I communications, vehicles may access various resources from nearby road side units (RSUs) which act as access routers connecting the vehicles to the underlying infrastructure. In other words, the RSUs, which are deployed alongside a road, are intermediates between the vehicles on the road and the infrastructure for bidirectional data dissemination. In V2V communications, vehicles communicate with each other directly without using the infrastructure. During the movement of the vehicles, the valuable information such as network traffic on the road, weather conditions, or safety alarms is shared among the vehicles either in a peer-to-peer (P2P) manner or with the aid of deployed RSUs.

Broadly, there are two regions of interest for data dissemination: sparse and urban. In sparse regions, the connectivity among the vehicles is less due to a minimum existence of infrastructure support. However, in urban regions, the density of the vehicles on the road is high, so large amount of data is transmitted among the vehicles and infrastructure. In such urban regions, vehicles are required to provide a number of services to the users for road safety, e.g., lane changing warning, signal violation warning, and emergency services. Not only this, traffic efficiency and comfort services such as traffic information, weather information, location of the nearest gas station/police station, and onboard Internet access are provided. All these services require efficient data dissemination among the vehicles and infrastructure so that the data can be transmitted to their destination without delay and transmission error.

Although vehicular communications have been studied for years, there are still challenges for data dissemination, which needs special attention. First, the information is shared by clusters of vehicles [2]. Thus, the information in such networks is transmitted in a store and forward manner to increase the efficiency of the data dissemination among the vehicles, but it causes a significant delay, i.e., query delay. Due to this fact, there is performance degradation with respect to query delay or network congestion during the data dissemination. It has been found in the literature that cooperative caching can be an attractive choice for providing information to vehicles [3]–[6]. In cooperative sharing, vehicles communicate with each other for data dissemination and information sharing while sharing the contents from the nearest RSUs. The data may be cached either at the nearest RSUs or in the vehicles as per requirements.

Manuscript received July 6, 2012; revised May 23, 2013; accepted September 22, 2013. Date of publication November 8, 2013; date of current version November 20, 2014. The work of J.-H. Lee was supported by a 2013 Research Grant from Sangmyung University. *(Corresponding author: J.-H. Lee.)*

N. Kumar is with the Department of Computer Science and Engineering, Thapar University, Patiala 147004, India (e-mail: neeraj.kumar@thapar.edu).

J.-H. Lee is with the Department of Computer Software Engineering, Sangmyung University, Cheonan 330-720, Korea (e-mail: jonghyouk@smu.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSYST.2013.2285611

Examples of such systems are Traffic View [7] and Self Organizing Traffic Information System (SOTIS) [8], [9]. Second, with an increase in the density of vehicles on roads, data collection and dissemination would be a challenging task to route packets in a timely manner due to the congestion in the network. The density of the vehicles varies according to the day and time as, during the early time of the starting day of a week, it is on peak as compared to the normal day time. An intelligent approach to select the best route for data delivery without delay is thus required. For instance, nodes including vehicles and RSUs should pass the information in a collaborative manner to the neighboring nodes so that communication situations and conditions are quickly adopted and treated accordingly. However, this process should not increase the load on the networks as the network, i.e., infrastructure, aims at providing uninterrupted services to all the moving vehicles. Accordingly, a development of a specific mechanism is desired that allows nodes to share the information with each other without causing a significant load on the network.

To tackle the aforementioned issues and to access the contents during the movement of the vehicles, caching can be used. In caching, the most frequent data are accessed from some intermediate sites to reduce the load on a server. As clients' requests are not processed from the server, there is less delay in accessing the service. However, maintaining cache information for providing data to the moving clients is a major challenge, since data items that are updated on the mobile clients must be updated in the cache of a nearest storage point so that, if any other clients access the same information, then the latest updated version of the data gets accessed.

Thus, keeping in view of the aforementioned challenges and constraints, in this paper, we propose a new P2P cooperative caching (P2PCC) scheme for urban vehicular communications. The traffic information among vehicles in a network is shared in a P2P manner probabilistically. Whenever two vehicles want to share the most commonly used information, they do the same using cooperative caching in a P2P manner without putting the extra load on a server. If the requested information is not found in their cache, then they will consult to the centralized control. The hit ratio in the proposed scheme is found to be higher, while the latency in accessing the data and congestion are less than those of earlier schemes of its category.

The rest of this paper is organized as follows. Section II discusses the most related works. Section III provides the description of our traffic model and problem formulation. Section IV describes the proposed P2PCC scheme. Section V provides the analytical analysis of the proposed scheme. Then, simulation results and discussion are presented in Section VI. Section VII concludes this paper with future directions.

II. Related Works

In the literature, various caching schemes have been proposed over the years with the goals of reducing the network congestion and providing uninterrupted services to end users. Biswas *et al.* [10] proposed information dissemination in self-organizing intervehicle communications. The authors have illustrated a new data dissemination scheme for data transfer among the vehicles. Yang *et al.* [11] proposed a multihop peer-communication protocol with fairness guarantee for IEEE 802.16-based vehicular networks. In the proposal, if a vehicle cannot directly send its data to the nearest RSU, it can relay its data to other vehicles in a P2P manner until the data are reached to the RSU using a multihop transmission strategy. Wischhof *et al.* [12] proposed a V2V communication model for travel and traffic information. In this proposal, the authors have used a concept of timestamp that allows vehicles on the road to find the freshness of the messages received from other vehicles. Korkmaz *et al.* [13] proposed an urban multihop broadcast scheme for vehicular ad hoc networks (VANETs). The proposed scheme removes the broadcast storm, hidden station, and reliability problems existing in most of the earlier existing schemes. Without knowledge of the topology, the proposed scheme efficiently forwards and receives the messages from the vehicles in a particular segment of the road. The proposed scheme has high packet forwarding rate and better channel utilization than the existing schemes. Shafiee and Leung [14] proposed an enhanced intersection mode data dissemination mechanism, which keeps emergency messages in an intersection long enough to ensure that the messages are forwarded to all the intersecting road segments. Naumov and Gross [15] proposed a position-based routing protocol called connectivity-aware routing, which introduces an end-to-end route discovery process. The discovered route contains a set of anchor points formed by the velocity vectors of the relaying vehicles along the path which is an integration of position and topology-based routing protocols and has provision of finding the location of the destination as well as determining the connected path between the source and the destination. Wisitpongphan *et al.* [16] proposed empirical vehicle traffic data which are used to develop a comprehensive analytical framework to study the disconnected network phenomenon and its network characteristics. In the study, depending on the sparsity of vehicles on a road, the vehicles can be disconnected from the network for a few seconds to several minutes.

