

Scalable TDB based RSUs deployment in VANETS

Ramneek Kaur and Ravreet Kaur

Department of Computer Science,
Guru Nanak Dev University,
Amritsar, Punjab, India

Copyright © 2013 ISSR Journals. This is an open access article distributed under the ***Creative Commons Attribution License***, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT: Vehicular ad hoc networks (VANETS) are the flaming topic of research. VANET comprises of moving vehicles communicating with each other. VANETS involve three types of communication: vehicle to vehicle (V2V), vehicle to roadside (V2R) or vehicle to infrastructure (V2I) communication. VANETS consist of some vital components: RSU, OBU and Trusted Authority. Among them Roadside units (RSUs) are one of the fundamental components of Vehicular ad hoc network (VANET). Roadside Units (RSUs) are placed across the road for infrastructure communication. But the deployment cost of RSUs is very high, so to deploy more and more number of RSUs across roads is quite expensive. Thus, there is a need to optimally place a limited number of RSUs in a given region or road in order to achieve maximum performance. In this paper, we present a solution to this problem using parallel processing. A so-called scalable TDB based RSUs deployment algorithm with a goal of minimizing the parallel time taken to place roadside units in a given area and to attain high efficiency and cover maximum area has been presented. The performance of the proposed algorithm and optimization strategy is assessed by evaluating different parameters like efficiency, power consumption, serial elapsed time, parallel elapsed time, speedup and overheads incurred in running the algorithm in parallel.

KEYWORDS: VANETS, RSUs, Scalable TDB, Efficient, Deployment, Speedup, Overhead, Scalability.

1 INTRODUCTION

Vehicular networks are being used for a plethora of applications including enabling automotive safety. Intelligent transportation systems can also leverage these vehicular networks to enable applications such as traffic congestion prediction, mitigation and dissemination [1]. Vehicular Networks deploy the concept of continuously varying vehicular motion. The nodes or vehicles in VANETS can move around with no boundaries on their direction and speed. Vehicular adhoc network (VANET) involves vehicle to vehicle (V2V), vehicle to roadside (V2R) or vehicle to infrastructure (V2I) communication [2], [6]. VANET generally consists of On Board Unit (OBU) and Roadside Units (RSUs). OBUs enables short-range wireless ad hoc network to be formed between vehicles. Each vehicle comprises of hardware unit for determining correct location information using GPS. Roadside Units (RSUs) are placed across the road for infrastructure communication [6]. The number of RSU to be used depends upon the communication protocol. However due to the high cost of deployment of RSUs, it is desirable to cover maximum area with minimum cost.

