MMCD: Cooperative Downloading for Highway

MMCD: Cooperative Downloading for Highway VANETs

Kaoru Ota, *Member, IEEE,* Mianxiong Dong, *Member, IEEE,* Shan Chang, *Member, IEEE,*
and Hongzi Zhu, *Member, IEEE*

Abstract—Advances in low-power wireless communications and micro-electronics make a great impact on a transportation system and pervasive deployment of road-side units (RSU) is promising to provide drive-thru Internet to vehicular users anytime and anywhere. Downloading data packets from the RSU, however, is not always reliable because of high mobility of vehicles and high contention among vehicular users. Using inter-vehicle communication, cooperative downloading can maximize the amount of data packets downloaded per user request. In this paper, we focus on effective data downloading for real-time applications (e.g., video streaming, online game) where each user request is prioritized by the delivery deadline. We propose a cooperative downloading algorithm, namely MMCD, which minimizes an average delivery delay of each user request while maximizing the amount of data packets downloaded from the RSU. The performance of MMCD is evaluated by extensive simulations and results demonstrate that our algorithm can reduce mean delivery delay while gaining downloading throughput as high as that of a state-of-the-art method although vehicles highly compete for access to the RSU in a conventional highway scenario.

Index Terms—Drive-thru Internet; cooperative downloading; vehicular networks

1 Introduction

With the rapid advance of wireless communication technologies, vehicular networks are emerging as a new landscape of mobile ad hoc networks, aiming to provide a wide spectrum of safety and comfort applications to drivers and passengers [1]–[4]. In the vehicular networks, vehicles equipped with wireless communication devices can transfer data with each other (V2V: vehicle-to-vehicle communications) as well as with roadside infrastructures (V2I: vehicle-to-infrastructure communications). Because of these technologies, the needs of using the Internet, checking email, and watching videos during the driving time have increased more and more. The recent penetration of LTE/Wimax/3G networks makes it possible for users to access the Internet even while they are in motion. However, there are still some reasons to fully utilize the vehicular networks to assist the drive-thru Internet access. First, due to tremendous traffic generated by cellular networks even at this moment, capacity of the cellular networks is near to the limits [5] and also cost of the Internet access via the cellular networks remains high, e.g., average 60 USD/7GB in Japan [6] and 10 USD/1GB in Canada [7]. Second, since a mobile phone's screen is generally smaller than one embedded with the vehicle, it is more convenient to use the Internet via the vehicle especially for entertainment scenarios. Last, the mobile phones are energy-limited [8]. Frequent use of the mobile phone for accessing the Internet costs the fast battery usage not to mention for video streaming. Although it can be charged during the driving time, it is no doubt that the equipped device in the vehicle is not only more convenient to use than the mobile phones but also more safer for the user in motion.

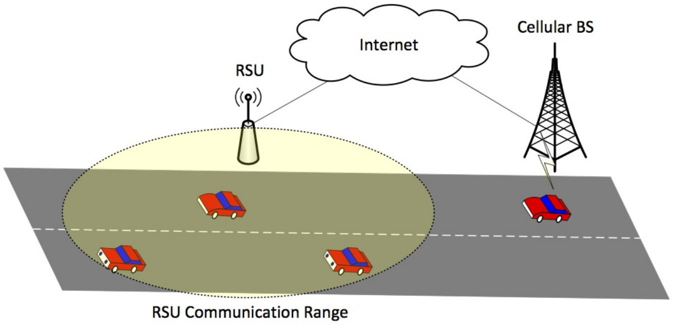


Fig. 1. A model of drive-thru Internet systems

In general, the Internet access in vehicular networks is composed by RSUs, and vehicles with wireless communication devices. RSUs are connected with the Internet by wired backbone (Fig. 1). Provisioning Internet service in the vehicular networks, however, is quite challenging because of the high mobility of the vehicles which makes the connectivity intermittent. Due to this highly dynamic nature, real-time downloading is very hard in such kind of networks. Several researches pay attention on

- *K. Ota and M. Dong are with Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan.*
  E-mail: ota@csse.muroran-it.ac.jp, mx.dong@csse.muroran-it.ac.jp
- *S. Chang is with Department of Computer Science and Technology, Dong Hua University, China.*
  Email: changshan@dhu.edu.cn
- *H. Zhu is with Department of Computer Science and Engineering, Shanghai Jiao Tong University, China.*
  Email: hongzi@cs.sjtu.edu.cn

maximization of the overall throughput for downloading [9]. In this paper, we focus on a real-time application scenario in the vehicular networks where each packet has deadline to be downloaded. Our approach is from two parts. We first analyze the throughput to be maximized, and then study the characteristics for minimizing the delivery delay. Based on these results, we propose MMCD which elegantly integrates both these two characteristics. The extensive simulation results reveal that MMCD performs well than a state-of-art algorithm.

The remainder of our paper is organized as follows. Section 2 reviews related work in the literature and section 3 presents the system model. Section 4 discusses cooperative downloading and flow scheduling, followed by proposing our algorithm MMCD. Section 5 evaluates the performance of MMCD and presents the results. In section 6, we further discuss a trade-off relationship between the download throughput and delay on the experimental results and present one solution to optimize the performance in MMCD under a dense traffic scenario. Finally, we present concluding remarks and outline the directions for future work in section 7.

2 Related Work

The vehicular networks attract much attention by academia, industry, and even government in recent years. For example, Japanese government has promoted deployment of Intelligent Transportation Systems (ITS) and more than 1000 RSUs have been deployed mainly around a highway [10]. Those RSUs provide information service to passing vehicles using the 5.9 GHz Dedicated Short Range Communications (DSRC) spectrum. Currently, safety-related information is dominated in the service, however; it is promising to offer more choices including entertainment purposes such as online shopping and game. In industry, V2V communication testing has been already started and testbed systems have been developed by worldwide automakers, e.g., Toyota [11], Honda [12], General Motors [13], Volvo [14], and BMW [15].

