2011 2nd International Conference on Networking and Information Technology
IPCSIT vol.17 (2011) © (2011) IACSIT Press, Singapore

A Motion Prediction Based Cooperative Scheduling Scheme for Vehicle-Roadside Data Access

Yiqing Gui[1] and Edward Chan[2]

Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong
[1] yiqinggui2@student.cityu.edu.hk, [2] csedchan@cityu.edu.hk

Abstract—Timely data dissemination to vehicles is a core function in VANETs, and the use of Road Side Unit (RSU) has been proposed by researchers to augment vehicle to vehicle communication. RSU can serve as data buffer point from which vehicles can request to access data items. When scheduling vehicle-RSU data access, the mobility pattern of vehicles can be used to facilitate multi-RSU data dissemination. In this paper, we propose a motion prediction based scheduling scheme for vehicle-RSU data access which enables cooperative work among a set of road side units. Data requests can be transferred and thus balanced among a group of road side units, which helps to result in better scheduling performance. Simulation results show that our scheme outperforms simple scheduling schemes under various conditions.

Keywords-Vehicular Ad-Hoc Networks, Mobility Model, Road Side Unit, Scheduling, Simulation

1. Introduction

Vehicular Ad-Hoc Network (VANET) is receiving increasing research focus these days. Timely data dissemination to vehicles is a core function in VANETs, and the use of Road Side Unit (RSU) has been proposed by researchers to augment vehicle to vehicle communication. An RSU is typically installed at busy road junctions, serves as data buffer point so that vehicles can upload and download data when it passes through the coverage area of an RSU [1]. Location and time dependent data can be temporarily stored at certain RSUs, which can be useful to provide services including location-based advertisement, real-time traffic notification, digital map downloading and so on.

Vehicle to road-side communication has its own characteristics due to the fast moving speed of running vehicles. Requests in vehicle-roadside data access has strict time constrains in nature since it must get completely served before the vehicle leaves the transmission range of the RSU. An efficient scheduling scheme called *D*S/N* is proposed in [1] to address this challenge. However, only a single RSU is considered. In real life, a set of RSUs are required to enable effective VANET data dissemination [2]. Hence, it is useful to design a data access scheme which involves a group of RSUs.

In this paper, we propose a cooperative scheduling scheme called the Motion Prediction Optimization (MPO) scheme. In MPO, vehicle requests can be transferred among different RSUs based on certain conditions, which helps to balance work load between RSUs as well as to better utilize RSU bandwidth.

The rest of this paper is organized as follows: Section II summarizes related work. Section III provides background information including our system model and performance metric. The proposed MPO scheme is explained in detail in Section IV. Section V presents simulation results where performance of the MPO scheme and other three schemes are compared. Section VI explores how the MPO scheme performs when requests are generated unevenly among different regions. Section VII concludes the paper.

2. Related Work

Vehicle to road-side data access has received considerable attention in research [9], [10], [11].

The idea of the employment of specialized but simple and inexpensive RSUs is proposed in [2]. The authors found that if a small number of RSUs is installed in a city and interconnected via some backbone network, the dissemination performance improves dramatically for VANETs.

The placement scheme of RSUs is considered in [3]. An RSU placement scheme is designed to improve connectivity for a given number of RSUs on the road network of a Korean city.

An efficient scheduling scheme specially tailored for vehicle-roadside data access involving one RSU and many vehicles is proposed in [1]. In this model, vehicles listen to the communication channel once they enter the transmission range of an RSU. Vehicles can request to either upload data or download data and both operations compete for the same bandwidth. Download broadcasting is made use of to yield better scheduling performance. However, only a single RSU is considered in this work.

3. Background

3.1 System Model

We model part of a city area by a 19 by 19 square road grid (Fig. 1). Each road contains two lanes and vehicles are allowed to move in either direction. We assume that the distance between each two roads is D meters. RSU servers are placed at road junctions once every other roads (black spots). So there are totally a hundred RSUs serving vehicles passing by. The transmission radii of RSU servers are set to be less than D meters. As a result, there exists some area in the grid which is not covered by the transmission range of any RSU.

