This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2016.2598253, IEEE Transactions on Vehicular Technology

A New Comprehensive RSU Installation Strategy for Cost-Efficient VANET Deployment

Donghyun Kim, *Senior Member, IEEE,* Yesenia Velasco, *Student Member, IEEE,* Wei Wang, R.N. Uma, *Member, IEEE,* Rasheed Hussain, *Member, IEEE,* Sejin Lee

Abstract—Recently, the studies on vehicular adhoc network (VANET) are booming due to the huge potential. Road side unit (RSU) is a key component of the VANET infrastructure connecting mobile vehicles and the rest of the infrastructure. To maximize the availability of RSUs, RSUs should be densely deployed. Otherwise, blind spots may exist in which vehicles lose the connection to the infrastructure. Unfortunately, the massive deployment of RSUs to seamlessly cover the whole area of interest, which could be a vast metropolitan, can be very expensive. As the effectiveness and the benefits of the VANET are not fully proven yet, such large scale deployment can hardly be a viable option as of today. Motivated by this observation, this paper investigates a new strategy to best deploy RSUs so that their spatio-temporal coverage is maximized under a limited budget. In detail, for the first time in the literature, we consider an innovative RSU deployment framework, which is a well-balanced combination of three different approaches, deploying RSUs on static locations, public mobile transportation, and fully controllable vehicles owned by the local government. We first introduce a new strategy to abstract a map of city area into a grid graph. Then, we formulate the problem as a new optimization problem and show its NP-hardness. To solve this problem, we transform this problem into another optimization problem. Then, we propose a new polynomial running time approximation algorithm for the problem and show that the performance ratio (the ratio between the quality of an output of the proposed algorithm and the quality of the best possible solution) is at least half of the best possible ratio. We also conduct simulations under various setting to study the effectiveness of the proposed approach.

Index Terms—Vehicular ad-hoc networks, road side unit deployment, approximation algorithm, graph theory, optimization.

1 Introduction

These days, vehicular ad hoc network (VANET) is getting more attention due to its huge potential. Originally, the concept of VANET has been introduced to improve driving safety. Recently, VANET is being considered as a platform to enable a wide range of commercial applications such as remote vehicle personalization and diagnostics, Internet access, digital map downloading, real time video relay, and value-added advertisement [6].

In the literature, the term, VANET, refers to a wireless adhoc network of mobile vehicles with optional infrastructure support. An (infrastructure-independent) VANET can be constructed spontaneously by a group of VANET nodes (mobile vehicles) moving nearby without relying on any infrastructure. On the other hand, lots of emerging applications of (infrastructure-dependent) VANET exploits various public and private infrastructure such as public/private cloud, government authority server, etc. In this paper, we consider the later types of VANETs. Compared to the infrastructure-independent VANET, infrastructure-dependent VANET is uniquely characterized by the heavy presence of back-end infrastructure, in particular, road side units (RSUs) [7]. Generally speaking, RSUs are relay nodes connecting the VANET to the outside networks such as the Internet as well as providing a hybrid routing path combining wired and wireless links for high-speed large-capacity communication among distant VANET nodes. Naturally, RSU is a key component for cooperative and distributed applications in VANET. Nowadays, RSUs are also considered for different roles such as traffic directories, data disseminators, security management, location servers, and service proxies.

Due to the significance of the roles of RSUs in VANET, the proper distribution of RSUs is of great importance to improve the service quality of VANET. Naturally, this issue has attracted a lot of attention. In [8], Barrachina et al. identified that the cost of densely deploying RSUs could be a major hinderance to make VANET service ubiquitously accessible. Their report also shows that as the traffic density of an area differs over time, maximizing the utility of fixed RSUs is very challenging. In [9], Aslam et al. considered to deploy a limited number of

- D. Kim is with Department of Computer Science, College of Computing and Software Engineering, Kennesaw State University, Marietta, GA 30060, USA. E-mail: donghyun.kim@kennesaw.edu.
- Y. Velasco and R.N. Uma are with Department of Mathematics and Physics, North Carolina Central University, Durham, NC 27707. E-mail: yvelasco@eagles.nccu.edu, ruma@nccu.edu.
- W. Wang is with School of Mathematics and Statistics, Xi'an Jiaotong Univ., Xi'an, China. E-mail: wang_weiw@163.com.
- R. Hussain is with Department of Computer Science, Innopolis University, Kazan, Russia. E-mail: rasheed1984@gmail.com.
- S. Lee is with Division of Mechanical and Automotive Engineering, Kongju National University, Cheonan, 31080, South Korea. E-mail: sejiny3@kongju.ac.kr.
- S. Lee is the corresponding author.

- *The preliminary version of this paper has been appeared in Proceedings of International Workshop on Computing, Networking and Communications (CNC) [5].*

fixed RSUs to maximize their utilization over time. Later, Wu et al. [10], Liang et al. [11] considered similar issues of best deploying fixed RSUs. More recently, several researches are conducted to use public infrastructure such as buses and taxis to disseminate messages [12], [13] or route messages by using them as data mules [14], [15], [16].

While deploying RSUs on static locations or on public transportation were reasonable ways to deploy RSUs, another plausible scenarios would be deploying RSUs on government vehicles, which are fully controllable. However, to the best of our knowledge, such problem has not been considered yet. To fill this void, this paper proposes an innovative strategy to unify and complement the existing approaches to best deploy available RSUs to maximize their coverage. More specifically, we assume that there is a budget limitation to deploy RSUs, known costs to deploy each RSU on (a) a fixed location, (b) a public transportation such as a bus and a light rail, whose routes are known in advance, but not controllable, and (c) a fully controllable vehicle, which is owned by the local government, as well as the statistical information of the traffic density over each area. Then, we introduce a new strategy to best deploy RSUs using the three different types of deployment strategies under the limited budget.

The main contribution of this paper is that to the best of our knowledge, this is the first paper in the literature to consider three different RSU deployment strategies on a unified framework; static, mobile but not controllable, and mobile and fully controllable. Under the assumption that there is only light traffic and therefore, traffic jam is negligible, we introduce a new strategy to abstract a given metropolitan map into a grid graph such that when a fixed RSU is deployed over a point on the grid graph, then the whole region corresponding to the point in the map can be covered by the RSU placed on the center of the region. Then, we convert the problem of our interest into a new NP-hard optimization problem, namely the generalized budget coverage problem (GBCP), and show that it is NP-hard. Next, we transform GBCP to a new optimization problem called the budgeted maximum coverage problem with cardinality constraint and propose a new polynomial time approximation algorithm for it. Most of all, *we show that the performance ratio of this algorithm is at least half of the best possible*. We conduct simulation to study the performance of the proposed approach under different parameter setting. Our result shows that the cost of deploying each type of RSU has a strong impact on the coverage of the RSU. Also, the simulation result shows that our framework provides a cost-effective solution compared to the case adopting a single deployment strategy. It also shows that our algorithm works well under moderate traffic jam.

The rest of this paper is organized as follows. Related work is discussed in Section 2. The formal definition of the problem and its justification are in Section 3. We introduce a transformation of the problem into another optimization problem in Section 4 and propose a new polynomial time approximation algorithm for it in Section 5. In Section 6, we present our simulation result and analyze it. Finally, we conclude this paper in Section 7.

2 Related Work

In this section, we outline the state of the art regarding the role of the public buses and the deployment of RSUs in a VANET. First we start with the RSUs deployment in VANET. VANET is composed of basically two major kind of entities, i.e. mobile vehicles and the roadside infrastructure. The movement of the mobile vehicles is limited by the road topology and the RSUs must be deployed in optimized locations at roadside for maximum performance. Roadside infrastructure is used for a number of purposes ranging from data ferrying and routing to location-based services [7]. To date, there have been proposed a number of schemes that suggest different RSUs deployment strategies.

In [37], Yang et al. studies a RSU deployment problem which aims to minimize the number of static RSUs while satisfying a given objective. By structure, this is a dual problem of ours, which aims to maximize the (spatio-temporal) coverage with a given number (budget) of static RSUs. Therefore, a solution for their problem is not applicable to ours.

The problem in [36] aims to deploy a given number (budget) of static RSUs so that their own objective function is maximized. We would like to emphasize that the static RSU deployment strategy in our comprehensive RSU placement algorithm places RSUs to the locations with highest weights in a greedy manner, and therefore it is an optimal algorithm and no other static RSU deployment algorithm can work better than ours under our performance metric. Furthermore, Section 6.3 shows that our comprehensive approach outperforms such greedy deployment strategy of static RSUs. This proves that our algorithm works better than [36] in our performance metric. Besides, the algorithm in [36] assumes that it knows how many cars will be on the street, and what are the confrontation probabilities among them at each location of the city. This is highly privacy violating information and hard to collect in the real world. As we do not assume such information available, the result in [36] cannot be used for our purpose.