In the vehicular networks, a large scale of vehicle nodes transmit/receive data packets simultaneously and continuously and data transmission inevitably suffers from the limited link bandwidth, intensive delay variance, and severe packet loss due to high mobility of the nodes. There also exist security and privacy concerns because data packets exchanging via V2V communication may include personal information of vehicular users. In addition, it is possible that vehicular location information (e.g., GPS) is improperly disclosed and used by any malicious user. To overcome the challenges, several efforts have been made in this research field [16]–[19]. [16] studied V2V communication efficiency and proposed a scheme to minimize the end-to-end delay while reducing the network traffic by using either contact-level or social-level scale of vehicular mobility. In [17], authors considered information gathering/dissemination in V2V communication and proposed a clustering approach where neighboring vehicle nodes make a group to efficiently sharing information within a cluster as well as between clusters. [19] studied location privacy issues in delay tolerant networks and proposed a cooperative location privacy protection scheme based on a game theory. [18] proposed a cooperative downloading strategy to maximize total throughput of user requests by utilizing both V2I and V2V communication, however; it did not take into account the delivery delay from a source node to a destination node. As mentioned previously, next-generation ITS systems offer many types of applications and some of them want to avoid any delivery delay occurred by vehicular communications, e.g., real-time streaming multimedia applications. In this paper, we consider cooperative downloading to minimize the delivery delay while maximizing downloading throughput from the RSU.

3 System Model

We consider a part of vehicular networks where an RSU is deployed in a straight road and vehicles pass through the front of the RSU in a single lane of the road at the same constant speed (e.g., highway scenario). We assume that a communication range of the RSU and the vehicles is the same and every node (RSU and vehicle) can communicate with each other only when entering its range. The RSU periodically broadcasts a beacon to let the vehicles know the downloading service available at the RSU. Upon receiving the beacon, a vehicle sends back a request when it has data to download from the RSU. Without loss of generality, we assume any contents requested by vehicular users are available at the RSU which obtains the contents in advance via the Internet by data prefetching methods [20]. The RSU can deal with the requests only one by one (i.e., unable to transfer data packets to multiple vehicles concurrently).

The IEEE 1609.4 standard for Wireless Access in Vehicular Environments (WAVE) has been proposed for multi-channel operations over DSRC spectrum to support safety-related as well as non-safety realated applications [21], [22]. We assume that every node has two wireless interfaces: control interface and data interface. The control interface is used for control packets and safety messages. The data interface is used for all other kinds of data packets and operates in different channels from one of the control interface, so that data traffic does not interfere the control channel. In our system model, we mainly focus on the data interface and consider separately the data packets and control packets (e.g., beacon message).

A data rate in V2I communication is mainly determined by a distance from a vehicle to the RSU [23] and the vehicle gains higher data rate when it locates closer to the RSU. According to a road segmentation model presented in [18], the road is divided into k segments denoted as $S_j(j = 1, 2, ..., k)$ and the length of each segment is denoted as $||S_j||$. The RSU is located in the center segment such as $S_{\lceil k/2 \rceil}$. Each segment is assigned corresponding data rate r_j and the vehicle is supposed to download data at r_j in S_j.

We assume each vehicle will download the data packets from the RSU with probability α (*download probability*). A vehicle is denoted as *busy vehicle* when it requests the RSU for data downloading otherwise *idle vehicle*. A probability that there are n busy vehicles in S_j is formulated as $\frac{(\alpha\rho||S_j||)^n}{n!}$ where $\rho = \lambda/v$ is the traffic density when the vehicle arrival rate is λ vehicles/sec. and the speed of the vehicles in the road is v. Then, the probability follows the normalized Poisson distribution such that [24]:

$$P_j^B(n) = \frac{(\alpha\rho||S_j||)^n/n!}{\sum_{i=0}^{C_j^B}(\alpha\rho||S_j||)^i/i!} \tag{1}$$

where C_j^B is the physically-possible maximal number of busy vehicles in S_j.

We define the downloading throughput and the delivery delay in this paper as follows.

- *Downloading throughput* is the amount of total data packets downloaded from the RSU. Our motivation of this study is how to fully utilize V2I and V2V communication for data downloading service in order to avoid access to expensive cellular networks. Thus, we aim at maximizing the average throughputs of all user requests.
- *Delivery delay* is the latency from the expected time to receive all the data packets (denoted as deadline) to the actual time to receive it (denoted as completion time). This metric is important when data downloading is necessary for realtime applications such as video streaming and online gaming service.

4 Cooperative Downloading and Flow Scheduling

In this section, we discuss cooperative downloading and flow scheduling. We study two strategies to maximize the throughput and minimize the delivery delay and then propose our algorithm MMCD which elegantly integrates both these two characteristics.

4.1 Maximizing the Downloading Throughput

According to the system model presented in section 3, higher data rate can be achieved when the vehicle is closer to the RSU. Leveraging this feature of the model, Cooperation-aided Max-Rate First (CMRF) has been proposed in [18], which greedily employs more vehicles located in road segments with higher data rate. First, the RSU orders requests from busy vehicles in order of decreasing the data rate and selects a request from a busy vehicle located in a road segment with the highest data rate of all. The RSU further searches idle vehicles which can be cooperators of the selected busy vehicle. The cooperators are supposed to locate within a communication range of the busy vehicle so that they can download data from the RSU and relay it to the busy vehicle later via V2V communication. Then, the RSU selects one of the cooperators if it is closer to the RSU than the busy vehicle (i.e., higher data rate is achieved). We only consider one hop relay from the cooperator to the destination of the busy vehicle and thus it does not relay the data to other vehicle nodes in multi-hop manner.

As assumed that all vehicles drive at constant speed, if any V2V connection is stable once connected, expected throughput with CMRF method is formulated as [18]:

$$\begin{aligned} T_C \;=\; & \frac{1}{1-P_{idle}}\left(\sum_{j=1}^{\lceil k/2\rceil} r_j^P P\{N_j^B + N_{k+1-j}^B > 0\}\right. \\ & \left.\cdot P\Big\{\sum_{i=j+1}^{k-j} N_i^B = 0\Big\} + r_{\lceil k/2\rceil} P\{N_{\lceil k/2\rceil}^B > 0\}\right) \end{aligned} \tag{2}$$

where N_j^B is a random variable representing the number of busy vehicles in S_j. r_j^P is data rate gained after involving the help of a cooperator in S_j so that $r_j \le r_j^P$. P_{idle} is the probability that there is no busy vehicles in S_j, given by:

$$P_{idle} = \prod_{j=1}^{k} P_j^B(0) \tag{3}$$

Equation 2 indicates that a vehicle at a road segment with lower data rate is selected as a cooperator only if there exists no other vehicle closer to the RSU. Also, the vehicle is always selected as a cooperator if it is at the middle of the road where the RSU is located in front, i.e., when the distance to the RSU is the shortest and the highest data rate can be gained.