Namboordiri and Gao [17] proposed a prediction-based routing protocol that determines a path between a given source vehicle and a destination vehicle with only the wireless local area network capability. With the location and velocity information, the lifetime of the path can be also predicted. He *et al.* [18] proposed an on-demand differentiated reliable routing protocol, which finds a number of link disjoint paths for each application in order to guarantee its reliability requirement and utilize the RSUs to reduce the overhead while improving the probability of successful connection setup. Su and Zhang [19] proposed a cluster-based communication scheme in V2V environments. If a vehicle can receive a valid invite-to-joint message from a cluster head within a specific time period, it will become a cluster member. Otherwise, it will become a cluster head by itself. A cluster can be merged with another cluster if the cluster head receives a valid invite-to-joint message from a neighboring cluster head which has more members.

Bronsted and Kristensen [20] proposed a scheme for efficient data dissemination among the vehicles using two new protocols called as zone flooding and zone diffusion. The two protocols aggregated and diffused the data to the vehicles within a geographical region bounded by its diameter. The scheme can be used in wide areas of applications in VANETs. Ibrahim and Weigle [21], [22] presented an information dissemination protocol similar to Traffic View [7] and SOTIS [8], [9]. The proposed scheme assumes that vehicles form clusters on a highway by exchanging the speed and position information within the cluster. A strict dropping policy and life cycle are applied for removing the old data in a caching for the new incoming data.

Wu *et al.* [23] proposed a mobility centric data dissemination algorithm for VANETs. The authors described vehicle mobility for data dissemination and combined the idea of opportunistic forwarding, trajectory-based forwarding, and geographical forwarding. Lim *et al.* [1] proposed a cache invalidation scheme that introduces various push- and pull-based strategies. An aggregate cache on demand has been introduced with modified two pull-based strategies which were earlier used for cellular networks. Safa *et al.* [3] have proposed a new selective adaptive cache invalidation scheme, which utilizes a new technique for generating invalidation reports to address the issue of false invalidation existing in most of the previous caching schemes. Lim *et al.* [4] have proposed a new caching scheme to improve the performance of VANETs. The authors have described a simple cache scheme which reduces the average latency when accessing the video content through wireless networks for mobile nodes. The proposed scheme increases the cache hit ratio which affects the overall performance of various applications in VANETs.

Cao [24], [25] proposed a power-aware adaptive cache management for mobile computing systems to address the issues such as low bandwidth, query latency, and frequent disconnections due to the mobility of the nodes. The author proposed the use of caching and indexing techniques for continuous topology changes in the mobile computing environment. Chand *et al.* [26] have proposed a synchronous stateful cache invalidation scheme to minimize cache overhead. Lau *et al.* [27] proposed a new mechanism for cooperative caching of multimedia data objects for mobile nodes. The authors have proposed an architecture in which an application manager is placed between the application and the network layers at each of the participating nodes in the network. The task of the manager is to find the best route for caching without creating additional load on the network. Lyer *et al.* [28] proposed a P2P-based decentralized Web caching scheme. The authors described a new technique for enabling the Web browsers on desktop machines so that local caches can be shared with the other nodes. Adler *et al.* [29] proposed a context-adaptive information dissemination algorithm for VANETs. The authors proposed a self-organizing and context-adaptive dissemination framework which is capable of combining the advantages of existing routing mechanisms. Message selection and medium access are based on a relevance function that makes use of the current context and the contents of the messages. Singh *et al.* [30] proposed cache invalidation with respect to location-dependent and location-independent data collected by the vehicles.

Although there exist many proposals in the literature for caching the data in the context of vehicular communications, but to the best of our knowledge, none of the existing ones have considered the caching of data with respect to the movement of vehicles. The following are the key contributions in this paper:

1) a new P2PCC algorithm;
2) an analytical model developed to estimate the delay for placing the cache data at different locations such as on RSUs, on gateway, or in the vehicles;
3) a comprehensive performance analysis for the proposal.

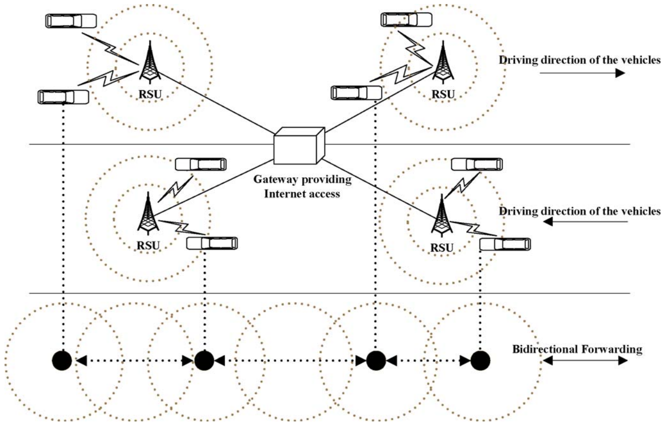

RSU
RSU
RSU
RSU
Driving direction of the vehicles
Gateway providing Internet access
Driving direction of the vehicles
Bidirectional Forwarding

Fig. 1. Considered traffic model.

III. Traffic Model and Problem Formulation

A. Traffic Model

Fig. 1 shows the considered 2-D traffic model in which vehicles can move in both directions and cache data can be placed either at an RSU, at a gateway, or in vehicles. A gateway is connected to the Internet to provide various services to the moving clients, e.g., users may download/upload the contents during mobility. For V2I, RSUs are deployed in different geographical regions. The RSU sends/receives the information to the centralized control continuously or after a finite interval of time. For V2V, each vehicle communicates and shares the information with the other vehicle in a P2P manner. We have considered the second type of communication in the proposed scheme in which vehicles communicate with each other for information sharing in a P2P manner without causing the load on the centralized control which improves the performance of the system to a substantial amount.

