An Effective RSU Allocation Strategy for Maximizing Vehicular Network Connectivity

Jeonghee Chi, Yeongwon Jo, Hyunsun Park, Taehyeon Hwang and Soyoung Park[1]

Department of Internet and Media Engineering, Konkuk University

{jhchi, choyw87, phs8004, joinbiro, soyoungpark}@konkuk.ac.kr

Abstract

Roadside Unit (RSU) in Vehicular Ad-hoc Network (VANET) is an essential unit usually placed at intersections to collect and analyze traffic data given from smart vehicles. Due to the costs of RSU setup, effective RSU allocation is one of important issues in VANET. In this paper, we provide an optimal RSU allocation algorithm with a concept of intersection-connectivity between intersections. We initially find optimal RSU candidate intersections which can cover the entire intersections, in the meantime minimizing the overlapped transmission coverage of RSUs. Then we repeatedly eliminate RSU candidates by the value of intersection-connectivity between RSU candidate intersections. The intersection-connectivity between two intersections is basically measured by the number of vehicles with the same moving pattern such that pass through both intersections. Thus, a high value of intersection-connectivity between two intersections stands for that the traffic information obtained at one intersection can be carried-and-forwarded to the other intersection with a high probability by those vehicles. With this feature, we repeatedly remove RSUs having high intersection-connectivity with its neighbors. Finally, we provide simulated analyses of our algorithms using real urban roadmaps of JungGu and Seochogu in Seoul. We analyze how our algorithms work in different types of roadways with real traffic data, and find the optimal number and positions of RSUs in these areas.

Keywords: *RSU Allocation, Optimal Algorithm, Intersection Connectivity, VANET*

1. Introduction

A Vehicular Ad-hoc NETwork (VANET) is a specialized mobile ad-hoc network which allows vehicle-to-vehicle communication and vehicle-to-infrastructure communication for providing with the driving safety, traffic efficiency and driver's entertainment, and so on. Since smart vehicles in VANET can exchange traffic information with other vehicles and infrastructural units, vehicles can collaborate with each other for making the traffic flow more fluent. Most of all, vehicles can react to accidents or specific threats to safety in advance by producing and forwarding alarms or warning messages [7, 2]. With smart vehicles, Roadside Unit (RSU) is an essential infrastructural static unit in VANET for collecting and analyzing traffic data. RSUs assist vehicle's efficient and secure driving by relaying traffic data and broadcasting locally analyzed traffic data. Thus, RSUs are basically supposed to be installed at intersections for maximizing data collection and distribution [1, 2, 16, 13]. An economical and efficient RSU placement for maximizing vehicular network connectivity is inevitable for supporting real-time traffic analysis and instant responses to VANET elements.

A lot of researches on the effective placement of RSUs in VANET [3, 4, 9, 10, 12] have been accomplished and most of them are mainly focusing on finding optimal locations of a

[1] Corresponding Author

limited number of RSUs due to the costs for RSU setup. Our goal is the same as others but our main interest is how to place RSUs as to cover the entire intersections. For real-time traffic analyses and responses to VANET elements, RSUs should be able to make use of traffic data collected at every intersection. In this paper, we introduce an effective RSU placement strategy which can cover as many intersections as possible in order to maximize network connectivity with the least number of RSUs. Our primary principles to allocate RSUs are (1) placing RSUs preferentially at important intersections, (2) distributing RSUs as even as possible, (3) minimizing the number of RSUs, and (4) maximizing the intersection coverage of RSUs. In order to figure it out, we use two concepts of intersection priority and intersection connectivity. The intersection priority represents the importance of each intersection. We have introduced the concept of the intersection-priority in our previous work [11]. The priority is computed by various traffic factors including traffic volume at the intersection, the intersection's geographical popularity, and the intersection's particularity, *etc.* Secondly, intersection connectivity introduced in this paper shows the possibility of data delivery between two intersections. The intersection-connectivity between two intersections is basically measured by the number of vehicles with the same moving pattern such that pass through both intersections. Thus, any two RSUs located at intersections with a high value of intersection-connectivity can collect and exchange traffic information obtained at each other's intersection with a high probability by vehicles passing through both intersections. With the concept of intersection priority, we find initial RSU candidate intersections that satisfy the first two principles. Then we try to reduce the number of RSUs based on the intersection connectivity between existing RSUs. Consequently, an RSU having a high value of intersection connectivity with its neighboring RSUs will be eliminated in sequence.

Finally, we provide simulated analyses of the proposed algorithms using real urban roadmaps based on real field traffic data. We have chosen two major districts: JungGu and SeochoGu in Seoul due to their characteristic road styles. JungGu represents an old district having very complex roadways with lots of unevenly distributed intersections, whereas SeochoGu is a finely planned district with a grid type of roadways and evenly distributed intersections. We analyze how the proposed algorithms work differently in such different types of roads, and find the optimal number and positions of RSUs in such areas.