4.2 Minimizing the Delivery Delay

Assume vehicle i can download all requested data when it is in S_j, the delivery delay of vehicle i's request is given by:

$$l_i^j = max(t_i + \frac{D_i}{r_j} - d_i, 0) \tag{4}$$

where t_i is time to start downloading data from the RSU while d_i is deadline of the request. Expected average delivery delay of all requests is formulated as:

$$L = \frac{1}{1-P_{idle}} \sum_{n_1=0}^{C_1} \cdots \sum_{n_k=0}^{C_k} \frac{\sum_{i=0}^{n_j}\sum_{j=1}^{k} l_i^j}{\sum_{j=1}^{k} n_j} \prod_{j=1}^{k} P_j^B(n_j) \tag{5}$$

Consider L in the worst case when the SPTF(CMRF) policy is applied. Assume the requests are ordered in decreasing order of deadline d_i (i.e., $d_1 > d_2 > \cdots > d_i$) while transmission time $\frac{D_i}{r_j}$ is in increasing order. SPTF policy is optimal for finding minimum total communication duration but not for finding minimum average delivery delay because it is highly possible to maximize delivery delay l_i^j of requests in latter of the queue.

To minimize the maximum delivery delay of each request, it is simple and effective to use a traditional method of *Earliest Due Date (EDD)* scheduling policy [25]. As following EDD policy, the RSU orders the requests in increasing order of their deadline and always serves the earliest one at first.

4.3 MMCD: Max-throughput and Min-delay Cooperative Downloading

We have introduced two existing methods: CMRF scheme and EDD scheduling policy, which can maximize the download throughput and minimize delivery delay, respectively. These methods are effective and simple, however; they have also disadvantages as follows.

First, CMRF scheme is not always effective to minimize the delay of each request. Since the RSU orders the requests by data rate, the requests are sequenced nearly according to their processing time, which is *Shortest Processing Time First (SPTF)* scheduling policy [25]. Here, the processing time means duration of communication between the RSU and each vehicle. The RSU always selects a vehicle in a road segment with the highest data rate so that V2I communication duration is expected to be shorter than selecting one with the lowest data rate. Note that the RSU does not always follow SPTF policy since the first selected request may take more processing time than the successive ones because of the size of data packets. For example, the communication duration can be $\frac{D_1}{r_{j_1}} > \frac{D_2}{r_{j_2}}$ if $D_1 >> D_2$ and $r_{j_1} > r_{j_2}$ where D_1 is the size of data packets and r_{j_1} is data rate for the first selected request and D_2 and r_{j_2} for the next selected request.

Second, EDD scheduling policy is unsuitable for maximizing the download throughput because the RSU will transfer data on the basis of round robin scheduling. The operation of the RSU is divided into time slots and each flow is assigned into a slot based on the deadline. When we assume that a duration of each time slot is Δt, the total duration of RSU operation is $\sum_{j=1}^{k}(n_j \Delta t)$ and the total amount of data transferred by the RSU is $\sum_{j=1}^{k}(n_j r_j \Delta t)$. Then, expected throughput with EDD policy is formulated as [18]:

$$T_T = \frac{1}{1-P_{idle}} \sum_{n_1=0}^{C_1} \cdots \sum_{n_k=0}^{C_k} \frac{\sum_{j=1}^{k} n_j r_j}{\sum_{j=1}^{k} n_j} \prod_{j=1}^{k} P_j^B(n_j) \quad (6)$$

where n_j denotes the number of busy vehicles in S_j. It is $T_C >> T_T$ verified by numerical analysis presented in [18].

Thus, we propose MMCD (Max-throughput and Min-delay Cooperative Downloading), a cooperative downloading algorithm in vehicular networks. The main idea of our algorithm is to take advantages of both strategies which maximizes the throughput by actively employing vehicles in a segment with higher data rate and minimizes delivery delay by giving a higher priority to a request which deadline is earlier, respectively.

More specifically, the RSU takes following four phases to schedule each flow and make effective cooperative downloading in drive-thru Internet.

1) The RSU orders the request of vehicular users based on their deadlines by following EDD scheduling policy.
2) After the RSU selects one busy vehicle with the earliest deadline, it further searches a cooperator who is located in a road segment closer to the RSU than the busy vehicle. (i.e., seek higher data rate)
3) If the RSU finds such a cooperator of the busy vehicle, it transfers the data packets to the cooperator within a certain time slot.
4) The RSU returns to the first phase and operates the new transmission (may serve the same vehicle at the previous step but transfer other data packets) at the next time slot.

The above phases of flow scheduling are summarized as shown in Algorithm 1.

Algorithm 1 Flow scheduling

```
if RSU receives a new message $msg_i(request, location, deadline)$ from busy vehicle $i$ then
    add $msg_i$ to a request queue and order it by EDD policy;
end if
select $msg_j$ of busy vehicle $j$ at the top of the queue;
$cooperators_j \leftarrow find\_cooperators(j)$
if $cooperators_j \neq NULL$ then
    $cooperator_j \leftarrow MIN(cooperators_j.distance)$
    // distance to the RSU
    $next\text{-}hop \leftarrow cooperator_j$;
else
    $next\text{-}hop \leftarrow j$;
end if
transfer data packets to $next\text{-}hop$ based on $msg_j.request$ within time slot $\Delta t$;
```

In V2V communication, the cooperator relays the data packets to the busy vehicle while both of them do not have ongoing communication with any other node. The cooperator can be a relay node for multiple destinations and manages a queue of data downloaded from the RSU. Data in the queue is also ordered by EDD scheduling policy so that the cooperator preferentially forwards data packets with the earliest deadline to a destination of the busy vehicle.

5 PERFORMANCE EVALUATION

We evaluate the performance of proposed MMCD by extensive experiments in simulator NetLogo [26]. The performance of MMCD is compared with a state-of-art cooperative downloading protocol, CMRF. Performance metrics are throughput and delivery delay as defined in Section 4, and *on-time arrival rate* which indicates how many requests are delivered to a final destination without any delay in all requests such as: $A = \frac{n_s}{N}$ where n_s is the number of requests delivered to each user until a deadline and N is the number of all the user requests. Since those performance metrics are highly affected by a network topology of vehicles, we create ten network examples for every experiment and derive an average of them as a final result.

5.1 Simulation Settings

We consider a highway scenario where there is a straight road with a single lane and vehicles go ahead until reaching at the right end of the road. The length of the road is 8000 m and an RSU is located at 1000 m away from the left end. 200 vehicles are injected from the left end by following a Poisson distribution with λ vehicles per second and the speed of the vehicles is 20 m/sec. The communication range of the RSU is 400 m and four data rates are used corresponding to the distance from the RSU to each road segment as shown in Table. 2. The communication range of a vehicle is also 400 m in V2I communication and 200 m in V2V communication because of each connection built in different communication modes: infrastructure mode and ad hoc mode, respectively. The connection in ad hoc mode is unstable because of high mobility of the vehicles. Thus, we assume the vehicles cannot directly communicate with each other when the distance between them is long as more than 200 m.