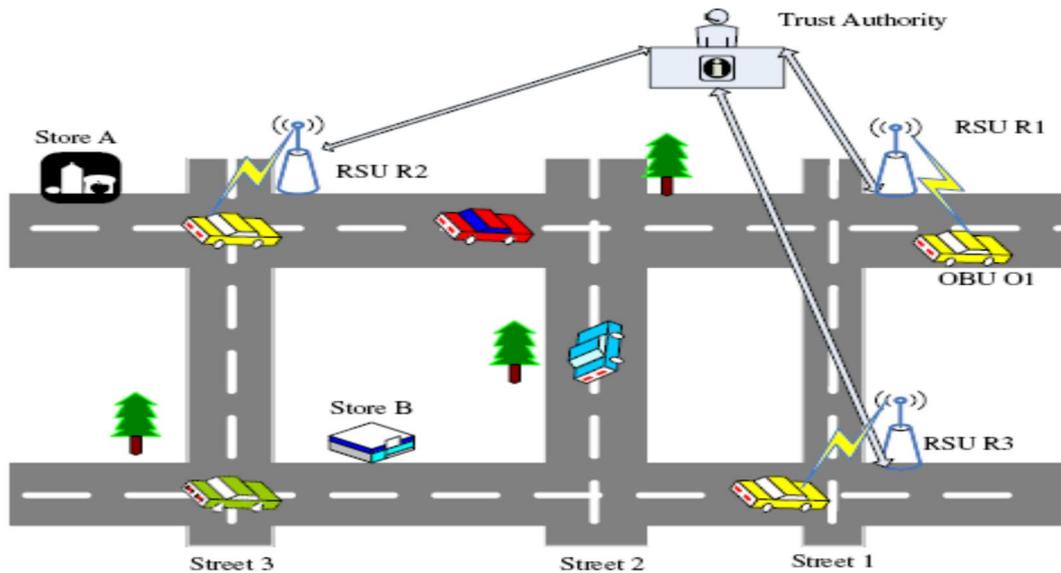


Fig. 1. System Model

VANET provide assistance to vehicle drivers for communication and coordination among themselves in order to avoid any critical situation through Vehicle to Vehicle communication [2] e.g. road side accidents, traffic jams, speed control, free passage of emergency vehicles and unseen obstacles etc. Besides safety applications VANET also provide comfort applications to the road users. Due to the dynamic nature of nodes in VANET the routing of data packets is much complex. Several factors like the type of the road, daytime, weather, traffic density and even the driver himself affect the movements of vehicles on a road. Hence, the network topology change frequently, and the routing protocol used has to adapt itself to these instantaneous changes continuously [2].

2 SYSTEM MODEL

As illustrated in Fig. 1, a typical VANET consists of three entities in city scenarios: the top TA, the fixed RSUs along the road side, and the mobile OBUs equipped on the running vehicles [6].

2.1 TA

TA is in charge of the registration of the RSUs and OBUs. TA can reveal the real OBU identity of a safety message and publishes the CRL periodically to the RSUs. Moreover, TA can be a road authority, such as the government. It has the basic information about streets and traffic statistics, and proposes the RSUs deployment plan according to the tradeoff between the requirements of most OBUs and the investment budget.

2.2 RSU

RSUs are erected at intersections for the considerations of power and management. RSUs use the same communication technology and the deployment cost is constant at any intersections. RSUs connect with TA by wired links, and act as certificate proxies of TA. An RSU can issue short-time certificates for the OBUs with valid membership.

2.3 OBU

Each OBU has a long-term unique identity. OBUs mainly communicate with each other for sharing local traffic information, and with the RSUs for updating the short time certificates. Digital maps are available for the OBUs. It provides the street-level map, the communication coverage of RSUs and the traffic statistics such as vehicle speed on roads, and traffic signal schedule at intersections.

3 RELATED WORK

Earlier works in optimal placement in VANET [6] include:

Y. Liang et. al in [1] has proposed a novel optimization framework for Roadside Unit (RSU) deployment and configuration in a vehicular network. Their objective is to minimize the total cost to deploy and maintain the network of RSU's. A user specified constraint on the minimum coverage provided by the RSU is also considered. The efficiency and scalability of the optimization procedure for large scale problems are also studied.

Sun et al. optimize the location of RSUs such that vehicle can reach an RSU within some timing constraint, given by sum of driving time and an overhead time (for adjusting the route), to update short term certificates. The optimization scheme may require vehicles to change their route which may have effects on local traffic condition. We do not have any route changing condition; we optimally place the RSUs considering the vehicles current routes only.

Lee et al. [4] seek optimal placement of RSUs to improve connectivity. Each intersection is considered as a potential RSU location. These potential locations are then ordered based on number of vehicle-reports received within communication range of each RSU. The placement scheme only considers taxi location reports and does not consider speed or density of all vehicles.

Li et al. [5] consider the optimal placement of gateways, which connect RSUs (access points - AP) to the Internet, while minimizing the average number of hops from APs to gateways. They consider pervasive APs such that every vehicle is connected to an AP. They do not consider vehicle speed, density or movement patterns.

Lochert et al. [7] use genetic algorithm for optimal placement of RSUs for a VANET traffic information system. The optimal placement is to minimize travel for some fixed landmarks and may not be useful for travel between any two points in an area.

Zhao et al. [8] optimize placement of Thowboxes, standalone units that act as relays, to improve contact and data-rate/throughput within context of a delay tolerant network. They aim at improving V2V communication and not the V2I communication.

Fiore et al. [9] optimally place RSUs (Access Points -AP) in an urban environment to improve cooperative download of data among vehicles. They aim at placing the APs at point where maximum vehicles cross each other, this helps in relaying the data from AP to a downloading vehicle via other vehicles.