Kitani et al. [17] proposed a public transportation based mechanism for message ferrying. They aim at the low density urban areas where buses provide connectivity and provide the other vehicles with traffic information. Another similar approach was followed by Luo et al. [18] where they used buses as a backbone network for data delivery and next-hop relays in the routing decisions like Lai et al. [19] who also use buses in routing decisions. Nevertheless this information is local and we assume that other vehicles on their own can have such traffic information through beacon messages.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2016.2598253, IEEE Transactions on Vehicular Technology

On the other hand Dow et al. [20] proposed bus-based service information discovery where bus routes are used to create a backbone structure to avoid the broadcast storm problem. Buses have also been used for efficient broadcast strategies by Holzer et al. [21]. Their scheme namely BROADTRIP uses network coding technique to reduce the number of retransmission in a location-based broadcasting.

A multi-modal message dissemination scheme with the help of public transportation is proposed by Zhang et al. [12]. In their scheme, public buses assist the message dissemination in public taxis and they studied the interconnection time among vehicles. In another work by Xu et al. [13] public buses have been used to detect the traffic congestion in the urban localities. They detect the traffic congestion of the area based on the travel time of the bus. Lan et al. [14] studied the feasibility of the public buses as data mules for traffic monitoring. Their proposed scheme is based on the assumption that all buses, bus stops and the traffic lights installed on the roads, are equipped with wireless devices in order to communicate with each other and the neighbors. For granularity, they calculate the average travel time of the bus for certain time.

Jiang et al. [15] proposed a bus vehicular network where buses are used with deployed RSUs for traffic information dissemination for better coverage. Another bus-based next-hop forwarding scheme has been proposed by Huang et al. [16]. They analyze the upper and lower bounds of the multicast capacity of bus-assisted VANET where ordinary vehicles when sending messages to other nodes, select the buses as next-hop forwarding nodes. Recently Tonguz et al. [24] proposed another scheme where cars are used as RSUs in VANET. Their scheme is also inspired by the initial deployment of the dedicated short range communications (DSRC) and the problems faced by such deployment. Therefore they use the ordinary cars as temporary RSUs. Whenever a car acts as a temporary RSU, it makes brief stops during which they act as Communication Bridge for other vehicles in the network. However, while this scheme seems practical, the stops of the ordinary vehicles (temporary RSUs) still leave a question mark on the robustness and reliability of the system.

Fillipini et al. [26] proposed a game theoretic-based technique for RSU deployment. Their scheme is based on concurrent decision on part of the operators to deploy RSU in an optimized fashion. In another work, Tao et al. [27] target the message propagation efficiency and power consumption in VANET, and devise the RSU deployment strategies to improve upon the aforementioned factors. Similarly Liu et al. [28] proposed RSU deployment mechanism to smoothen the content distribution in VANET and thus-forth their proposed scheme covers single dimension, i.e. contents distribution in VANET. Another similar scheme has been proposed by Mehar et al. [34] where they make the RSU deployment decision for the delay-sensitive VANET applications. Their aim is to deploy RSUs in such an optimal way that can improve the end-to-end delay for the applications as well as to reduce to deployment cost. Farsi and Szczechowiak [29] targeted the car density and traffic data to decide on the optimal locations for RSU deployment. They input these matrices to the algorithm to find out the optimal spots for static RSU deployment in VANET. Another optimal RSU deployment mechanism is proposed by Patra et al. [30] where they use analytic hierarchy process (AHP). They also take RSU to RSU communication delay as a performance metric. Nonetheless, their scheme is static and does not take the variations in the mobility into account.

Aslam and Zou [31] proposed a balloon optimization method to deploy RSUs along the highways in the initial stages of VANET with minimum budget and limited resources. Their aim is to minimize the message propagation delay from one RSU to another neighbor RSU. Similarly Rizk et al. [32] proposed a greedy method to deploy RSUs in both urban and rural areas based on overlap-based greedy method (OGM). They mainly consider two dimensions for the optimal RSU deployment, sites of interests with higher probability and RSU coverage radius. Another similar scheme has been proposed by Makkavi et al. [33] where they consider sites of interest for RSU deployment. Their scheme is cumulative weight based method (CWM) where CWM decides on the highest weight first, in the RSU distribution process. However, these factors may not always guarantee full coverage for RSUs in both rural and urban localities.

Based on our comprehensive survey, we can conclude that our work is the first effort to consider the comprehensive deployment approach of insufficient number of RSUs using three different different strategies at the same time.

3 Problem Statement

In this paper, we investigate how to deploy various static and mobile RSUs on a metropolitan area so that the coverage of RSUs can be maximized under a limited budget. We make the following assumptions.

(a) RSUs can be deployed on a static location (D-Type 1), on mobile public transportation such as buses and light rails, which are mobile but not controllable (D-Type 2), and/or on government vehicles which are fully controllable by need (D-Type 3).
(b) The cost to deploy an RSU on each deployment type is fixed and known in advance.
(c) In case of Deployment D-Type 2, each mobile transportation does not suffer from any delay, and their travel schedule is known. Note that this is mostly true for light rails, as well as for buses within a city area without heavy traffic jam. This assumption implies that the location of each transportation at any moment of a day is known in advance.
(d) D-Type 3 does not suffer from traffic jam. In practice, this can be handled by constructing their travel schedule under very low speed.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2016.2598253, IEEE Transactions on Vehicular Technology

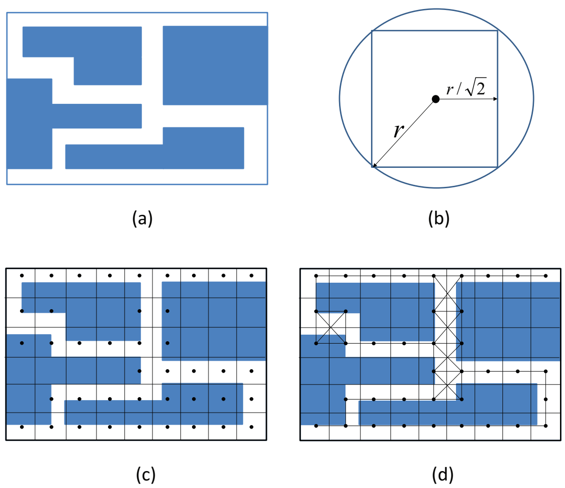


Fig. 1: Abstraction of a metropolitan area map M into a graph.

(e) The significance of each region (e.g. traffic load) within the metropolitan area is available. This can be obtained by collecting the relevant statistical information over time.

In the following, we first introduce a new way to abstract a metropolitan area map M into a graph model. Then, we formulate the problem of our interest as an optimization problem on the graph. Last, we show the problem of our interest is NP-hard.

3.1 Abstraction of Topology

In this paper, we assume the shape of the map M of a metropolitan area of our interest is a rectangle, e.g. Fig. 1(a). Suppose the communication range of both VANET nodes and RSUs are equal to r. Next, we partition M into regular squares whose height and width are $(r/\sqrt{2}) \times 2$, e.g. Fig. 1(b). Observe that the RSU at the center of the grid square with this length and height is accessible from a VANET node in any location within the grid square. Now, we represent each grid square as a point and obtain a set of grid points representing the whole map, e.g. the points in Fig. 1(c). Finally, we construct a topology graph $G = (V, E)$ such that V is the set of central points of the grid squares. For each pair of points $u, v \in V$, $(u, v) \in E$ if the two squares, whose central points are u, v, are adjacent in M (i.e. the grid squares share exactly one common edge), e.g. Fig. 1(d).

Once the topology graph is constructed, then we assign a weight on each node, which implies the importance of the node, e.g. business of the grid square. By definition, if a grid space includes a popular spot with more traffic, then the corresponding node will get a higher weight. Note that the weight of a node may vary during the day as the traffic situation of a metropolitan area changes over time. Therefore, if we divide a day into T consecutive time slots, we can obtain a set of temporal graphs [25] $\mathcal{G} = \{G_0 = (V_0, E_0, w_0 : V_0 \to R^+), G_1 = (V_1, E_1, w_1 : V_1 \to R^+), \cdots, G_T = (V_T, E_T, w_T : V_T \to R^+)\}$ as an input of our problem of interest. Note that $V_0 = V_1 \cdots = V_T$ and $E_0 = E_1 \cdots = E_T$, but the graphs only differ in node weights.

We would like to emphasize that in the real world scenario, we actually need some additional number of RSUs due to various physical effects such as non-Line-of-Sight issue. However, such issues can be easily solved by adding a few extra RSUs, and therefore, we will proceed our discussion with the $\mathcal{G}$.