B. Problem Formulation

Let $V = \{V_1, V_2, \ldots, V_n\}$ denote the vehicles in the network. $D = \{D_1, D_2, \ldots, D_n\}$ and $Q = \{Q_1, Q_2, \ldots, Q_n\}$ are the data identifications (IDs) and queries generated for particular data that may be cached during the transmission, respectively.

Each query for a data item has the following parameters $\{V_i, D_i, \theta, \rho, \tau_{\text{interval}}\}$, where θ and ρ are the access frequency and probability of availability of the data item in a time interval τ_{interval}, respectively. This query is broadcasted to all the vehicles falling under the same RSU. Upon receiving the query about the data item, the vehicles check their local cache for the availability of the data item. If the data are available in their local cache, the vehicles reply. Otherwise, the query is

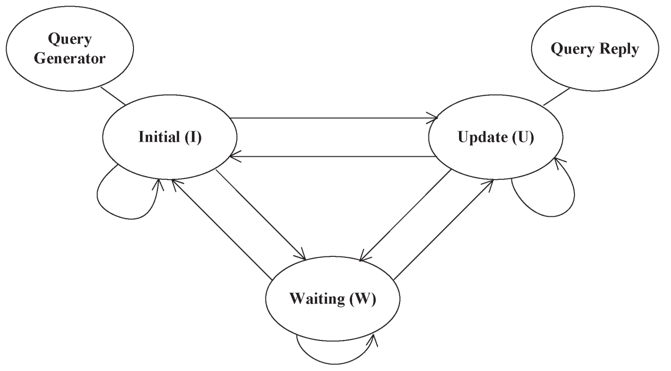


Fig. 2. Markov chain model with three states: Initial (I), waiting (W), and update (U).

put into a waiting state. Whenever the data are available, the query is replied back to the vehicle from where it comes. If more than one query reply comes, then these are put in a queue and then processed accordingly. The size of the queue S_{queue} is set with respect to the minimum threshold $S_{\min}$ and maximum threshold $S_{\max}$. Progress of each query is also monitored by the start time τ_{start} and update time τ_{update}. The time interval for processing the query is taken as the difference between τ_{start} and τ_{update}. We have not considered any location dependence algorithms in the proposed scheme and assume as proposed in [29]. Each vehicle generates invalidation reports for those data items which are not relevant for its use and need to be replaced with the fresh copies. Also, each query is marked with the timestamp τ_{stamp} to keep track of the time at which that query has been generated.

Let N_{attempts} be the number of attempts made for accessing a particular data item and N_{hits} be the total number of hits in the cache. Then, ρ is computed as $\rho = N_{\text{hits}}/N_{\text{attempts}}$. Accordingly, an objective function is defined as

$$\begin{aligned} \text{maximize} \quad & \rho \\ \text{subject to} \quad & S_{\min} \leq S_{\text{queue}} \leq S_{\max} \\ & \tau_{\text{start}} \leq \tau_{\text{interval}} \leq \tau_{\text{update}}. \end{aligned}$$

IV. P2PCC

A. Overview

The proposed P2PCC scheme is illustrated with the Markov chain model with three states, as shown in Fig. 2.

As shown in Fig. 2, each vehicle's state has a certain probability to pass a query from one state to another or to stay back into the current position, i.e., the vehicles share the data in an interactive manner by exchanging the query with each other. In this process, the vehicle's state may remain in the same state or it may jump to other state such as the waiting state or update state, depending upon the probability, i.e., the vehicle caches the data obtained from other vehicles depending upon its probability.

In the initial state, a query is accepted from a query generator, which is a client in this case. Depending upon the request in the query, the data for the query may be available in a vehicle's cache. If they are available, the query result is passed at once to the client, and the process is over. However, if the data are not available, it is passed to some other vehicles until the reply of the query from other vehicle cache is not available, and the process enters into the waiting state. After the query result is passed to the requesting vehicle, the state of the system is changed to the update state or to the initial state to receive the next query.

The transition from one state to another is probabilistic in nature, and the associated probability is thus calculated from the cache transition probability matrix (CTPM). Denote S as the associated probability to cache the data from some other vehicles. Then, S is calculated as

$$S = \begin{bmatrix} S_{I,I} & S_{I,W} & S_{I,U} \\ S_{U,I} & S_{U,U} & S_{U,W} \\ S_{W,I} & S_{W,U} & S_{W,W} \end{bmatrix}.$$

Suppose that t_M is the target mean time in accessing the cache, t_R is the required mean delay due to the mobility of the vehicles, and t_C is the cumulative mean delay, i.e., total delay occurred in accessing the cache with passage of time. The time to live (TTL) is divided into different time slots as TTL $= \{t_1, t_2, \ldots, t_l\}$. Then, t_C is calculated as

$$t_C = N_{\text{hops}} \left[\frac{t_R - \sum_{i=1}^{l} t_i t_M}{N_{\text{attempts}}} \right] \tag{1}$$

where N_{hops} is the number of hops traveled by the vehicle. Then, the value of each state $S_{i,j}$ in the CTPM is computed as

$$S_{i,j} = \begin{cases} \frac{t_R - \sum_{i=1}^{l} t_i t_M}{N_{\text{attempts}}}, & i = j \\ t_C, & i \neq j \end{cases} \tag{2}$$

where $i \in \{I, U, W\}$ and $j \in \{I, U, W\}$.