Trullols et al. [10] optimally deploy RSUs (Dissemination Points – DPs) in an urban area to maximize the number of vehicles that contact the DPs.

Malandrino et al. [11] optimally deploy the RSUs (APs) to maximize the system throughput. They consider both the V2I (or I2V) and V2V communications for optimal placement of APs. Vehicle trajectory information (time and location) forms basis of this optimization which may not be available in many cases.

Zheng et al. [12] optimally deploy APs to improve contact opportunity; defined in terms of time for which a user remains in contact with an AP. These optimizations aims at transfer of data from RSUs to vehicles whereas, our optimization aims at transfer of data from vehicles to RSUs with an area coverage constraint. Also, we do not consider V2V communication in our optimization problem.

4 PARALLEL PROCESSING

Parallel processing is an efficient approach to meet the computational constraints of a large number of the current and emerging applications. Parallel processing is an efficient form of information processing which emphasizes the exploitation of concurrent events in the computing process. Parallel processing demands concurrent execution of many programs in the computer. It is in contrast to sequential processing. It is a cost-effective means to improve system performance through concurrent activities in the computer. So, we have used this approach to deploy RSUs efficiently. The main steps adopted during implementing the process are as:

- Application Specification : RSUs deployment in VANETs is chosen as application
- Subtask Decomposition: Fork and Join Construct is used to divide the problem into subtasks.
- Scheduling: Scheduling is done using Scalable Task Duplication Based RSU Deployment Algorithm (given later on).
- Programming: Writing executable program both in serial and in parallel.

5 PROBLEM STATEMENT

This paper deals with utilizing the fork and join algorithms in VANETs, for efficiently deploying RSUs in given area or road. As shown in literature survey the cost of the RSUs are too high so we are not able to deploy more and more RSUs to cover the given road, so need of the hour is to deploy them optimistically, such that the minimum number of RSUs can cover maximum range. But we have found that optimistic deployment of RSUs takes too much time i.e. serial time. So in order to reduce the amount of time required doing the same, we have proposed a new strategy of RSUs deployment which will use fork and join algorithms using TDB in such a way, so that it will result in the reduction of execution time. The overall objective is to deploy RSUs in the minimum time, to cover more and more area and to improve efficiency.

Throughout the paper emphasizes is on the parallel algorithm in VANETs, so no other VANET problems are considered in this research work. The proposed algorithm is also scalable. It gives better results till 45 no. of RSUs.

6 PROPOSED METHODOLOGY

Scheduling and allocation is a highly important issue since an inappropriate scheduling of tasks can fail to exploit the true potential of the system and can offset the gain from parallelization [12]. The objective of scheduling is to minimize the completion time of parallel application by properly allocating the tasks to the processors.

In this paper task duplication based scheduling is applied for optimizing the placement of roadside units. The task duplication scheduling provides greater efficiency and minimum make span time as compared to other scheduling techniques. The main idea behind the task duplication based scheduling is utilizing processor idle time to duplicate predecessor tasks [13]. This can avoid the transfer of data from a predecessor to a successor thus reducing the communication cost, network overhead and potentially reduce the start times of waiting task Task duplication scheduling provides better results than the serial placement of roadside units.

The new '**Scalable TDB based RSU deployment**' algorithm has been proposed. The scalability of the algorithm can be increased up to 50 number of RSUs. This algorithm works by using fork and join technique. The fork divides the job into equal parts and on each part the algorithm is applied simultaneously. The flag bit is assigned to each processor the when which will get free first will set its flag bit and the coming process will be allocated to that processor.

Following steps are included in the proposed algorithm:

- Step I:** Define the number of RSUs, the dimensions of the vanet area or road and initially place the first seed. Rest of the seeds will be placed randomly accordingly.
- Step II:** Deploy network considering parameters in step I.
- Step III:** Apply Task Duplication Based Scheduling Parallel algorithm using fork method. Fork will divide the job into equal parts on the basis of the number of matlab clients in the matlabpool.
- Step IV:** Apply optimistic RSUs deployment algorithm on each set separately. The solutions thus obtained will be joined.
- Step V:** If the result thus obtained is not optimal then repeat from stepIII, until the optimistic deployment is achieved.
- Step VI:** Evaluate parallel parameters and compare the results.
- Step VII:** End.

