Simulation Study of On-Demand Scheduling Algorithms in Road Side Units (RSUs)-Based Vehicular Ad Hoc Networks (VANETs)

Syeda Khairunnesa Samantha and Nusrat Nur Afrose

***Abstract*—Facilitation of data dissemination in Vehicular Ad Hoc Networks (VANETs) demands for a robust scheduler in this regard. High mobility of vehicles exhibits less connectivity problem establishing the idea ofRoad Side Units (RSUs) as a buffer point in VANETs. In this paper,firstly we formulate a RSU-based dynamic VANETs environment by specifying the constraints and secondly apply different on-demand scheduling algorithms in this environment. We have executed a series of simulations and analyzedthe performance of different on-demand scheduling algorithms against different performance metrics having high workload and tight time constraints. Finally, we suggest which on-demand scheduling algorithmis most adaptablein this highly mobile and sparsely connected network environment based on our experimental result.**

***Index Terms*—On-demand scheduling algorithms, road side units (RSUs), vehicular ad hoc networks (VANETs) etc.**

I. Introduction

Researchers have given meticulous attention to a lot of applications (road safety, internet access, entertainment etc.) in Vehicular Ad Hoc Networks (VANETs) [1]-[3]. Efficient data dissemination mechanism is a key challenge to provide successful VANETs applications. In VANETs, usually vehicles move pretty fast leading to short vehicle to vehicle connectivity time; moreover in the case of VANET roll out phase (when vehicles density is low, night time/off-peak hour, highways etc.), there is very little chance to get the required information from other vehicles. Hence, installing RSU at the important places in a planned way [4] and get responses from it is an important consideration in this environment.

RSU is a stationary substance unit having wireless access point (Dedicated Short Range Communication (DSRC) [1]), memory storage and computational capabilities. As RSU transmission range is short and vehicles are always on the move, request and response time is brief as well. To achieve better performance in this circumstance, an RSU needs to provide services to the vehicles so that it can balance among minimum deadline miss rate, high throughput and minimum response time.

Manuscript received July 12, 2012; revised August 16, 2012.

Syeda Khairunnesa Samantha is with the Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh (e-mail: syeda.samantha@ yahoo.com).

Nusrat Nur Afrose is with the Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh (e-mail: nusratnurafroseshoma @ yahoo.com).

Unlike unicasting, broadcasting is an efficient approach in here as many vehicles' requests can be served by this. Broadcasting can be done in two ways: 1) Periodic broadcasting and 2) On-demand broadcasting. Periodic broadcasting is not scalable for handling large database [5], [6]. Moreover, client access patterns are not same all the time. So, on-demand broadcasting is more suitable than periodic broadcasting in VANETs. Our major contributions in this work are:

- We formulate a RSUs based VANETs environment and constraints.
- We apply different on-demand scheduling algorithms, compare and analyze the experimental results and finally, based on these results we recommend which algorithm is the best suited in this environment.

The rest of the paper organized as follows. Section 2 describes related work; section 3 shows our VANETs system model, section 4 describes used on-demand scheduling algorithms, section 5 and 6 exhibits simulation model and experimental results respectively. We finish by a discussion and stating our future work.

II. Related Work

Many efforts have been carried out to find an efficient data dissemination procedure. Due to the high mobility of vehicles which is a unique characteristic of VANETs, many researchers try to adopt different techniques for finding a stable data dissemination procedure. Chen *et al.* [7] propose messages relayed technique where data is stored at the moving vehicles until favorable data delivering opportunities come. MDDV [8] also uses the intermediate nodes to buffer the data and carry it until any of the forwarding approaches (opportunistic, trajectory based and geographical forwarding) is encountered by the environment. VADD [9] uses the store and forward procedure and to reduce the data delivery delay it considers predictable traffic pattern and road layout. In DP-IB [10] technique data are periodically broadcasted to vehicles; vehicles buffer that data and rebroadcast it at the intersections. T. Nadeem*et al.* [11] propose periodic broadcast approach to disseminate both generated and relayed data. Lochert*et al.* [4] recommend few wired connected RSUs to provide better data dissemination than many standalone RSUs. Zhang *et al.* [1] propose a Two-step scheduling to balance the upload and download services.