In the proposed scheme, a replacement of cache entries is performed when the cache is full, and a new entry is required to be added. The deletion of old entries and the addition of new entries in the cache are performed in a probabilistic nature. The data item that has a minimum probability is selected for a replacement for the new incoming entries. We have considered the duration of the query request in the waiting state as a trigger to find the data item to be replaced. The greater the time spent in the waiting time, the greater the frequency of the data to be accessed by the vehicle and hence the lesser its probability of replacement. The data item which will be available at once may not be in demand by the other vehicles, and hence, it has higher chances of replacement. Moreover, the data item which has a higher frequency of access in τ_{interval} is also taken into account. The higher frequency of access of the data item tells that it is in great demand and it keeps the other vehicles in the waiting state as described earlier. Keeping in view of all these factors, a probability of the replacement π is calculated as

$$\pi = \frac{1}{t_W D_i \theta \tau_{\text{interval}}} \tag{3}$$

where t_W is the time to stay in the waiting state for particular data D_i. From (3), the higher the value of this variable and the frequency of its access, the lesser the probability of its replacement.

B. Algorithm

The algorithm for data selection and cache replacement is presented in the following. The complete description of the designed algorithm is as follows. Every data item is identified with unique ID for which request may be generated from the vehicles. Initially, the data item is searched in the local cache of the vehicles followed by nearest access points/RSUs. As vehicles have limited memory, thus corresponding to each request for new data ID, entries in the cache of the vehicles are inserted/deleted in a probabilistic manner, as presented in (3).

Algorithm 1 Finding the required data from caches

```
if Request to find $D_i$ arrived at $V_i$ then
    Check the local cache of $V_i$
    if $D_i$ found then
        Return $D_i$ found
        Exit
    else
        Generate $Q_i$ corresponding to $D_i$
        Set $S_{\text{queue}} \leftarrow S_{\text{queue}} + 1$
        if ($S_{\min} \leq S_{\text{queue}} \leq S_{\max}$) is not satisfied then
            Perform a cache replacement with $\pi$
            Continue
        else
            Broadcast $Q_i$ to all other vehicles
            while Wait for an acknowledgement with $D_i$ do
                if $D_i$ acknowledged then
                    Compute $t_C$
                    Return $D_i$ acknowledged
                    Exit
                end if
                if $D_i$ not acknowledged in a given time then
                    Drop $Q_i$
                    Exit
                    end if
            end while
        end if
    end if
end if
```

The probability is computed with respect to different states as described in Fig. 2. For faster response, a data item is searched in the nearby vehicle cache, and accordingly, attributes of the cache query are set up, e.g., vehicle ID, start and end times, and data ID. The algorithm has been designed keeping in view of the two processes (cache query and updates and cache replacement) as defined earlier. The algorithm starts with query generation for data ID from vehicles/clients. A queue with the upper and lower thresholds is maintained by each vehicle to keep record of all the queries generated for same/different data ID. If the queue size is between the upper and lower thresholds, it is accepted, and then, a search for the requested data item begins with the vehicle ID and data ID along with the time to start the query. The vehicle broadcasts the query for specified data ID to all the vehicles if the data ID is not found in the local cache and waits for the acknowledgement from some vehicles during the fixed time interval. The procedure is as follows: A broadcast message is sent to all the vehicles with vehicle ID and waits until the acknowledgment comes. If nobody replies, then the call is dropped; else, each intermediate vehicle checks its local cache for the data item in the cache query and replies back if it succeeds, or passes the same to the next vehicle in a P2P manner. This process continues until all the vehicles in a particular geographical region are searched. Now, the second process starts where the space for the new data ID is located after a receipt of the acknowledgement message in which the local cache is searched. If an enough space is not available for caching the data ID, then the request is forwarded to the nearest RSU. If a reply about caching is received from the nearest RSU, then the message is sent to the demanding vehicle with its own ID. Otherwise, it enters into the update time with an increment of the value of update time. This process is done to ensure that replacement of queries from the existing records in the vehicles cache. If the queue size is greater than the maximum size, then it enters into the waiting state and replaces some of the items from the cache with the vehicle ID, frequency of access, and data ID.

The cache replacement requires the vehicle ID, frequency of data access, and data ID. The probability of each data item is calculated using (3), and that data item is selected for the replacement which has a minimum value of the probability as computed in (3).

V. Cost Analysis

This section provides an analysis of cost computation of accessing the cache. We have considered a bidirectional vehicle traffic model in which vehicles are moving in both directions. It is assumed that RSUs are located at the center of the circle with radius R so that the RSUs can serve the vehicles on both sides of the road. In what follows, we compute the cost of placing and accessing the cache using the query generator for three different cases.

A. Vehicle to RSU

Suppose that $\text{Query}^{\text{succ}}$ is a query successful rate from an RSU. Let $\text{Query}^{\text{gen}}$ denote the rate at which queries are generated. It is assumed that the query generation rate follows a Poisson distribution and the query successful rate is exponentially distributed. Then, the query cost from a vehicle to an RSU C^{V-R} is computed as

$$C^{V-R} = \frac{\text{Query}^{\text{succ}}}{\text{Query}^{\text{gen}}} t_{\text{MR}} \tag{4}$$

where t_{MR} is the migration time of the queries from the vehicle to the RSU.

We assume that there are n segments $S = \{S_1, S_2, \ldots, S_n\}$ in an urban region. Suppose that $t = \{t_1, t_2, \ldots, t_n\}$ is a set of different time intervals in which events between vehicles and RSUs occur. Let λ and ϖ be the rate of queries and the density of the vehicles in the segments defined earlier, respectively. Then, we define a function $f(\cdot)$ as

$$f(\cdot) = \frac{\sum_{i=1}^{n} (\lambda \varpi) S_i}{\nu^{\max}} \tag{5}$$

where $\nu^{\max}$ is the maximum speed of the vehicles in a particular segment. Note that (5) implies that the density of the vehicles increases when the velocity of the vehicles decreases. Then, the probability of the query transmitted to RSUs is modeled as

$$P(\text{Query}^{\text{gen}}) = \frac{e^{-Sf}(Sf(\cdot))^n}{n!}, \qquad n = 1, 2, \ldots, N. \tag{6}$$