The above algorithm is developed and designed in matlab. The experimental results are obtained by running the proposed algorithm on matlab. Matlab provides an interactive software package. It has inbuilt toolboxes which we have used in calculating results. The variance in the results is achieved by running the algorithm serially and in parallel for deploying the RSUs. Serial and Parallel deployment of RSUs is compared using some metrics. Some of them are:

1. **Sequential Time:** This gives the serial elapsed time experienced by a particular job when run on a given system. It is denoted by T_s .
2. **Parallel Elapsed Time:** It is the time required to run the program on an n-processor parallel computer. It is denoted by T_p .

3. **Speed up:** Speedup is the ratio between sequential execution time and parallel execution time where the sequential time execution time is sum of total computation time of each task and parallel time execution is the scheduling length on limited number of processors.

$$S_p = (\sum_{i=1}^n T_i) / T_p$$

4. **Efficiency:** The efficiency of a parallel program is a measure of processor utilization.

$$EFF = S_p / N_p$$

S_p : Speed up and N_p : Number of processors

5. **Power consumption:** It gives the total power consumed by the unit. It can be calculated by adding up the power of each individual unit.

$$P_c = \sum (\text{power consumed by each unit})$$

6. **Overheads:** It gives the overhead incurred in running the algorithm in parallel on multiple clients.

$$O = \text{parallel time} / (\text{serial time} / \text{no. of processors or clients})$$

7 RESULTS AND DISCUSSION

We have carried out the experimental work in Matlab technology. Various parameters have been evaluated by taking different values of road side units, side dimensions and random seed placement. Comparisons on the basis of serial elapsed time, parallel elapsed time, speed up and efficiency has been done. The results are as:

7.1 RESULTS AND ANALYSIS OF ROAD SIDE UNITS

Figure below shows the variance of serial elapsed time and parallel elapsed time with the increase in number of road side units.

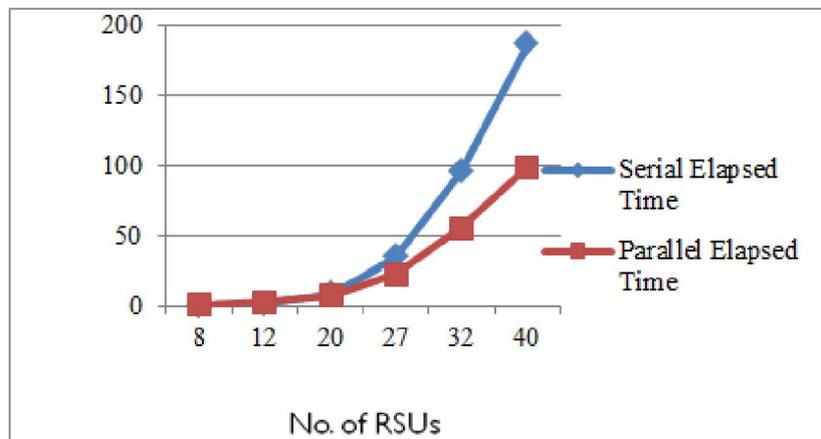


Fig. 2. Plot showing no. of RSUs and serial and parallel time

Fig. 2 shows that the serial elapsed time increases significantly with increase in the number of road side units. Initially for 8 and 12 rsus the serial elapsed time is more than the parallel elapsed time. After 20 rsus the serial elapsed time is much more than the parallel elapsed time.