Every vehicle has data packets to download from the RSU with download probability α. The size and deadline of the data packets vary depending on each busy vehicle, which follow a Poisson distribution with mean size of 5 to 50 MB and with mean time of 10 sec., respectively. Note that we set a clock on each vehicle in this simulation and the clock starts ticking when the vehicle enters to communication range of the RSU. ”The deadline is 10 sec.” means the data requires to be transferred to a busy vehicle until its clock shows the time of 10 sec. We assume the RSU calculates the deadline of all data packets when it receives requests from each user. Main parameters are summarized in Table 1.

5.2 Simulation Results

5.2.1 *Download Probability*

We evaluate the performance of MMCD (our algorithm) and CMRF with two different traffic density: sparse and dense, while changing download probability α. In this set of experiments, we fix the mean size of data packets as 50 MB. Fig. 2 and Fig. 4 show the average throughput and delivery delay per user in sparse traffic ($\lambda = 0.1$), respectively. In Fig. 2, the average delivery delay slowly increases in MMCD, comparing to that of CMRF. Especially when the download probability is 0.9, MMCD successfully reduces the delay less than about 20% of the average delay in CMRF. On the other hand, the average throughput in MMCD gets slightly lower than CMRF when the download probability is more than 0.3 as shown in Fig. 4. This is because more vehicular users require to download data (more busy vehicles) when the download probability increases and they compete for access to the RSU. With CMRF strategy, the RSU always selects a vehicle with the highest data rate that maintains high download throughput on average although the number of access to the RSU is increased. However, MMCD gains only less than 8% of the average throughput in CMRF ($\alpha = 0.9$) and thus the result is still acceptable. We conclude that MMCD can reduce the delivery delay while gaining high enough throughput in the sparse traffic scenario where vehicles highly compete for access to the RSU.

TABLE 1
Main parameters used in the simulation

Parameters	Value
Length of a road (m)	8000
Number of vehicles	200
Speed of vehicle (m/sec.)	20
Communication range of RSU (m)	400
Communication range of vehicle (m)	200(V2V), 400(V2I)
Data rate (Mbps)	$r_j = \{3, 6, 12, 24\}$
Traffic density (vehicles/sec.)	$\lambda = \{0.1, 0.5\}$
Download probability	$\alpha = \{0.1, 0.2, ..., 0.9\}$
Mean size of data packets (MB)	$D = \{5, 10, ..., 50\}$
Mean time of deadline (sec.)	10

TABLE 2
Parameters of road segments

Road segment	S_1	S_2	S_3	S_4	S_5	S_6	S_7
Distance from RSU (m)	300	150	75	0	75	150	300
Length (m)	200	100	50	100	50	100	200
Data rate (Mbps)	3	6	12	24	12	6	3

We also consider a dense traffic scenario ($\lambda = 0.5$) and Fig. 5 and Fig. 3 show the average throughput and delivery delay per user in the scenario, respectively. The performance of both methods degrades more significantly according to the download probability because of higher contention of downloading. However, as can be seen in Fig. 3, MMCD greatly reduces the average delay comparing to CMRF when the download probability increases. On the other hand, in Fig. 5, MMCD’s performance gets worse according to the download probability. Meanwhile CMRF more greedily uses vehicles with higher data rate so that the high throughput is maintained, MMCD gives a priority to a request with a shorter deadline and it tends to be difficult for a vehicle to find cooperators which is idle as well as stays in a road segment with high data-rate when the download probability increases. We conclude that MMCD greatly reduces the delivery delay while maintaining minimal downloading throughput in the dense traffic scenario.

Fig. 6 shows the on-time arrival rate and compares the

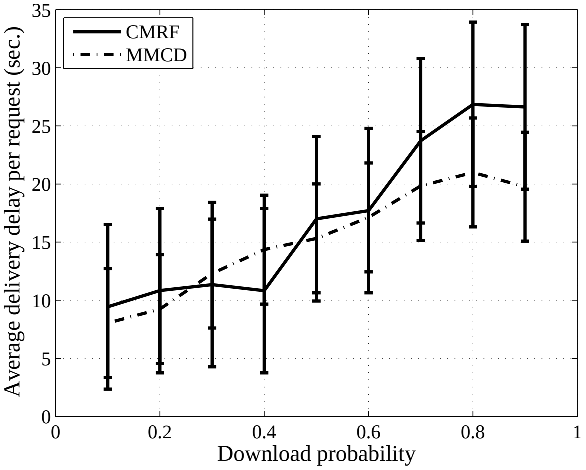


Fig. 2. Average delay vs. download probability in sparse traffic ($\lambda = 0.1$)

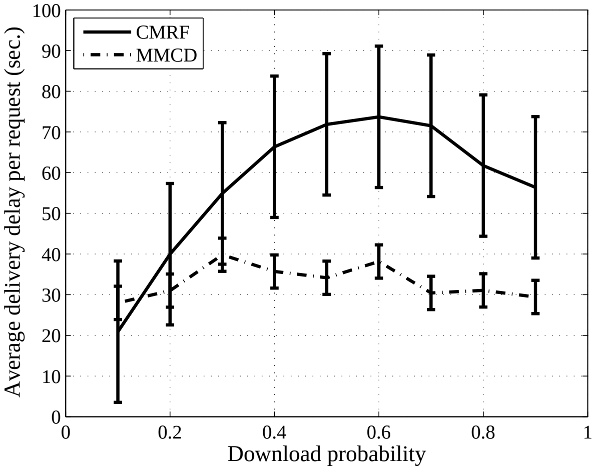


Fig. 3. Average delay vs. download probability in dense traffic ($\lambda = 0.5$)

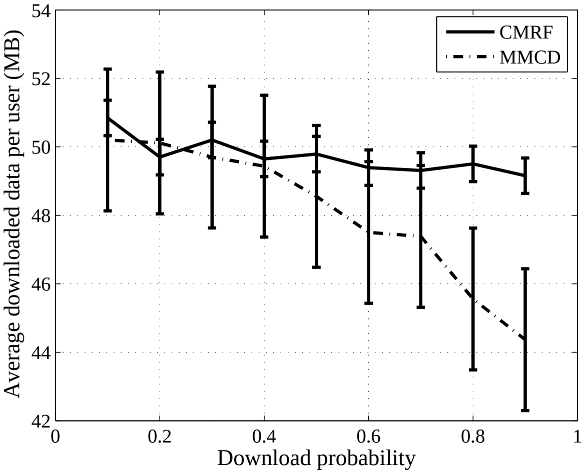


Fig. 4. Average throughput vs. download probability in sparse traffic ($\lambda = 0.1$)

