Joint Roadside Unit Deployment and Service Task Assignment for Internet of Vehicles (IoV)

Yuanzhi Ni, Student Member, IEEE, Jianping He, Member, IEEE, Lin Cai, Senior Member, IEEE, Jianping Pan, Senior Member, IEEE, and Yuming Bo

Abstract—Internet of vehicles (IoV) is a promising Internet of Things (IoT) application, where Roadside Unit (RSU) plays an important role for network service provisioning. How to select the number and locations of RSUs to deploy and allocate the traffic load to them is a critical and practical open problem. Most of the existing work focused on one-dimensional (1D) scenarios assuming unlimited RSU capacity, while a more practical 2D case with limited RSU capacity has not been fully considered yet. In this paper, we investigate an RSU deployment problem for 2D IoV networks considering the expected delivery delay requirements and task assignment. We formulate a novel utility-based maximization problem to solve the RSU deployment problem, where the utility function indicates the total benefit from the RSU deployment. We observe that each RSU has an irregular service area, which makes the problem much more difficult than the traditional facility location problem. Then, we design a utility-based RSU deployment algorithm (URDA), a Linear Programming (LP) based clustering algorithm, to solve the problem. The gap between URDA and the optimal solution has been analyzed, which proved that the proposed URDA is near optimal if the deployment cost is low. Extensive simulations have been conducted to demonstrate the effectiveness and superiority of the proposed solution for IoV network service guarantee over other approaches.

Index Terms—Internet of Vehicles (IoV), service-centric architecture design, service load management, RSU deployment, 2D IoV networks, delivery delay requirement.

I. INTRODUCTION

INTERNET of vehicles (IoV) is a promising Internet of Things (IoT) application, and it attracts extensive attention from both the academic and industrial communities. Numerous safety-related services and infotainment applications can be supported in the new IoV paradigm, thanks to the fast development of Vehicle-to-Everything (V2X) communication technologies [1]–[7]. Roadside Unit (RSU) is of great value in IoV given its high communication capacity and complementary features compared with vehicles [8]–[13]. For example, RSUs can be used as content dispatchers exchanging the information with nearby vehicles reliably at fixed locations.

Y. Ni and Y. Bo are with the School of Automation, Nanjing University of Science and Technology, Nanjing, China. E-mail: {ywni, byming}@njust.edu.cn.
J. He is with the Department of Automation, Shanghai Jiao Tong University, Shanghai, China. E-mail: jpehe@sjtu.edu.cn.
L. Cai is with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada. E-mail: cai@ece.uvic.ca.
J. Pan is with the Department of Computer Science, University of Victoria, Victoria, BC, Canada. E-mail: pan@uvic.ca.

Copyright (c) 2012 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

14]. [15]. Recently, Cellular-Vehicle-to-Everything (C-V2X) technologies have been developed [16]–[18], where both RSUs and Evolved Node B (eNB) can provide services to vehicles and other users in Intelligent Transportation Systems (ITS) [19]. In an area with heavy data traffic, deploying RSUs can be an effective solution to relieve the cellular network from severe congestion. How to optimize the placement of RSUs in two-dimensional (2D) IoV networks is a crucial and practical issue.

In this paper, the delivery delay, a general and important performance metric, is selected as the Quality of Service (QoS) index. The solution of this paper can be easily extended if other types of QoS are considered. There are great efforts devoted to studying the RSU deployment problem by combining the delay analysis and system constraints [20]–[24]. Also, many works in the literature have studied the relationship between the RSU deployment and the network connectivity, service coverage, etc. [9], [25]–[27]. However, most of the existing work focused on one-dimensional (1D) roads and assumed RSUs with unlimited capacity. In the urban scenario, vehicles move in a 2D area, and RSUs need to serve the vehicles driving towards different directions. Simply applying the existing 1D solutions to 2D scenarios may lead to substantial performance degradation. Clearly, the service load from the vehicles, the road topology, and the RSU locations are coupled in the 2D area. The modeling of the RSU service area and the deployment strategy design along with the service task assignment should be jointly optimized, which motivates this work.

To solve the RSU deployment problem in 2D IoV networks, it is necessary to consider the trade-off between the benefit brought by the RSU service and the RSU deployment cost. How to model the benefit is difficult. Furthermore, due to the uneven vehicle densities in different roads, given the delivery delay requirement, the effective service area of each RSU is irregular. We thus cannot directly utilize solutions to the traditional facility location problem to solve the RSU deployment problem.

To address these challenges in this paper, we consider a practical 2D RSU deployment problem and propose an efficient and effective algorithm to acquire the deployment strategy. The major contributions are listed as follows.