The rest of this paper is organized as follows: We review some related works in Section 2, then describe assumptions, definitions and notations in Section 3. In Section 4, we give concrete descriptions on the proposed algorithms, and analyze the simulated results for our algorithm in Section 5. Finally, we conclude the paper in Section 6.

2. Related Work

There are several works addressing RSU deployment on VANET. Baber Aslam *et al.*, [9] presented two different optimization methods as a Binary Integer Programming (BIP) and Balloon Expansion Heuristic (BEH) for placement of a limited number of RSUs in an urban region. BIP method utilizes branch and bound approach to find an optimal analytical solution, and BEH method uses balloon expansion analogy to find an optimal or near optimal solution. Lochert *et al.*, [12] presented optimal placement of RSUs for a VANET traffic information system. They use the genetic algorithm to minimize travel for some fixed landmarks. Sun *et al.*, [3] proposed a significance ranking model for RSU localization and three kinds of Significance Degree (SD) computing strategies. They evaluated their model based on a VII test bed of the Olympic Park network in Beijing. Rebai *et al.*, [15] proposed the mathematical linear programming formulation to modeling the total road coverage problem in hybrid VANET-Sensor networks. They focused on hybrid VANET-Sensor network maintaining connectivity between sensors and RSUs. Rashidi *et al.*, [14] studied the trade-offs between the size of the gaps between RSUs and other system parameters such as data delivery ratio, data collection update interval and size of measured data, and proposed some heuristic method that can be used while deciding on the distance between neighboring RSUs. Trullols *et al.*, [4] proposed a maximum coverage approach for modeling the problem of deploying RSU. They optimally deploy RSUs as Dissemination Points (DPs) in an urban area to maximize the number of vehicles that contact the DPs. Xiong *et al.*, [6] studied the vehicular mobility pattern to find the optimal deployment places, and proposed a graph model to characterize the observed mobility pattern. Their approach is closely related to our works. However, even considering mobility pattern of vehicular, their approach does not take into account other information such as a situations or importance of road.

Most of these works are focused on maximizing the throughput and minimizing travel times by optimally deploying a limited number of RSUs, whereas we focus on finding the optimal number and positions of RSUs that can cover all intersections and that can maximize the connectivity between RSUs.

3. Assumption and Definition

In this section, we describe our main assumptions, definitions and notations used through the entire paper.

3.1 RSU Configuration

An RSU is a static unit installed at an intersection. It is basically assumed to be equipped with a transmitter for wireless communications, storage for collecting traffic data, and a computational device for creating traffic messages and analyzing traffic data. 75MHz of Dedicated Short Range Communication (DSRC) spectrum at 5.9GHz is adopted for the wireless communication technology. RSUs are usually supposed to transmit traffic data given from smart vehicles. In addition, RSUs keep analyzing local traffic situations then broadcast the analyzed data periodically and disseminate abnormal events occasionally. RSUs can communicate their traffic information with neighboring RSUs for further traffic analysis.

3.2 Intersection Priority

Intersection priority is an essential key to decide initial RSU candidate locations. Among the entire intersections, RSUs are sequentially deployed at intersections in descending order of the intersection priority. The priority can be measured by some traffic-related factors such as vehicle density, intersection's locational popularity (located within popular or famous areas), intersection particularity (located within attention regions such as an accident-prone area), and so on.

Let P_i be an intersection priority of the i^{th} intersection for $1 \leq i \leq n$. The value of P_i is determined as follows:

$$P_i = w_1 * f_{i1} + w_2 * f_{i2} + \cdots + w_m * f_{im} \tag{1}$$

where f_{ij} is a normalized value obtained by the j^{th} traffic factor for the i^{th} intersection, and w_j is a weight for each traffic factor for $1 \leq j \leq m$. Here, $w_1 + w_2 + \cdots + w_m = 1$. Further detailed descriptions on the intersection priority can be obtained in [11].

3.3 Intersection Connectivity

We define that two intersections are connected if messages obtained from one intersection can be delivered to the other intersection by vehicles without loss of data, and vice versa. Hence, if the traffic volume of vehicles passing by both two intersections is pretty big then, those vehicles can carry-and-forward traffic data between the two intersections. Thus, the connectivity between any two intersections can be measured by the number of vehicles with the same moving pattern that pass through both intersections. However, it is infeasible to track the whole moving patterns of every vehicle on the roads to calculate exact connectivity between every pair of intersections. Therefore, we approximately compute the connectivity with an average traffic volume between two intersections. In order to find the average traffic volume, we consider two cases: (1) two intersections are adjoining each other and (2) otherwise. For the first case, the traffic volume can be directly estimated as the total number of vehicles heading for a target direction. Since each intersection has usually four directional roadways, it causes twelve moving patterns of vehicles in total at each intersection. Then, the connectivity from intersection *A* to *B* is a total sum of vehicles merged into *B* from 3 other roadways intersecting at *A*, such as making a right, left and straight at *A*.