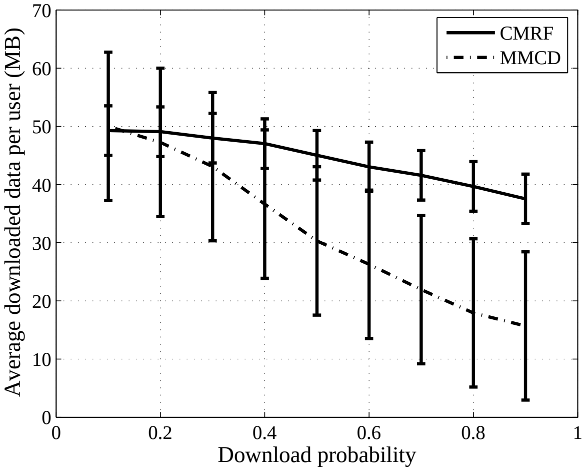


Fig. 5. Average throughput vs. download probability in dense traffic ($\lambda = 0.5$)

performance of CMRF and MMCD in the sparse and dense traffic scenario, respectively. Not many requests are delivered without any delay such that at most 16% of all requests and at most 10% of all requests are arrived on time when using CMRF and MMCD, respectively in the sparse traffic scenario. The on-time arrival rate becomes much lower in both CMRF and MMCD for the dense traffic scenario as shown in Fig. 6. This is because we consider that the mean size of data is 50 MB in this set of experiments. It is not small enough to complete downloading via drive-through Internet systems where the highest-data rate in a road segment is 24 Mbps as shown in Table 2. The impact of the data size on the downloading performance will be addressed in the next section 5.2.2.

It is particularly worth nothing that the on-time arrival rate of MMCD is lower than that of CMRF in the sparse traffic scenario especially when the download probability is more than 0.5, while the delivery delay of MMCD is less than that of CMRF as shown in Fig. 2. The results demonstrate that MMCD properly manages data packets of requests based on the deadline and optimizes the whole system by minimizing the average delivery delay. Meanwhile, CMRF achieves the higher arrival rate but the average delivery delay increases, that indicates CMRF can reduce the delivery delay for only some "randomly selected" requests by sacrificing any others which have to be waited for a longer period. It is not suitable for the system including requests from real-time applications where time is the most important factor.

5.2.2 *Mean size of data packets*

We examine the impact of the size of data packets on the performance of MMCD and CMRF. In this set of experiments, we fix the download probability as $\alpha = 0.9$ and change the mean size of data packets from 5 to 50 MB. Fig. 8 shows the mean delivery delay of the two methods in the sparse ($\lambda = 0.1$) and dense ($\lambda = 0.5$)

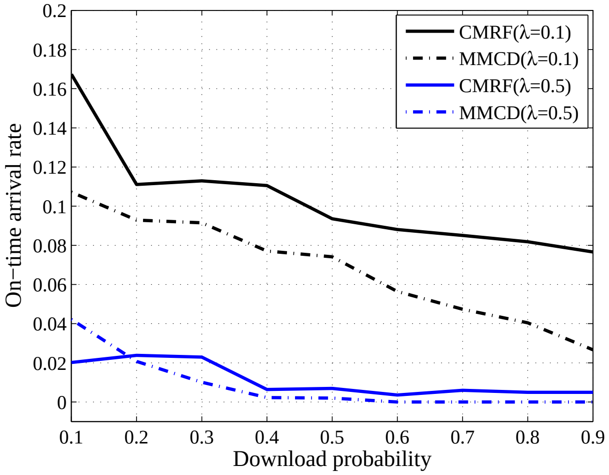


Fig. 6. On-time arrival rate vs. download probability when the mean size of data packets is 50 MB ($D = 50$)

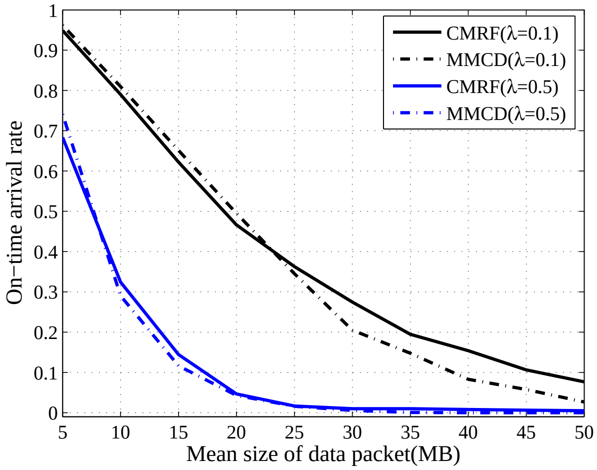


Fig. 7. On-time arrival rate vs. mean size of data packets when the download probability is 0.9 ($\alpha = 0.9$)

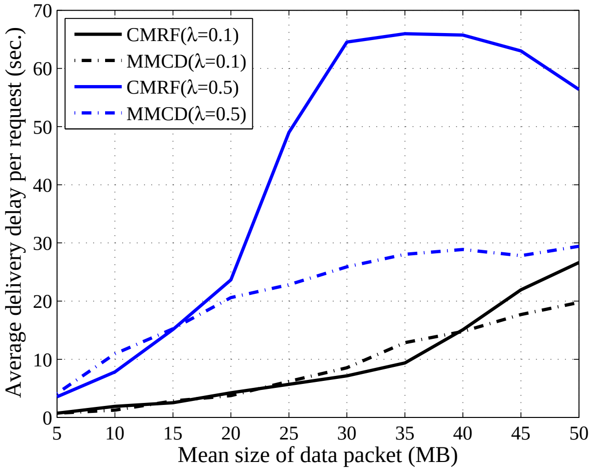


Fig. 8. Average delay vs. mean size of data packets when the download probability is 0.9 ($\alpha = 0.9$)