We assume that, corresponding to each query generated from vehicles with respect to the conditions as above, the probability of success is ρ during the time interval $(t_2 - t_1)$. Then, we can define the probability of success of at least a single query having a Poisson distribution with parameter $1 - e^{-f(\cdot)\rho}$. Thus, we can write the probability of this event as

$$P(\text{Query}^{\text{succ}}) = e^{-f(\cdot)\rho}\left(1 - e^{-f(\cdot)\rho}\right). \tag{7}$$

Some of the vehicles are moving toward and others are moving away from a particular RSU. Thus, let us assume that, at time t_1, all the vehicles are D distance apart from the RSU and having a constant velocity ν_1. Therefore, the distance covered in time $(t_2 - t_1)$ is $\nu_1(t_2 - t_1)$. The required time to send and receive the query from the RSU is as follows:

$$\text{MR}^T = 2\left(\frac{D \pm \nu_1(t_2 - t_1)}{R} t_T + C^{\text{HO}}\right) \tag{8}$$

where t_T is the travel time of the request from the vehicle to the RSU. It depends upon the distance and transmission speed of the request, and C^{HO} is the cost associated with a handoff mechanism. Note that the handoff cost consists of signaling and tunneling costs and is computed like in [3]. By putting the values of (6)–(8) into (4), we get the cost associated with the query retrieval from the RSUs with migration of the vehicles to and away from the RSUs.

B. Vehicle to Gateway

As most of the activities for moving vehicles are monitored and controlled by the RSUs, thus, less congestion would be on a gateway to answer the queries sent from the moving vehicles. The gateway is located at far distance compared to RSUs, so the response to the query generated from the vehicles is increased by the factor $D^{R-G}/\nu^{\max}$, where D^{R-G} is the distance between an RSU and a gateway. Thus, it will be costly to place and access the cache from the gateway. The query cost from a vehicle to a gateway C^{V-G} is thus computed as

$$C^{V-G} = \frac{\text{Query}^{\text{succ}}}{\text{Query}^{\text{gen}}} t_{\text{MR}} D^{R-G}. \tag{9}$$

C. V2V

In this case, accessing the content from a vehicle is performed in a P2P cooperative manner without an involvement of RSUs or a gateway. Accordingly, the delay would be less as compared to the other cases. As all the vehicles are in the same segment and moving with constant speed within the transmission range of each other, the cost of cache placement and accessing at peer nodes consists of only the processing delay. The query cost from a vehicle to a vehicle C^{V-V} is thus computed as

$$C^{V-V} = \frac{\text{Query}^{\text{succ}}}{\text{Query}^{\text{gen}}}. \tag{10}$$

TABLE I
USED PARAMETERS AND VALUES

Parameter	Value
Number of vehicles (nodes)	500
Average velocity of vehicles	20-40 miles/h
Period of traffic lights	60 sec
Wireless communication range	300 m
Query interval	10 sec
Query update interval	10 sec
Size of a cache	4 MB

VI. ANALYSIS RESULTS AND DISCUSSION

A. Simulation Environment

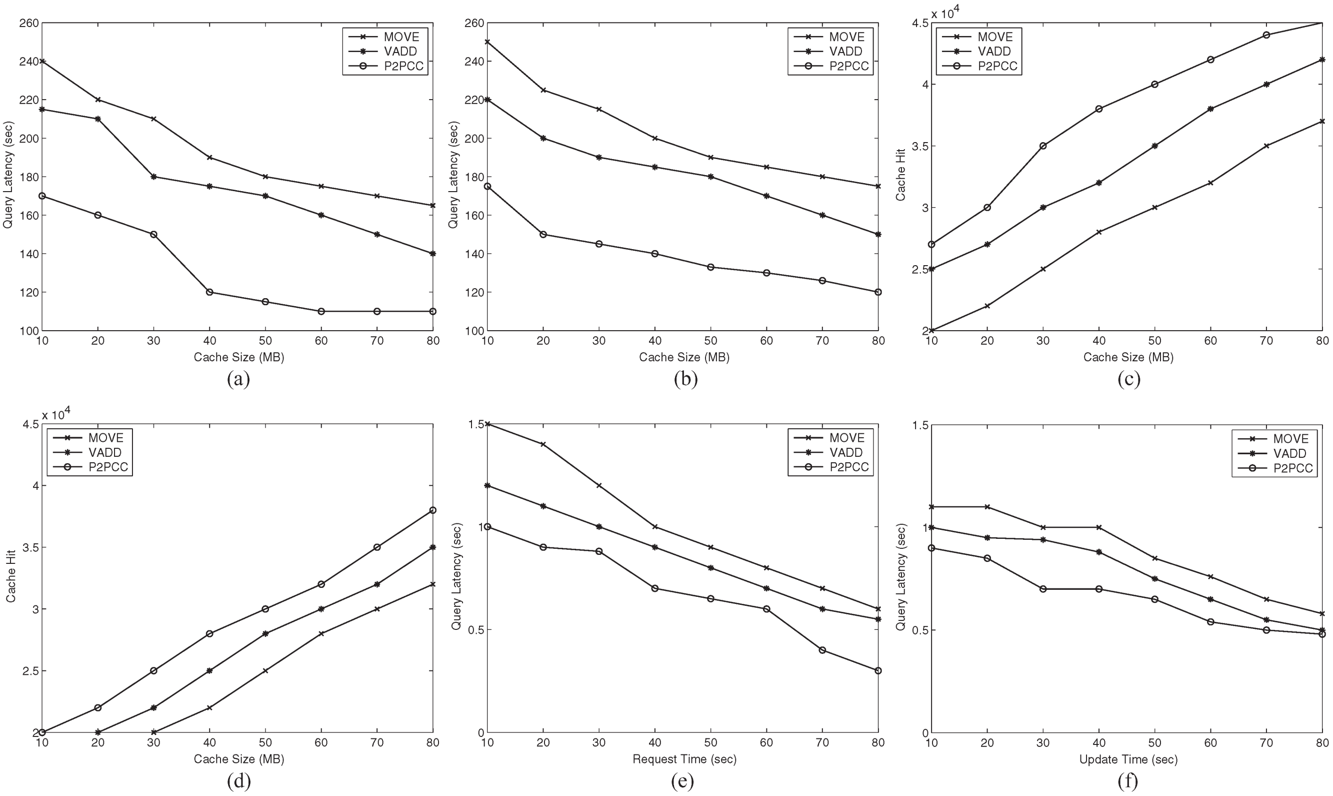
The simulation tool dedicated to vehicular communications, VanetMobiSim [32], has been used to evaluate the proposed scheme compared to the proposals MOVE [33] and VADD [34]. The VanetMobiSim has built-in vehicle motion models in which it can import maps from the existing database and adds support for multilane roads, separate directional flows, differentiated speed constraints, and traffic signs at intersections. It implements new mobility models for providing V2V and V2I communications. Thus, vehicles can regulate their speed depending on nearby cars, overtake each other, and act according to traffic signs in the presence of intersections. A road is divided into different segments, and each segment has different density of the vehicle at any instant of time. Each vehicle's movement pattern is determined by a hybrid mobility model. The vehicles move according to the roadways from their start position to their end position. The simulation is continued until all of these packets are either delivered or dropped due to the TTL expiration. The simulation parameters are given in Table I.