For the second case, there exists a path between two intersections passing through other intersections in the middle. In such case, we should consider the moving pattern of vehicles driving through the path. Our approximate strategy is to compute in sequence the connectivity of two adjacent intersections in the path. Then, it computes an average of the all. For an example, suppose that, between *A* and *D*, there exists a path of intersections such like {*A*, *B*, *C*, *D*}. Hence, the connectivity between *A* and *D* becomes an average of the values of connectivity between *A* and *B*, between *B* and *C*, between *C* and *D*. The difference with the first case is that, the connectivity from *B* to *C* should count only vehicles that passed through both *A* and *B* then move toward to *C* at the same time. It is not the sum of all vehicles merging in *C* from three other roadways, but is only vehicles that move *A* to *C* via *B*. And the connectivity from *C* and *D* is the same as the case of *B* and *C*. The connectivity from *C* to *D* counts only vehicles that move from *B* to *D* via *C*. One more consideration is that, vehicles within the region between *B* and *C* can share their information during driving. That is, all vehicles merged into *C* can share the information about *A* by the vehicles that passed through *A* and *B*. Thus, the connectivity from *C* to *D* does not need restrict to vehicles driving with a moving path of {*A*, *B*, *C*, *D*}, but simply consider vehicles moving from *B* to *D* via *C*. The followings describe a formal definition about the intersection connectivity.

Let I_i be the i^{th} intersection for $1 \leq i \leq n$, and $\overline{I_i I_j}$ be a path of intersections from I_i to I_j. If I_j is adjacent to I_i then $\overline{I_i I_j}$ is $\{I_i, I_j\}$. $\left|I_i I_j\right|$ is a length of the path from I_i to I_j. DI_i denotes a set of four roadways intersecting at I_i. We denote connectivity from I_i to I_j as $\overrightarrow{I_\iota I_J}$. $\overrightarrow{I_\iota I_J}$ is computed as follows:

(1) If $\left|I_i I_j\right| = 1$, then $\overrightarrow{I_\iota I_J} = \sum_{\{k \neq I_j \mid \forall\, k \in DI_i\}} T_{k,I_j}$,

where T_{k,I_j} is the number of vehicles entering from k into a roadway heading for I_j.

(2) If $\left|I_i I_j\right| \geq 2$, then $\overline{I_i I_j}$ can be represented as $\{I_i, I_{i+1}, \ldots, I_{j-1}, I_j\}$. Then,

$$\overrightarrow{I_\iota I_J} = \frac{\overrightarrow{I_\iota I_{\iota+1}} + \sum_{x=i+1}^{j-1} \alpha_x \cdot \overrightarrow{I_x I_{x+1}}}{\left|I_i I_j\right|} \text{ where } \alpha_x = \frac{T_{x-1,x+1}}{\overrightarrow{I_x I_{x+1}}}.$$

3.4 RSU Connectivity and Coverage

RSU's connectivity is defined in two ways such as direct connection and indirect connection. If any two RSUs are located within each other's wireless transmission range then the two RSUs are directly connected to each other. In other case, RSUs connectivity is also determined by the real traffic passing through those RSUs. Therefore, the indirect connectivity of two RSUs is identical to the intersection connectivity between intersections where RSUs are installed.

We define that an intersection is covered by an RSU if the intersection is located within the transmission range of the RSU. Hence, the coverage of a single RSU defines a set of intersections within its transmission range.

3.5 Notation

The following Table 1 summarized major notations used in this paper.

Table 1. Notations

Notation	Description
I	A set of all intersections
I_i	The i^{th} intersection in I where i={1,….,n}
P_i	The intersection priority of I_i
$\overline{I_i I_j}$	A path of intersections from I_i to I_j
$\lvert I_i I_j \rvert$	A length of the path from I_i to I_j
DI_i	A set of four roadways intersecting at I_i
$\overrightarrow{I_i I_j}$	An intersection connectivity from I_i to I_j
N_i	A set of intersections within the transmission range of an RSU installed at I_i
RSET	A set of intersections where RSUs are placed
CSET	A set of all intersections covered by RSUs at *RSET*

4. Intersection Connectivity-based RSU Allocation

Now we describe the concrete algorithms on our RSU allocation strategy. Our basic strategy is divided into two parts. The first part is to decide initial RSUs installation candidates according to the intersection priority until they cover all intersections. And the second part is to eliminate some of the candidates based on the intersection connectivity. In order to find the elimination candidates in the second part, we first create RSU connectivity network among the initial RSU candidates based on the intersection connectivity. Therefore, we explain the details for the three major steps such as (1) initial RSU candidate selection, (2) RSU connectivity network construction, and (3) RSU candidate elimination.