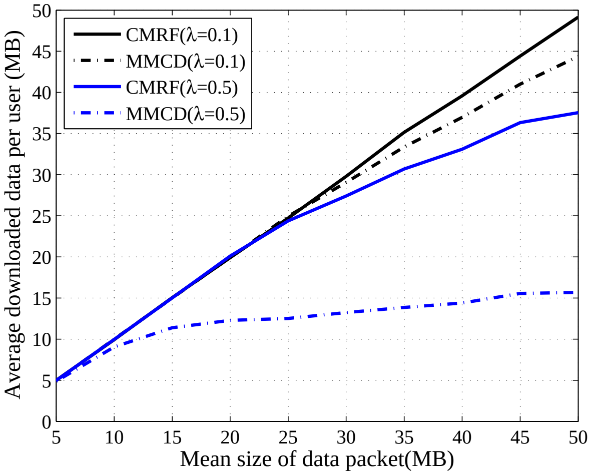


Fig. 9. Average throughput vs. mean size of data packets when the download probability is 0.9 ($\alpha = 0.9$)

traffic scenario, respectively. In the sparse traffic scenario, the delay of CMRF is lower than or equal to that of MMCD when the data size is small such as $D \leq 40$. This is because CMRF takes a shorter time to download packets by employing higher-data-rate vehicles than MMCD, that results in saving enough time to deliver the packets to a final destination until the deadline. However when the data size becomes larger, downloading itself consumes more time and download scheduling is necessary to meet a deadline of each request. MMCD functions adequately in regard to this point and thus outperforms CMRF when $D > 40$. Likewise in the dense traffic scenario, CMRF outperforms MMCD when the data size is relatively small because of the same reason. However when the data size becomes larger such as $D > 20$, the delay of CMRF drastically increases while that of MMCD slowly increases. The results demonstrate that MMCD can reduce the delivery delay even when the data size to be downloaded from the RSU is large in the dense traffic scenario.

Fig. 9 shows the mean throughput of MMCD and CMRF in the sparse and dense traffic scenario, respectively. In the sparse scenario, we can see little difference between the two methods. With results of the delivery delay as shown in Fig. 8, we conclude that MMCD works better when the mean size of data packets is large in the sparse traffic scenario. On the other hand, in the dense traffic scenario, the data size causes a gap between MMCD and CMRF. CMRF maintains the high throughput while MMCD degrades its performance with the increasing the data size. With results of the delivery delay as shown in Fig. 8, we conclude that the delivery delay and throughput have a trade-off relationship under the dense traffic scenario when the mean size of data packet is relatively large such as $D > 20$. In practical use, the RSU may restrict the data size to be downloaded by each user in order to satisfy a user's requirement for deadline. For example, the average delivery delay can be minimized when each user requests to download up to 20 MB data packets.

We also examine the impact of the data size on the on-time arrival rate and Fig. 7 shows the performance of CMRF and MMCD in the sparse and dense traffic scenario, respectively. In the both traffic scenarios, there is not a big difference between MMCD and CMRF and the on-time arrival rate is relatively higher when the mean size of data packets is smaller. It is noteworthy that almost 0% of requests is delivered on time when the data size is large such as $D > 30$ in the dense traffic scenario although CMRF gains the high average throughput as shown in Fig. 9. This indicates a capacity of data flow in vehicular networks is reached. Thus to reduce the delivery delay, one solution is to restrict the data size in the same way as MMCD. Another solution could be to use a technique for improving a network throughput (e.g., network coding [27]) and/or for removing redundant data packets if neighboring users request the same information (e.g., in-network processing [28]), so that both high throughput and low delay are achieved even in the dense traffic scenario.

6 Discussion

In the previous section, experimental results demonstrate that our proposed algorithm MMCD minimizes the average delivery delay per user while satisfying the average throughput as high as that of CMRF in the sparse traffic scenario. This is reasonable because a conventional highway road is not congested with traffic and MMCD works well especially when vehicular users highly compete for access to the RSU as well as have a large size of data packets to be downloaded.

On the other hand, we also find the trade-off between the delivery delay and downloading throughput under the dense traffic scenario. MMCD can highly reduce the average delivery delay, however while it sacrifices the average downloading throughput when compared to CMRF. This means that we may need more sophisticated integration between MMCD and CMRF to optimize the performance.

We show an example to solve the problem here. In the dense traffic, vehicles compete for access to the RSU especially when the download probability is high. When applying CMRF under such a situation, it is guaranteed to always select a vehicle closest to the RSU so that the downloading throughput keeps high. However when applying MMCD, the order of a request deadline is much prioritized that can reduce the delivery delay but results in lower throughput than CMRF. In order to take more advantages of CMRF, we relax a condition of EDD policy in the flow scheduling at the RSU. Algorithm 2 shows a modified algorithm of the flow scheduling at the RSU and its brief summary is as follows. When the RSU receives a request from a vehicle, it checks whether a request deadline is less than a threshold value. If so, the request is added to EDD queue which is ordered by EDD policy. If not, it is added to another queue, called Max-Rate-First (MRF) queue, which is ordered according to a distance from the RSU (i.e., shortest distance first). If EDD queue is not empty, the RSU selects a request with the earliest deadline from EDD queue; otherwise it selects a request with the highest data rate from MRF queue. All other procedures are the same with MMCD. We assume that the threshold value can be flexibly set by a system administrator according to traffic conditions, user demands, and etc.

Algorithm 2 Flow scheduling with a threshold value

```
if RSU receives a new message $msg_i(request, location, deadline)$ from busy vehicle $i$ then
    if $msg_i.deadline < threshold$ then
        add $msg_i$ to EDD queue and order it by EDD policy;
    else
        add $msg_i$ to MRF queue and order it by MRF policy;
    end if
end if
if EDD queue is empty then
    select $msg_j$ of busy vehicle $j$ at the top of MRF queue;
else
    select $msg_j$ of busy vehicle $j$ at the top of EDD queue;
end if
$cooperators_j \leftarrow find\_cooperators(j)$
if $cooperators_j \neq NULL$ then
    $cooperator_j \leftarrow MIN(cooperators_j.distance)$
    // distance to the RSU
    $next\text{-}hop \leftarrow cooperator_j$;
else
    $next\text{-}hop \leftarrow j$;
end if
transfer data packets to $next\text{-}hop$ based on $msg_j.request$ within time slot $\Delta t$;
```

We conduct simulation experiments for the modified algorithm where simulation settings are the same in section 5.2.1. We set the threshold value as five sec. which is a half period of the mean deadline of requests (see Table 1). Results are shown in Fig. 10 and Fig. 11. As we can see, modified MMCD reduces the delivery delay rather than CMRF while the throughput gains higher than original MMCD. This means that the performance of MMCD can be optimized by properly tuning parameter values according to traffic conditions, i.e., dense traffic scenario. Not only the traffic conditions, but also others such as QoE (Quality of Experience) would be