Vehicles listen to the transmission channel when they enter the transmission range of an RSU. We assume that a vehicle generates request only when it is within the transmission range of a certain RSU. When driving through regions that are not covered by any RSU, no requests are generated by a vehicle. Request can either be a downloading one or uploading one and each contains a deadline. We also assume that the RSUs serve vehicle request non-preemptively.

The movement pattern of vehicles within the square region follows the *Manhattan Mobility Model* [4].

During each scheduling round, the RSU will scan through the whole set of waiting requests and picks the one with the highest priority to serve based on the scheduling scheme. Besides, requests that are not likely to be served before their deadline will either be deleted or transferred depending on certain conditions. These conditions may involve the work load of nearby RSUs, the velocity of the vehicle that generates the request, as well as the request deadline. The details will be specified in the next section in which our algorithm is discussed. Each RSU will try to transfer the transferrable request to one of the nearby RSUs based on the motion prediction of the vehicle. Motion prediction is based on the probabilistic mobility nature of the *Manhattan Mobility Model*. If the motion prediction is correct, the transferred request will have a chance to be served after its issuing vehicle notified its arrival to the RSU where the request is transferred to. A transferred request will be considered in each scheduling round only after its issuing vehicle notifies its arrival to the RSU where the request is transferred to. We also assume that no transfer overhead will be incurred, and requests can be transferred instantly among RSUs with no delay. By employing such a cooperative scheduling scheme, work load can be balanced if certain RSU server suffers from overloading and RSU bandwidth can be better utilized. As a result, there is expected to be increase in overall scheduling performance.

We also set some real life deadlines for each request rather than assuming it to be the time when the issuing vehicle leaves the RSU transmission range.

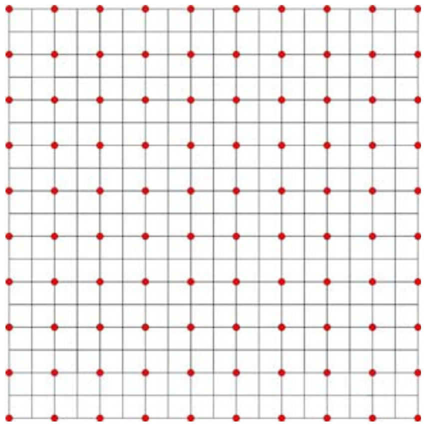

Figure 1. System Model

A non-transferred request is considered successfully served if it is completely served on or before its deadline. For a transferred request to be considered successfully served, the following two conditions need to hold. Firstly, the motion prediction should be correct, say the vehicle should enter the transmission range of the RSU where its request is transferred to. Secondly, the request must be completely served within the time interval from the time the issuing vehicle enters the transmission range of this RSU to the time it leaves.

3.2 Performance Metric

We will use *service ratio* as our performance metric. It is defined as the ratio of the total number of successfully served requests to the total number of requests that are generated by all the vehicles. Obviously, the higher the service ratio, the better the algorithm performs.

4. Motion Prediction Optimization Scheme

To make use of the mobility pattern in the *Manhattan Mobility Model*, we proposed the MPO scheme, which enables cooperative work among a set of inter-connected RSUs. The MPO scheme is based on the *D*S/N* scheme proposed in [1].

Before the formal description of these two algorithms, we define some notations and terminologies first.