4.1 Initial RSU Candidate Selection

The goal of our initial RSU candidate selection is to find the minimum number of RSUs and their locations, which can cover all intersections. The two main principles for the initial selection are (1) to put RSUs firstly at highly prioritized intersections and (2) to minimize the overlapped coverage of RSUs. To achieve this, we use the hybrid algorithm [11] proposed in our previous work. The algorithm firstly chooses intersections according to the order of intersection priority. In addition, the algorithm also considers replacing a chosen-high prioritized candidate with a relatively lower prioritized intersection if such substitution can reduce the overlapped area. This is for making RSUs distributed as even as possible while keeping the order of intersection priority. That is, for each intersection I_i, the algorithm chooses an optimal solution between two options; (1) placing RSU at I_i by a simple greedy approach or (2) replacing previously-chosen RSU candidates with I_i. The second option happens at a situation that I_i's coverage includes any RSU candidate intersections. In such case, I_i will be excluded from the candidate by the greedy approach. But, the algorithm attempt to remove the chosen candidate intersection from the candidate set, then adds I_i having a lower priority as a new candidate. The replacement happens if and only if the following two conditions are all satisfied: (1) if the substitution reduces the size of the overlapped coverage of RSUs, and (2) either if the average of intersection priorities of modified candidates does not change or if the change is negligible. The detailed algorithm can be obtained in [11].

4.2 RSU Connectivity Network Construction

Once initial RSU locations are determined, we construct RSU connectivity network based on the intersection connectivity described in Section 3.3. In order to do this, we firstly calculate the intersection connectivity of every pair of intersections. At this stage, along with the Floyd algorithm that finds shortest paths among every vertex, we find maximum intersection connectivity and the corresponding path between every intersection. Because, between any two intersections, there could be many paths via different intersections, the connectivity between the two intersections is determined as the optimal path which has the biggest connectivity. Consequently, we can draw a connected bidirectional weighted graph of RSUs with such optimal paths. Vertices of the graph represent RSUs, and every path among RSUs is represented as an edge. The weight on each edge is the connectivity among RSUs. Here, an edge between two RSUs is drawn as a direct edge if there is no other RSU in the path between the two RSUs. Otherwise, the edge is represented as a path that passes through other RSUs included in the path. For an example, suppose that a path $\overline{I_iI_j} = \{I_i, I_x, I_y, I_z, I_j\}$ with a maximum connectivity is given, and RSUs are placed at I_i, I_y, and I_j. The RSU connectivity graph is drawn as Figure 1. Because the path contains R_y in the middle, an edge from R_i to R_y and another edge from R_y to R_j are inserted in the graph. There is no direct edge between R_i and R_j in this case. The graph can be drawn in this way for every pair of RSUs.

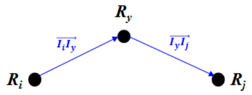


Figure 1. An example of drawing edges in the RSU connectivity graph

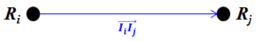


Figure 2. The change of RSU connectivity after removing R_y

The completed graph shows the initial RSUs connectivity network that can cover all intersections along with the minimum number of RSUs and the maximum connectivity between RSUs.

4.3 RSU Candidate Elimination

The RSU connectivity network is completed, we begin to remove RSUs. RSU elimination creates a new connectivity between the eliminated RSU's adjacent RSUs. In other words, by removing an RSU, the corresponding vertex is removed in the graph then every edge linked to the removed vertex is deleted. Instead, a new edge between the removed vertex's adjacent vertices is inserted. Here, the new edge implicitly stands for a new path where the removed vertex is included in the middle. The main tasks for the elimination are that (1) find an RSU for the elimination, (2) update with a new connectivity, and (3) re-compute the RSU coverage using newly updated RSUs. Any RSU can be removable but an RSU, such that can produce a new maximal connectivity after removing it, should be eliminated at first because our goal is to maximize the RSU connectivity and coverage. Therefore, the RSU removal requirements are as follows: The elimination should

(1) produce the second best connectivity after the removal;

(2) satisfy a minimum connectivity requirement in order to guarantee successful message delivery, where the minimum connectivity requirement is denoted as a threshold σ;

(3) provide still a satisfactory RSU coverage after the removal.

Consequently, any RSU that satisfies the above three conditions is eliminated. And the elimination can be repeated until we get a proper number of RSUs or until no more elimination is allowed according to the RSU connectivity and the RSU coverage. Either if any connectivity newly added in the graph is not bigger than σ or if the coverage goes down under a predefined threshold, then the elimination terminates.

The detailed elimination algorithm is given below. In order to find the best elimination candidate, for every RSU with the minimum vertex degree, the algorithm computes all possible connectivity among adjacent RSUs after removing each RSU. For an example, suppose that R_y in Figure 1 is chosen for the elimination. Then the algorithm computes again the connectivity $\overrightarrow{R_iR_j}$ between R_i and R_j about the path $\overline{R_iR_j} = \{R_i, \ldots, R_y, \ldots., R_j\}$. If $\overrightarrow{R_iR_j}$ is still bigger than σ and the best then R_y can be removed. As a result, the graph is changed to as shown in Figure 2. If there were already an edge between R_i and R_j, then edge replacing can occur. The underlying connectivity is about the path $\overline{R_iR_j} = \{R_i, \ldots, R_j\}$. It does not include R_y. Thus, the algorithm compares the new connectivity about $\overline{R_iR_j} = \{R_i, \ldots, R_y, \ldots., R_j\}$ with the existing connectivity about $\overline{R_iR_j} = \{R_i, \ldots, R_j\}$. Then, if and only if the new connectivity is better than the old one then existing edge is replaced with a new edge having a new path and connectivity.

Algorithm 1: RSU Elimination Algorithm

```
The initial RSU connectivity graph $G_R$ is given.
$V$ = A set of all vertices in $G_R$.
$E$ = A set of all edges in $G_R$. Here, $E_{V_iV_j}$ represents a single edge from $V_i$ to $V_j$
Initialize a degree of vertex $DG$ = 2;
While (! Terminate Condition) {
    Set $MAX$ = 0;
    For $\forall R_i \in V$ with a degree of $DG$ {
        Set $IN$ = A set of $R_i$'s adjacent vertices with an edge entering into $R_i$;
        Set $OUT$ = A set of $R_i$'s adjacent vertices with an edge outgoing from $R_i$;
        For $\forall$ pair of $V_iV_j$ where $V_i \in IN$, $V_j \in OUT$ and $V_i \neq V_j$
            Compute $\overrightarrow{V_iV_j}$ where $\overrightarrow{V_iV_j} = \{V_i, \ldots, R_i, \ldots, V_j\}$;
        If ($\forall \overrightarrow{V_iV_j} > \sigma$) && (max{$\forall \overrightarrow{V_iV_j}$} > $MAX$) then {
                $MAX$ = max{$\forall \overrightarrow{V_iV_j}$};
                $R_M = R_i$;
                $IN_M = IN$;
                $OUT_M = OUT$;    }
    }
    If ($R_M$ is found) then {
        If ($\exists E_{V_iV_j}$ for all $V_i \in IN_M$ and $V_j \in OUT_M$) then {
            If ($\forall$ new $\overrightarrow{V_iV_j}$ > existing $\overrightarrow{V_iV_j}$) then {
                $V - \{R_M\}$;
                $E - \{\forall E_{V_iR_M}\} - \{\forall E_{R_MV_j}\}$;
                Replace $E_{V_iV_j}$ with a new $E_{V_iV_j}$having a new path $\{V_i, \ldots, R_i, \ldots, V_j\}$ and a new connectivity;  }
        }   else {
                $V - \{R_M\}$;
                $E - \{\forall E_{V_iR_M}\} - \{\forall E_{R_MV_j}\}$;
                $E \cup \{E_{V_iV_j}\}$ having a new path $\{V_i, \ldots, R_i, \ldots, V_j\}$ and a new connectivity;   }
        $DG$ = 2;                    // this is for searching again from an RSU with the minimum degree
    } else $DG$++;
}
```

The RSU coverage at the initial RSU allocation is 100%. The coverage is a percentage about (the number of intersections covered by RSUs) / (a total number of intersections). As RSUs are removed, however, the coverage will be reduced. When R_y is removed, some intersections in N_y will not be managed by R_i and R_j because those intersections are out of transmission range of both of them, then those intersections may be excluded from *CSET*. At this point, we can consider one more situation such that, if some outside intersections are still included in the path $\overline{R_iR_j} = \{R_i, \ldots, R_y, \ldots., R_j\}$ then those intersections can be supposed to be indirectly managed by R_i and R_j. Thus, the actual intersections excluded from *CSET* are $N_y - \{\forall\, I_j \in \overline{R_iR_j}\}$ where $\overline{R_iR_j} = \{R_i, \ldots, R_y, \ldots., R_j\}$.

5. Simulation Results

In this section, we provide the simulation results of the proposed algorithm using real urban roadmaps based on real field traffic data. We have used the roadmaps of SeochoGu and JungGu in Seoul due to their characteristic road styles. SeochoGu has a grid form of roadways and evenly distributed intersections, whereas JungGu has very complex roadways with lots of unevenly distributed intersections. We analyze how our algorithm works differently in such different types of road networks, and provide the optimal number and positions of RSUs in such areas.

5.1 Simulation Environment

We have used Google Earth to obtain the geographical information of intersections in those areas. SeochoGu has 89 intersections distributed as a grid form, and JungGu has 147 intersections. We have used real traffic data about 12 moving patterns of vehicles collected at each intersection.

Table 1. Simulation dataset

Areas	The Total # of Intersections	Traffic Volume at each Intersection (the total # of vehicles at a single intersection)			Traffic Volume for each Direction (the # of vehicles on each moving direction)		
		Min	Max	Avg	Min	Max	Avg
SeochoGu	89	890	20,564	5,543	49	6,924	1,414
JungGu	187	1,190	9,034	4,080	1	2,922	814

In order to determine the intersection priority, we have considered two traffic factors such as the density of vehicles and the popularity of an intersection. For the density of vehicle, we have used the total number of vehicles passed by each intersection during an hour. In addition, we have used the number of vehicles moving along each moving pattern at every single intersection for obtaining the intersection connectivity. The real traffic data are obtained from the traffic analysis reports provided by the SeochoGu office and the JungGu city office, respectively. Table 2 summarizes the traffic density in two simulated areas. For the popularity, we have used trajectories of public bus lines. The popularity of each intersection is measured by the number of different bus lines passing by the intersection [11].

We have run the simulation for various experimental such as RSU's transmission range and the minimum connectivity requirement denoted as threshold. Table 3 summarizes the simulated values of the parameters.

Table 3. Simulation parameters

Simulation Parameters		Values
Transmission range of an RSU (*TR*)		250m, 350m, 450m, 550m
Threshold (σ) (Minimum Connectivity Req.)	SeochoGu	100, 300, 500, 800, 1000, 1500 (the average of traffic volume for each direction)
	JungGu	100, 300, 500, 800 (the average of traffic volume for each direction)

5.2 Analysis of Simulation Results

The Figure 3 and 4 show the progress of each step of our algorithm with real roadmaps. Figure 3(a) and 4(a) show the initial distribution of RSUs in two areas, respectively. The yellow colored marks are the locations of initial RSUs. Figure 3(b) and 4(b) show the initial RSU connectivity network in those areas. Figure 3(c) and 4(c) show the changes of the connectivity graphs after removing RSUs in two areas. The white colored mark indicates removed RSUs.

Our initial RSU selection algorithm generated 84 RSUs in JungGu and 75 RSUs in SeochoGu for the transmission range of 250m. When *TR* is 550m, 34 RSUs are required in JungGu and 37 in SeochoGu. For SeochoGu, the first scanning reduces RSUs at least 16% in the worst case (*TR*=250m) but reduces RSUs more than 58% in the best case (*TR*=550m). For JungGu, it reduces almost 60% of RSUs in the worst case (*TR*=250m), and reduces more than 81% of RSUs in the best case (*TR*=550m). Figure 5 shows the number of RSUs after elimination according to the different transmission ranges and the different threshold values. When *TR*=250m and *σ*=100, JungGu needs at least 67 RSUs, but it requires 83 RSUs when σ=800. In SeochoGu, 44 RSUs are required with *TR* =250m and *σ*=100. With *σ*=1500, 73 RSUs are required. In the case of JungGu, by the elimination, the number of RSUs has been reduced more than 20% compared to the number of initial RSU candidates. In SeochoGu, the RSUs have been reduced up to 41%.

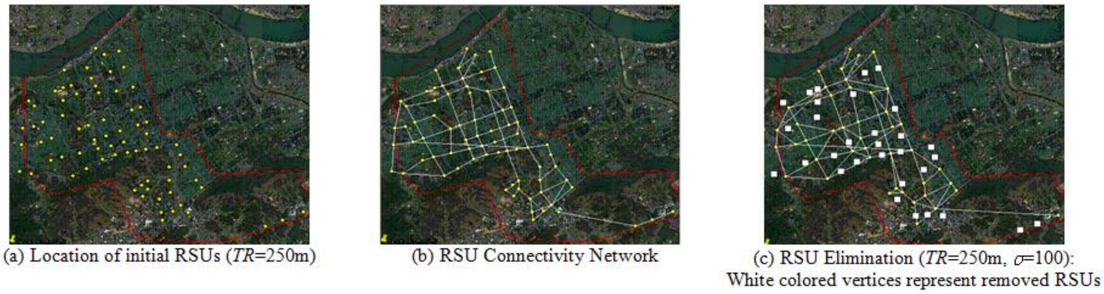

(a) Location of initial RSUs (*TR*=250m)
(b) RSU Connectivity Network
(c) RSU Elimination (*TR*=250m, *σ*=100): White colored vertices represent removed RSUs

Figure 3. Locations of RSUs in SeochoGu

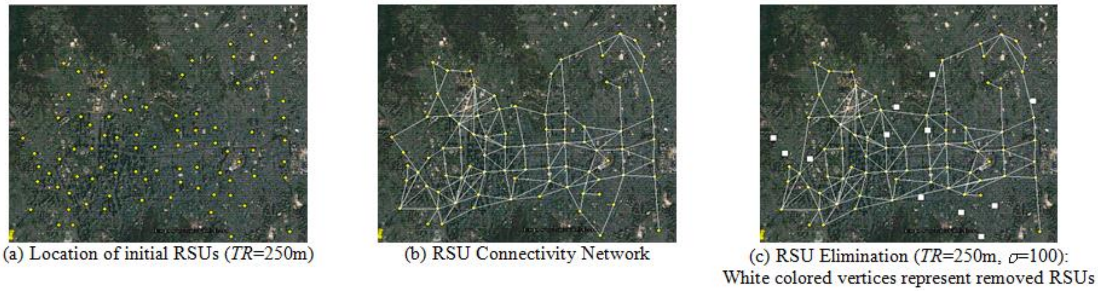
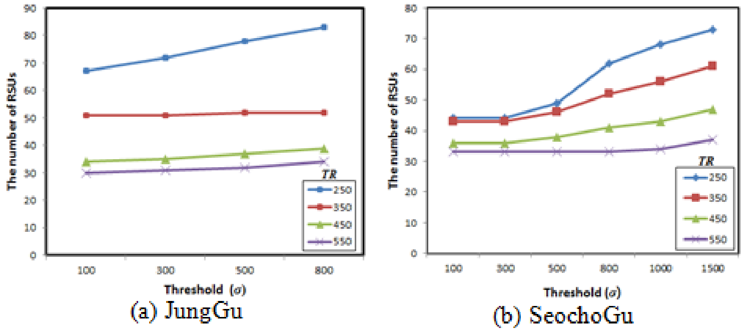
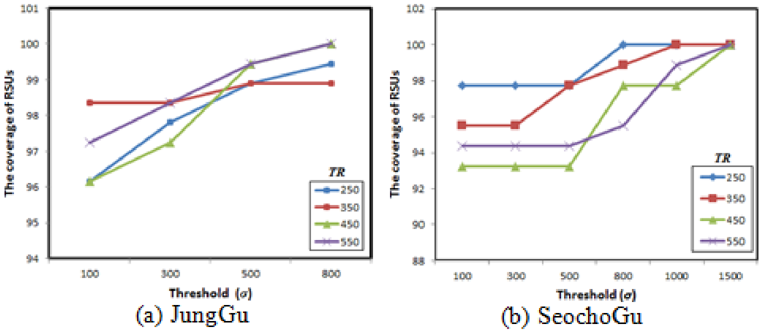

(a) Location of initial RSUs (*TR*=250m)
(b) RSU Connectivity Network
(c) RSU Elimination (*TR*=250m, *σ*=100): White colored vertices represent removed RSUs

Figure 4. Locations of RSUs in JungGu



(a) JungGu (b) SeochoGu

Figure 5. The number of RSUs according to *TR* and *σ*



(a) JungGu (b) SeochoGu

Figure 6. The change of the RSU coverage according to *TR* and *σ*

Since our initial RSU selection algorithm primarily minimizes the number of RSUs, the ratio of reduction by the second RSU elimination is relatively small. However, compared to the total number of intersections, our final results remarkably reduce the number of RSUs.

Figure 6 shows the changes of the RSU coverage according to the minimum connectivity threshold. The experimental results show that the RSU coverage is at least 93% in the worst case.

6. Conclusion

In this paper, we have proposed intersection-connectivity based RSU placement algorithms to distribute RSUs for covering all intersections with a maximal connectivity between RSUs while minimizing RSU setup costs. The advantage of our method is to find an optimal number and positions of RSUs for their initial distribution to cover the entire intersections. Then it reduces RSU candidates while keeping the best connectivity between RSUs and the RSU coverage. We have analyzed our strategy with real road maps of JungGu and SeochoGu in Seoul. The simulation results show that JungGu needs at least 67 RSUs with *TR*=250m and *σ*=100, and that 44 RSUs are required in SeochoGu for the same parameters. Compared to the total number of intersections, our strategy reduces RSUs more than 50% in average.

In the future, we need to improve and formalize our intersection connectivity model that can consider more specific traffic factors to provide more accuracy. In addition, we need further researches on the minimum connectivity requirement to guarantee data delivery without loss of data.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2011-0013931).

References

[1] X. Lin and H. Chen, "A secure and efficient RSU-aided bundle forwarding protocol for vehicular delay tolerant networks", Wireless Communications and Mobile Computing, vol. 11, no. 2, (**2011**), pp. 187-195.

[2] M. Raya and J. Hubaux, "Securing vehicular ad hoc networks", Journal of Computer Security, vol. 15, no. 1, (**2007**), pp. 39-68.

[3] J. Sun and Y. Yang, "RSU localization model and simulation optimization for VII network", Transport, vol. 26, no. 4, (**2011**), pp. 394-402.

[4] O. Trullols, M. Fiore and C. Casetti, "Planning roadside infrastructure for information dissemination in intelligent transportation systems", Computer Communications, vol. 33, no. 4, (**2010**), pp. 432–442.

[5] R. Whitaker, L. Raisanen and S. Hurley, "The infrastructure efficiency of cellular wireless networks", Computer Networks, vol. 48, no. 6, (**2005**), pp. 941–959.

[6] Y. Xiong, J. Ma, W. Wang and J. Niu, "Optimal Roadside Gateway Deployment for VANETs", Przeglad Elecktrotechnicznyt , ISSN 0033-2097, R. 88 NR 7b (**2012**).

[7] H. Hartenstein and K. Laberteaux, "VANET Vehicular Applications and Inter-Networking Technologies", Wiley, (**2010**).

[8] E. Amaldi, A. Capone, M. Cesana and F. Malucelli, "Optimizing WLAN radio coverage", Proceedings of IEEE ICC, (**2004**).

[9] B. Aslam, F. Amjad and C. C. Zou, "Optimal Roadside Units Placement in Urban Areas for Vehicular Networks", Proceedings of ICSS 2012, (**2012**), pp. 423-429.

[10] E. Cavalcante, A. Aquino, G. Pappa and A. Loureiro, "Roadside unit deployment for information dissemination in a VANET: an evolutionary approach", Proceedings of GECCO (Companion), (**2012**), pp. 27-34.

[11] J. Chi, Y. Jo, H. Park and S. Park, "Intersection-Priority based Optimal RSU Allocation for VANET", to appear in Proceedings of IEEE ICUFN 2013, (**2013**).

[12] C. Lochert, B. Scheuermann, C. Wewetzer, A. Luebke and M. Mauve, "Data Aggregation and Roadside Unit Placement for a Vanet Traffic Information System", Proceedings of ACM VANET, (**2008**).

[13] S. Park, B. Aslam and C. Zhou, "Long-term Reputation System for Vehicular Networking based on Vehicle's Daily Commute Route", Proceedings of IEEE CCNC 2011, (**2011**), pp. 457-462.

[14] M. Rashidi, I. Batros, T. K. Madsen, M. T. Riaz and T. Paulin, "Placement of Road Side Units for floating car data collection in highway scenario", Proceedings of Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), (**2012**), pp. 114-118.

[15] M. Rebai, L. Khoukhi, H. Snoussi and F. Hnaien, "Optimal placement in hybrid VANETs-sensors networks", Proceedings of Wireless Advanced (WiAd), (**2012**), pp. 54-57.

[16] A. Studer, E. Shi, F. Bai and A. Perrig, "Tacking Together Efficient Authentication, Revocation and Privacy in VANETs", Proceedings of SECON, (**2009**), pp. 1-9.

[17] K. Tutschku, "Demand-based radio network planning of cellular mobile communication systems", Proceedings of IEEE INFOCOM, (**1998**).

[18] M. Wright, "Optimization methods for base station placement in wireless applications", Proceedings of IEEE VTC, (**1998**).

Authors

Jeonghee Chi

2006. 2, Ph.D at Chungbuk National University, Korea

2007. 3 ~ Present, (Lecture) Professor, Konkuk University, Korea

Research Interests: Data management in mobile ad-hoc networks, Location-based information systems, Vehicle ad hoc network

Yeongwon Jo

2007. 3 ~ Present, Undergraduate Student, Konkuk University, Korea

Hyunsun Park

2009. 3 ~ Present, Undergraduate Student, Konkuk University, Korea

Taehyeon Hwang

2008. 3 ~ Present, Undergraduate Student, Konkuk University, Korea

Soyoung Park

2006. 2, Ph.D at Ewha Womans University, Korea

2006. 1. ~ 2010. 1, Post-doctoral Researcher, University of Central Florida, U.S.A

2010. 3 ~ 2012. 2, Full-time Lecturer, Ewha Womans University, Korea

2012. 3 ~ Present, (Lecture) Professor, Konkuk University, Korea

Research Interests: Cryptography, Network security, Vehicular network