• The RSU deployment problem is modeled as an RSU service coverage problem given the expected delivery delay requirement. The effective service area of a single RSU is obtained based on the delay analysis. Given the varying vehicle density, the service area is irregular.
This paper proposes a comprehensive utility function evaluating the RSU deployment strategy in the 2D dynamic traffic environment, where the utility denotes the difference between the total benefit to all clients and the cost of deployment.

A Linear Programming (LP) based clustering algorithm is proposed to solve the utility maximization problem. The gap between the obtained solution and the optimal one has been analyzed, which proved that the proposed algorithm is near optimal if the deployment cost is low. Extensive simulations have been conducted to demonstrate the effectiveness and superiority of the proposed algorithm over the existing approaches.

The remainder of the paper is organized as follows. Section II introduces the related work. In Section III, the preliminaries including the scenario and network modeling are provided. The utility maximization problem is formulated in Section III followed by the proposed algorithm in Section V. Section VI analyzes the gap between the obtained solution and the optimal one. Performance evaluations by simulation are presented in Section VII. Section VIII concludes the paper and discusses the future work.

II. RELATED WORK

The approach and analysis of message dissemination with RSUs in IoV networks have been widely studied considering the following two aspects.

First, the importance of RSUs in vehicular network message dissemination has been studied extensively in [8], [28]–[33]. In [8], Reis et al. investigated the benefit of RSU deployment in highway scenarios where both connected RSUs and disconnected RSUs were considered in the analytical model. Wang and Wu [28] proposed an adaptive algorithm to maximize users’ satisfaction of downloading by offloading traffic from the cellular links to RSUs in IoV networks. Salvo et al. [29] proposed three forwarding algorithms for disseminating the message originated from the RSU in an extended service area in urban scenarios. It was also proposed in [30], [32] that the parked vehicles can serve as RSUs to provide service to users. [30] exploited the parked vehicles to extend the content downloading service coverage of RSUs while considering the freshness of the content, the efficiency of the radio resource utilization and the fairness in the vehicle energy consumption. In the vehicular cloud computing (VCC) system, RSUs are applied to collect the computing tasks and then offload to the associated vehicles based on the proposed multi-task replication policy [31]. In [32], a self-organizing network approach was proposed to select the minimum number of parked cars while maximizing the coverage of the support networks. In [33], RSUs at the intersections were used to control data congestion by clustering the message and selecting appropriate parameters for different clusters.

Second, the RSU deployment problem with different requirements and objectives has been considered [9], [20], [21], [23]–[27]. Cavalcante et al. [9] formulated a maximum coverage problem with time constraints for the RSU deployment for information dissemination in IoV networks. To provide Internet access for the passengers in the vehicle, [20] studied the gateway deployment problem aiming at minimizing the deployment cost while guaranteeing the probability of finding a network path greater than the threshold. Wang et al. [21] studied the message delivery problem on a bidirectional road segment and proposed a mathematical model indicating the relationship between the message delivery delay and the RSU deployment distance. Based on the proposed analytical results, the maximum deployment distance was obtained given an information delivery constraint. In [23], Zheng et al. studied the access point placement problem to provide guarantees on the quality of data service in the urban network. A geometry-based coverage optimization problem maximizing the coverage ratio in urban scenarios was considered in [25]. Considering the uneven distribution of the vehicle traffic, [26] proposed a density-based RSU deployment policy and compared it with the minimum cost and the uniform mesh deployment policies. The proposed solution outperforms the latter two when the expected vehicle density is greater than certain thresholds. By proposing a Dynamic Programming (DP) and dimension enlargement based algorithm, He et al. [24] obtained the optimal RSU deployment strategy for message delivery in a large-scale vehicular network. For the 2D urban or suburban RSU deployment problem, [27] applied the 0-1 Knapsack algorithm to maximize the total centrality of RSU deployment given a limited deployment budget. More generally, the facility location problem has been studied with different variants since the early 1960s [34]–[36]. For the uncapacitated facility location problem, [34] proposed an algorithm based cost scaling and greedy local improvement, and achieved a bicriteria approximation tradeoff for facility cost versus service cost. The LP-rounding algorithm, which is based on solving the LP relaxation and rounding the obtained fractional solution into integers, has been studied extensively. [35] presented a polynomial-time algorithm based on the filtering and rounding technique for the facility location problem. The proposed solution provided the first constant performance guarantee for this problem. For the similar problem, [36] proposed an improved approximation algorithm by using the randomized rounding on the optimal solution to the linear program relaxation. The proposed solution significantly improved the approximation guarantee to (1+2/e).

Previous works have justified the importance of RSUs in IoV networks. However, the RSU deployment problem in a 2D area considering the irregular service areas and limited service capacity has not been discussed yet. Furthermore, an analytical framework is needed to investigate the effectiveness of the deployment strategy in delivery delay guarantee.

III. SYSTEM MODELING

A. Scenario

This paper investigates where and how many RSUs should be deployed in a 2D vehicular network in order to guarantee the expected delivery delay. As shown in Fig. 1, the deployed RSU can disseminate the message to nearby vehicles by Vehicle-to-Infrastructure (V2I) communication links. After the vehicle receives a message from the RSU, it broadcasts the
The capacity of an RSU is limited so the number of tasks it can handle simultaneously is limited. Otherwise, the queue of service tasks may be unstable, or many tasks may be dropped due to congestion.

B. Network Modeling

Graph Model: A 2D area can be modeled as a graph $G = \{V, E\}$, where intersections are abstracted to nodes and road segments between intersections are abstracted to edges. Let $V = \{v_i : i = 1, 2, \ldots, N\}$ be the node set and $E = \{e_j : j = 1, 2, \ldots, M\}$ be the edge set. $N$ and $M$ are the number of nodes and edges, respectively.

Message Delivery: In $G$, a 2D graph, messages delivered from an RSU to a vehicle may have multiple paths. For each path, the message is first broadcast to the vehicles within the V2I communication range. Let $D_b$ be the delay of broadcasting the message to vehicles through V2I communication links. Then, according to the analysis provided in [38], the expected message delivery delay in the 2D traffic environment through V2V links can be divided into two categories, i.e., the expected road propagation delay $D_p$ and the expected intersection transfer delay $D_i$. The expected road propagation delay is the average time of a message being delivered from one end of a road segment to the other. The expected intersection transfer delay is the average time it takes for a message being forwarded from the current road segment to an adjacent road segment once the message carrier arrives the intersection. The intersection transfer delay usually comes from the time of searching for a nearby vehicle going to the intended direction. We refer readers to [38] for more details. Since the delivery path $P$ from the RSU to the vehicle may consist of several edges and nodes, the expected entire delivery delay $D_s$ is the sum of the broadcast delay, the road propagation delays and the intersection transfer delays, i.e.,

$$D_s = D_b + \sum_{e \in P} D_p(e) + \sum_{v \in P} D_i(v). \quad (1)$$

$D_s$ can be calculated using the existing methods [21], [22], [38]–[40], assuming that the vehicles in each road segment are distributed randomly, and thus the details are omitted in this paper. Compared with the road propagation delay and the intersection transfer delay which are usually in seconds, the queuing delay and the processing delay at the RSUs are negligible in the studied scenario, so they are omitted in this paper.

RSU Service Area: With the requirement of the delivery delay, the message transmitted from an RSU can be propagated no farther than a continuous area consisting of intersections and road segments, which is the RSU service area defined in Definition 1.

Definition 1 (RSU Service Area). The region to where the message can be delivered starting from the RSU through road segments and intersections within the required average delivery delay requirement.

The road propagation delay defined here is different from the definition of the electromagnetic waveform propagation delay, which is the time for an electromagnetic wave to propagate from the sender to its intended receiver.
Fig. 2: The illustration of RSU radio coverage and service area. The dotted green circle in Fig. 2(a) is the radio coverage of an RSU. It is assumed that the vehicle within the radio coverage can directly communicate with the RSU. The green solid lines Fig. 2(b) represent the RSU service area set $C_1$ deployed at $v_1$. Although part of the vehicles at edge $e_7$ can receive the message from the RSU within the required delay, $e_7$ is excluded by $C_1$ for brevity.

The RSU service area includes not only the area within its V2I communication range but also the area where a message can arrive through multi-hop V2V links with tolerable delay, i.e., the average delay for a message from the RSU to reach the area is below the delay bound. $\forall v_i \in V, i = 1, 2, \ldots, N$, its service area can be presented as an edge set $C_i$ defined as

$$C_i = \{e_j^i\} = \left[ e_1^i \quad e_2^i \quad \cdots \quad e_\text{M}_i \right],$$

where $e_j^i, j = 1, 2, \ldots, M_i$ is the edge that $v_i$ can cover, and $M_i$ is the number of the edges in $C_i$. Vehicles traveling on $\forall e_j^i \in C_i$ are able to receive messages disseminated from $v_i$ within the average delivery requirement. If part of an edge is not covered, this edge is excluded in the coverage set of the RSU, so that the expected delivery delay constraint can be satisfied in our solution. Note that, in vehicle networks, typically the message should be delivered to a certain area, no matter which vehicles are there. Thus, the coverage area defined above is to cover the area including road segments and intersections rather than a certain vehicle.

To obtain the service area of each RSU, the following steps are conducted. First, assuming an RSU is deployed at an intersection, the initial delivery delay from the RSU to this intersection is 0. Each RSU broadcasts the message to different directions. If the current delivery delay plus the expected delivery delay from the current intersection to the next intersection is within the required delay, we include the latest road segment into the RSU service area and update the expected delivery delay. The investigation on the neighboring road segments of the newest road segment continues until any of the following conditions happens: 1) it reaches the boundary of the studied area; 2) the expected delivery delay exceeds the required delivery delay. Finally, the set of road segments in the RSU service area is obtained.

The comparison of the radio coverage and service coverage is shown in Fig. 2. Fig. 2(a) presents the traditional radio coverage of an RSU while Fig. 2(b) illustrates the RSU service area defined in this section. Compared with the traditional radio coverage, the service area is a flexible model to represent the relationship between the QoS and the locations. First, the service area can statistically ensure the QoS of delay-tolerant information dissemination. It is adjustable to QoS requirements. Second, it is important that the service area can provide the flexibility to fit different road topologies and traffic conditions, especially in the 2D urban scenario. The service area shows that how QoS varies with locations while the radio coverage does not contain such information.

Remark 1 (Irregular RSU Service Area). Different from the traditional coverage problem, the RSU service area highly depends on the vehicle traffic density and road topology, as well as its own communication ability. Due to the different vehicle traffic densities, the propagation speed of the message delivery in different places is different. Furthermore, the RSU service area must follow road geometry. Thus, the RSU service area is usually irregular.

Given the irregular RSU service area considered in this paper, many traditional approaches cannot be applied directly. For example, how to evaluate the importance of candidate RSUs and assign the service task between RSUs and the covered road segments at the same time is the main challenge.

Deployment and Assignment: Let $Y = \{y_i\}$ be the RSU deployment strategy where $i = 1, 2, \ldots, N$.

$$y_i = \begin{cases} 1, & \text{if } v_i \text{ is deployed}, \\ 0, & \text{otherwise}. \end{cases}$$

Let $X = \{x_{ij}\}$ be the service task assignment strategy, and $x_{ij} \in [0, 1]$ is a continuous variable indicating the portion of service tasks from edge $e_j$ that is assigned to RSU $v_i$. $X$ and $Y$ are a pair of feasible solutions if the message dissemination tasks of all edges in $G$ are served.

IV. Utility-based RSU Deployment Problem

To evaluate the deployment strategy, we should consider two aspects, i.e., the benefit and the cost. Specifically, the deployment strategy is aiming at improving the network performance using the minimum number of RSUs. Thus, a utility-based approach is proposed in this section.

A. Strategy Evaluation

Naturally, the more RSUs the vehicular network deploys, the better performance it has. However, with the increase of the RSU density, the extra benefit of deploying one more RSU may decrease. Hence, the key issue is to maximize the total utility while fulfilling the expected delivery delay requirement.

Utility: In the RSU deployment problem, the RSU deployment utility $U$ includes two parts, i.e., the benefit of serving the vehicles traveling on $\forall e_j^i \in C_i$ are able to receive messages disseminated from $v_i$ within the average delivery requirement. If part of an edge is not covered, this edge is excluded in the coverage set of the RSU, so that the expected delivery delay constraint can be satisfied in our solution. Note that, in vehicle networks, typically the message should be delivered to a certain area, no matter which vehicles are there. Thus, the coverage area defined above is to cover the area including road segments and intersections rather than a certain vehicle.

To obtain the service area of each RSU, the following steps are conducted. First, assuming an RSU is deployed at an intersection, the initial delivery delay from the RSU to this intersection is 0. Each RSU broadcasts the message to different directions. If the current delivery delay plus the expected delivery delay from the current intersection to the next intersection is within the required delay, we include the latest road segment into the RSU service area and update the expected delivery delay. The investigation on the neighboring road segments of the newest road segment continues until any of the following conditions happens: 1) it reaches the boundary of the studied area; 2) the expected delivery delay exceeds the required delivery delay. Finally, the set of road segments in the RSU service area is obtained.

2For simplicity, we use $v_i$ to represent the RSU deployed at the node $v_i$.
data dissemination tasks depending on the assignment strategy $X$ and the cost of the deployment strategy $Y$, as follows

$$F_U(X, Y) = F_B(X) - F_C(Y) = \sum_{i=1}^{N} \sum_{j=1}^{M} \tilde{b}_{ij} x_{ij} - \sum_{i=1}^{N} f_i y_i,$$

where $\tilde{b}_{ij}$ is the benefit from the service to $e_j$ provided by $v_i$ and $f_i$ is the cost of deploying an RSU at $v_i$. Considering both the installation and maintenance cost, $f_i$ is the depreciation cost plus the maintenance cost of RSU $v_i$ over time period $T$. It is noted that both benefit and cost are measured over the same time period of $T$.

**Benefit Evaluation:** To evaluate the benefit of the task assignment strategy, the following issues are considered in the modeling. 1) If the expected delivery delay is smaller, the benefit of the message is higher. 2) If the network can serve more vehicles, the total benefit is higher. 3) The benefits of message being received by different vehicles are additive. In this work, the benefit indicates the RSU service gain without considering the cost, and it is always positive. The utility indicates the total net profit of the deployment and services, which is the difference between the benefit and the cost. Thus, as shown in (3), the utility is positive only when the benefit is larger than the cost.

With the above principles, a comprehensive metric of the deployment benefit can be defined as the sum of the benefit for all vehicles in the network over the time period $T$. Hence, $\forall e_j \in C_i$, $\tilde{b}_{ij}$ is defined as follows

$$\tilde{b}_{ij} = T \cdot \int_{0}^{L_j} f(d_i(x))dx \cdot r_j, \quad \forall e_j \in C_i$$

where $L_j$ and $r_j$ are the length and the amount of service task of $e_j$, respectively, $d_i(x)$ is the expected delivery delay from $v_i$ to the position $x$ of $e_j$ if the message is delivered through the shortest-delay path, and $f(\cdot)$ is a decreasing function of delay $d(\cdot)$ indicating the relationship between the benefit and the delay. The shortest-delay routing in [38] is applied to obtain the expected delivery delay over the shortest-delay path. For those $e_j \notin C_i$, $\tilde{b}_{ij}$ is set to a negative number indicating the benefit loss for violating the QoS requirement. Note that the delivery delay may vary due to the time-varying vehicle traffic and randomness in wireless communications. In this paper, the expected delivery delay is applied for delay-tolerant applications in vehicle networks. The influence of the delivery variance in real cases will not be significant for two reasons. First, the variances of the road propagation delay and the intersection transfer delay decrease with the increase of the vehicle density. In most of the 2D scenarios, e.g., the urban area, the vehicle density is relatively high, and thus the variance of the delivery delay will be less significant. On the other hand, for a connected network, the vehicle reduction in a road segment usually means an increase in nearby areas, and vice versa. Thus, the positive and negative variances of the delays following a specific path are expected to compensate each other. To verify the above assumption, we have also investigated the delivery delay distribution in the simulation in Section VII. The results depict that the variance is acceptable and the vast majority of the message can be delivered within the required delay.

**Delivery Delay from RSU to Vehicle:** Before a message is received by a vehicle, the message first passes one end of the road segment where the vehicle is currently located. Thus, to calculate $d(\cdot)$, $d_{ij}$, the expected delay over the shortest-delay path from $v_i$ to $v_j$ (where $v_j$ is one end of an edge within $C_i$), should be obtained first. Based on Algorithm 1, $d_{ij}$ is calculated given the deployed RSU $v_i$ and the corresponding service area set $C_i$.

First, from each deployed RSU, if the node is within its coverage, it corresponds to the smallest expected delivery delay obtained by Algorithm 1. Then, the delivery delay from an RSU to a vehicle can be obtained as follows. Let $ep_1$ and $ep_2$ denote the two endpoints of edge $e$ and the smallest expected delays from the RSU to $ep_1$ and $ep_2$ are $dp_1$ and $dp_2$, respectively. $dp_1$ and $dp_2$ can be obtained using Algorithm 1. As shown in Fig. 3, the road segment can be divided into two parts following a portion $\alpha \in [0, 1]$. For the vehicles located in the left portion, the expected delay from $ep_1$ will be smaller than that from $ep_2$ by definition, and vice versa. $\alpha$ can be obtained by

$$\alpha = \begin{cases} 1, & dp_1 - dp_2 < -dp_{12}; \\ \frac{dp_{21} - dp_{12} + dp_{2}}{dp_{12} + dp_{21}}, & dp_1 - dp_2 \in [-dp_{12}, dp_{21}]; \\ 0, & dp_1 - dp_2 > dp_{21}, \end{cases}$$

where $dp_{12}$ ($dp_{21}$) denotes the expected delivery delay from $ep_1$ ($ep_2$) to $ep_2$ ($ep_1$). Denote the distance between $ep_1$ and the vehicle by $x$ as shown in Fig. 3. Thus, the expected

---

**Algorithm 1 Shortest Delay Calculation Algorithm**

**Input:** $v_i$, $C_i$

1: Generate a graph $G_i$ based on $C_i$
2: $\forall i \neq j, d_{ij} \leftarrow \infty, d_{ii} = 0$
3: $n \leftarrow \text{number of nodes in } G_i$
4: $\text{count} \leftarrow 0$
5: while $\text{count} < n - 1$ do
6: for $\forall v_j \in G_i$ do
7: if $v_j$ has received the message then
8: Send the message to its neighbors
9: Compare and update $d_{ij}$
10: end if
11: end for
12: $\text{count} \leftarrow \text{count} + 1$
13: end while

**Output:** $d_{ij}, \forall v_j \in G_i$

---

Fig. 3: Road segment division
delivery delay can be calculated as

\[
d(x) = \begin{cases} 
    d_{p1} + \frac{x}{L_j}d_{p12}, & x < \alpha L_j, \\
    d_{p2} + \frac{x - \alpha L_j}{L_j}d_{p21}, & x \geq \alpha L_j.
\end{cases}
\]  

Combining (3), (4), (5) and (6), the benefit is obtained.

**B. Problem of Interest**

To obtain a deployment strategy with the maximum utility, by plugging (4) into (3), an optimization problem is formulated as follows

\[
\begin{align*}
    \max & \quad \sum_j \sum_i r_j b_{ij} x_{ij} - \sum_i f_i y_i \\
    \text{s.t.} & \quad \sum_i x_{ij} = 1, \quad \forall j, \\
    & \quad x_{ij} \leq y_i, \quad \forall i, j, \\
    & \quad \sum_j r_j x_{ij} \leq u_i y_i, \quad \forall i, \\
    & \quad 0 \leq x_{ij} \leq 1, \quad y_i \in \{0, 1\}, \quad \forall i, j
\end{align*}
\]  

(P0)

where \(r_j\) and \(u_i\) indicates the task of \(e_j\) and the capacity of \(v_i\), respectively.

In this problem, the basic constraint is to use RSUs to serve all the data dissemination tasks in \(G\) within the required expected delivery delay. In addition, we also want to maximize the deployment utility. The formulated problem not only focuses on whether the requirement is met but also takes into account the resulting service benefit for all clients and the deployment cost into account.

**V. DEPLOYMENT STRATEGY DESIGN**

In this section, we design an LP-based clustering algorithm to solve (P0). First, the formulated problem is analyzed. Then, the single-node instance is introduced as the preliminary of the algorithm design. Finally, the algorithm details are provided.

To design an effective RSU deployment algorithm, there are some aspects should be taken into account. First, although the coverage of one RSU does not affect that of others, the overlapping area of the coverage does affect the task assignment, as the vehicles in the overlapped area may receive services from multiple RSUs and how to properly select the server and client under the capacity constraints will significantly affect the delivery delay.

Second, there are some aspects should be taken into account. First, although the coverage of one RSU does not affect that of others, the overlapping area of the coverage does affect the task assignment, as the vehicles in the overlapped area may receive services from multiple RSUs and how to properly select the server and client under the capacity constraints will significantly affect the delivery delay. Second, there are \(2^N\) possible deployment strategies given \(N\), the number of location candidates. For each feasible deployment strategy, the service load assignment and the corresponding utility are different. To obtain the optimal solution, all the feasible strategies should be compared, so the complexity increases exponentially with \(N\). Thus, an approximation algorithm with performance guarantee is preferred. Last, to maximize the total utility, there is a trade-off between the benefit and cost. Furthermore, the assignment is highly related to the deployment strategy as shown in the constraint (7b) and (7c). The algorithm needs to comprehensively consider the relationship between the utility maximization and delivery delay requirement, and take care of both deployment and assignment simultaneously.

**Algorithm 2 Deployment Algorithm for Single-node Problem**

**Input:** \(R, f_i, b_i, u_i\)

1. \(x_i, y_i = 0, \forall i\)
2. **while** \(\sum_i y_i = 1 < 1\) **do**
3. \(k \leftarrow \arg \max_{i, y_i = 0} (b_i - \frac{L_i}{u_i})\)
4. \(x_k \leftarrow \min(u_k/R, 1 - \sum_i y_i = 1 x_i)\)
5. \(y_k \leftarrow Rx_k/u_k\)
6. **end while**

**Output:** \((x, y)\)

Based on the above analysis, we propose a utility-based RSU deployment algorithm (URDA) in this section.

**A. Single-node Problem**

Before introducing the deployment algorithm in details, we first study the single-node capacitated facility location problem (SNCFL) where there is only one road segment needed to be served by multiple RSUs. The original problem (P0) with only one road segment is simplified as follows

\[
\begin{align*}
    \max & \quad \sum_i R b_i x_i - \sum_i f_i y_i \\
    \text{s.t.} & \quad \sum_i x_i = 1, \\
    & \quad R x_i \leq u_i y_i, \quad \forall i, \\
    & \quad 0 \leq x_i, y_i \leq 1, \quad \forall i
\end{align*}
\]  

(P1)

where \(y_i\) is relaxed to a continuous variable within \((0, 1)\). If \(y_i = 1\), the RSU \(v_i\) is fully opened while \(0 < y_i < 1\) indicates the RSU \(v_i\) is fractionally opened. It is noted that the capacity and cost of the fractionally open RSUs are also scaled down. \(b_i\) and \(x_i\) is the benefit from the service provided by \(v_i\) and the portion of service task assigned to \(v_i\), respectively. \(R\) is the amount of the service tasks.

Given any feasible solution \((x, y)\), we can set \(\hat{y}_i = \frac{R x_i}{u_i}\) and obtain a feasible solution \((x, \hat{y})\) with no less total utility. Thus, the objective function can be replaced by \(\max \sum_i R \left(b_i - \frac{R}{u_i} x_i\right)\), and the corresponding constraints (8b) and (8c) are changed to \(R x_i \leq u_i\) for all \(i\). By adding the RSUs following the decreasing order of \(b_i - \frac{R}{u_i}\) and assigning the maximum possible demand to the selected RSUs, i.e., the smaller one among the capacity of the RSU and the remaining demand, until all the demands are satisfied, (P1) is thus solved in a greedy manner as shown in Algorithm 2. The obtained solution is the optimal one and there is at most one RSU is fractionally open [41]. The property is used in the algorithm introduced below.

**B. Algorithm Design**

In this subsection, we introduce URDA, an LP-based clustering algorithm to solve (P0). By relaxing the solution region
from integer to continuous variables, the original problem can be transformed as follows

\[
\begin{align*}
\max & \sum_j \sum_i r_j b_{ij} x_{ij} - \sum_i f_i y_i \quad (P2) \\
\text{s.t.} & \sum_i x_{ij} = 1, \quad \forall j,
\end{align*}
\]

and its dual problem is

\[
\begin{align*}
\min & \sum_i z_i - \sum_j \alpha_j \\
\text{s.t.} & \alpha_j \leq -r_j b_{ij} + \beta_{ij} + r_j \gamma_i, \quad \forall j,
\end{align*}
\]

where intuitively, \( \alpha \) shows the contribution of each road segment to the total utility. Let \((x, y)\) and \((\alpha, \beta, \gamma, z)\) be the optimal solutions to (P2) and (P3), respectively. Let \(F = \{v_i : y_i > 0\}\) be the opened facilities in \((x, y)\) and \(F_j = \{v_i : x_{ij} > 0\}\) be the facilities in \(F\) that fractionally serve \(e_j\). The algorithm is conducted in three steps as shown in Algorithm 3: 1) RSU clustering, 2) reducing to the single-node instance and 3) assigning the tasks.

1) **RSU clustering:**

**S1:** Let \(C\) be the set of the current cluster centers, which is initially empty, and \(N_k\) denote the RSUs clustered around the edge \(e_k \in C\). For those edges \(e_j \notin C\), let \(B_j\) be the set of the unclustered RSUs that are more beneficial to \(e_j\) than any cluster center, i.e., \(B_j = \{v_i \in F_j : i \notin \cup_{e_k \in F} N_k \text{ and } b_{ij} \geq \max_{e_k \in C} b_{ik}\}\). Let \(S\) be the set containing all the edges that could be chosen as the cluster centers which send at least half of their demands to the RSUs in \(B_j\), i.e., \(S = \{e_j \notin C : \sum_{v_i \in B_j} x_{ij} \geq \frac{1}{2}\}\). We repeatedly select \(e_j \in S\) with the smallest \(\alpha_j\) and form the cluster with \(N_j = B_j\). Then, we update the sets \(C\) and \(S\).

**S2:** After the above process, there may still leave some RSUs in \(F\) have not been clustered around any \(e_j \in C\). For these RSUs, we assign them to the existing cluster center to whom the RSU is most valuable, i.e., \(N_j \leftarrow N_j \cup \{v_i\}\) where \(e_j = \arg\max_{e_k \in C} b_{ik}\). The tasks served in each cluster are defined as the total tasks served by all the RSUs in it. Note that not only the tasks from the cluster center \(e_j\) but also from other edges fractionally served by the RSUs in \(N_j\) are counted as the cluster demand. Let \(L_k = \{v_i \in N_k : 0 < y_i < 1\}\) be the set of remaining RSUs and \(R_k = \sum_{e_k \in L_k} \sum_j r_j x_{ij}\) be the total demand, respectively. The greedy algorithm in Section V-A is used to find the optimal solution \((x^{(k)}, y^{(k)})\) for the SNCFL. Let \(O_k^*\) be the value of the optimal solution. We open all RSUs with \(y_i^{(k)} > 0\). Together with the RSUs opened at the beginning of this step (i.e., those with \(y_i = 1\)), the RSUs opened now have sufficient capacity to serve all the demand \(\sum_{v_i \in N_k} \sum_j r_j x_{ij}\) and thus the total demand can be served. Combine the solutions for all clusters and we have all the RSUs either fully opened or not opened.