3.2 Problem Definition using Abstracted Topology

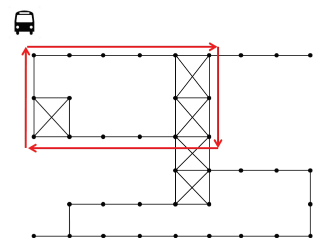


Fig. 2: Location of a mobile public infrastructure varies over time.

Previously, we explained how to obtain a set of temporal graphs $\mathcal{G}$ which represents the logical relationships among the subregions (grid squares) in the metropolitan area and their corresponding significance over time. We can easily observe that some deployment strategies (D-Type 1 and D-Type 2) over the set of temporal graphs have the following characteristics.

(a) In case of a static RSU deployment (D-Type 1), once we decide to deploy one RSU on a node in G_0, it also represents the deployment of the same node on the rest of the graphs in $\mathcal{G}$ as the RSU is not mobile. This is because by our construction, the graphs are different only in node weights.
(b) In case of D-Type 2, as we can see in Fig. 2, the location of a mobile public transportation varies over time. Therefore, once we decide to deploy an RSU on such mobile unit, the exact location of the RSU may be changed in each graph in $\mathcal{G}$.
Still, as the travel schedule of the RSU is known in advance (following Assumption 3, see Fig. 3), we know which node in each $G_i \in \mathcal{G}$ will be covered by an RSU on the mobile transportation exactly. This means that when we deploy multiple RSUs using D-Type 2 strategy, the group of nodes covered by the RSUs changes over time, but we can compute what they are at each moment exactly.

Now, let us introduce one related problem.

Definition 1 (Budgeted Maximum Coverage Problem)**.** *Given a budget B, a set $S = \{s_1, s_2, \cdots, s_n\}$, their corresponding weights $W = \{w_1, w_2, \cdots, w_n\}$, a collection $\mathcal{S}$ of subsets $\{S_1, S_2, \cdots, S_m\}$ of S, and their corresponding cost $C = \{c_1, c_2, \cdots, c_m\}$, the budgeted maximum coverage*

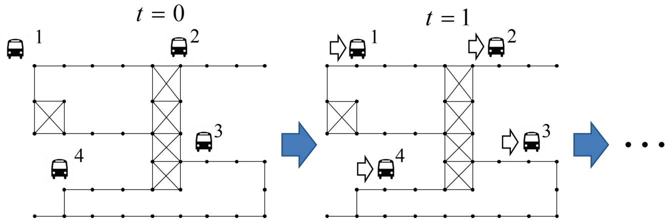


Fig. 3: The subset of nodes covered by the RSUs attached to mobile public infrastructures are changing over time, but can be predicted.

problem is to find a subset $\mathcal{S}' \subset \mathcal{S}$ *to maximize* $\sum_{i:s_i \in S} w_i \cdot y_i$ *subject to*

(a) $\sum_{1 \le j \le m} c_j \cdot x_j \le B$, *// the cost of selecting the subsets is no greater than the budget limit.*

(b) $\sum_{j:s_i \in S_j} x_j \ge y_i$ *for each* s_i, *//* $y_i = 0$ *if for every* S_j *containing* s_i, $x_j = 0$.

(c) $0 \le y_i \le 1$, *//* $y_i = 1$ *if* s_i *is covered, i.e.* $x_j = 1$ *for some* S_j *including* s_i.

(d) $x_j \in \{0, 1\}$. *//* $x_j = 1$ *only if* S_j *is selected, i.e.* $S_j \in \mathcal{S}'$.

In the definition, $x_j = 1$ if the subset S_j is selected, otherwise 0 (Constraint (d)). Also, $y_i = 1$ if s_i is covered by any subset, otherwise 0 (Constraint (c)). By Constraint (b), if no subset including s_i is selected, $\sum_{j:s_i \in S_j} x_j$ becomes 0, and this enforces $y_i = 0$. Otherwise, to maximize the objective goal, $y_i = 1$ is always selected. Note that the budgeted maximum coverage problem is a known NP-hard problem [23].

Based on our discussion so far (without considering D-Type 3), we can construct a budgeted maximum coverage problem instance as follows: Given a budget B, we first construct a set $S = V(G_0) \bigcup V(G_1) \bigcup \cdots \bigcup V(G_T)$, where $V(G_i)$ is the set of nodes in G_i with corresponding node weights. Note that the weight of each node in S is known in advance and we have W. Then, we create an empty collection $\mathcal{S}$ of subsets of S. Then, for each node $v_i \in G_0$ and its identical nodes in the rest of the temporal graphs $G_1, \cdots, G_T$, we create a subset S_i and add it to $\mathcal{S}$. Then, each S_i represents a grid point covered by a static RSU deployed on the v_i over time period $[0, \tau]$, where $\tau = T \times \mu$, and μ is the length of each time slot. Next, for each public mobile transportation, we collect the set of grid nodes covered over time period $[0, \tau]$ by an RSU over the mobile transportation and construct a new subset. Then, we also add this subset to $\mathcal{S}$. This subset represents the set of grid points covered by the RSU attached to the mobile transportation over time. For each subset in $\mathcal{S}$, we assign the known cost to the subset which is corresponding to the cost to deploy and operate an RSU to cover the nodes in the subset over time, then we have C. Then, we obtain a budgeted maximum coverage problem instance.

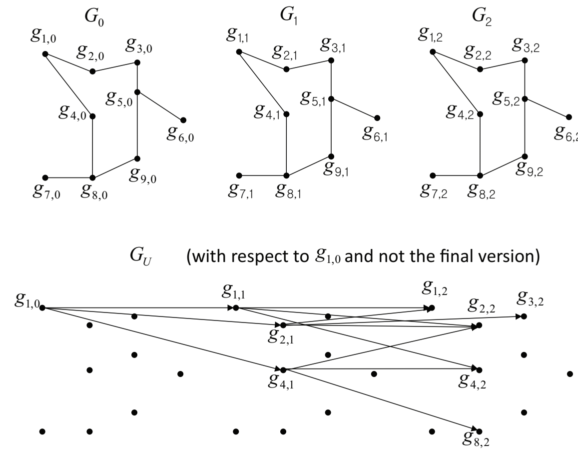


Fig. 4: This figure shows the movement of a fully controllable mobile node over time becomes one subset in S in the budgeted maximum coverage problem instance. For instance, if we decide to move the node through $g_{1,0} \rightarrow g_{1,1} \rightarrow g_{2,2}$, a corresponding subset $\{g_{1,0}, g_{1,1}, g_{2,2}\}$ is added to $\mathcal{S}$. This movement implies that a mobile node starts from $g_{1,0}$ and stays there for one more time unit and then finally moves to $g_{3,2}$.

Finally, let us discuss how the consideration of D-Type 3 deployment strategy (fully controllable) will impact our formulation so far. Given a starting location of fully controllable node with an RSU in G_0, the node can always stay at the same location or move to adjacent location in the next graph in the set of temporal graphs. From this observation, we can construct a new directed acyclic graph (DAG) $G_U = (V_U, E_U)$ such that $V(G_U) \leftarrow \bigcup_{G_i \in \mathcal{G}} V(G_i)$ and for each $v \in V(G_i)$ and $u \in V(G_{i+1})$ pair, there exists a directional edge from v to u in E_U only if $v = u$ or v and u are adjacent in G_i (which also means that they are adjacent in G_{i+1}). Fig. 4 shows that under such construction, a feasible path of a mobile node which is located at $g_{1,0}$ is a path from G_0 to G_T, and the number of such paths is exponential. This means that while our problem of interest is similar to the budgeted maximum coverage problem, it is significantly more challenging as there are so many choices to construct a subset for each fully controllable mobile node, which becomes a new subset into $\mathcal{S}$ in the formulation of the budgeted maximum coverage problem.

Below is the formal definition of our problem of interest.

Definition 2 (Generalized Budgeted Maximum Coverage Problem)**.** *Given*

(a) a DAG $G_U = (V_U, E_U)$ *with their corresponding node weights* W,

(b) a collection $\mathcal{S}$ *of subsets of* V_U *and their corresponding cost* $\mathcal{C}$,

(c) a budget B,

the generalized budget coverage problem is to construct a subcollection of subsets of $\mathcal{S}$ *by (a) computing subsets, each of* *which represents the grid points covered by fully controllable mobile nodes with an RSU which moves over G_U and (b) selecting additional subsets, each of which represents the set of grid points covered by either a static RSU or an RSU attached to a public mobile transportation, such that the total cost to deploy the RSUs is under the limited budget B and the coverage of RSUs over time is maximized, which is the sum of the weight of each grid point in any $G_i \in \mathcal{G}$ covered by the subsets in the subcollection.*

It is worthwhile to notice that the constraints (a) and (b) are not necessarily in that order. That is, we may compute the paths for fully controllable mobile nodes first and then select some more subsets from $\mathcal{S}$, or vice versa. Also, it is allowed to do interchangeably. Meanwhile, it is easy to see that the generalized budget coverage problem is NP-hard as its simplest version without any fully controllable mobile node (this is possible by assuming the cost to operate each fully controllable mobile node is greater than the given budget), is equivalent to the budgeted maximum coverage problem.

Remark 1. *The cost to deploy a fully controllable mobile node with an RSU and operating them for whole one year can be much higher than deploying an RSU on a fixed location or a mobile public transportation. This means that the maximum number k of fully controllable mobile nodes under a limited budget may not be huge.*

Due to Remark 1, in the rest of this paper, we will focus on a variation of generalized budgeted maximum coverage problem, in which the number of available fully controllable mobile nodes is specifically given as a positive integer k. Clearly, by solving this problem in polynomial time, we can solve the original version in polynomial time by selecting the best result among all possible choices of the number of fully controllable mobile nodes, which is bounded by the limited budget B divided by the cost of deploying one fully controllable mobile node.

4 Budgeted Maximum Coverage Problem with Cardinality Constraints

Now, we reformulate the generalized budgeted maximum coverage problem, whose input is $\langle \mathcal{G}, W, C, k, B, T\rangle$ to a new optimization problem namely the budgeted maximum coverage problem with cardinality constraints (BMCP-CC), see Definition 3, as follows. First, from $\mathcal{G}$ and W, we can construct a DAG $\Gamma = (V', E')$ as follows: $V' \leftarrow V(G_0) \cup V(G_1) \cup \cdots \cup V(G_T)$. For any two nodes u and v, there exists a direct edge from u to v in E' only if (a) $u \in G_i$ is same as (a copy of) $v \in G_{i+1}$ or (b) $u \in G_i$ and $v' \in G_i$ are adjacent in G_i, where v' is a copy of $v \in G_{i+1}$. Note that this construction is similar to the construction of G_U. Without loss of generality, suppose

$$V' = \{v_1^0, v_2^0, \cdots, v_n^0; v_1^1, v_2^1, \cdots, v_n^1; \cdots, v_1^T, v_2^T, \cdots, v_n^T\},$$

where each v_i^j represents the i-th node of V in the j-th moment $t = j$ for $j = 0, 1, \cdots, T$, i.e. v_i^j is a copy of v_i in $V(G_j)$. There is a directed edge from v_i^j to v_k^{j+1} only if either nodes v_i and v_k are adjacent in graph G_i or $i = k$; see Fig 5 for example.

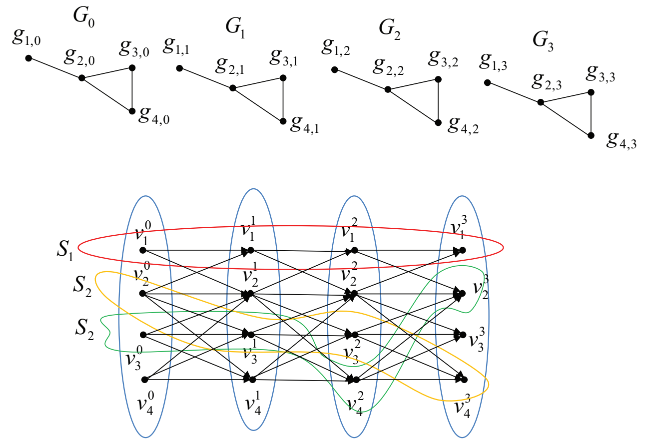


Fig. 5: This figure shows how $\mathcal{G}$ from a generalized budgeted maximum coverage problem instance is used to construct Γ in a budgeted maximum coverage problem with cardinality constraints instance.

Now, we are given a collection of subsets

$$\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2 = \{S_1, S_2, \cdots, S_m\} \cup \{S_1', S_2', \cdots, S_l'\},$$

(a) where $\mathcal{S}_1$ **represents the set of all possible trajectories for fully controlled vehicles in** Γ, which consists of all possible subsets of nodes constituting a directed path from some nodes in time $t = 0$ to time $t = T$ in graph Γ. Note that the cardinality of $\mathcal{S}_1$ can grow exponentially in terms of T and n, and cannot be given explicitly.

(b) where each subset S_i' in $\mathcal{S}_2$ contains $T + 1$ nodes from V', which forms a directed path from some nodes in time $t = 0$ to some nodes in time $t = T$ in graph Γ corresponding to the trajectory of the stationary (and mobile) RSUs. The cost $c(S_i')$ of S_i' is given in advance, the coverage benefit of S_i' is $w(S_i') = \sum_{v \in S_i'} w(v)$.

Definition 3 (Budgeted Maximum Coverage Problem with Cardinality Constraints, BMCP-CC)**.** *Using notations above, given a positive integer k and a budget B, we are asked to select k subsets from $\{S_1, S_2, \cdots, S_m\}$ and some subsets from $\{S_1', S_2', \cdots, S_l'\}$ with total cost no more than B, such that the total weights of nodes covered is maximized.*

5 A New Approximation Algorithm for BMCP-CC

As BMCP-CC is NP-hard, in this section, we introduce a new α-approximation algorithm for it, where $\alpha = \frac{1}{2}(1 - \frac{1}{e})$. This means that the quality of the output of the algorithm is at least the half of the best possible solution. The basic idea follows from the existing work [23]. However, since the number of subsets in $\mathcal{S}_1$ can be exponentially large, even a simple greedy strategy

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2016.2598253, IEEE Transactions on Vehicular Technology

does not work (it takes exponential time if we simply enumerate all possibilities). This means that it is not possible to directly apply the modified greedy strategy in [23] to solve BMCP-CC. Fortunately, we manage to make the greedy strategy work, by exploiting the special structures of Γ (which is a directed acyclic graph), based on dynamic programming strategy and a clever node weight reassignment procedure.

The basic idea of our algorithm is based on the greedy strategy. The algorithm mainly consists of two independent stages. In the first stage, we apply the greedy algorithm for the Maximum k Coverage Problem with $\mathcal{S}_1$ as the input. The second stage uses greedy strategy to solve the budgeted maximum coverage problem with input $\mathcal{S}_2$ and budget B. The algorithm takes the union of solutions obtained in two stages as the outputs.

Algorithm 1. Algorithm for BMCP-CC
Input: $(G = (V, E), T, B, k, \mathcal{S}_1)$
Output: k subsets from $\mathcal{S}_1$ and a sub-collection $\mathcal{S}' \subset \mathcal{S}_2$ with cost at most B.

- Step 1 (**Greedy Algorithm for Maximum k Coverage**). $\mathcal{A}_1 \leftarrow \emptyset$; Select subsets in $\mathcal{S}_1$ into $\mathcal{A}_1$ in a greedy manner, i.e., first find a subset $S \in \mathcal{S}_1$ which covers nodes with maximum total weight, then at each round, pick a subset in the remaining subsets in $\mathcal{S}_1$ which covers the uncovered nodes in V with maximum total weight, until there are k subsets selected in $\mathcal{A}_1$. Let $\mathcal{A}_1 = \{S_1, S_2, \cdots, S_k\}$ be the collection of selected subsets in this step.
- Step 2 (**Greedy Algorithm for the Budgeted Maximum Coverage Problem**). Apply the $(1 - 1/e)$-approximation algorithm in [23] for the budgeted maximum coverage problem over BMCP-CC with $k = 0$ (which means we have no fully controllable mobile nodes. Under this condition, BMCP-CC becomes a traditional budgeted maximum coverage problem). Suppose after running this algorithm, we obtain $\mathcal{A}_2 = \{\hat{S}_1, \hat{S}_2, \cdots, \hat{S}_p\}$, for some $p \le l$.
- Step 3. Let $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$, output $\mathcal{A}$.

The main consideration is how to efficiently select subsets from $\mathcal{S}_1$ such that the greedy strategy works. Next, we show Step 1 can be done in polynomial time, by using dynamic programming. We remark, however, that finding the longest paths in general graphs is NP-hard.

Step 1(a). Finding the longest directed path in Γ (starting from a node in $t = 0$ and ending at a node in $t = T$).
Input: $(G = (V, E), w, T,)$
Output: the longest path staring from $t = 0$ and ending at $t = T$.

(i) Transform Γ into an edge-weighted graph as follows: Let $e_{ij} = (v_i, v_j)$ be a directed edge. Then $w(e_{ij}) = w(i) + w(j)/2$ if $i = 0$ and $j \neq T$; $w(e_{ij}) = (w(i) + w(j))/2$, if $1 \le i, j \le T - 1$; $w(e_{ij}) = w(i)/2 + w(j)$, if $i \neq 0$ and $j = T$; and $w(e_{ij}) = w(i) + w(j)$ if $i = 0$ and $j = T = 1$. In this way, we induce the graph from Γ such that the weight of each edge in the induced graph is the sum of the half of the weights of the two endpoints of the corresponding edge in Γ. In this way, we can transform a problem on a node weighted graph into another problem on an edge-weighted graph.

(ii) (Dynamic Programming) In the (node weighted) temporal graph, the sum of nodes on the path from a node (could be static or mobile) in $t = 0$ to another node in $t = T$ represents its coverage. After the temporal graph is changed into the edge-weighted graph, the sum of edge weight of the same path on the induced graph is equivalent to the sum of the node weights on the path in the temporal graph. As a result, for each node in $t = 0$, the path with maximum edge weight (or mathematically equivalent to the longest path) to another node $t = T$ in the induced graph is the best trajectory for the node. Therefore, we shall find the longest path between any pair of nodes starting from time $t = 0$ and ending at time $t = T$. First, define the outgoing neighbors of a node v in Γ to be $N_{out}(v) = \{u | (v, u) \in E(\Gamma)\}$ and the incoming neighbors of v to be $N_{in}(v) = \{u | (u, v) \in E(\Gamma)\}$. Let $dist(v_i^0, v_k^j)$ be the distance of the longest path between v_i^0 and v_k^j. Then $dist(v_i^0, v_k^0) = 0$, $\forall i, k = 1, 2, \cdots, n$. Generally,
$$dist(v_i^0, v_k^j) = \max_{r \in N_{in}(v_k^j)} \{dist(v_i^0, r) + w(r, v_k^j)\},$$
for $i, k = 1, 2, \cdots, n$ and $j = 1, \cdots, T$. Based on above recursive relation, we can find all the longest paths from v_i^0 to any node v_k^T at time $t = T$.

(iii) After computing all the longest distance between v_i^0 and v_j^T for any $i, j = 1, 2, \cdots, n$. We can choose among them the longest paths from a node in $t = 0$ to a node in $t = T$.

Step 1(b). Procedure(Γ, T, B, k)
Input: (Γ, T, B, k)
Output: k subsets in $\mathcal{S}_1$.

(i) In the edge weighted graph Γ, find the first longest path, say $P_1 = v_{i_1}^0 v_{i_2}^1 \cdots v_{i_n}^T$;

(ii) Reset the node weights of P_1 to be zero in graph Γ, then reconstruct the edge weight of Γ according to the rule in Step 1(a). Find the longest path, say P_2, in Γ with new edge weights, by using Step 1(a);

(iii) Repeat the above process until k paths have been selected.

Now we have the following theorems.

Theorem 1. *Algorithm 1 runs in polynomial time.*

Proof. We show that both Step 1 and Step 2 in Algorithm 1 can be done in polynomial time.

First, let us estimate the time complexity of Step 1 (a), which consists of two sub-steps (i),(ii) and (iii). Clearly, the weight re-assignment process (i) can be done in $O(|E(\Gamma)|) = O(n^2 T)$, where n is the number of nodes. Sub-step (ii) is a standard dynamic programming which is used to compute the longest paths in graph Γ with varied edge weights. For each fixed $i = 1, 2, \cdots, n$, we maintain a table of size at most $n \times T$ in which the distances computed so far have been recorded. For each $1 \le j \le n$, $dist(v_i^0, v_j^k)$ can be computed within time $O(|v_{in}(v_j^k)| = O(n)$. So the k-th column of the table can be computed within $n \times |v_{in}(v_j^k)| = O(n^2)$. It follows that for each i, the table can be completed in $O(n^2)T$. Therefore, in (iii), all pairs of distances between v_i^0 and v_k^T can be computed in $O(n^3T)$, for $i, k = 1, 2, \cdots, n$. So the total complexity of Step 1 is $O(kn^3T)$.

Now we show Step 2 can also be done in polynomial time. Note that each subset in $\mathcal{S}_2$ represents a trajectory of a stationary (and mobile) RSU. Let N be the total number of trajectories of mobile RSUs available. Then the number of subsets in $\mathcal{S}_2$ is bounded by $n + N$. Thus, according to the polynomial time $(1 - \frac{1}{e})$-approximation algorithm in [23], the running time of Step 2 is bounded by a polynomial in n and N. □

Theorem 2. *Algorithm 1 is a $\frac{1}{2}(1 - \frac{1}{e})$-approximation for BMCP-CC, which is guaranteed to produce a solution at least $\frac{1}{2}(1 - \frac{1}{e})$ times the optimal solution.*

Proof. Let $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2 = \{S_1, S_2, \cdots, S_k\} \cup \{\hat{S}_1, \hat{S}_2, \cdots, \hat{S}_p\}$ be the solution of BMCP-CC obtained by Algorithm 1. Let $\mathcal{A}^* = \mathcal{A}_1^* \cup \mathcal{A}_2^* = \{S_1^*, S_2^*, \cdots, S_k^*\} \cup \{\hat{S}_1^*, \hat{S}_2^*, \cdots, \hat{S}_q^*\}$ be an optimal solution of BMCP-CC. Let OPT_1 be the cost of an optimal solution of the Maximum Coverage Problem by selecting k subsets from $\mathcal{S}_1$, and let OPT_2 be the cost of an optimal solution of the Budgeted Maximum Coverage Problem by only selecting some subsets from $\mathcal{S}_2$ with total costs at most B. Then it follows from [22] and [23] respectively that

$$w(\mathcal{A}_1) = w(S_1 \cup S_2 \cup \cdots \cup S_k) \ge (1 - 1/e)OPT_1, \quad (1)$$

$$w(\mathcal{A}_2) = w(\hat{S}_1 \cup \hat{S}_2 \cup \cdots \cup \hat{S}_p) \ge (1 - 1/e)OPT_2. \quad (2)$$

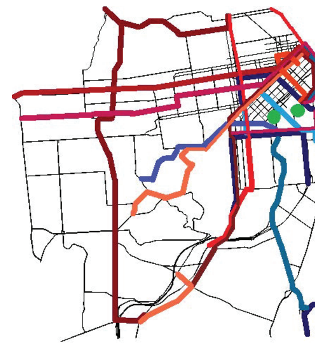
Note $w(\mathcal{A}_1 \cup \mathcal{A}_2) \ge \max(w(\mathcal{A}_1), w(\mathcal{A}_2)) \ge \frac{w(\mathcal{A}_1) + w(\mathcal{A}_2)}{2}$. It follows that

$$\begin{aligned} w(\mathcal{A}_1 \cup \mathcal{A}_2) &\ge \frac{w(\mathcal{A}_1) + w(\mathcal{A}_2)}{2} \\ &\ge \frac{1}{2}(1 - 1/e)(OPT_1 + OPT_2) \\ &\ge \frac{1}{2}(1 - 1/e)(w(\mathcal{A}_1^*) + w(\mathcal{A}_2^*)) \\ &\ge \frac{1}{2}(1 - 1/e)w(\mathcal{A}_1^* \cup \mathcal{A}_2^*) \\ &= \frac{1}{2}(1 - 1/e)w(\mathcal{A}^*), \end{aligned}$$

where $w(\mathcal{A}_1^*) = w(S_1^* \cup S_2^* \cup \cdots \cup S_k^*) \le OPT_1$ and $w(\mathcal{A}_2^*) = w(\hat{S}_1^* \cup \hat{S}_2^* \cup \cdots \cup \hat{S}_q^*) \le OPT_2$ follow from the fact that OPT_1 and OPT_2 are optimal solutions, respectively. This completes the proof. □

By the theorem, the performance ratio of our algorithm is $\frac{1}{2}(1 - 1/e)$. Similar to [23], as our problem without the fully controllable mobile node and the budget constraint (or with a huge budget) is essentially a maximum coverage problem, it is not possible to approximate no better than $(1 - 1/e)$ [35]. As a result, our algorithm achieves at least half of the best possible.

(a) Initial Graph



(b) This figure illustrates an example output of our algorithm. The 10 distinct red lines represents the trajectories of the RSUes on the buses, the 7 blue lines represents the trajectories of the fully controllable RSUes, and the 2 green circles represents the static RSUes. The overall coverage of 0.48 is achieved.

Fig. 6: An output of our algorithm under a small budget of 330 with the deployment cost 10, 10, and 30 for each of static, mobile, and fully controllable RSUs, respectively. The time frame considered is $T = 24$ hours, with increments of $t = 1$ minute.

6 Simulation Result and Analysis

In this section, we evaluate the performance of the proposed algorithm through a simulation.

6.1 General Simulation Setting

To construct the map on which our algorithm will feed on, we first obtain a map data from Open Street Map database [1], filter it using the Osmfilter command line tool [1], and produce a simplified map which only includes primary roads from the original map, e.g. Fig. 6(a). Once the simplified graph is obtained (shown in Fig. 6(a)), we applied the grid space on the map (with $r = 280 \times \sqrt{2}$ meters in Fig. 1(b), where the size of the whole map is 283,360 square meters and produce an

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2016.2598253, IEEE Transactions on Vehicular Technology

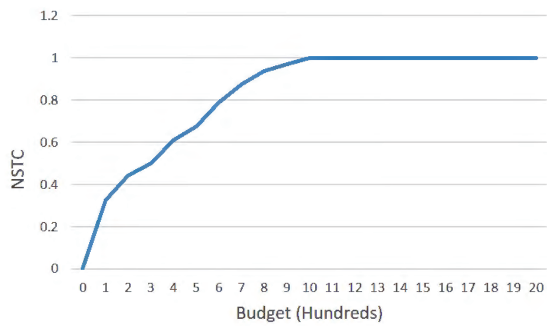


Fig. 7: The NSTC achieved by our algorithm changes over budget.

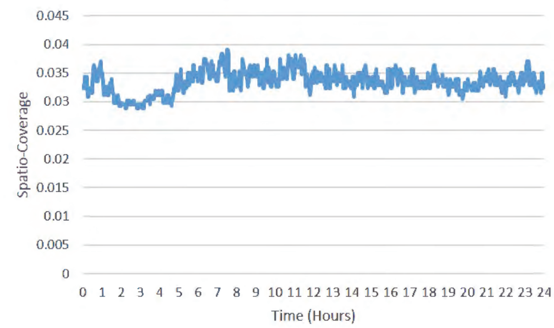


Fig. 8: The change of the normalized spaito-coverage achieved by our algorithm over a time span of 24 hours is given.

abstract graph (e.g. Fig. 1(d)) with 13,046 nodes. This means that for a vehicle to move from one grid square to another grid square, its speed should be around 35mph, which is a setting as a car in a big metropolitan area.

Public transportation routes are computed using information from the Metropolitan Transportation Commission (MTC) of San Francisco where data on routes, route times, and stops are available [2]. The routes are incorporated into the map with the aid of an open traffic simulation suite known as the Simulation of Urban Mobility (SUMO) [3]. Lastly, the weight assigned to each node is calculated using a reference of the citys traffic density distribution by the San Francisco County Transportation Authority [4]. The simulation code was coded using the C++ language (Visual studio 2015) on a Lenovo U310 personal computer (3rd Generation Intel Core i5-3317U (1.70GHz, 3MB), 4GB DDR3 RAM, Samsung 850 EVO 500GB, SATA III Internal SSD, Windows 10 Operating System), reading in xml files outputted by SUMO as the graph and routes of the mobile RSUs.

Under the settings, we run the algorithm and determine how many RSUs on static locations, on public transportation, and on fully controllable mobile nodes we need, respectively, to maximize the total normalized spatio-temporal coverage (NSTC). While computing the optimal numbers, at the same time, the algorithm also determines which public transportation should be selected and what the trajectory of each fully controllable mobile node with an RSU should be. Once the output comes out, each RSU is used during the period and the NSTC is computed. Note that during the rest of this section, we use hundred dollars as the unit for the budget and cost. The time frame considered is $T = 24$ hours, with increments of $t = 1$ minute. Fig. 6(b) illustrates an output of our algorithm under a certain parameter setting.

6.2 Performance of Proposed Algorithm using Real Data

In this section, we evaluate the performance of the proposed algorithm with respect to the budget increase. We consider the deployment cost 10, 10, and 30 for each

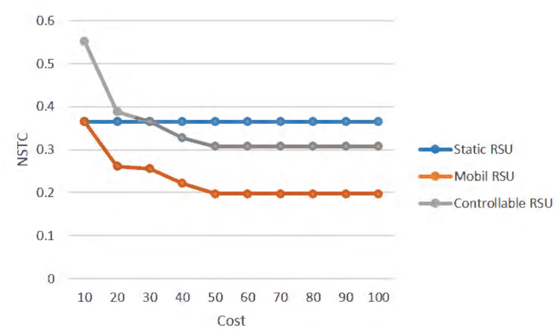


Fig. 9: NSTC change over cost.

of static, mobile, and fully controllable RSUs, respectively. Fig. 7 illustrates the change in NSTC as the budget increases. The rise in NSTC becomes less apparent with each augmentation of the budget, thus following a logarithmic curve. As shown in the figure, a budget of 4000 yields approximately the same results as that if the budget were 5000. The lack of a stark difference in spatio-temproal coverage would make 4000 the better choice. Once the map is fully covered, adding onto the budget will bring no added benefit and NSTC reaches at 1.

Fig. 8 shows the change of normalized spatio-coverage over time 8 under a small budget of 330. The fluctuations in normalized spatio-coverage are primarily due to mobile RSUs traveling to higher or lower weighted areas and routes not being continually offered at all times. In our simulation, the hours from 2 A.M to 4 A.M have the lowest normalized spatio-coverage since most mobile routes are not offered at that time. With a peak normalized spatio-coverage of 0.0391 and a low of 0.0288, our solution has a satisfactory level of stability with a range of 0.0103 and a mean of 0.0335. Higher stability could be achieved if a larger number of static RSUs are selected or fully controllable RSUs such that routes are continually offered.

Fig. 9 depicts the effects an increasing cost of deploying of an RSU type has on the overall NSTC given an initial condition. Such a condition is a budget of 120, cost of deploying a static and mobile RSU being 10, and cost of a fully controllable RSU being 30. Since the fully controllable RSUs provide much added coverage, lowering the cost of deployment will increase the number of controllable RSUs selected, and thus increase the overall coverage of the solution under the same budget. This means that despite of the higher deployment cost, if we adequately operate the fully controllable RSUs, the NSTC per cost achieved by them can exceed that of the other two RSU deployment approaches. The cost of deploying a static and mobile RSU are initially relatively low in comparison. Thus increasing their cost will make it so less of them can be deployed. Static RSUs were not initially selected in great numbers due to the selection of mobile RSUs, so increasing their cost will not negatively affect the NSTC significantly. Mobile RSUs, on the other hand, were the primary selection due to their great coverage and low cost. Thus increasing the cost of mobile RSUs leads the solution to depend primarily on static RSUs and the costly controllable RSUs. As shown in the figure, this decreases the overall NSTC greatly.

6.3 Performance of Proposed Algorithm against Base Strategies

In the previous section, we conducted a simulation to evaluate the performance of the proposed algorithm, which aims to identify a balance of three different RSU deployment strategies under a limited budget to maximize the overall NSTC, based on real world topology and traffic data. On the other hand, from the simulation, it remains unclear which deployment strategies is more influential. To fill this void, in this section, we apply each of the deployment strategies as base strategies and compare them with the proposed algorithm.

Fig. 10(a) illustrates our solution given a budget of 120, cost to deploy static and mobile RSUs being 10, and fully controllable RSUs being 30, which achieved an NSTC of 0.37. In comparison, deploying only mobile RSUs preformed below our solution with an NSTC of 0.30, despite mobile RSUs being relatively inexpensive (Fig. 10(b)). Fig. 10(c) and Fig. 10(d) further show that with the given budget and costs, a single deployment strategy is not optimal, but rather, it is a blend of strategies that provides for a better solution. Fig. 10(c) deployed only static RSUs with an NSTC of 0.02, showing that static RSUs provide the least influence in weighted coverage, although as described in Section 6.2, they do provide the highest stability in NSTC. Fig. 10(d) to Fig. 10(f) vary the cost of fully controllable RSUs to better illustrate the influence this deployment strategy has on the overall solution. With the cost of fully controllable RSUs being as high as they are in Fig. 10(d), very few can be deployed which leads to an NSTC of 0.20. Lowering the cost to 20, a higher NSTC of 0.24 is achieved as more RSUs are able to be deployed under the given budget. With a cost of 10, equaling that of mobile and static RSU costs, fully controllable RSUs outperformed all other deployment strategies with an NSTC of 0.48. This shows that fully controllable RSUs bring the most weighted benefit to the solution and thus are the most influential. This is not to disregard the need for other deployment strategies since mobile and static RSUs are reasonably cheaper to deploy, a combination of these strategies leads to the highest coverage given reasonable costs.

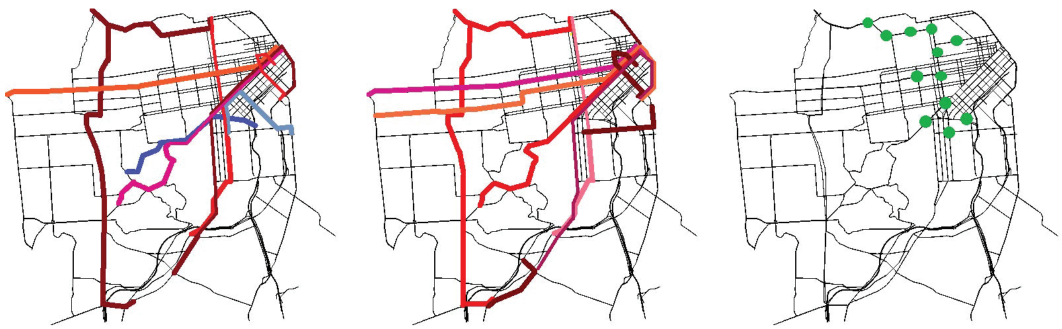
TABLE 1: **NSTC outputs when a probability of traffic delay is introduced. Parameters include a budget of 330, deployment costs of 10, 10, and 30 for each static, mobile and fully controllable RSU deployment strategy respectively.**

Probability of Traffic Delay (p)	NSTC
0.00	0.48
0.10	0.50
0.20	0.55
0.30	0.57
0.40	0.59
0.50	0.60
0.60	0.55
0.70	0.49
0.80	0.35
0.90	0.31
1.00	0.29

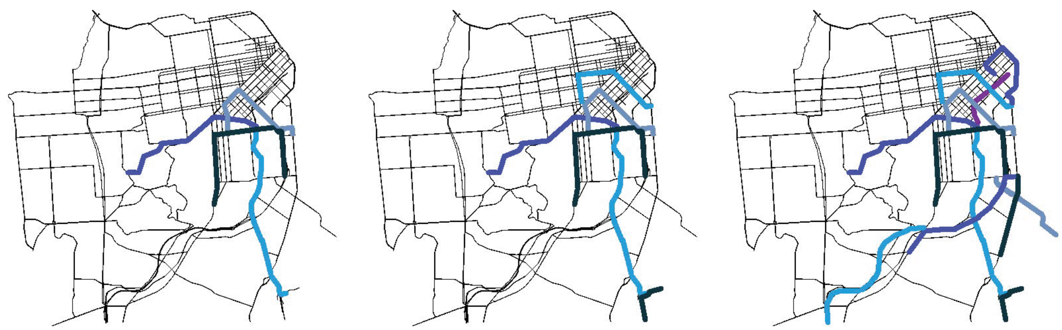
6.4 Impact of Traffic Delay on The Performance of The Proposed Algorithm

So far, we have been studying the problem of our interest under the assumption that the traffic is very light and there exists no traffic jam. While there are many places in which this assumption holds, this may not be the case in the well-known big cities. In this section, we conduct another set of simulation and see how our algorithm works when the area of interest is suffering from zero to moderate traffic jams. In detail, we first apply our algorithm and obtain an output, which determines how RSUs should be deployed on static positions and mobile public transportation, as well as the route of each fully controllable RSU. Then, we adopt the probability of traffic delay p in a way that a public transportation or a fully controllable RSU can move from the current node to another neighboring node with the inverse weighted probability $q = (1 - p * w)$, where w is the normalized weight (importance) of the current node. Remember that w is higher if w has more traffics. As a result, q becomes higher if the area (represented by the current node) has less chance to suffer from a traffic jam.

Table 1 shows our simulation result and from which we can observe that the performance of our algorithm improves when the traffic increases from zero to moderate. This is primarily because as traffic jam increases, an mobile RSU (on a bus or on a fully controllable node) will have a better chance to reside in a node with high weight for a longer period of time. As a result, the algorithm achieves a higher NSTC as the degree of traffic jam increases up to some moderate level. On the other hand, when the traffic jam becomes really serious, the all of the mobile RSUs (both controllable



(a) Our Solution with NSTC of 0.37 (b) Mobile RSUs deployed with NSTC of 0.30 (c) Static RSUs deployed with NSTC of 0.02



(d) Fully controllable RSUs deployed with cost of 30 and NSTC of 0.20 (e) Fully controllable RSUs deployed with cost of 20 and NSTC of 0.24 (f) Fully controllable RSUs deployed with cost of 10 and NSTC of 0.48

Fig. 10: Our solution is compared with different deployment strategies in order to better realize with strategy has the most influence in the overall solution.

and uncontrollable) suffer and behave almost like static RSUs. As the cost to deploy fully controllable RSUs is higher than the static, our algorithm actually pays higher cost for almost static RSUs. As a result, its performance goes down.

From this simulation, we can conclude that our algorithm works well under zero to moderate traffic situation even it does not consider the impact of the traffic jam explicitly. At the same time, we can learn that under very heavy traffic situation, one may generate a better comprehensive RSU deployment algorithm by exploiting very detail traffic data. However, as this is out of the scope of this work, we leave this part as our future work.

7 Conclusion

In this paper, we propose a new strategy to deploy RSUs under the limited budget. Given that the cost-effectiveness of VANET is not sufficiently recognized by general public yet, we believe massive deployment of RSUs all over the wide metropolitan area, which will incur high cost, is difficult in the near future. This paper is intended to provide a new way to test the viability of VANET. In particular, we consider three different RSU deployment strategies, static, mobile but not controllable, and fully controllable, each of which will cost differently. Then, we propose a new optimization problem to best deploy RSUs under a limited budget, and propose a new approximation algorithm whose performance ratio is at least half of the best possible. As a future work, we plan to investigate the tightness of our algorithm and further investigate the existence of approximation algorithms with better performance ratio to close the gap. We are also interested in introducing a new model and corresponding strategy to deal with heavy traffic.

Acknowledgement

This work was supported in part by US National Science Foundation (NSF) HRD-1345219. This research was jointly supported by National Natural Science Foundation of China under grants 11471005.

References

[1] openstreetmap.org

[2] http://transit.511.org/schedules/index.aspx#m1=S&m2=bus

[3] http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/

[4] http://www.sustainablecommunitiesindex.org/city_indicators/view/18

[5] D. Kim, Y. Velasco, Z. Yang, W. Wang, R. Hussain, and R.N. Uma, "Cost Effective Mobile and Static Road Side Unit Deployment for Vehicular Adhoc Networks," *Proceedings of International Workshop on Computing, Networking and Communications (CNC)*, February 15-18, 2016, Kauai, Hawaii, USA.

[6] V. Kumar, S. Mishra, and N. Chand, "Applications of VANETs: Present & Future," Communications and Network, vol. 5, issue 12-15, 2013.

[7] K. Mershad, H. Artail and M. Gerla, "ROAMER: Roadside Units as message routers in VANETs," *Ad Hoc Networks*, vol. 10, no. 3, pp. 479-496, 2012.

[8] J. Barrachina, P. Garrido, M. Fogue, F.J. Martinez, J.C. Cano, C.T. Calafate and P. Manzoni, "Road Side Unit Deployment: A Density-Based Approach," *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 3, pp. 30-39, 2013.

[9] B. Aslam, F. Amjad and C.C. Zou, "Optimal roadside units placement in urban areas for vehicular networks," *Proc. of 2012 IEEE Symposium on Computers and Communications (ISCC)*, pp. 423-429, 2012.

[10] T.-J. Wu, L. Wanjiun and C. Chung-Ju, "A Cost-Effective Strategy for Road-Side Unit Placement in Vehicular Networks," IEEE Transactions on Communications, vol. 60, no. 8, pp. 2295-2303, 2012.

[11] L. Yingsi, L. Hui and D. Rajan, "Optimal Placement and Configuration of Roadside Units in Vehicular Networks," *Proc. of 2012 IEEE Vehicular Technology Conference (VTC 2012-Spring)*, pp. 1-6, 2012.

[12] Z. Lei, Z. Yanyan, P. Jianping, L. Kaur and Z. Hongzi, "Multimodal Message Dissemination in Vehicular Ad-hoc Networks," *Proc. of 2012 1st IEEE International Conference on Communications in China (ICCC)*, pp. 670-675, 2012.

[13] X. Yuwei, W. Ying, X. Jingdong and S. Lin, "Efficient Detection Scheme for Urban Traffic Congestion Using Buses," *Proc. of 2012 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 287-293, 2012.

[14] L. Kun-chan and W. Ze Ming, "On the Feasibility of using Public Transport as Data Mules for Traffic Monitoring," *Proc. of 2008 IEEE Intelligent Vehicles Symposium*, pp. 979-984, 2008.

[15] J. Xiaoxiao and D.H.C. Du, "A BUS Vehicular Network Integrated with Traffic Infrastructure," *Proc. of 2013 International Conference on Connected Vehicles and Expo (ICCVE)*, pp. 562-567, 2013.

[16] H. Yan, G. Xin, C. Zhipeng and T. Ohtsuki, "Multicast Capacity Analysis for Social-proximity Urban Bus-assisted VANETs," *Proc. of 2013 IEEE International Conference on Communications (ICC)*, pp. 6138-6142, 2013.