B. Query Latency Versus Cache Size

We have evaluated the query latency of the three different schemes with two different area scenarios such as a less congestion area and a congestion area (i.e., urban area). Fig. 3(a) shows the variation of the query latency with different cache sizes in the less congestion area, whereas Fig. 3(b) shows the variation in the congestion area. As shown, the proposed scheme, P2PCC, has less query latency in both the scenarios compared to the other schemes. For instance, with an increase in the cache size, the query latency is reduced up to 30% in the proposed scheme compared to the other schemes, owing to the proposed scheme's probabilistic selection for caching and searching of contents in the cache. The contents are searched in a P2P manner, so there is no need of contacting the server that helps to reduce the network load and access time.

C. Cache Hit Versus Cache Size

The impact of the cache size on the cache hit is presented. Fig. 3(c) shows the variation of the cache hit with different



(a) (b) (c)

(d) (e) (f)

Fig. 3. Simulation results (1/2). (a) Query latency versus cache size (less congestion). (b) Query latency versus cache size (congestion). (c) Cache hit versus cache size (less congestion). (d) Cache hit versus cache size (congestion). (e) Query latency versus request time. (f) Query latency versus update time.

cache sizes in the less congestion area, whereas Fig. 3(d) shows the variation in the congestion area. The cache hit increases in all the schemes when the cache size increases. The proposed scheme still outperforms in both the scenarios. An interesting point is that, with an increase in the cache size, the proposed scheme's cache hit performance is more increased in the less congestion area than in the congestion area (i.e., urban area). This phenomenon happens owing to the fact that the congestion area means a high density of vehicles in the area and the high density causes a high failure to find suitable vehicles for query searching and caching. Hence, there is a decrease in the cache hit for a particular query in the congestion area.

D. Query Latency Versus Request/Update Time

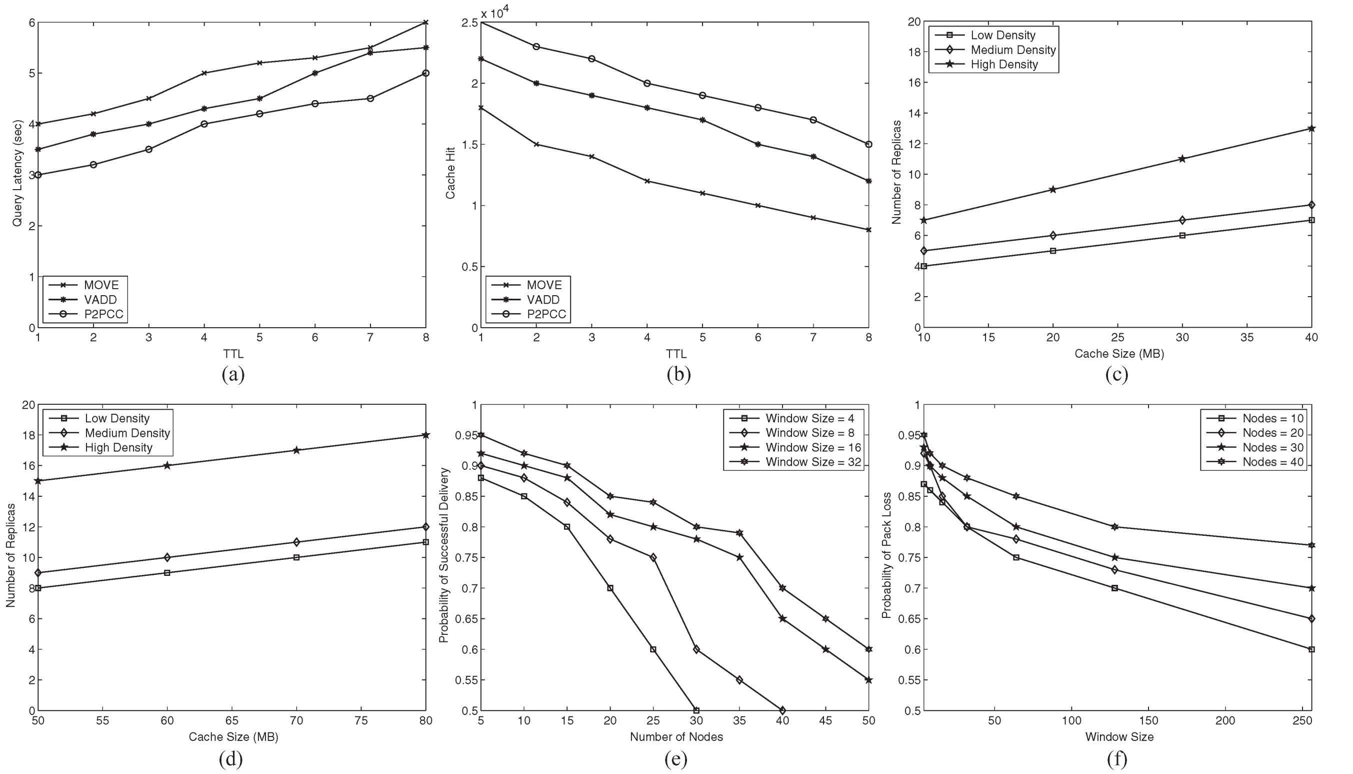
The impact of the request/update time on the query latency is presented. The request time is the time taken to process a query and the update time is the time taken to update the contents when certain operations are performed on the query. As shown in Fig. 3(e) and (f), the proposed scheme provides better performance than the others as it utilizes the Markov chain model with three states to efficiently handle a query about particular data from a vehicle in the network.