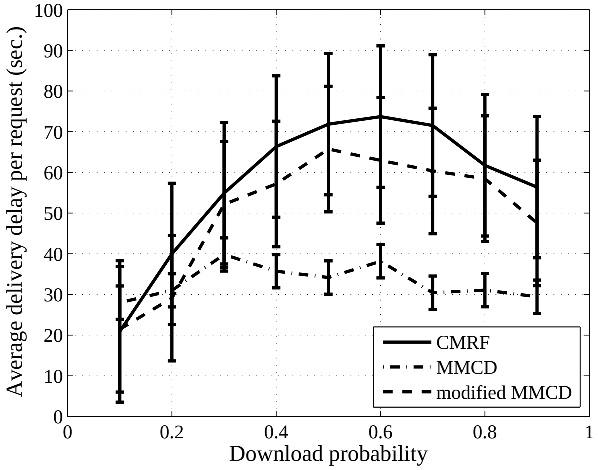


Fig. 10. Average delay vs. download probability in dense traffic ($\lambda = 0.5$)

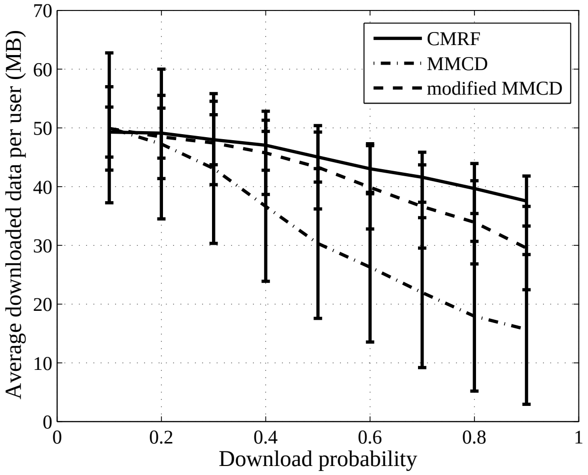


Fig. 11. Average throughput vs. download probability in dense traffic ($\lambda = 0.5$)

considered to create more satisfying user experiences. This is the first step in designing cooperative downloading system for highway VANETs and gives a clue to develop effective highway drive-thru Internet systems using cooperative V2V communication. As our future work, we will analyze the impact of several factors (e.g., traffic density pattern, vehicle mobility, type of service) on the performance and find characteristics and relationships between the factors and performance to develop a optimal solution.

7 Conclusion and Future Work

In this paper, we study cooperative downloading for drive-thru Internet systems using vehicular networks and propose an effective cooperative downloading algorithm called MMCD. It minimizes the average delivery delay of each request of vehicular users while maintaining the high downloading throughput in highway scenarios where vehicles highly compete for access to the RSU. The extensive simulations evaluate the performance of MMCD and show the efficiency of our algorithm by comparing to the performance of a state-of-the-art cooperative downloading algorithm in a sparse traffic scenario. We also find a trade-off relationship between the delivery delay and downloading throughput under the dense traffic scenario. We address how to obtain an optimal solution and give an initial clue to design an alternative based on MMCD.

For our future work, we will further verify the performance of MMCD by conducting simulations with real measurements of GPS traces [29]. Moreover, we will consider various kinds of scenarios including changeable traffic patterns, different vehicular speed, multiple lanes in the same road, various road patterns, and so on.

Acknowledgment

This work is partially supported by NSFC Grant No. 61450110085, the Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China (No. ICT1407), JSPS KAKENHI Grant Number 25880002, 26730056, and JSPS A3 Foresight Program.

References

[1] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil, "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions," *IEEE Communications Surveys Tutorials*, vol. 13, no. 4, pp. 584–616, Fourth 2011.

[2] J. Liu, H. Nishiyama, N. Kato, T. Kumagai, and A. Takahara, "Toward modeling ad hoc networks: current situation and future direction," *IEEE Wireless Communications*, vol. 20, no. 6, pp. 51–58, December 2013.

[3] J. Chen, X. Cao, Y. Zhang, W. Xu, and Y. Sun, "Measuring the performance of movement-assisted certificate revocation list distribution in vanet," *Wireless Communications and Mobile Computing*, vol. 11, no. 7, pp. 888–898, 2011. [Online]. Available: http://dx.doi.org/10.1002/wcm.858

[4] T. Luan, L. Cai, J. Chen, X. Shen, and F. Bai, "Engineering a distributed infrastructure for large-scale cost-effective content dissemination over urban vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 3, pp. 1419–1435, March 2014.

[5] U. Paul, A. Subramanian, M. Buddhikot, and S. Das, "Understanding traffic dynamics in cellular data networks," in *Proceedings of IEEE 2011 International Conference on Computer Communications (INFOCOM)*, 2011, pp. 882–890.

[6] NTT Docomo, Inc., "NTT Docomo Home," http://www.nttdocomo.co.jp/.

[7] Fido Solutions, "Mobile phones, plans and more reasons to love Fido — Fido.ca," http://www.fido.ca/.

[8] J. Parkkila and J. Porras, "Improving battery life and performance of mobile devices with cyber foraging," in *Proceedings of 2011 IEEE 22nd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, 2011, pp. 91–95.

[9] L. Gao, G. Iosifidis, J. Huang, and L. Tassiulas, "Economics of mobile data offloading," in *Proceedings of IEEE 2013 International Conference on Computer Communications (INFOCOM)*, 2013, pp. 3303–3308.

[10] Japanese Ministry of Land, Infrastructure, Transport and Tourism, "Japanese Ministry of Land, Infrastructure, Transport and Tourism, Road Bureau ITS Homepage," https://www.mlit.go.jp/road/ITS/.

[11] J. Fukuyama, "A delay time analysis for multi-hop v2v communications over a linear vanet," in *2009 IEEE Vehicular Networking Conference (VNC)*, Oct 2009, pp. 1–7.

[12] R. Miucic, Z. Popovic, and S. Mahmud, "Experimental characterization of dsrc signal strength drops," in *12th International IEEE Conference on Intelligent Transportation Systems, 2009 (ITSC '09)*, Oct 2009, pp. 1–5.

[13] General Motors Company, "General Motors — Emerging Vehicle Technology — GM.com," http://www.gm.com/vision/design_technology/emerging_technology.html.

[14] M. Nilsson, P. Hallbjorner, N. Araback, B. Bergqvist, and F. Tufvesson, "Multipath propagation simulator for v2x communication tests on cars," in *2013 7th European Conference on Antennas and Propagation (EuCAP)*, April 2013, pp. 1342–1346.