[17] T. Kitani, T. Shinkawa, N. Shibata, K. Yasumoto, M. Ito and T. Higashinoz, "Efficient VANET-Based Traffic Information Sharing using Buses on Regular Routes," *Proc. of 2008 IEEE Vehicular Technology Conference (VTC 2008-Spring)*, pp. 3031-3036, 2008

[18] L. Jie, G. Xinxing, Z. Tong and Y. Wei, "MI-VANET: A New Mobile Infrastructure Based VANET Architecture for Urban Environment," *Proc. of 2010 IEEE Vehicular Technology Conference (VTC 2010-Fall)*, pp. 1-5, 2010.

[19] L. Wei Kuang, Y. Kai-Ting and L. Meng-Chong, "Bus Assisted Connectionless Routing Protocol for Metropolitan VANET," *Proc. of 2011 Fifth International Conference on Genetic and Evolutionary Computing (ICGEC)*, pp. 57-60, 2011.

[20] D. Chyi-Ren, H. Pa, L. Yu-Hong, L. Yi-Tung and H. Ching-Yin, "An Efficient Data Circulation and Discovery Scheme in VANETs using Public Transportation Systems," *Proc. of 2010 International Conference on Network and Service Management (CNSM)*, pp. 286-289, 2010.

[21] A. Holzer, S. Maaroufi and S. Pierre, "BROADTRIP: Broadcast for Transit in Platoons," *Proc. of 2011 IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 301-306, 2011.

[22] D.S. Hochbaum, ed. "Approximation Algorithms for NP-Hard Problems", PWS Publishing Company, 1997.

[23] S. Khuller, A. Moss, and J. Naor, "The Budgeted Maximum Coverage Problem," *Information Processing Letters*, vol. 7, issue 1, pp. 39-45, 1999.

[24] O.K. Tonguz and W. Viriyasitavat, "Cars as Roadside Units: a Self-organizing Network Solution," IEEE Communications Magazine, vol. 51, no. 12, pp. 112-120, 2013.

[25] V. Kostakos, "Temporal graphs, *Physica A: Statistical Mechanics and its Applications*, vol. 388, issue 6, 1007-1023, 2009.

[26] I. Filippini, F. Malandrino, G. Dan, M. Cesana, C. Casetti, and I. Marsh, "Non-cooperative RSU deployment in vehicular networks," *Proc. of 2012 9th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, pp. 79-82, Jan. 9-11, 2012.

[27] J. Tao, L. Zhu, X. Wang, J. He, Y. Liu, "RSU deployment scheme with power control for highway message propagation in VANETs," Proc. of 2014 IEEE Global Communications Conference (GLOBECOM), pp. 169-174, Dec. 8-12, 2014.

[28] Y. Liu, J. Ma, J. Niu, Y. Zhang, W. Wang, "Roadside units deployment for content downloading in vehicular networks," Proc. of 2013 IEEE International Conference on Communications (ICC), pp. 6365-6370, June 9-13, 2013.

[29] A. Farsi and P. Szczechowiak, "Optimal deployment of Road Side Units in urban environments," Proc. of 2014 International Conference on Connected Vehicles and Expo (ICCVE), pp.815-820, Nov. 3-7, 2014.

[30] M. Patra, S. Mishra, and C.S.R. Murthy, "An Analytic Hierarchy Process Based Approach for Optimal Road Side Unit Placement in Vehicular Ad Hoc Networks," Proc. of 2014 IEEE 79th Vehicular Technology Conference (VTC Spring), pp. 1-5, May 18-21, 2014.

[31] B. Aslam and C.C. Zou, "Optimal roadside units placement along highways," *Proc. of 2011 IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 814-815, Jan. 9-12, 2011.

[32] R. Rizk, R. Daher, and A. Makkawi, "RSUs placement using overlap based greedy method for urban and rural roads," 2014 7th International Workshop on Communication Technologies for Vehicles (Nets4Cars-Fall), pp. 12-18, Oct. 6-8, 2014.

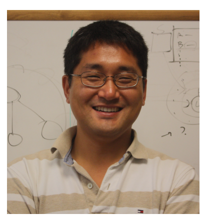
[33] A. Makkawi, R. Daher, and R. Rizk, "RSUs placement using cumulative weight based method for urban and rural roads," Proc. of 2015 7th International Workshop on Reliable Networks Design and Modeling (RNDM), pp.307-313, Oct. 5-7, 2015.

[34] S. Mehar, S.M. Senouci, A. Kies, and M.M. Zoulikha, "An Optimized Roadside Units (RSU) placement for delay-sensitive applications in vehicular networks," Proc. of 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), pp.121-127, Jan. 9-12, 2015.

[35] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions I," *Mathematical Programming* vol. 14, pp. 265294, 1978.

[36] H. Omar, W. Zhuang, and L. Li, "Gateway placement and packet routing for multihop in-vehicle Internet access," *IEEE Transactions on Emerging Topics in Computing (TETC)*, vol. 3, no. 3, pp. 335-351, Sept. 2015.

[37] W. Yang, P. Li, H. Liu, J. Li, "A RSU deployment scheme based on hot spot in vehicular Ad Hoc Networks," *Unifying Electrical Engineering and Electronics Engineering*, Springer, June 2013.



Donghyun Kim received the BS degree in electronic and computer engineering from the Hanyang University, Ansan, Korea (2003), and the MS degree in computer science and engineering from Hanyang University, Korea (2005). He received the PhD degree in computer science from the University of Texas at Dallas, Richardson, USA (2010). Currently, he is an assistant professor in the Department of Computer Science at Kennesaw State University, Marietta, USA. From 2010 to 2016, he was an assistant professor in the Department of Mathematics and Physics at North Carolina Central University, Durham, USA. His research interests include security and privacy, social computing, mobile computing, cyber physical systems, wireless and sensor networking, and algorithm design and analysis. He is an associate editor of Discrete Mathematics, Algorithms and Applications. He is a member of ACM and a senior member of IEEE.

Yesenia Velasco is a senior student in computer science at North Carolina Central University, Durham, NC, USA, and conducting researches on vehicular technologies, computational complexity, and algorithm design and analysis under the supervision of Dr. Donghyun Kim and Dr. R.N. Uma. Previously, she held intern positions at Oak Ridge National Laboratory, Oak Ridge (2014), CatVehicle REU, University of Arizona, Tucson, AZ (2015), Idaho National Laboratory, Idaho Falls, ID (2016). Her research interests include vehicular adhoc networks, mobile computing, computational complexity, and algorithm design and analysis. She is a student member of the IEEE.

Wei Wang received the BS degree in applied mathematics from ZheJiang University, Hangzhou, China (1991). He received the MS degree in computational mathematics (1994) and PhD degree in mathematics from Xian Jiaotong University, Xian, China (2006). He is currently a professor at School of Mathematics and Statistics, Xian Jiaotong University. His research interests include algebraic graph theory and approximation algorithm design and analysis.

R.N. Uma received her BE degree in mathematics from University of Madras, Chennai, India in 1990, ME degree in computer science from Indian Institute of Science, Bangalore, India in 1994 and PhD degree in computer and information science from Polytechnic University, NY in 2000. She is an associate professor in the Department of Mathematics and Computer Science at the North Carolina Central University, Durham, USA. Her research interest includes data science, scheduling and resource allocation with applications to cloud computing, robotics, wireless & sensor networks, multimedia networking, and large logistics problems. She is a member of the IEEE and the ACM.

Rasheed Hussain received his B.S. in Computer Software Engineering from N-W.F.P University of Engineering and Technology, Peshawar, Pakistan in 2007 and MS and PhD degrees in Computer Engineering from Hanyang University, South Korea in 2010 and 2015, respectively. He also worked as a Postdoctoral Research Fellow at Hanyang University, South Korea from March 2015 till August 2016. He is currently working as assistant professor in the Institute of Information Sciences at Innopolis University, Innopolis, Russia. He is also working as a consultant for Innopolis University and guest researcher in the Department of Informatics at University of Amsterdam (UvA), Netherlands. His main research interests include information security and privacy, applied cryptography, and vehicular ad hoc networks. He is a member of the IEEE.

Sejin Lee received the B.S. degree in Mechanical Engineering from Hanyang University, Ansan, South Korea in 2003 and the M.S. and Ph.D. degrees in Mechanical Engineering from Pohang University of Science and Technology, South Korea in 2005 and 2009, respectively. Currently, he is an assistant professor in the Division of Mechanical and Automotive Engineering at Kongju National University, South Korea. From 2011 to 2013, he was an assistant professor in robotics at Kyungil University, Degu, South Korea. His research interest includes vehicular technologies, mobile robots, simultaneous localization and mapping, sensor fusion, extended Kalman filter-based Localization, map building and management, indoor navigation, unmanned surface vehicle, underwater terrain mapping, and 3D point cloud description.