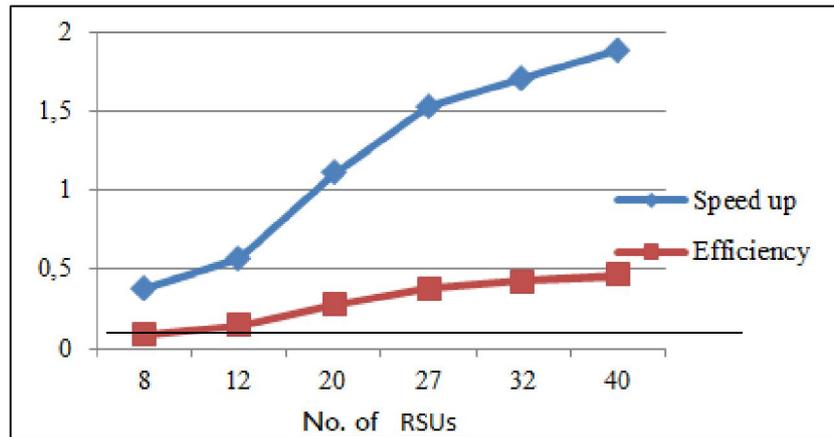


Fig. 3. Plot showing no. of RSUs and speedup and efficiency

The above plot shows the variation of speedup and efficiency with increase in number of RSUs.

The speedup increases significantly with increase in number of RSUs. But increase in efficiency is small.

7.2 RESULTS AND ANALYSIS OF SIDE DIMENSIONS

The figure4 shows the variation of serial time and parallel time against the changing side dimensions. It describes that what will be the effect of changing side dimensions on the serial elapsed time and parallel elapsed time.

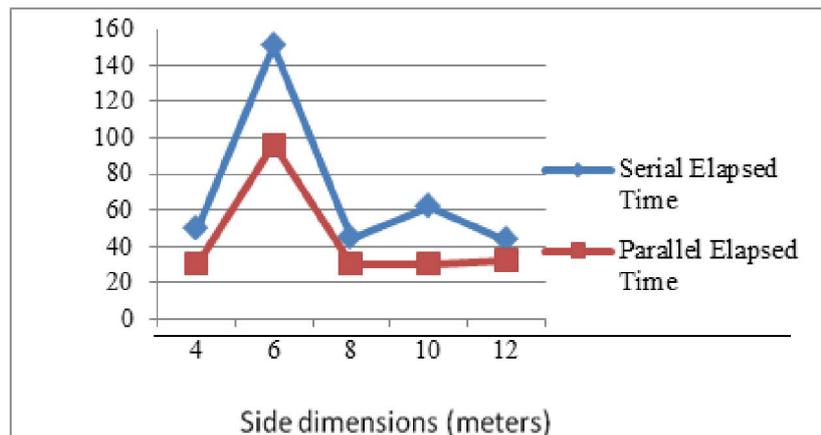


Fig. 4. Plot showing variance in serial and parallel time

The results of figure 4 shows that with increase in side dimension serial elapsed time increase to certain limit and then vary irregularly. As in fig. we can see that serial elapsed time increases up to 6 meters side and after that at 8meters dimension it falls and at 10 meters it again rises but a little.

Similarly parallel time increases but not much as compared to serial time and then falls and become almost constant. It concludes that you can get better parallel results by increasing side dimensions up to a certain limit.

Figure 5 shows the measure of speedup and efficiency against the increasing side dimension. The fig5 is showing that the speedup upto a certain limit decreases with increase in side dimensions after that it rises and again falls to a certain fixed value. The decrease in speedup shows that increase in side dimension up to a certain limit gives less parallel RSU deployment time. The efficiency is almost constant. It shows small fluctuations with increase in side dimensions.