4.1 Notations and Terminology

1) *request:* A request is defined as a 9-tuple: <data_size, x_cor, y_cor, generation_time, valid_time, deadline, direction, duration, finish_time>, where

a) *data_size:* the size of the data item that is requested by the vehicle

b) *x_cor, y_cor:* the coordinates on the map where the request is genereted. The lower left corner of the square region is set to be the origin.

c) *generation_time:* the time at which the request is generated by the vehicle

d) *valid_time:* the period of time after the generation of the request, within which the request is valid

e) *deadline:* the deadline of the request, which is equal to *generation_time* plus *valid_time*

f) *direction:* the moving direction of the vehicle generating the request. The direction can be one of the following four values: North(N), South(S), West(W) and East(E).

g) *duration:* the total service time of a request, which is equal to *data_size* devided by the bandwidth of the RSU server

h) *finish_time:* the RSU will sort the request set in the waiting queue based on the scheduling scheme. For the first request in the waiting queue after sorting, the finish_time is equal to its *duration*. The finish_time of the second request is defined as the finish_time of the first request plus the *duration* of the second request. The finish_time of the following requests can be calculated by this recurrence relation. It can be seen from the definition that the finish_time of the request gives an appriximation by what time the request will be completely served at the RSU under the current workload.

2) *qualified request:* A request is said to be qualified if the *finish_time* of this request is less than or equal to its *deadline*. If not, the request is said to be *unqualified*. Qualified requests are likely ro be served completely on or before their *deadline* because their *finish_time* is earlier than *deadline*. In other words, they

will be successfully served according to their current positions in the waiting queue of the RSU where they are queuing. Requests that are unqualified are likely to miss their *deadline* since they are scheduled to be served in a late time.

3) RSU_available_time: it is the biggest *finish_time* of all the *qualified* requests in the waiting queue of an RSU.

4) RSU_x: the x coordinate of the RSU in the map

5) RSU_y: the y coordinate of the RSU in the map

4.2 D*S/N Scheme

In the *D*S/N* scheme, each request is given a value called the *DSN_value* defined as:

$$DSN\_value = \frac{(deadline - current\_time) \times data\_size}{Number}$$

The *Number* attribute is equal to the number of downloading requests that request for the same data item if the request is a downloading one. For uploading requests, this attribute is always 1.

The *D*S/N* algorithm computes *DSN_value* for each request in the waiting queue. It sorts the requests by their *DSN*_value and the request with the minimum *DSN_value* is chosen to be served first. Requests that miss their deadlines are discarded during each scheduling round. This scheme takes both *data_size* and *deadline* into consideration. It also takes advantage of download broadcasting, where a single broadcast can serve the whole set of downloading requests that request for the same data item.

4.3 MPO Scheme

The MPO scheme is based on the *D*S/N* scheme, in which the RSU scans through the whole set of the requests and choose the qualified request with the smallest *DSN_value* to serve first during each scheduling round. What's different is that requests that are likely to miss their deadline are attempted to be transferred in MPO. The RSU will transfer *unqualified* requests to nearby RSUs based on the motion prediction of their issuing vehicles if the situation falls in one of the following scenarios.

Scenario A: $x\_cor \leq RSU0\_x$ for a given request R

The *unqualified* request is generated by a vehicle moving towards east within the transmission range of RSU0 and to the left of RSU0 (Fig. 2). In this case, MPO will check the *RSU_available_time* of RSU2 first since the vehicle will have the highest probability to move in this direction under the *Manhattan Mobility Model*. If the predicate *RSU2_available_time* + *duration* ≤ *deadline* holds, then this request will be transferred to RSU2. We choose this predicate to ensure that the request *deadline* is relative long so that there will be a smaller probability for the *deadline* to be

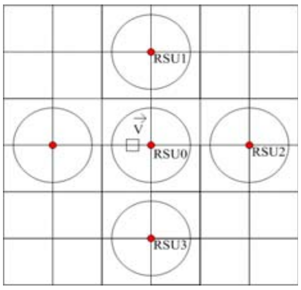

Figure 2. Scenario A

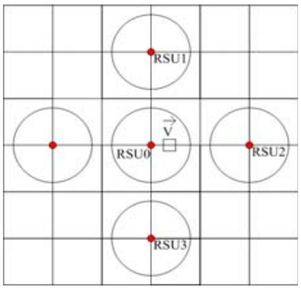

Figure 3. Scenario B

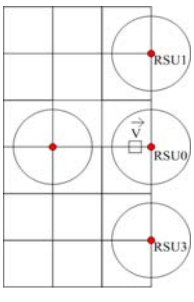

Figure 4. Scenario C

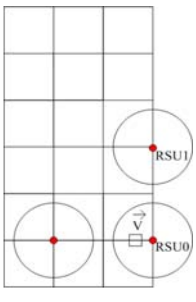

Figure 5. Scenario D

missed even before the issuing vehicle arrive at RSU 2. This predicate also helps to reduce the probability that the transferred request will cause overloading problem for RSU 2 since the request can be successfully served even if it is queued after the last *qualified* request in the waiting queue according to the current work load of RSU2.

If this inequality does not hold, the *RSU_available_time* of RSU1 and RSU3 will be checked. The request will be transferred to one of these RSUs if the above inequality holds for either of them. If both of the RSU1 and RSU3 satisfy the condition, MPO will randomly pick one of them. If none of them satisfies the inequality, the request will be discarded.

Scenario B: x_cor > RSU0_x for a given request R

The unqualified request is generated by a vehicle moving towards east within the transmission range of RSU0 and to the right of RSU0 (Fig. 3). In this case, MPO will check the *RSU_available_time* of RSU2 only. If the predicate *RSU2_available_time* + *duration* ≤ *deadline* holds, then this request will be transferred to RSU2. If not, the request will be discarded.

Scenario C: RSU is on the edge for a given request R

The *unqualified* request is generated by a vehicle moving towards east within the transmission range of RSU0 and RSU0 is on the edge (but not on the corner) (Fig. 4). In this case, MPO will check the *RSU_available_time* of both RSU1 and RSU3. The request will be transferred to one of these RSUs if the same inequality defined in the above scenario holds for either of them. If none of them satisfies the inequality, the request will be discarded.

Scenario D: RSU is on the corner for a given request R

The *unqualified* request is generated by a vehicle moving towards east within the transmission range of RSU0 and RSU0 is on the corner (Fig. 5). In this case, MPO will check the *RSU_available_time* of RSU1 only. If the predicate *RSU1_available_time* + *duration* ≤ *deadline* holds, then this request will be transferred to RSU1. If not, the request will be discarded.

Finally, if none of the above four cases are met, the request will be deleted.

Note that in each of the above four scenarios, the corresponding north, south and west cases can be derived similarly by symmetry.

A transferred request is stored in a separate buffer at the target RSU, and it will be moved to the waiting queue of the RSU only when its issuing vehicle arrives at the transmission range of the target RSU.

To address the problem of motion prediction failure, where the transferred request will be stored in the RSU buffer forever, the transferred request will be deleted from the buffer when its *deadline* is missed.

5. Performance Evaluation

5.1 Experiment Setup

We have built a C-SIM [5] based simulator to evaluate the performance of our proposed scheduling scheme and three other schemes including the First Come First Serve (FCFS) scheme, the Smallest Datasize First (SDF) scheme and the Earliest Deadline First (EDF) scheme.

The distance between two roads is set to be 400 meters. So the whole simulation area becomes a 7200m by 7200m square grid with 100 RSUs installed at road junctions once every other roads.

Vehicle movement pattern follows the *Manhattan Mobility Model* with the inter-vehicle and intra-vehicle relationship defined below:

1) $V(t+1) = V(t) + \text{Normal Dist.} (\mu, \sigma)$, where $V(t)$ is the velocity of one vehicle at time slot t

2) The distance between two vehicles running on the same street in the same direction is at least *Safe Distance* meters.

Vehicles can generate either downloading or uploading requests during the simulation. The request generation time follows the exponential distribution with mean value *T_request* seconds for each node. The data access pattern is based on the *Zipf* distribution [6], which is commonly used to model data access pattern in

Mobile Ad Hoc Networks [1][7]. In *Zipf* distribution, the probability of accessing the i[th] data item is represented as:

$$P(i) = \frac{1}{i^{\theta} \sum_{k=1}^{n} 1/k^{\theta}}$$, where n is the number of data items

when $\theta = 1$, it is strictly *Zipf* distribution. When $\theta = 0$, it becomes uniform distribution. The θ value is chosen to be 0.8 in our model according to studies on real web trace [8].

Request valid time also follows the normal distribution.

Each server has its own waiting queue. They run scheduling algorithm to serve vehicle requests and transfer work load among them during the simulation. Each RSU server has access to the full set of data items. We assume that the data set which can be accessed by vehicles remains fixed. Vehicles can only upload and download data item from a fixed list of data items that are stored in the RSU. They are not allowed to add new entries into this data set.

We also assume no background load for servers.

Simulation parameters and their default values are listed in Table 1.

TABLE I. SIMULATION PARAMETERS

Parameter	Default Value
Simulation Time	900s
RSU Bandwidth	625 KB/s
Vehicle Velocuty	V(0) = 15m/s, μ = 0, σ = 0.03
Max. Vehicle Speed	20m/s
Min. Vehicle Speed	12m/s
RSU Trans. Radius	300m
Data Size	50K ~ 5M
Data Set Size	25
Zipf Parameter theta	0.8
Inter-arrival Time between Requests	Mean = 5s
Request Valid Time	μ = 15s, σ = 2
Number of Vehicles in the Simulation	80
Safety Distance	2m

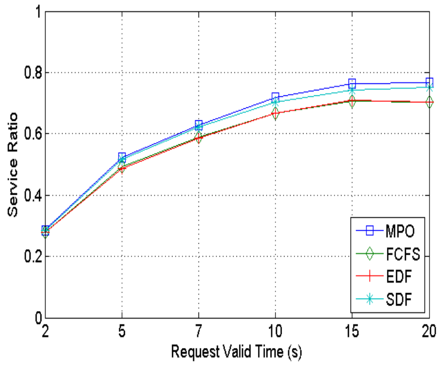

Figure 6. Effect of Request Valid Time

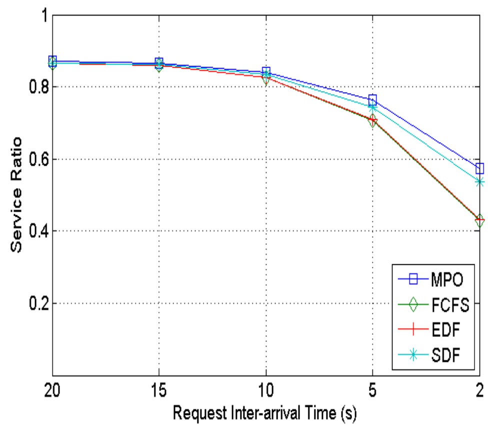

Figure 7. Effect of Request Inter-arrival Time

5.2 The Effect of Request Valid Time

It is shown in Fig. 6 that in general, all four algorithms performs better when request valid time increases since it is more likely for each request to be served before its *deadline* when it is valid for a longer time. And the MPO scheme outperforms other three scheduling schemes under various request valid time as is expected.

5.3 The Effect of Request Inter-arrival Time

When request inter-arrival time decreases, more requests will be generated within a certain period of time, which results in heavier RSU work load. We can see from Fig. 7 that the performance of the MPO scheme degrades more slowly than other three schemes when RSU work load increases. This result shows that work load transfer enables the MPO scheme to serve more requests within a certain period of time.

6. Hotspot Effect

When driving in downtown area, people may be attracted by the breaking news shown on the huge screen at busy cross roads. Many of them may try to learn more about the news by accessing data through nearby RSUs. Similarly, when there is a big jam on the road, many drivers will try to know what has happened and will try to find out which way to go at the next crossroads to get out of this jam by accessing real time traffic data via RSUs along the road.

As a result, vehicles may generate many more requests within a specific region than they will in other areas. We call these specific regions hotspots.

Hotspots are common in real life and in this section we evaluate the performance of the MPO scheme under the effect of hotspots. The major idea underlies the MPO scheme is the motion prediction based work transfer. Under this optimization, requests that are *unqualified* in one RSU may have a chance to be served successfully by nearby RSUs.

Under hotspot effect, the MPO scheme is expected to perform even better than other three schemes. With the presence of hotspots, RSUs will have a much bigger difference in work load so that overloaded and non-overloaded RSUs will scatter in the region. Work transfer will have a higher chance to be successful since it is more likely to find non-overloaded RSUs around overloaded RSUs. But the other three schemes cannot benefit from this change.

To verify this, we modified our simulation model in the last section. We select some of the RSUs to be hotspot RSUs and we set the request inter-arrival time within the region that is 100m from hotspot RSUs to be much lower than that of other regions. So when vehicles move across hotspot RSUs, they will generate a lot more requests compared to non-hotspot regions.

We examine the performance of the MPO scheme and compare it with EDF, FCFS, SDF under two different location distribution of hotspot RSUs, the line distribution and two cluster distribution (Fig. 8 and 9). Hotspot RSUs are shaded.

Simulation parameters are the same as in Table 1 except that the inter-arrival time between requests in non-hotspot region is set to be 50s but only 0.5s for hotspot region.

As shown in Fig. 10 and 11, MPO performs the best in both two location distribution of hotspot RSUs. The performance difference between MPO and other three schemes is larger compared to Fig 6. as expected. Hence, the MPO does perform even better under the hotspot effect.

Comparing Fig. 10 and 11, it is interesting to see that the performance gap is larger when hotspots RSUs are distributed in line shape. This can be explained by the ratio be

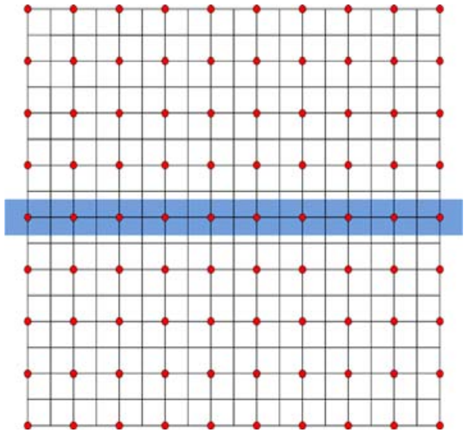

Figure 8. Line Distribution of Hotspot RSUs

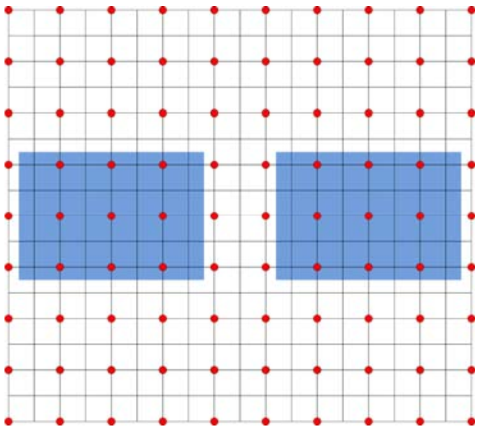

Figure 9. Two Cluster Distribution of Hotspot RSUs

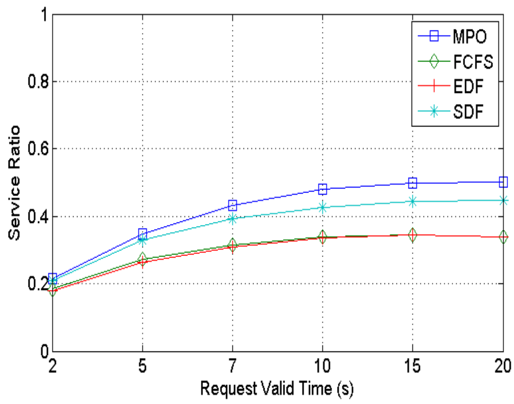

Figure 10. Simulation Result of the Line Distribution

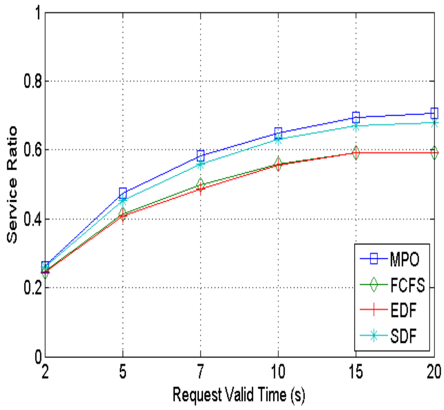

Figure 11. Simulation Result of the Two Cluster Distribution

tween the total number of non-hotspots RSUs which are next to hotspot RSUs to the total number of hotspot RSUs. The bigger the ratio, the more likely the work transfer will be successful since the chance for finding a nearby non-overloaded RSU will be higher. As mentioned before, MPO leverages on successful work transfer to achieve good performance. Hence, the performance gap will increase as the ratio increases.

In our case, the line distribution has a ratio of 2 (20/10) while in the two cluster distribution, the ratio is only 1.33 (24/18). This explains the wider performance gap in Fig. 10.

7. Conclusion

In this paper, we proposed a cooperative scheme for scheduling data access in VANETs based on the motion prediction of vehicles. Vehicle mobility is used to predict future vehicle position, which enables more accurate work load transfer among RSUs. Request transfer enables more requests to be successfully served, which helps to balance RSU workload and better utilize the RSU bandwidth. Our proposed scheme performs even better under hotspots effect where requests are generated unevenly among different regions, which is the typical case in real life.

8. References

[1] Y. Zhang, J. Zhao and G. Cao, "On scheduling vehicle-roadside data access", in *Proc. of 4th ACM International Workshop on Vehicular Ad Hoc Networks (VANET '07)*, 2007, pp. 9-18

[2] C. Lochert, B. Scheuemann, M. Caliskan and M. Mauve, "The feasibility of information dissemination in vehicular ad-hoc networks", in *Proc. of Wireless on Demand Network Systems and Services (WONS'07)*, 2007, pp. 92-99

[3] J. Lee and C. M. Kim, "A roadside unit placement scheme for vehicular telematics networks", in *Proc. of AST/UCMA/ISA/CAN*, 2010, pp. 196-202

[4] F. Bai, N. Sadagopan and A. Helmy, "IMPORTANT: a framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks", in *Proc. of IEEE INFOCOM*, 2003, vol.2, pp. 825 – 835

[5] CSIM Simulation Library from *Mesquite Software*: http://www.mesquite.com/

[6] G. Zipf, "Human behavior and the principle of least effort," *Addison-Wesley*, 1949

[7] L. Yin and G. Gao, "Supporting cooperative caching in ad hoc networks", *IEEE Transaction on Mobile Computing*, Jan. 2006, vol. 5, issue 1, pp. 77-89

[8] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and *Zipf*-like distributions: evidence and implications," in *Proc. of IEEE INFOCOM*, 1999, vol. 1, pp. 126-134

[9] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan and S. Madden, "A measurement study of vehicular internet access using in situ Wi-Fi networks", in *Proc. of 12th International Conference on Mobile Computing and Networking (MobiCom '06)*, 2006, pp. 50-61

[10] K. Liu and V. C. S. Lee, "RSU-based real-time data access in dynamic vehicular networks", in *Proc. of 13th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2010, pp. 1051-1056

[11] C.L. Chang and J. R. Luo, "The scheduling algorithm in road side unit for vehicular ad hoc applications", in *Proc. of Computer Symposium (ICS)*, 2010, pp. 85-90