[15] T. Mangel, O. Klemp, and H. Hartenstein, "A validated 5.9 ghz non-line-of-sight path-loss and fading model for inter-vehicle communication," in *2011 11th International Conference on ITS Telecommunications (ITST)*, Aug 2011, pp. 75–80.

[16] H. Zhu, M. Dong, S. Chang, Y. Zhu, M. Li, and X. Shen, "Zoom: Scaling the mobility for fast opportunistic forwarding in vehicular networks," in *Proceedings of IEEE 2013 International Conference on Computer Communications (INFOCOM)*, 2013, pp. 2832–2840.

[17] F. Chiti, R. Fantacci, and G. Rigazzi, "A mobility driven joint clustering and relay selection for ieee 802.11p/wave vehicular networks," in *Proceedings of the IEEE International Conference on Communications (ICC 2014)*, 2014, pp. 348–353.

[18] S. Yang, C. K. Yeo, and B.-S. Lee, "Maxcd: Efficient multi-flow scheduling and cooperative downloading for improved highway drive-thru internet systems," *Computer Networks*, vol. 57, no. 8, pp. 1805–1820, 2013.

[19] S. Du, H. Zhu, X. Li, K. Ota, and M. Dong, "Mixzone in motion: Achieving dynamically cooperative location privacy protection in delay-tolerant networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4565–4575, Nov 2013.

[20] F. Malandrino, C. Casetti, C. Chiasserini, and M. Fiore, "Offloading cellular networks through its content download," in *Proceedings of 2012 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2012, pp. 263–271.

[21] Y. Bi, K.-H. Liu, L. X. Cai, X. Shen, and H. Zhao, "A multichannel token ring protocol for qos provisioning in inter-vehicle communications," *IEEE Transactions on Wireless Communications*, vol. 8, no. 11, pp. 5621–5631, Nov. 2009.

[22] Q. Wang, S. Leng, H. Fu, and Y. Zhang, "An ieee 802.11p-based multichannel mac scheme with channel coordination for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 449–458, June 2012.

[23] D. Hadaller, S. Keshav, T. Brecht, and S. Agarwal, "Vehicular opportunistic communication under the microscope," in *Proceedings of the 5th international conference on Mobile systems, applications and services (MobiSys)*. New York, NY, USA: ACM, 2007, pp. 206–219.

[24] W. L. Tan, W. C. Lau, O. Yue, and T. H. Hui, "Analytical models and performance evaluation of drive-thru internet systems," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 207–222, 2011.

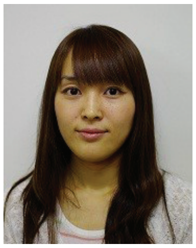
[25] J. Bazewicz and E. Al, *Scheduling Computer and Manufacturing Processes*, 2nd ed. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2001.

[26] S. Tisue and U. Wilensky, "Netlogo: A simple environment for modeling complexity," in *Proceedings of International Conference on Complex Systems*, 2004, pp. 16–21.

[27] M. Li, Z. Yang, and W. Lou, "Codeon: Cooperative popular content distribution for vehicular networks using symbol level network coding," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 223–235, January 2011.

[28] H. Paul, J. Fliege, and A. Dekorsy, "In-network-processing: Distributed consensus-based linear estimation," *IEEE Communications Letters*, vol. 17, no. 1, pp. 59–62, January 2013.

[29] K. Ota, M. Dong, H. Zhu, S. Chang, and X. Shen, "Traffic information prediction in urban vehicular networks: A correlation based approach," in *Proceedings of 2011 IEEE Wireless Communications and Networking Conference (WCNC)*, 2011, pp. 1021–1025.



Kaoru Ota received M.S. degree in Computer Science from Oklahoma State University, USA in 2008 and Ph.D. degree in Computer Science and Engineering from The University of Aizu, Japan in 2012. She is currently an Assistant Professor with Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan. From March 2010 to March 2011, she was a visiting scholar with BBCR group at University of Waterloo, Canada. Also she was a prestigious Japan Society of the Promotion of Science (JSPS) research fellow with Kato-Nishiyama Lab at Graduate School of Information Sciences at Tohoku University, Japan from April 2012 to April 2013. She has joined JSPS A3 foresight program as one of primary researchers since 2011 which is supported by Japanese, Chinese and Korean government. She serves as a Guest Editor of IEEE Wireless Communications, IEICE Transactions on Information and Systems, Editor of Peer-to-Peer Networking and Applications (Springer), International Journal of Embedded Systems (Inderscience) and Journal of Cyber-Physical Systems. Her research interests include wireless sensor networks, vehicular networks, and ubiquitous computing.

Mianxiong Dong received B.S., M.S. and Ph.D. in Computer Science and Engineering from The University of Aizu, Japan. He is currently an Assistant Professor with Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan. Before join Muroran-IT, he was a Researcher with National Institute of Information and Communications Technology (NICT), Japan. He was also a JSPS Research Fellow with The University of Aizu, Japan and was a visiting scholar with BBCR group at University of Waterloo, Canada supported by JSPS Excellent Young Researcher Overseas Visit Program from April 2010 to August 2011. Dr. Dong was selected as Foreigner Research Fellow (A total of 3 recipients all over Japan) by NEC C&C Foundation in 2011. He is the Best Paper Award winner of IEEE HPCC 2008, IEEE ICESS 2008 and ICA3PP 2014. Dr. Dong is currently a research scientist with A3 Foresight Program (2011-2016) funded by Japan Society for the Promotion of Sciences (JSPS), NSFC of China, and NRF of Korea. His research interests include wireless sensor networks, vehicular ad-hoc networks and wireless security.

Shan Chang was born in Xi'an, China, in 1982. She received the B.S. degree in computer science and technology from the Xi'an Jiaotong University in 2004 and the Ph.D. degree in computer software and theory from the Xi'an Jiaotong University in 2013. From 2009 to 2010, she was a visiting scholar with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. She was also a visiting scholar with BBCR research lab, Electrical and Computer Engineering Department, University of Waterloo from 2010 to 2011. Since 2013, she has been an assistant professor with the Department of Computer Science and Technology, Donghua University, Shanghai. Her research interests include security and privacy in mobile networks and wireless sensor networks. She is a member of the IEEE Computer, Communication and Society.

Hongzi Zhu received his PhD degree in computer science from Shanghai Jiao Tong University (SJTU) in 2009. Now he is an associate professor working at the Department of Computer Science and Engineering in SJTU. His research interests include vehicular networks, smartphone applications and mobile computing. He is a golden core member of the IEEE Computer Society and also a member of the IEEE Communication Society.