E. Query Latency/Cache Hit Versus TTL

The variations of the query latency and cache hit are presented with respect to the TTL in Fig. 4(a) and (b). As shown, by varying the TTL from one hop to eight hops, the query latency decreases in all the schemes. Similarly, for all the schemes, the cache hit is also decreased with an increase in the TTL. In other words, as the TTL increases, the latency to access the remote information must be increased. The proposed scheme still outperforms the other schemes, owing to the probabilistic selection for caching and searching of contents in the cache. In addition, the proposed scheme's P2P content search minimizes the impact of the TTL to the proposed scheme.

F. Replicated Data Versus Cache Size

The proposed scheme is analyzed in terms of the replicated data. Fig. 4(c) and (d) shows the impact of varying the cache size on the replicated data. As shown, with an increase in the cache size and the node density, the average number of replicas increases. This is due to the fact that, with an increase in both these values, there is an increase in space for data items, and a large space would be available to store the data items. This reduces the time spending in the waiting state of the proposed Markov chain model. In other words, it helps to reduce the access delay for a particular query. Hence, a large replica of data can be built with an increase in the cache size and node density. The advantage of creating this replica with an increase in the cache size is that this will create more chances of hits in the cache and also an increase in the probability of accurate query selection. This also increases the effectiveness of the proposed scheme compared to the other schemes. Hence, the proposed scheme has higher chances of query selection with a reduction in latency as compared to the others.



(a) (b) (c)

(d) (e) (f)

Fig. 4. Simulation results (2/2). (a) Query latency versus TTL. (b) Cache hit versus TTL. (c) Replicated data versus cache size. (d) Replicated data versus cache size. (e) Probability of successful delivery versus number of nodes. (f) Probability of packet loss versus window size.

G. Probability of Successful Delivery/Packet Loss Versus Number of Nodes/Window Size

The impacts of varying the number of nodes and the contention window size on the probability of successful packet delivery and the probability of packet loss are investigated in Fig. 4(e) and (f). We have divided the TTL value into different slots with searching and update operations performed in a particular time slot. We have varied the window size with an increase in number of nodes. As shown in Fig. 4(e), with an increase in the window size and the number of nodes, the probability of successful delivery increases. This is owing to the fact that, with an increase in the window size, more time would be provided to searching and updating operations, and there is an increase in the probability of successful delivery. Moreover, with increasing the window size of the contention window and the number of nodes, the probability of packet loss decreases. As if we will increase the size of the contention window, then the chances of collision are less and packet loss is less which will increase the probability of successful packet delivery and decrease the probability of packet loss.

VII. Conclusion

To avoid congestion and to minimize the load on the network, caching is widely used. To achieve these goals and by keeping in view of the mobility of nodes, we have proposed a new P2PCC scheme for urban vehicular communication environments. The information among vehicles in the network is shared in a P2P manner. The vehicles share the information of taking routing decisions, and this information is cached in an efficient manner in the proposed scheme. This information is then used in the future when the query about it is generated from some other vehicles in a particular region. The performance of the proposed scheme is evaluated in comparison to those of the existing schemes with respect to the metrics such as network congestion, query delay, and hit ratio. The results show that the proposed scheme has reduced the congestion and query delay by 30% and an increase in the hit ratio by 20%. The work presented in this paper can be also enhanced with proxy mobile caching in which a proxy server is used for accessing the contents during the movements of vehicles.

References

[1] S. Lim, W.-C. Lee, G. Cao, and C. Das, "Cache invalidation strategies for Internet-based mobile ad hoc networks," *Comput. Commun.*, vol. 30, no. 8, pp. 1854–1869, Jun. 2007.

[2] Z. Zhang, G. Mao, and B. Anderson, "On the information propagation process in mobile vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 5, pp. 2314–2325, Jun. 2011.

[3] H. Safa, H. Artail, and M. Nahhas, "A cache invalidation strategy for mobile networks," *J. Netw. Comput. Appl.*, vol. 33, no. 2, pp. 168–182, Mar. 2010.

[4] S. Lim, W.-C. Lee, G. Cao, and C. R. Das, "A novel caching scheme for improving Internet-based mobile ad hoc networks performance," *Ad Hoc Netw.*, vol. 4, no. 2, pp. 225–239, Mar. 2006.

[5] C. Lochert, B. Scheuermann, and M. Mauve, "A probabilistic method for cooperative hierarchical aggregation of data in VANETs," *Ad Hoc Netw.*, vol. 8, no. 5, pp. 518–530, Jul. 2010.

[6] J. Rybicki, B. Scheuermann, and M. Mauve, "Peer-to-peer data structures for cooperative traffic information systems," *Pervasive Mobile Comput.*, vol. 8, no. 2, pp. 194–209, Apr. 2012.

[7] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode, "Traffic View: Traffic data dissemination using car-to-car communication," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 8, no. 3, pp. 6–19, 2004.

[8] L. Wischhof, A. Ebner, H. Rohling, M. Lott, and R. Halfmann, "SOTIS—A self organizing traffic information system," in *Proc. IEEE VTC*, 2003, pp. 2442–2446.

[9] L. Wischhof, A. Ebner, and H. Rohling, "Information dissemination in self-organizing intervehicle networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 1, pp. 90–101, Mar. 2005.

[10] S. Biswas, R. Tatchikou, and F. Dion, "Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety," *IEEE Commun. Mag.*, vol. 44, no. 1, pp. 74–82, Jan. 2006.