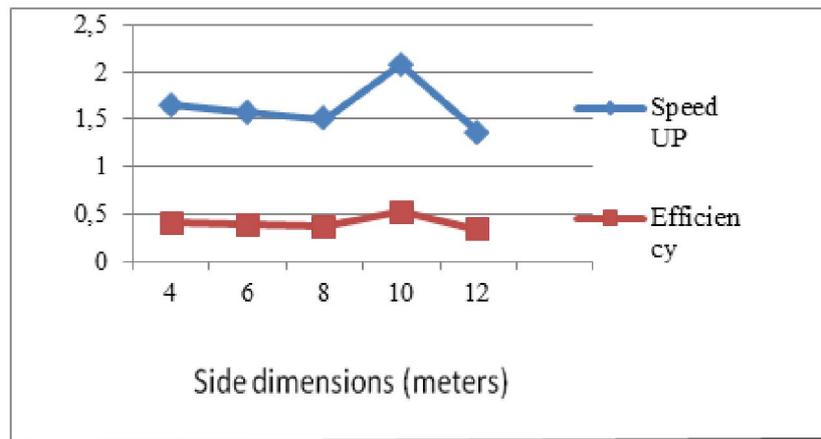


Fig. 5. Plot showing speedup and efficiency against changing dimensions

8 CONCLUSION AND FUTURE WORK

The main contribution of this paper have been the proposal of an approach for optimizing the placement of networked roadside infrastructure—supporting units—based on ‘Scalable TDB based RSUs deployment’ algorithm. The maximization of the network coverage and minimization of cost are considered as conflicting objectives to be achieved. It is shown that the task duplication scheduling provides greater efficiency and minimum make span time. The main idea behind the task duplication based scheduling is utilizing processor idle time to duplicate predecessor tasks. This paper gives related analysis for the optimistic deployment of RSUs. The results show that all the parameters vary with the change in the number of roadside units and side dimensions. With the increase in number of RSUs the increase in parallel elapsed time is less as compared to the large increase in serial elapsed time. Thus the proposed algorithm has great implication for large number of RSUs. The speedup and efficiency also increases with scalability in RSUs. The increase in side dimensions is favorable upto certain limit after that there will be only minor fluctuations with further increase in side dimensions.

In the near future research work can be extended in making this approach more suitable for realistic world. In this paper more emphasis is on parallel processing. But there are great number of issues in VANETs e.g. road side accidents, traffic jams, speed control, free passage of emergency vehicles and unseen obstacles and several other factors like the type of the road, daytime, weather, traffic density etc. which can be addressed to in future.

REFERENCES

- [1] Y. Liang, H. Liu, and D. Rajan, “Optimal Placement and Configuration of Roadside Units in Vehicular Networks”, *Proceedings of VTC Spring 2012*.
- [2] Anup Dhamgaye and Nekita Chavhan, “Survey on security challenges in VANETS”, *International journal of computer science and network*, Vol. 2, No. 1, pp. 88-96, 2013.
- [3] Y. Sun, X. Lin, R. Lu, X. Shen, and J. Su, "Roadside Units Deployment for Efficient Short-time Certificate Updating in VANETs", *IEEE ICC proceedings*, 2010.
- [4] J. Lee and C. Kim, “A roadside unit placement scheme for Vehicular Telematics networks”, *AST*, 2010.
- [5] P. Li, X. Huang, Y. Fang, and P. Lin, “Optimal placement of gateways in Vehicular Networks”, in *IEEE Transactions on Vehicular Technology 2007*, Vol. 56, No. 6, pp. 3421-3430, 2007.
- [6] Baber Aslam, Faisal Amjad, and Cliff C. Zou, “Optimal Roadside Units Placement in Urban Areas for Vehicular Networks”, *University of Central Florida*, 2011.
- [7] C. Lochert, B. Scheuermann, C. Wewetzer, A. Luebke, and M. Mauve, “Data aggregation and roadside unit placement for a VANET traffic information system”, *ACM*, 15 Sept, 2008
- [8] W. Zhao, Y. Chen, M. Ammar, M. Corner, B. Levine, and E. Zegura, “Capacity enhancement using Throw boxes in DTNs”, *MASS 2006*.
- [9] M. Fiore and J. Barcelo-Ordinas, “Cooperative download in urban vehicular networks,” *MASS 2009*.
- [10] O. Trullols, M. Fiore, C. Casetti, C.F. Chiasserini, and J.M. Barceló Ordinas, “Planning roadside infrastructure for information dissemination in intelligent transportation systems”, *Computer Communications*, Vol. 33, No. 4, pp. 432-442, March 2010.

- [11] F. Malandrino, C. Casetti, C. Chiasserini, and M. Fiore, "Content downloading in vehicular networks: What really matters," *INFOCOM*, 2011.
- [12] Z. Zheng, Z. Lu, P. Sinha, and S. Kumar, "Maximizing the Contact Opportunity for Vehicular Internet Access," *INFOCOM*, 2010.
- [13] Y.-K. Kwok and I. Ahmad, "Benchmarking the task graph scheduling algorithms," *IPPS/SPDP*, 1999.