In this paper, we use on-demand broadcast which is also called pulled based strategy to provide opportunities to the vehicles making their demand and then serve them from RSU database using on-demand scheduling algorithms. To implement these algorithms in VANETs, the main challenge from the traditional database on-demand scheduling is to maintain the strict time constraint for vehicles high mobility. Because, after a vehicle lefts the RSU transmission range, broadcast becomes meaningless. Considering this issue, we apply those on-demand algorithms in here aiming to find a stable one in this highly sparse and frequently disconnected environment with experimental results and analysis.

III. System Model

A. System Architecture

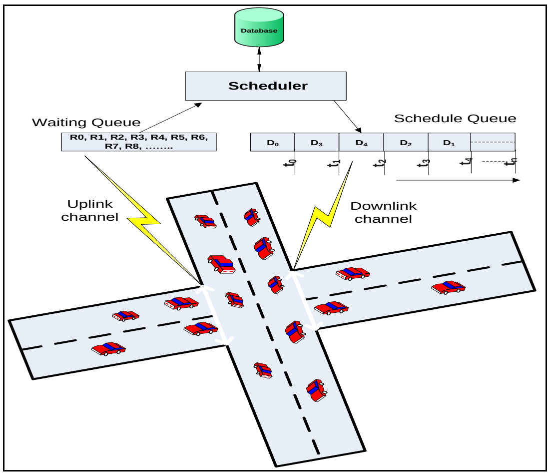


Fig. 1.The system architecture.

Our system model is similar to Fig. 1. We assume RSU server already has updated data item. When a vehicle is in the transmission range of a RSU, it can generate requests for updated data item through the uplink channel. Requests will be queued in the waiting queue for being serviced. The scheduler will decide the most appropriate data item for broadcast according to the vehicles' requests in the waiting queue. Scheduler broadcasting decision fully depends on the used underlying scheduling algorithm. Many vehicles can generate requests for same data item and a vehicle can send requests and get responses until it passes the transmission range of a RSU.

B. Notations and Assumptions

When a vehicle sends a request to a RSU, it sends the following information with that request.

Request R_i = {V_{ID}, R_{ID}, D_{ID}, $R_{Deadline}$}, where V_{ID} and R_{ID} are vehicle and request identifier respectively. D_{ID} is the data item identifier that is requested by the request R_i and $R_{Deadline}$is the deadline assigned by the vehicle, when this time will be expired that request will be dropped.

Suppose at time t, we have n requests in the waiting queue, then

Definition 1. The un-ordered request set, R = {$R_i(D_i)$, $R_j(D_j)$, …….. , $R_n(D_n)$}, which means the request $R_i$requests *i* data item, request R_j, *j* data item and so on.

*Definition 2.*The set of deadline of n requests is,

Deadline = $Dead_{R_1}, Dead_{R_2}, \dots \dots \dots, Dead_{R_n}$.

If any n requests ask for the same data item *D*, only request *R* which has minimum deadline value will be considered for making broadcast decision, i.e. $R = \{R_i \,|\forall i \neq j, R_i \in R \;\&\; D_i = D_j \,\&\min(Dead_{R_i}, \dots, Dead_{R_j})\}$.

*Definition 3.*The set of popularity of data item in the database is, Popularity = {$Popu_{D_1}, Popu_{D_1}, \dots\dots, Popu_{D_N}$}, where $N \leq |Database|$ and $Popu_{D_1}$is the current popularity of data item D_1. When a new request asks for data item D_1: $Popu_{D_1} = Popu_{D_1} + 1$ and when data item D_1 is broadcast: $Pop_{D_1} = 0$.

*Definition 4.*If the communication range of a RSU is *D* and average speed of a vehicle is S, then the maximum deadline of a request is:

$$Deadline_{Max} = \frac{2D}{S} - T.$$

where, *T* is the request generation time. As there is a certain possibility a vehicle may stop within the transmission range of a RSU for any reason, the general assigned deadline of a request is:

$$Deadline = \min\{random(\omega_{min}, \omega_{max}) * Service\ Time, Deadline_{Max}\}.$$

where, $Service\ Time = \frac{Data\ item\ size}{Channel\ bandwidth}$ and ω is a random number.

C. Performance Metrics

We use the following performance metrics to evaluate the performance of different on-demand scheduling algorithms in our model.

1. Deadline Miss Rate: It measures the percentage of requests missed the deadline to the total number of requests received by the RSU. If the deadline miss rate is low, means scheduling algorithm is better.

2. Throughput: Throughput is the number of requests successfully served by a RSU in unit time. Hence, if a scheduler broadcasts the most popular data item, many requests will be served concurrently and throughput increased. High throughput means better system performance.

3. Average Response Time: The average amount of time required to get the response from a RSU after submission of a request. Low average response time initiates system is improvement.

IV. On- demand Scheduling Algorithms

We adopt the following on-demand scheduling algorithms in our system and then compare and analysis their performances across above defined performance metrics.

1) First Come First Served (FCFS) [12]:This is a base line scheduling algorithm. It serves the requests according to their arrival order. We just use this to take into consideration the performance of other different on-demand algorithms; how far they vary from the base line scheduler.
2) Most Request First (MRF) [13]: This algorithm works according to the popularity of the data item. It broadcasts the data item from the database which has the maximum popularity.
3) Earliest Deadline First (EDF)[14]: EDF works according to the deadline of the requests. The data

item which is requested by the most-urgent-deadline-request in the requests waiting queue, will be served first.

4) Number of pending requests Multiply Waiting time (R×W) [15]: R×W works based on the two factors. R means popularity of the data item and W means waiting time. So, before each broadcast decision making, this algorithm calculates every requests R×W value, i.e. the multiplication of popularity of the requested data item and waiting time of the request. The request having maximum R×W value will be broadcasted first.
5) Longest Wait First (LWF) [16]: LWF measures the sum of the waiting time of all the outstanding requests for a data item. A data item with maximum LWF value will be chosen for broadcasting. LWF incorporates directly requests deadline and indirectly popularity of the data item.
6) Shortest Service Time First (SSTF): Yu *et al.* [17] study the performance of SSTF algorithm in heterogeneous environment. SSTF picks out the data item from the requested data item according to their service time. The data item which needs minimum service time to serve will be broadcasted first, *where service time is the time necessitated to broadcast a data item when the system is idle. SSTF directly depends on the data item size.*
7) Deadline Size Inverse Number of pending requests (DSIN): DSIN algorithm is used in [1] where they call it as D*S/N. DSIN combines deadline of the request, size and popularity of the requested data item. Before making broadcasting decision scheduler determines the DSIN value of the all the requests in the waiting queue and serves the request which has minimum DSIN value.

V. Simulation Model

Our simulation model is precisely like the system architecture shown in Fig. 1. We use CSIM19 [18] for simulation experience and the explicitly used parameters are shown in Table 1, other parameters are CSIM default. A vehicle can continuously generate requests after getting into the transmission range of a RSU till moving out irrespective of its previous requests successful or not. At a time a vehicle can send single item request and the vehicle request generation interval is exponentially distributed defined by IGT (Table 1). If IGT value is low, request generation interval period is short bringing to heavy load to RSU. Vehicles data item access pattern is distributed by Zipf [19] distribution. Here, the access probability of j^{th}data item is:

$$p(j) = \frac{\frac{1}{j^{\theta}}}{\sum_{n=1}^{N}\frac{1}{n^{\theta}}}$$

where 0≤θ≤1, 0 means uniform and 1 means strict Zipf distribution. For data item size, we use 3 different types distributions (INC, DEC and RAND) [5], [20] but for space limitation, in this paper we will only discuss about INC size distribution. So, for integrated Zipf and INC size distribution, vehicles' requested popular data item will be small sized and unpopular data item will be big sized. INC size distribution is:

$$DSize_i = DSMin + \frac{(i-1)\times(DSMax - DSMin + 1)}{|Database|}$$

where i = 1, 2, 3, ……….., |Database|

TABLE I. Simulation Parameters.

Parameter	Default	Range	Description
IGT	0.3	0.1-1.0	Request generation interval
NVehicle	100	25-200	Number of vehicles
THETA(θ)	0.7	0.0-1.0	Zipf distribution parameter
DBSize	500	--	Number of data items in the database
ChannelBW	100KB/sec	--	Broadcasting bandwidth
Commun Range[21]	350 m	--	RSU communication range
$\omega_{min}, \omega_{max}$	5, 10	--	Range of min. and max. randomnumber
DSMin, DSMax	15, 512 KB	--	Min. and max. size of data item

VI. Experimental Results

To mimic the real time traffic, we let the vehicles to enter and go forth the RSU transmission range; after doing so we let these vehicles persistently to pass in identical fashion until we get the stable traced data for the same parameter settings. Then, we take mean of 100 iterations for each graph plotting data. In the following portion we discuss our experimental outcomes.

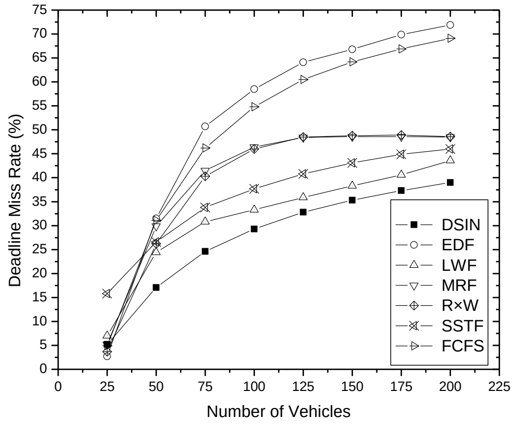
A. Impact of Workload

Fig. 2 shows the effect of varying workload in all of our seven on-demand algorithms in terms of deadline miss rate (Fig. 2(A)), throughput (Fig. 2(B)) and average response time (Fig. 2(C)) by increasing number of vehicles in the RSU transmission range.

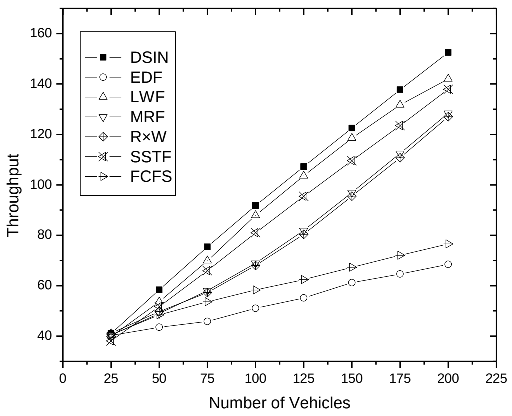
For Deadline Miss Rate: By building up workload, deadline miss rate increases for all the algorithms. When number of vehicles increases, number of requests generation also rises, as a consequence RSU gets many requests in the waiting queue. Then while RSU servicing a request many requests may miss their deadlines during that time period, hence overall deadline miss rate increased. From Fig 2 (A), EDF and FCFS suffer worst when number of vehicles increased. This is because EDF only takes the deadline of the requests into account neglecting the size of the data item; therefore while it serves a big sized data item with an urgent deadline request, it takes long time to serve, during that time many other urgent requests miss their deadlines. FCFS does not consider either deadline or data item size so it also suffers worst. As we use INC size distribution here, small sized data are most popular; hence considering size of the data item is an important metric to lessen deadline miss rate. SSTF care about data item size, thus it has moderate deadline miss rate. R×W and MRF both use popularity and their modest performance almost alike. Although LWF does not consider data item size, its indirect popularity measure helps to improve the deadline miss rate by broadcasting small sized data. However, DSIN which considers deadline, size and popularity outperforms all the other algorithms.

Its association of data item size and deadline metrics help to achieve better deadline miss rate in INC size distribution.

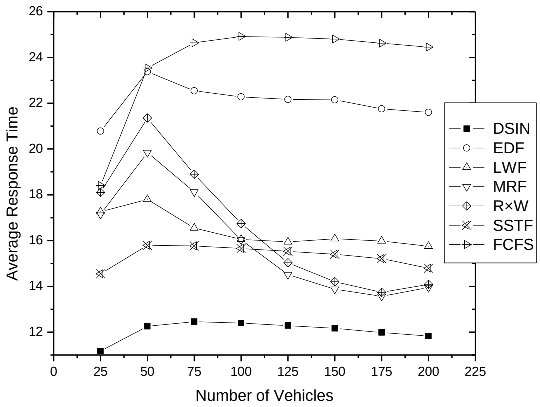
For Throughput: All algorithms' but FCSF and EDF's throughput significantly rises with increasing workload (Fig. 2(B)). We do this experiment by setting Zipf distribution parameter at default 0.7. Consequently, with increasing number of requests generation, many requests ask for the same data item and by disseminating that popular data item throughput increases dynamically.



(A) Deadline Miss Rate



(B) Throughput

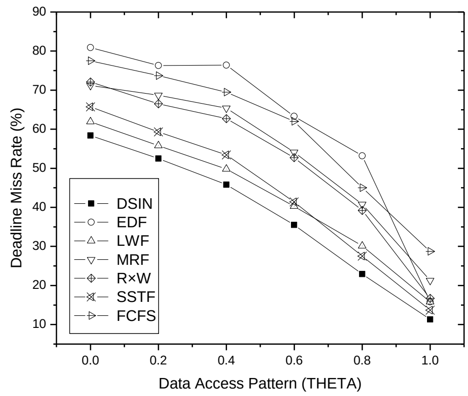


(C) Average Response Time

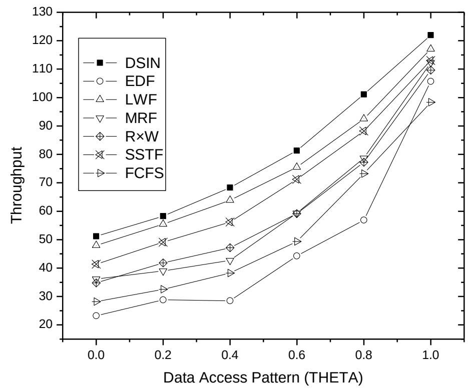
Fig. 2. Impact of workload by varying number of vehicles.

However, FCSF and EDF do not consider the popularity, so they have not much noteworthy improvement in throughput for increasing workload. By disseminating smallest sized data, SSTF, and popular data for long waited requests, LWF achieve better throughput. But DSIN achieves best throughput among all for disseminating smallest size popular data itemwith growing workload.

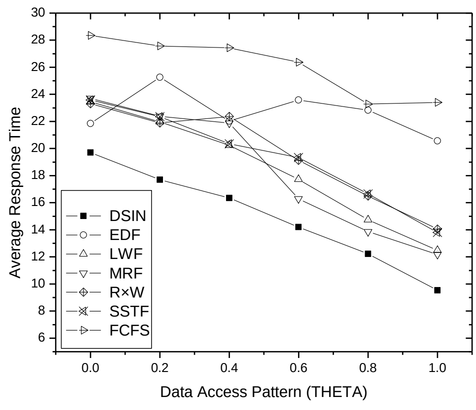
For Average Response Time: Except MRF and R×W algorithms, there is no major change for average response time with increasing workload. With workload ascending high, MRF and R×W get more popular small sized data item for broadcasting at θ value 0.7 and INC size distribution, hence MRF and R×W average response time decreases with increasing workload. All other algorithms' average response time value remain almost same except FCFS and EDF, their value slightly rise from the initial stage with high workload because they do not get the advantage either from popularity or small size of data item. Although with upgraded workload, waiting queue increases; DSIN, SSTF and LWF get the advantage for disseminating more popular small sized data item among the many waiting requests. Indeed they can retain their average response time value stable. However, DSIN can maintain the stable lowest average response time in high workload condition.



(A) Deadline Miss Rate



(B) Throughput



(C) Average Response Time.

Fig. 3. Impact of data access pattern(θ).

B. Impact of Data Access Pattern

Fig. 3 shows the impact of Zipf distribution parameter θ varying from 0.0 to 1.0. Almost all the algorithms' performance improve in terms of deadline miss rate, throughput and average response time with increasing θ value, which is similar finding from the data base community for on-demand algorithms in the hard real-time environment [12], [14]. Below we discuss the significance of deadline miss rate, throughput and average response time for varying θ value.

For Deadline Miss Rate (Fig. 3(A)): When θ is 0, vehicle data access pattern is purely random distribution, so all the algorithms have high deadline miss rate. But with increasing θ value, vehicle requests more popular data item, again in INC distribution the popular data items are small sized, so eventually by incrementing θ value, vehicles request popular small sized data item. Then by a single broadcast many requests been served with short time (small sized data takes short time to broadcast), hence performance increased dramatically. MRF and R×W shows modest performance improvement with increasing θ value for their popularity metric. However, here also DSIN algorithm performs better than all others for its combination (especially popularity and deadline metrics influence here much) request selection criteria.

For Throughput (Fig. 3(B)): As increasing θ value many requests ask for the same popular data item, by servicing such hot data items scheduler throughput increases appreciably. All the algorithms have much better rate when θ value exceeds 0.6. By broadcasting deadline urgent and popular date item DSIN algorithm outperforms all other on-demand algorithms in terms of throughput.

For Average Response Time (Fig. 3(C)):EDF and FCFS have no radical improvement for decreasing average response time with increasing θ, because they do not consider data item size or popularity which effect a lot for decreasing average response time specially in INC size distribution. As MRF and R×W counts the popularity for requests selection, by broadcasting popular data item (which are small sized too) they can reduce the average response time with increasing θ value. But DSIN achieves the best average response time value by broadcasting popular small sized data item.

VII. Discussion and Future Works

RSU based VANETs data dissemination has been researched by a number of researchers [1], [4], [21]. In this paper, we formulate the VANETs model which has three unique characteristics:(1) high vehicles mobility, (2) frequent disconnection of vehicles to RSU, and (3) strict time constraint for data dissemination. To find an efficient scheduler in this environment, we apply seven different existing on-demand scheduling algorithms to find a most adaptable one. We analyze their performances by varying workload and vehicles request access pattern in increment size data distribution. Our major outcomes of this analysis are: (1) Only deadline considering algorithm EDF, is not sustainable like FCFS algorithm in this environment but combination of request waiting time and popularity based algorithm LWF has modest performance. Popularity based algorithm MRF and R×W and service time related algorithm SSTF's performance lying between EDF and LWF. (2) Deadline, popularity and data item size based algorithm DSIN outperforms all the on-demand scheduling algorithms in the VANETs environment in terms of minimizing deadline miss rate and average response time and maximizing throughput both in high workload condition and varying vehicles requests access pattern. We do believe that our findings will boost us as well as other VANETs scheduling researchers to do more in this area.

In the future work, we want to apply DSIN scheduling algorithm in the multiple RSUs and incorporating upload requests from vehicles to RSUs with download requests from RSU to vehicles.

References

[1] Y. Zhang, J. Zhao, and G. Cao, "On scheduling vehicle-roadside data access," in *Proc. ACM VANET*, 2007.

[2] E. Schoch, F. Kargl, M. Weber, and T. Leimuller, "Communication patterns in VANETs," *IEEE Communication Magazine*, Nov. 2008.

[3] A. Boukerche, H. A. B. F. Olivera, E. F. Nakamura, and A. A. F. Laureiro, "Vehicular Ad Hoc Networks: a new challenge for localized-based systems," *Computer Communications*, vol. 31, pp. 2838-2849, 2008.

[4] C. Lochert, B. Scheuermann, M. Caliskan, and M. Mauve, "The feasibility of information dissemination in vehicular ad-hoc networks," in *Proc. 4th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, 2007.

[5] V. C. S. Lee, X. Wu, and J. K. Ng, "Scheduling real-time requests in on-demand data broadcast environments,"*Real-Time System*, vol. 34, no. 2, pp. 83-99, Oct. 2006.

[6] J. Chen, K. Liu, and V. C. S. Lee, "Analysis of data scheduling algorithms in supporting real-time multi-item requests in on-demand broadcast environments," in *Proc. IPDPS*, 2009.

[7] Z. D. Chen, H. T. Kung, and D. Vlah, "Ad hoc relay wireless networks over moving vehicles on highways," in *Proc. 2nd ACM international symposium on Mobile ad hoc networking and computing*, 2001.

[8] J. H. Wu, R. Fujimoto, R. Guensler, and M. Hunter, "MDDV: a mobility-centric data dissemination algorithm for vehicular networks," in *Proc.ACM VANET*, 2004.

[9] J. Zhao and G. Cao, "VADD: vehicle-assisted data delivery in vehicular ad hoc networks," *IEEE Transaction on Vehicular Technology*, vol. 57, no. 3, May 2008.

[10] J. Zhao, Y. Zhang, and G. Cao, "Data pouring and buffering on the road: a new data dissemination paradigm for vehicular ad hoc networks," *IEEE transaction on Vehicular Technology*, Nov. 2007.

[11] T. Nadeem, P. Shanakr, and L. Iftode, "A comparative study of data dissemination models for VANETs," in *Proc. ACM VANET*, 2006.

[12] J. W. Wong and M. H. Ammar, "Analysis of broadcast delivery in videotex system," *IEEE Transactions on Computers*, vol. C34, no. 9, Sept. 1985.

[13] J. W. Wong, "Broadcast delivery," in *Proc. of the IEEE*, vol. 76, no. 12, pp. 1566-1577, 1988.

[14] C. Liu and J. Layland, "Scheduling algorithms for multiprogramming in hard real-time traffic environments," *ACM*, vol. 20, no. 1, pp. 179-194, 1973.

[15] D. Aksoy and M. Franklin, "R×W: a scheduling approach for large-scale on-demand data broadcast," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 846-860, 1999.

[16] H. Dykeman and J. Wong, "A performance study of broadcast information delevery systems," *INFOCOM*, pp. 739-745, 1988.

[17] Y. Wu and G. Cao, "Stretch-optimal scheduling for on-demand data broadcasts," in *Proc. ICCCN*, 2001.

[18] H. Schwetman, "CSIM19: a powerful tool for building system models," in *Proc. 33th IEEE Winter Simulation Conference*, Arlington, VA, USA, 2001, pp. 250-255.

[19] G. Zipf, *Human Behaviour and the Principle of Least Effort: An Introduction to Human Ecology*,Cambridge, MA:Addison-Wesley, 1949.

[20] J. Xu, Q. Hu, W. C. Lee, and D. L. Lee, "Performance evaluation of an optimal cache replacement policy for wireless data dissemination," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 1, pp. 125-139, Jan. 2004.

[21] L. Z. Yi, L. Bin, Z. Tong, and Y. Wei, "On scheduling of data dissemination in vehicular networks with mesh backhaul," in *Proc. IEEE ICC*, 2008.

SyedaKhairunnesa Samantha is currently seeking for the opportunities to pursue her higher studies. She received her B.Sc. degree in Computer Science and Engineering from Khulna University of Engineering & Technology, Bangladesh. She published couple of papers in the international journals and conference proceedings. Her current research interests include computer networking, mobile computing and ad hoc networking with a focus on vehicular ad hoc networking.

NusratNurAfroseis currently working in a software firm in Bangladesh. She received her B.Sc. degree in Computer Science and Engineering from Khulna University of Engineering & Technology, Bangladesh.Her current research interests include quality assurance, computer networking, mobile computing and ad hoc networking.