[11] K. Yang, S. Ou, H.-H. Chen, and J. He, "A multihop peer-communication protocol with fairness guarantee for IEEE 802.16-based vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3358–3370, Nov. 2007.

[12] L. Wischhof, A. Ebner, H. Rohling, M. Lott, and R. Halfmann, "Adaptive broadcast for travel and traffic information distribution based on intervehicle communication," in *Proc. IEEE IV*, Jun. 2003, pp. 6–11.

[13] G. Korkmaz, E. Ekici, and F. Ozguner, "Urban multi-hop broadcast protocols for inter-vehicle communication systems," in *Proc. ACM VANET*, Oct. 2004, pp. 76–85.

[14] K. Shafiee and V. C. M. Leung, "A reliable robust fully ad hoc data dissemination mechanism for vehicular network," *Int. J. Adv. Sci. Technol.*, vol. 2, pp. 53–62, Jan. 2009.

[15] V. Naumov and T. R. Gross, "Connectivity-aware routing (CAR) in vehicular ad hoc networks," in *Proc. IEEE INFOCOM*, May 2007, pp. 1919–1927.

[16] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, and O. Tonguz, "Routing in sparse vehicular ad hoc wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp. 1538–1556, Oct. 2007.

[17] V. Namboordiri and L. Gao, "Prediction-based routing for vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 4, pp. 2332–2345, Jul. 2007.

[18] R. He, H. Rutagemwa, and X. Shen, "Differentiated reliable routing in hybrid vehicular ad-hoc networks," in *Proc. IEEE ICC*, May 2008, pp. 2353–2358.

[19] H. Su and X. Zhang, "Clustering-based multichannel MAC protocols for QoS provisionings over vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3309–3323, Nov. 2007.

[20] J. Bronsted and L. M. Kristensen, "Specification and performance evaluation of two zone dissemination protocols for vehicular adhoc networks," in *Proc. ANSS*, 2006, pp. 1–12.

[21] K. Ibrahim and M. C. Weigle, "Accurate data aggregation for VANETs," in *Proc. ACM VANET*, 2007, pp. 71–72.

[22] K. Ibrahim and M. C. Weigle, "Optimizing CASCADE data aggregation for VANETs," in *In Proc of MoVeNet*, 2008, pp. 724–729.

[23] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter, "MDDV: A mobility centric data dissemination algorithm for vehicular networks," in *Proc. ACM VANET*, 2004, pp. 47–56.

[24] G. Cao, "A scalable low-latency cache invalidation strategy for mobile environments," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 5, pp. 1251–1265, Sep. 2003.

[25] G. Cao, "Proactive power-aware cache management for mobile computing systems," *IEEE Trans. Comput.*, vol. 51, no. 6, pp. 608–621, Jun. 2002.

[26] N. Chand, R. Joshi, and M. Misra, "Energy efficient cache invalidation in a disconnected wireless mobile environment," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 2, no. 1/2, pp. 83–91, 2007.

[27] W. Lau, M. Kumar, and S. Venkatesh, "A cooperative cache architecture in supporting caching multimedia objects in MANETs," in *Proc. WoWMoM*, Sep. 2002, pp. 56–63.

[28] S. Lyer, A. Rowstron, and P. Druschel, "Squirrel: A decentralized peer-to-peer Web cache," in *Proc. ACM PODC*, Jul. 2002, pp. 213–222.

[29] C. Adler, R. Eigner, C. Schroth, and M. Strassberger, "Context-adaptive information dissemination in VANETs—Maximizing the global benefit," in *Proc. CSN*, 2006, pp. 7–12.

[30] A. Singh, N. Chand, and L. Awasthi, "Cache invalidation for location dependent and independent data in IVANETs," in *Proc. ACC*, 2011, pp. 609–614.

[31] M. C. Chuang, J.-F. Lee, and M.-C. Chen, "SPAM: A secure password authentication mechanism for seamless handover in proxy mobile IPv6 networks," *IEEE Syst. J.*, vol. 7, no. 1, pp. 102–113, Mar. 2013.

[32] J. Harri, M. Fiore, F. Fethi, and C. Bonnet, "VanetMobiSim: Generating realistic mobility patterns for VANETs," in *Proc. ACM VANET*, Sep. 2006, pp. 96–97.

[33] J. LeBrun, C. N. Chuah, D. Ghosal, and M. Zhang, "Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks," in *Proc. IEEE VTC*, May 2005, pp. 2289–2293.

[34] J. Zhao and G. Cao, "VADD: Vehicle-assisted data delivery in vehicular ad hoc networks," in *Proc. IEEE INFOCOM*, Apr. 2006, pp. 1–12.

Neeraj Kumar received the Ph.D. degree in computer science and engineering from Shri Mata Vaishno Devi University, Katra, India, and the Postdoctoral Fellow from the U.K.

He is currently an Assistant Professor with the Department of Computer Science and Engineering, Thapar University, Patiala, India. He has delivered invited talks and lectures in various IEEE international conferences. His research is focused on mobile computing, parallel/distributed computing, multiagent systems, service-oriented computing, routing, and security.

Jong-Hyouk Lee (M'07–SM'12) received the M.S. and Ph.D. degrees in computer engineering from Sungkyunkwan University, Suwon, Korea, in 2007 and 2010, respectively.

In 2009, he joined the project team Informatique, Mathématiques et Automatique pour la Route Automatisée (IMARA), Institut National de Recherche en Informatique et en Automatique (INRIA), where he undertook the protocol design and implementation for IPv6 vehicular (ITS) communication and security. In 2012, he started his academic profession at TELECOM Bretagne, Rennes, France, as an Assistant Professor. In September 2013, he moved to the Department of Computer Software Engineering, Sangmyung University, Cheonan, Korea, as an Assistant Professor. He is an Associate Editor of Wiley *Security and Communication Networks*. His research interests include authentication, privacy, and mobility management.

Dr. Lee was the recipient of the Best Paper Award at the 2012 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob) and was a Tutorial Speaker at the 2013 IEEE Wireless Communications and Networking Conference (WCNC). He is an Associate Editor of the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS.