Towards Connectivity-Aware Deployment and Adjustment for Roadside Units

Jiafeng Hu

Institute of Software, Chinese Academy of Sciences
Beijing 100190, China
University of Chinese Academy of Sciences
Beijing 100190, China
E-mail: acmhujiafeng@gmail.com

Beihong Jin

Institute of Software, Chinese Academy of Sciences
Beijing 100190, China
E-mail: beihong@iscas.ac.cn

Fusang Zhang, Zhaoyang Wang and Yuwei Yang

Institute of Software, Chinese Academy of Sciences
Beijing 100190, China
University of Chinese Academy of Sciences
Beijing 100190, China

Abstract: Deploying RSUs (RoadSide Units) is expected to improve the ratios of data delivery and reduce the delay of information dissemination over VANETs (Vehicular Ad hoc Networks). However, as for RSU deployment in urban scenarios, no appropriate theoretic model has been built as a guide so far. Meanwhile, the solution space for practical RSU deployment seems huge. Therefore, it is still a challenge to propose an ideal RSU layout. Considering that some applications may lead to a large amount of interaction between vehicles and POIs (Points Of Interest), this paper presents an RSU deployment and adjustment approach Volans so as to improve the performance of such interaction. Following Volans, the connectivity status of a VANET is first estimated by analyzing the historical trajectories of vehicles, and then the locations of a specified number of RSUs are calculated on the basis of the connectivity status. Moreover, the result layout of RSUs can be adjusted with the changes of POIs or trajectory data. Experimental results show that Volans is feasible and flexible and also demonstrate the performance improvement on data delivery brought by RSU deployment.

Keywords: Vehicular Ad hoc Networks; Roadside Units; Deployment; Adjustment; Connectivity Aware.

Reference to this paper should be made as follows: *International Journal of Autonomous and Adaptive Communications Systems*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Jiafeng Hu is an M.S. candidate at Institute of Software, Chinese Academy of Sciences in Computer Science. He received B.E. degree in 2011 from Jilin University, China. His research interests include mobile and pervasive computing, middleware and distributed systems.

Beihong Jin received B.S. degree in 1989 from Tsinghua University, China, M.S. degree in 1992 and Ph.D. degree in 1999 from Institute of Software, Chinese Academy of Sciences, all in computer science. Currently she is a professor at Institute of Software, Chinese Academy of Sciences. Her research interests include mobile and pervasive computing, middleware and distributed systems. She has obtained research support from the National Natural Science Foundation of China, the 863 plan of China MOST (Ministry of Science and Technology), etc. She has published over 80 research papers in international journals and conference proceedings and holds two China patents. She is a senior member of the CCF (China Computer Federation) and a member of the ACM and the IEEE.

Fusang Zhang is a research assistant at Institute of Software, Chinese Academy of Sciences. He received his M.S degree from Institute of Software, Chinese Academy of Sciences in 2013, and B.S degree from Xidian University in 2010. His research interests include vehicular ad hoc networks, mobile and pervasive computing, middleware and distributed systems.

1 Introduction

VANETs are a kind of mobile and self-organizing network built over two types of nodes, i.e., the vehicle nodes and the RSU nodes. They are susceptible to link disconnection and network partitioning due to the movement of vehicle nodes. However, RSU nodes are almost stationary, which can be fixed on certain points along the roads, and also can be moved from one place to another if necessary. Recently, the roles that should and could be played by RSUs have been extensively explored. In most cases, the RSUs are employed to temporally store and then relay data (Barrachina et al., 2012). Sometimes, the RSUs are used as gateways to the Internet and to the infrastructure of other systems such as an ITS (Intelligent Transportation System) (Kchiche and Kamoun, 2010). Occasionally, RSUs also assist in channel coordination and access in the MAC layer (Chung et al., 2011). Obviously, the more functions RSUs have, the more important the RSU deployment is. However, as far as costs are concerned, it is unrealistic to deploy a large number of RSUs along the roads. Therefore, there is a tradeoff between the RSU deployment (including the number of RSUs and their locations) and the performance of VANETs.

To date, it has been still a challenge to propose an ideal RSU layout for urban scenarios. Firstly, no researches are found to build a theoretical model to analyze and evaluate the performance benefit brought by different RSU deployment. The reason behind is the intrinsic complexity of road networks and vehicle behaviors. Secondly, the solution space for practical deployment is huge. We have observed that, along with an increasing number of RSUs and road segments, the number of possible combination of RSU locations becomes extraordinary large. On the other hand, the deployment of RSUs is related with various factors, including the topology of road networks, vehicle trajectories, vehicle speeds and densities, data routing mechanisms and application scenarios. All of these factors affect the size of the solution space.

We note that the POIs are often occurred in the VANET applications for urban scenarios where the interaction between vehicles and POIs is frequent. For example, as shown in Figure 1, an accident happens at some place, and then vehicles nearby need to deliver the accident message to the traffic rescue center over a VANET, where the rescue center is a POI. Similarly, the places reporting the unoccupied parking lots, and the places publishing air quality data are also the instances of the POI. Here, the POIs will evolve over time, some places may not be POIs anymore and some may become the new ones. Up to now, although some RSU deployment strategies have been proposed for different applications, for example, Zheng et al. (2010) aim at cooperative downloading over VANETs, no POI-related RSU deployment strategies have been proposed.

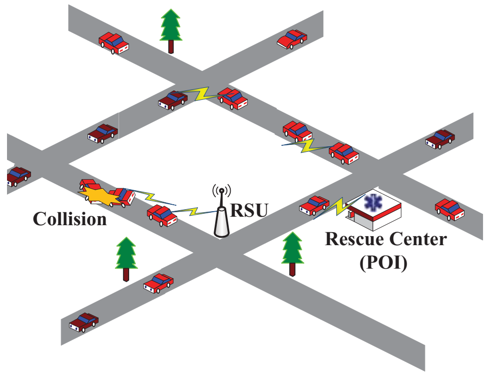


Figure 1: Example of application scenarios

The paper presents Volans, a connectivity-aware approach to RSU deployment and adjustment for POI-related applications. Following Volans, the connectivity status of a VANET is first estimated by analyzing the historical trajectories of vehicles. And then, the RSUs locations on the map can be calculated as long as the locations of POIs and the number of RSUs are specified. To efficiently get the RSU layouts, Volans gives two deployment strategies: stepwise strategy and greedy strategy. Volans can also be used to adjust the locations of RSUs while the POIs change or the newest trajectory data are obtained. In detail, if the total number of RSUs remains unchanged and the RSU number that need to be adjusted are specified as input, Volans can point out which RSUs need to be adjusted and where the new locations of these adjusted RSUs are. Volans can also give the locations for the newly-added RSUs on the existing layout if more RSUs are given.

The rest of this paper is organized as follows. Section 2 introduces the related work, while Section 3 builds a model for RSU deployment. The two strategies for RSU deployment and adjustment are described in Section 4. Section 5 shows our RSU deployment tool. The corresponding experimental results are reported in Section 6. Finally, Section 7 draws some concluding remarks.

2 Related Work

As for RSU deployment, the existing analytical models (Abdrabou and Zhuang, 2011; Sou and Tonguz, 2011; Reis et al., 2011) are all for highway scenarios. Abdrabou and Zhuang (2011) analyzes the vehicle-to-RSU delivery delay so as to capture the relationship between the delay and RSU-to-RSU distances. The analysis results can help to estimate the proper number of RSUs required to be deployed. The analytical model built in Sou and Tonguz (2011) can derive the delay of transmitting an event message to an RSU. However, the analytical model presented by Reis et al. (2011) is used for estimating the delay of delivering a message to a neighbor vehicle cluster with the aid of RSUs, and further evaluating the performance improvement brought by RSUs.

Some research work focuses on the practical RSU deployment strategies for urban scenarios. The deployment strategy presented by Lee and Kim (2010) is to look for the maximized coverage of RSUs, but totally ignoring the vehicle factors, e.g., vehicle trajectories, densities and speeds, etc. Kchiche and Kamoun (2010) present four strategies, selecting the locations of RSUs according to degree centrality, closeness centrality, betweenness centrality and equidistant metrics, respectively. The simulation experiments are conducted for the case of a relative small set of candidate RSU positions and the results show that the equidistant deployment achieves the best performance. Unfortunately, equidistant deployment has the same shortcoming as the strategy given by Lee and Kim (2010). Barrachina et al. (2012) present a density-based strategy D-RSU, where more RSUs are placed in the areas with lower vehicle densities. Obviously, if the data are forwarded along with high-density vehicle flows, then the deployed RSUs cannot aid in data delivery. Both Trullols et al. (2010) and Aslam et al. (2012) mention that the RSU deployment issue has a solution space with very high computational complexity. Trullols et al. (2010) formulates the issue of delivering information from RSUs to vehicles as NP-hard problems (e.g., a Maximum Coverage Problem) and then give the heuristic approaches. Aslam et al. (2012) give linear programming formalizations with the objective of minimizing average time duration from occurrence of an event until the event is delivered by vehicles to some RSU, given that the maximum number of RSUs and area coverage are set. After that, Aslam et al. (2012) present two approximation algorithms. The first algorithm uses a binary search tree to search all possible combinations. The second one is called balloon expansion heuristic algorithm. However, they assume that the RSUs are interconnected, which leads to an expensive and fixed layout.

Compared with the existing work, our Volans has the following features:

- It analyzes vehicle trajectories to obtain the data which reflect the connectivity status of a VANET and then makes use of these data to guide RSU deployment.
- In addition to obtaining an RSU layout according to the specified parameters, it also can adjust the locations of RSUs, which is triggered by the varying POIs or new trajectory data.
- It consists of two strategies which can achieve the similar performance improvement. Moreover, the two strategies can be used in a mixed way, i.e., using the greed strategy in the deployment phase and the stepwise strategy in the adjustment phase.

3 Problem Formulation

We assume that vehicles and RSUs are equipped with GPS receivers and digital maps, and they all can get the locations of POIs. Meanwhile, the trajectories of vehicles can be collected effectively.

We can extract from the trajectories the information related to the connectivity status of a VANET over a road network, and construct a weighted undirected graph $G(\mathbb{V}, \mathbb{E})$ for the VANET, where each intersection is mapped to a vertex in the set V and each road segment is mapped to an edge in the set E. Let $v_i \in \mathbb{V}$ be a tuple in the form of $< loc, rsu >$ where $v_i.loc$ is the location of v_i and $v_i.rsu$ is a Boolean variable that 1 means there is an RSU at v_i. Let L_e denote the length of edge $e \in \mathbb{E}$. Table 1 lists the notations used throughout the paper.

Table 1 Notation Used

Notation	Definition
$\mathbb{V}$	set of all intersections: $v_i \in \mathbb{V}$
$\mathbb{E}$	set of all road-segments: $e \in \mathbb{E}$
$v_i.loc$	the location of v_i
$v_i.rsu$	a Boolean variable that 1 means there is an RSU at v_i
$W(v_i, v_j)$	the weight of edge $e(v_i, v_j)$
L_e	the length of edge e
C_e	the average transmission distance ratio of edge e
R	the transmission radius of vehicles and RSUs
r	the number of RSUs needed to be deployed
$SP(v_i, v_j)$	the shortest path from v_i to v_j in graph $G(\mathbb{V}, \mathbb{E})$
N	the cardinality of $\mathbb{V}$
q_i	the weight of v_i used in Greedy Strategy
$h_k(t)$	the location of vehicle k at time t
T	the whole observation period

Assuming that at time t, there are $n(t)$ vehicles on the edge $e(v_i, v_j)$. These vehicles along edge e (from v_i to v_j) are denoted as $h_1, h_2, ..., h_n(t)$, respectively, and the location of any vehicle h_k ($k \in [1, n(t)]$) is denoted as $h_k(t)$. The weight $W(v_i, v_j)$ of edge $e(v_i, v_j)$ can be calculated as follows:

$$W(v_i, v_j) = \frac{normalize(L_e)}{C_e} = \frac{L_e - L_{min}}{C_e \times (L_{max} - L_{min})} \quad (1)$$

where L_{min} denotes the minimum length of all edges, L_{max} denotes the maximum length of all edges. If L_{max} is equal to L_{min}, then $normalize(L_e)$ is set to 0.5.

Then, we calculate C_e (the average transmission distance ratio) as follows:

$$C_e = \frac{\sum_{t=0}^{T} O_e(v_i, h_1(t), h_2(t), ..., h_k(t), ..., h_{n(t)}(t), v_j)}{L_e \times T} \quad (2)$$

where function $O_e()$ is defined in (3) to reflect the transmission distance on edge e.

$$\begin{aligned} & O_e(v_i, h_1(t), h_2(t), ..., h_k(t), ..., h_{n(t)}(t), v_j) \\ & = v_i.rsu \times f(v_i.loc, h_1(t)) + \sum_{k=1}^{n(t)-1} f(h_k(t), h_{k+1}(t)) \\ & \quad + v_j.rsu \times f(h_{n(t)}(t), v_j.loc) \\ & where, f(x,y) = \begin{cases} dist(x,y) & dist(x,y) \le R \\ 0 & dist(x,y) > R \end{cases} \end{aligned} \tag{3}$$

here, R denotes the transmission radius of vehicles and RSUs, x and y denote the locations of vehicles or RSUs, and $dist(x, y)$ is the Euclidean distance between location x and y.

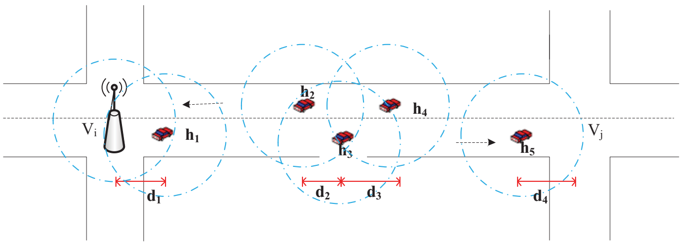


Figure 2: Illustration of transmission distance on edge $e(v_i, v_j)$ at time t

Figure 2 gives an illustration of calculating transmission distance. As shown in Figure 2, at time point t, there are 5 vehicles on edge $e(v_i, v_j)$, denoted as h_1, h_2, h_3, h_4, h_5 respectively. $v_i.rsu$ equals 1 because there is an RSU at v_i, while $v_j.rsu$ equals 0. Thus, the transmission distance can be calculated as follows:

$$\begin{aligned} & O_e(v_i, h_1(t), h_2(t), h_3(t), h_4(t), h_5(t), v_j) \\ & = v_i.rsu \times f(v_i.loc, h_1(t)) + \sum_{k=1}^{4} f(h_k(t), h_{k+1}(t)) \\ & \quad + v_j.rsu \times f(h_5(t), v_j.loc) \\ & = 1 \times d_1 + d_2 + d_3 + 0 \times d_4 \\ & = d_1 + d_2 + d_3 \end{aligned}$$

After we calculate the weights for all the edges, we get the weighted graph $G(\mathbb{V}, \mathbb{E})$. The weight of edge e reflects the delay of the road segment corresponding to e, which indicates the VANET connectivity status on that road segment.

Let's go back to the RSU deployment. According to the results given by Trullols et al. (2010), only intersections are our candidate locations for the deployment of RSUs. Assuming that $G(\mathbb{V}, \mathbb{E})$ has been built, the problem of selecting r intersections from all intersections to deploy RSUs can be converted into an equivalent one: select r vertexes from set $\mathbb{V}$ of graph $G(\mathbb{V}, \mathbb{E})$. However, the selection criteria and the optimization goals may be different, depending on different applications.

4 Strategies for RSU Deployment and Adjustment

Our strategies have a close relation with the target applications, i.e., POI-related applications. For the sake of simplicity, the POIs are supposed to have the same popularity and are located at the intersections. Moreover, the messages are permitted to be sent anytime, anywhere from any vehicle to any POI.

4.1 Stepwise Strategy

Keeping the POI-related applications in mind, we formulate the RSU deployment problem as a binary integer programming problem, named *RSU Deployment Optimization Problem* (RDOP in short). Given the graph $G(\mathbb{V}, \mathbb{E})$ introduced in Section 3 and r, the number of RSUs needed to be deployed, the optimization goal is to minimize the sum of the weights on edges in the shortest paths from each vertex to each POI. The formulization of $RDOP(\mathbb{V}, r)$ is given in (4).

$$\begin{aligned} & min \sum_{i=1}^{N} \sum_{j \in POISet} SP(v_i, v_j) \\ & s.t. \\ & SP(v_i, v_j) = \sum_{l=1}^{m} W(v_{a_{l-1}}, v_{a_l}), a_0 = i, a_m = j \\ & < a_0, a_1, ..., a_m > is\ the\ shortest\ path\ from\ v_i\ to\ v_j \\ & \sum_{i=1}^{N} v_i.rsu \le r, v_i.rsu \in \{0, 1\} \\ & W(v_i, v_j) is\ defined\ in\ Eq.\ (1) \\ & 1 \le i, j \le N, N = |\mathbb{V}| \end{aligned} \tag{4}$$

The RDOP is NP-hard because it can be proved to be a generalization of the *Set Cover Problem* (SCP), a well-known NP-hard problem (Cormen et al., 2001). The proof is as follows.

Proof: The input of a SCP instance includes a set $\mathbb{U} = \{u_1, u_2, ..., u_n\}$, m subsets of $\mathbb{U} : Set_1, Set_2, ..., Set_m \subset \mathbb{U}$, and a constant k. The SCP is to judge whether there exists a collection of at most k of these subsets whose union covers $\mathbb{U}$.

An instance of the RDOP I_{RDOP} can be constructed from an instance of the SCP I_{SCP} by:

1. For each element u_j in $\mathbb{U}$, create a vertex d_j;
2. For each subset Set_i, create a vertex c_i;
3. For each pair of vertex (i, j), if $u_j \in Set_i$, then connect c_i and d_j with an edge with initial weight 1;
4. Create a vertex p, for every subset Set_i, connect p and c_i with an edge with initial weight 0;
5. If vertex c_i is selected, the weight of edge (c_i, d_j) is set to 0, where the element u_j corresponding to d_j

belongs to Set_i, so that the weight of the shortest path from d_j to p is 0;

6. The problem is to judge whether it exists a collection of at most k vertexes in the new constructed graph that the sum of the weights on edges in the shortest paths from each vertex to p equals 0;

7. The new weighted undirected graph could be considered as a special instance of RDOP I_{RDOP}.

Obviously, if we can select k vertexes in I_{RDOP} that the sum of the weights on edges in the shortest paths from each vertex to p equals 0, then there exists a solution for I_{SCP}. On the other hand, if in I_{SCP} there exists k subsets whose union covers $\mathbb{U}$, then there exists a solution for I_{RDOP}.

The above construction completes the proof that RDOP is a generalization of SCP. Since SCP is clearly NP-hard, RDOP is also NP-hard. □

When the solution space of RDOP is small, it can be solved using naive search method within a tolerable running time. The naive search method $naive\_search(\mathbb{V}, r)$ can be described as follows:

1. For any possible combination $\mathbb{C}$ that select r vertexes from set $\mathbb{V}$:

   (a) For each $v_i \in \mathbb{C}$, set $v_i.rsu$ to 1, otherwise to 0;

   (b) Calculate the weights of edges in $G(\mathbb{V}, \mathbb{E})$ using Eq. (1);

   (c) Calculate the sum of the weights on edges in the shortest paths from each vertex to each POI denoted as $Opt_{\mathbb{C}}$.

2. Finally, select the combination $\mathbb{C}$ with minimum $Opt_{\mathbb{C}}$ as the solution.

However, in general, the solution space of RDOP is very huge. Therefore, we design a heuristic algorithm, named stepwise strategy. The basic idea in the stepwise strategy is to select intersections to deploy RSUs in an incremental way.

Let $\mathbb{V}$ denote the candidate vertex set, r is the number of RSUs needed to be deployed, and $\mathbb{S}$ is the resultant vertex set. At the beginning, $\mathbb{S}$ is empty. The steps of the stepwise strategy are as follows:

1. Find the maximum integer k, so that $RDOP(\mathbb{V}, k)$ whose candidate set is $\mathbb{V}$ can be solved through the naive search method within a tolerable running time;

2. If k is greater than r, solve $RDOP(\mathbb{V}, r)$, get the result set denoted as $\mathbb{S}'$, update $\mathbb{S}$ to $\mathbb{S} \cup \mathbb{S}'$ and return $\mathbb{S}$ as the ultimate result. Otherwise, denote the result of $RDOP(\mathbb{V}, k)$ as $\mathbb{S}'$, add $\mathbb{S}'$ into $\mathbb{S}$, remove $\mathbb{S}'$ from $\mathbb{V}$, and update r to $r - k$. Go to 3;

3. If r equals 0, return $\mathbb{S}$ as the ultimate result. Otherwise, go to 1.

Algorithm 1 Stepwise strategy for $RDOP(\mathbb{V}, r)$

```
S ← null;
while r > 0 do
    Find the maximum integer k, so that RDOP(V −
    S, k) can be solved using naive_search method
    if k > r then
        k ← r
    end if
    subset ← naive_search(V − S, k)
    S ← S ∪ subset
    r ← r − k
end while
return S
```

Algorithm 1 gives the pseudocode for the stepwise strategy.

When the newest trajectories data are obtained or POIs are updated, we have to adjust the positions of RSUs correspondingly. Based on the stepwise strategy, we design an RSU adjustment strategy named SBA (Stepwise-Based Adjustment).

Let $\mathbb{V}$ denote the candidate vertex set, and $\mathbb{S}$ is the set of all the RSUs which have been already deployed. Let c be the number of RSUs which needs to be adjusted, and $\mathbb{M}$ is the resultant set of vertexes. At the beginning, $\mathbb{M}$ is empty. The steps of the SBA are as follows:

1. For any possible combination $\mathbb{C}$ that select c vertexes from set $\mathbb{S}$, execute the next step.

2. Let $\mathbb{S}' = \mathbb{S} - \mathbb{C}$, run the stepwise strategy to select c vertexes from $\mathbb{V} - \mathbb{S}'$ to deploy RSUs, the result set is denoted as $\mathbb{C}'$. Let $\mathbb{M}' = \mathbb{S}' \cup \mathbb{C}'$, update $\mathbb{M}$ to $\mathbb{M}'$ if $\mathbb{M}$ is empty or $\mathbb{M}'$ is better than $\mathbb{M}$ in terms of Eq. (4).

SBA is feasible because in a realistic large-scale urban area a relatively small number of RSUs suffice (Wu et al., 2012).

4.2 Greedy Strategy

We consider while an RSU is placed on a vertex, the weights of its adjacent edges should be reduced. In the meantime, the nearer the location of an RSU is to a POI, the more important role it will take on, since the data will be transferred to the POI eventually. After balancing the tradeoff between these two factors and the performance of data delivery, we design a greedy strategy for RSU deployment.

First, define the priority q_i for vertexes v_i in $G(\mathbb{V}, \mathbb{E})$ as follows:

$$q_i = \alpha \times normalize(CF_i) + (1 - \alpha) \times normalize(DF_i) \quad (5)$$

where CF_i (Connectivity Factor) is the weight shift of $v_i s$ adjacent edges after an RSU is placed at v_i, DF_i

(Distance Factor) is the distance factor between v_i and each POI, α is the weight factor, and the *normalize* function normalizes CF_i and DF_i. The following is the formal definitions of CF_i and DF_i:

$$\begin{aligned} CF_i = &\sum_{(v_i,v_j)\in E} (W(v_i,v_j) - W(v_i',v_j)) \\ s.t.\ \ & v_i.loc = v_i'.loc,\ v_i.rsu = 1 \\ & v_i'.rsu = 0,\ v_j.rsu = 0 \end{aligned} \tag{6}$$

where $W(v_i,v_j) - W(v_i',v_j)$ is the weight shift of edge e which is one of v_i's adjacent edges.

$$DF_i = \frac{1}{\sum\limits_{j\in POISet} dist(v_i,v_j)} \tag{7}$$

Then, take the top r vertexes from a list of vertexes sorted by the priority to place RSUs.

Similarly, the greedy strategy has an adjustment strategy named GBA (Greedy-Based Adjustment). Assuming that c RSUs from $\mathbb{S}$ need to be adjusted, the steps are as follows:

1. Recalculate the priority q_i of each vertex $v_i \in \mathbb{V}$.
2. Select the top c lowest priority vertexes in $\mathbb{S}$ (which form a set named $\mathbb{C}$) as the vertexes which need to be adjusted, and record $\mathbb{S}' = \mathbb{S} - \mathbb{C}$.
3. Select top c highest priority vertexes in $\mathbb{V} - \mathbb{S}'$, denote the set of these vertexes as $\mathbb{C}'$, set $\mathbb{S}' = \mathbb{S}' \cup \mathbb{C}'$, and $\mathbb{S}'$ is the result set.

5 RSU Deployment Tool

According to our RSU deployment approach Volans, we build an RSU deployment tool whose GUI is shown in Figure 3. This tool is built based on an open source tool CityMob for Roadmaps (C4R) Fogue et al. (2012). RSU deployment strategies containing random strategy, high density strategy, stepwise strategy and greedy strategy are integrated in C4R. Our tool can generate RSU deployment layouts with the input of a road map, vehicle trajectory dataset, and the number of RSUs. Figure 3 shows the result that 20 RSUs are deployed with random strategy in a region of ZhongGuanCun district, Beijing, China.

6 Evaluation

In this section, we evaluate the strategies proposed in Section 4 by conducting the simulation experiments on different RSU layouts obtained from different strategies and then comparing the differences in data delivery performances. We choose a delivery ratio and delay as performance metrics, where the delivery ratio refers to a ratio of the number of messages received by POIs to the number of messages sent by vehicles, and the delivery delay refers to the duration from the moment when a message is sent from some vehicle to the moment when it is delivered to the destination POI.

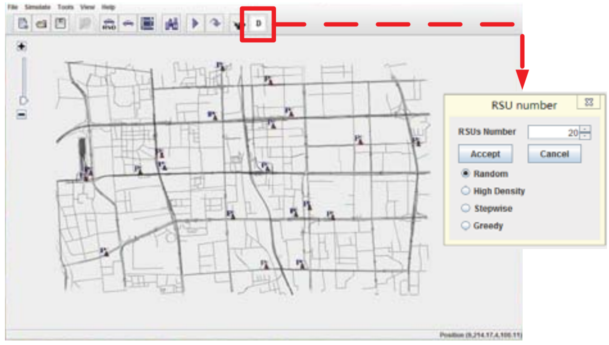


Figure 3: Screenshot of RSU deployment tool showing 20 RSUs are deployed in a region of ZhongGuanCun district

We use random strategy and high density strategy as comparison strategies. In random strategy, we randomly pick the specified intersections out of all intersections to deploy RSUs. In high density strategy, we choose intersections with higher traffic densities to deploy RSUs.

6.1 Experimental Setup

The road topology used in experiments is based on a real map of ZhongGuanCun district, Beijing, China. As shown in Figure 4, the experiment region is 4km*8km with a total road length of 145km, including 58 intersections and 86 two-lane road segments. The specifications of computer used for simulation are: *Processor* - Intel(R) Core(TM) quad-core i7-2600 @ 3.04GHz, *RAM* - 4GB, and *OS* - Ubuntu 12.04 32-bit.

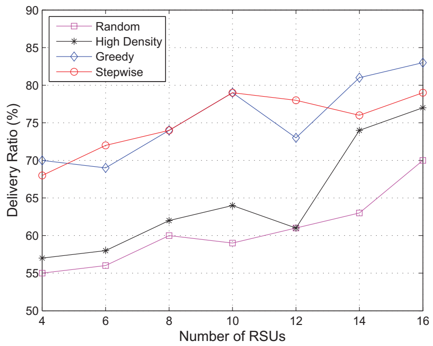
NS-3 (http://www.nanam.org) is adopted as the simulation platform in which 2Mbps 802.11 is used as the MAC protocol. In order to simulate roadside buildings in urban areas, barrier walls along the roadside are set to block wireless signals. SUMO-0.16.0 (Behrisch et al., 2011) simulator is used to generate the trajectories of vehicles whose movements are set to follow the IDM (Intelligent Driver Model). In the experiments, the initial positions of all vehicles are randomly generated, and vehicles make turns at intersections with a certain probability (probabilities of going straight, turning left and turning right are 0.6, 0.2, 0.2 respectively).

The process of a data delivery experiment is designated as follows. After the experiment begins, each vehicle generates one message and randomly chooses a POI as the messages destination every other second. During the period of the whole experiment, each vehicle will send 20 messages to each POI. The experiment lasts for 1000 seconds, recording the average delivery ratio and delay of messages at the end. In the experiments, RSU-Aided Data Dissemination Mechanism (RADDM) (Zhang et al., 2012) is employed for data routing. Table 2 summarizes the other parameters used in experiments.

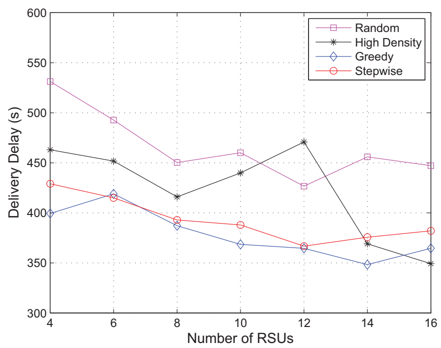
Figure 4: Snapshot of the region

Table 2 Experimental Parameters

Parameter	**Value**
Number of vehicles	50 ∼ 125
Vehicles velocity	20 ∼ 60 km/h
Transmission radius	200 m
Number of POIs	3
Number of RSUs	0 ∼ 58



(a) Delivery Ratio



(b) Delivery Delay

Figure 5: RSU deployment under different RSU numbers

6.2 Experimental Results and Analysis

6.2.1 RSU Deployment

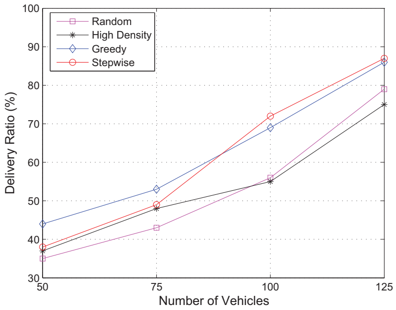
We conduct the first group of experiments with 100 vehicles. The first experiment is run on the region without deploying any RSU, and we get the delivery ratio is around 59.3% and delay is around 479 seconds. The second one is run under the situation of deploying RSUs on all the 58 intersections, where the delivery ratio is around 89.5% and delay is around 341 seconds. Figure 5 records the results of the other experiments, in each of which the RSU layout is set by one of the four deployment strategies.

In general, with the increase of the number of RSUs, all the four deployment strategies can improve the delivery ratio and decrease the delay. Compared with random strategy, stepwise and greedy strategies can acquire higher delivery ratios and lower delay. With 10 RSUs deployed, the delivery ratio of random strategy is about 20% lower than stepwise and greedy strategies, and the delivery delay is 70 seconds longer than stepwise strategy and 90 seconds longer than greedy strategy. With 16 RSUs (occupying 27.6% of the number of intersections) deployed using stepwise and greedy strategies, the average increase in delivery ratio is 66.7% of the increase obtained by deploying RSUs on all intersections.

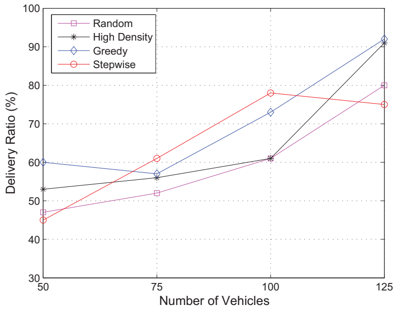
In the second group of experiments, we observe the effect of the number of vehicles on delivery ratio and delay under different number of RSUs (6 RSUs, 12 RSUs and 18 RSUs) deployed by the four strategies. The experimental results in Figure 6 show that the stepwise and greedy strategies outperform the random strategy and the high Density strategy. For instance, in Figure 6a and Figure 6d, with 125 vehicles, the delivery ratio of the random strategy is 8% lower than the stepwise strategy and 7% lower than the greedy strategy, and the delivery delay is 92 seconds longer than the stepwise strategy and 52 seconds longer than the greedy strategy. Meanwhile, the delivery ratio of high density strategy is 12% lower than the stepwise strategy and 11% lower than the greedy strategy, and the delivery delay is 140 seconds longer than the stepwise strategy and 91 seconds longer than the greedy strategy.

6.2.2 RSU Adjustment

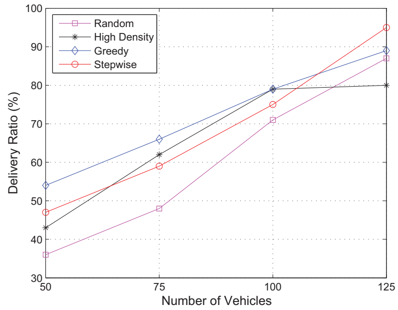
In order to inspect the effectiveness of SBA and GBA after the trajectories or POIs are updated, we carry out data delivery experiments where 1-6 RSUs are chosen from 6 deployed RSUs and their locations are adjusted by SBA or GBA. Figure 7 and Figure 8 show the experimental results after trajectories and POIs change, respectively. With the increase of the number of RSUs to be adjusted, the performances of experiments run on adjusted RSU layouts are improved. In terms of performance improvement, SBA is superior to GBA. In the case of POI variations, the effects of the two adjustment strategies are obvious. As shown in Figure 8a, the delivery ratio is improved by nearly 15% with only one RSU adjusted.



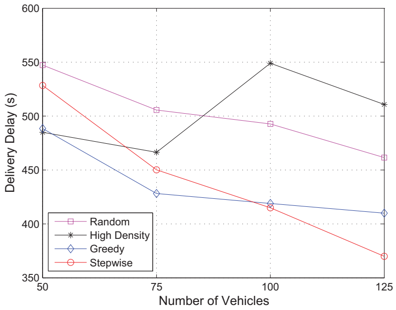
(a) Delivery Rate - 6 RSUs



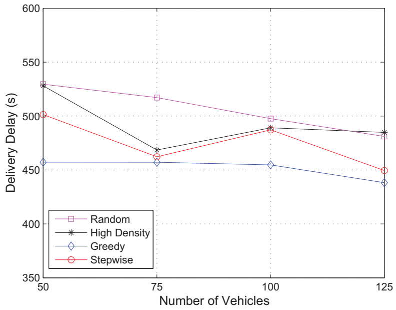
(b) Delivery Rate - 12 RSUs



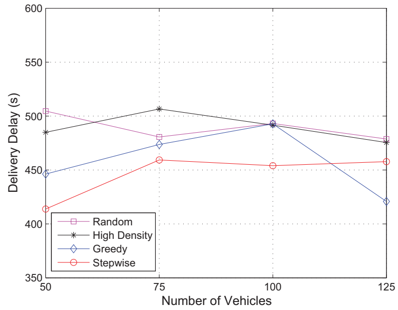
(c) Delivery Rate - 18 RSUs



(d) Delivery Delay - 6 RSUs

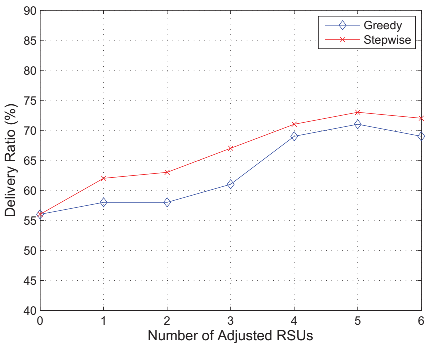


(e) Delivery Delay - 12 RSUs

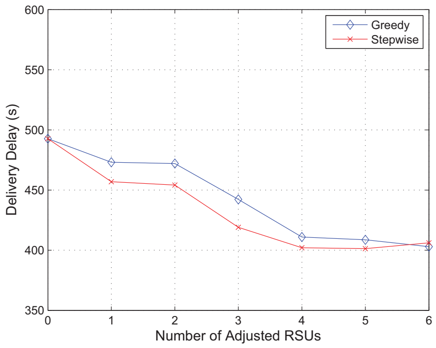


(f) Delivery Delay - 18 RSUs

Figure 6: RSU deployment under different vehicle densities

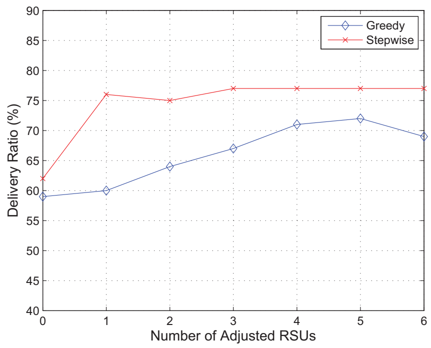


(a) Delivery Ratio

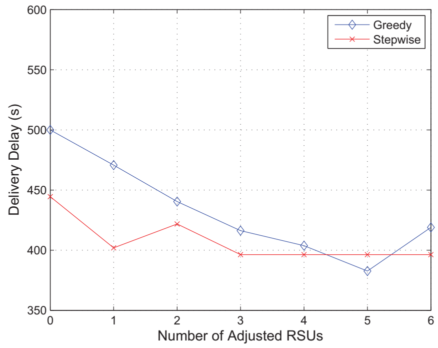


(b) Delivery Delay

Figure 7: RSU adjustment after trajectories change



(a) Delivery Ratio



(b) Delivery Delay

Figure 8: RSU adjustment after POIs change

7 Conclusion

In this paper, Volans, an RSU deployment and adjustment approach, is presented for the POI-related applications in urban scenarios. Volans adopts stepwise and greed strategies to solve the RSU layouts. Although the RSU layouts obtained by two strategies are different, the experimental results show that the performance improvement on data delivery is almost same. Our next work will examine the effects of different data routing mechanisms on RSU deployment, and explore the deployment strategies which can adapt to characteristics of data routing mechanisms.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant No. 60970027 and Grant No. 91124001.

References

Abdrabou, A. and Zhuang, W. (2011) ‘Probabilistic Delay Control and Road Side Unit Placement for Vehicular Ad Hoc Networks with Disrupted Connectivity’,*IEEE Journal on Selected Areas in Communications*, Vol. 29, No. 1, pp.129–139.

Aslam, B., Amjad, F. and Zou, C.C. (2012) ‘Optimal Roadside Units Placement in Urban Areas for Vehicular Networks’, *2012 IEEE Symposium on Computers and Communications (ISCC)*, pp.423–429.

Barrachina, J., Garrido, P., Fogue, M., Cano, J., Calafate, C.T. and Manzoni, P. (2012) ‘D-RSU: A Density-Based Approach for Road Side Unit Deployment in Urban Scenarios’, *Proceedings of the 2012 IEEE Intelligent Vehicles Symposium Workshops*, pp.1–6.

Behrisch, M., Bieker, L., Erdmann, J. and Krajzewicz, D. (2011) ‘SUMO - Simulation of Urban MObility: An Overview’, *SIMUL 2011, The Third International Conference on Advances in System Simulation*, pp.63–68.

Chung, J.M., Kim, M., Park, Y.S., Choi, M., Lee, S.W. and Oh, H.S. (2011) ‘Time Coordinated V2I Communications and Handover for WAVE Networks’, *IEEE Journal on Selected Areas in Communications*, Vol. 29, No. 3, pp.545–558.

Cormen, T.H., Leisersen, C.E. and Rivest, R.L. (2001) ‘Introduction to Algorithms, Second Edition’, *MIT Press*, ISBN 0-26203-384-4.

Fogue, M., Garrido, P., Martinez, F.J., Cano, J.C., Calafate, C.T. and Manzoni, P. (2012) ‘A Realistic Simulation Framework for Vehicular Networks’, *Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques (SIMUTools ’12)*, pp.37–46.

Kchiche, A. and Kamoun, F. (2010) ‘Centrality-based Access-Points Deployment for Vehicular Networks’, *17th International Conference on Telecommunications*, pp.700–706.

Lee, J. and Kim, C.M. (2010) ‘A Roadside Unit Placement Scheme for Vehicular Telematics Networks’, *Advances in Computer Science and Information Technology Lecture Notes in Computer Science*, Vol. 6059, pp.196–202.

Reis, A.B., Sargento, S. and Tonguz, O.K. (2011) ‘On the Performance of Sparse Vehicular Networks with Road Side Units’, *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pp.1–5.

Sou, S. and Tonguz, O.K. (2011) ‘Enhancing VANET Connectivity through Roadside Units on Highways’, *IEEE Transactions on Vehicular Technology*,Vol. 60, No. 8, pp.3586–3602.

The NS-3 network simulator, http://www.nanam.org

Trullols, O., Fiore, M., Casetti, C., Chiasserini, C.F. and Barcelo Ordinas, J.M. (2010), ‘Planning Roadside Infrastructure for Information Dissemination in Intelligent Transportation Systems’, *Computer Communications*, Vol. 33, No. 4, pp.432–442.

Wu, Y., Zhu, Y. and Li, B. (2012) ‘Infrastructure-Assisted Routing in Vehicular Networks’, *Proceedings of IEEE INFOCOM 2012*, pp.1485–1493.

Zhang, F., Jin, B., Zhuo, W., Wang, Z. and Zhang, L. (2012) ‘A Content-Based Publish/Subscrsibe System for Efficient Event Notification over Vehicular Ad hoc Networks’, *Proceedings of the 9th IEEE International Conference on Ubiquitous Intelligence and Computing*, pp.64–71.

Zheng, Z., Lu, Z., Sinha, P. and Kumar, S. (2010) ‘Centrality-based Access-Points Deployment for Vehicular Networks’, *INFOCOM, 2010 Proceedings IEEE*, pp.1–9.