2) **Reducing to the single-node instance:** First, \(\forall v_i \in N_k\), \(v_i\) is opened if \(y_i = 1\) for all clusters. Then for each cluster, an SNCFL is formed with the remaining RSUs and service tasks. Let \(L_k = \{v_i \in N_k : 0 < y_i < 1\}\) be the set of remaining RSUs and \(R_k = \sum_{v_i \in L_k} \sum_j r_j x_{ij}\) be the total demand, respectively. The greedy algorithm in Section V-A is used to find the optimal solution \((x^{(k)}, y^{(k)})\) for the SNCFL. Let \(O_k^*\) be the value of the optimal solution. We open all RSUs with \(y_i^{(k)} > 0\). Together with the RSUs opened at the beginning of this step (i.e., those with \(y_i = 1\)), the RSUs opened now have sufficient capacity to serve all the demand \(\sum_{v_i \in N_k} \sum_j r_j x_{ij}\) and thus the total demand can be served. Combine the solutions for all clusters and we have all the RSUs either fully opened or not opened.

3) **Assigning the tasks:** After opening enough RSUs, we redo the task assignment by solving (P0) to maximize the total utility.

The algorithm operation includes four steps. First, solve a relaxed LP problem (P2) and its dual problem (P3), which can be done by well-studied tools. Second, based on the solutions of (P2) and (P3), group the RSUs into different clusters. Here, as shown in Algorithm 3, the number of clusters will not exceed the number of RSU candidates. Third, solve single-node problem (P1) for each cluster in a greedy manner with a low complexity. Last, combine the solution of all clusters together and solve the task assignment problem which is another LP problem. Overall, the complexity is moderate with the above analysis and will not increase significantly with the network scale. Thus, the algorithm is viable even when the network scale is large.
In this section, the gap between the proposed URDA and the optimal one is studied. The analysis uses the following two facts. First, Lemma 1 shows that the solution of (P1) is not far away from its optimal solution. Second, Lemma 3 shows that the solution of (P2) obtained by assembling the solutions to each single-node instances is not far away from the optimal solution to the original problem. For convenience, we assume that the service demand $r_j = 1, \forall j$. It is straightforward to find the following result also holds for varying $r_j$. Let $U^* = F_L(x, y)$ and $B^* = F_B(x)$ be the optimal utility and the corresponding total benefit for (P2), respectively.

Recall that $L_k = \{v_i \in N_k : y_i < 1\}$, $(\hat{x}^{(k)}, y^{(k)})$ is the optimal solution to (P1) found by the greedy algorithm for the single-node instance corresponding to this cluster, and $O_k^*$ is the value of this solution.

**Lemma 1.** For each $e_k \in \mathcal{C}$, the optimal value

$$O_k^* \geq \sum_{j} b_k x_{ij} - \sum_{v_i \in L_k} f_i y_i,$$

and hence,

$$\sum_{e_k \in \mathcal{C}} O_k^* \geq \sum_{j} \sum_{i : y_i < 1} b_k x_{ij} - \sum_{i : y_i < 1} f_i y_i. \quad (12)$$

**Proof.** First, we propose a feasible solution $(\hat{x}, \hat{y})$, where $\hat{y}_i = y_i$, and $\hat{x}_i = \sum_{j} x_{ij}$ for all $v_i \in L_k$. Note that $\sum_{v_i \in L_k} \hat{x}_i = \sum_{v_i \in L_k} x_{ij} = R_k$. The facility cost of this solution is at most $\sum_{v_i \in L_k} \hat{y}_i = \sum_{v_i \in L_k} f_i y_i$. The service benefit is $\sum_{v_i \in L_k} f_k \hat{x}_i = \sum_{v_i \in L_k} f_k x_{ij}$. Combining this with the bound on the facility cost, we have that

$$O_k^* \geq \sum_{v_i \in L_k} f_k \hat{x}_i - \sum_{v_i \in L_k} f_i y_i \quad (13)$$

Since $N_k$ are disjoint, by summing up all clusters, we have

$$\sum_{e_k \in \mathcal{C}} O_k^* \geq \sum_{j} \sum_{i : y_i < 1} b_k x_{ij} - \sum_{i : y_i < 1} f_i y_i. \quad (14)$$

Lemma 1 shows that the sum of the optimal solutions of the clusters has a bounded gap to the global optimal solution.

**Lemma 2.** The cost of opening the (at most one) extra RSU in cluster $N_k$ is at most $2 \sum_{v_i \in N_k} f_i y_i$.

**Proof.** $\sum_{v_i \in N_k} y_i \geq \sum_{v_i \in N_k} x_{ik} \geq \frac{1}{2}$ since $N_k$ was established around $k$ in step S1, and no RSU is removed from $N_k$ in step S2. We open at most one extra RSU from $N_k$ according to the property of the greedy algorithm introduced in Section V-A. Since all RSUs have the same cost $f$, the cost of opening this facility is $f \leq f \cdot 2 \sum_{v_i \in N_k} y_i = 2 \sum_{v_i \in N_k} f_i y_i$. (This is the only place that we assume the RSU costs are all the same.)

Lemma 2 demonstrates that the cost of opening the extra RSU is also bounded.

Let $\hat{y}$ be the 0-1 vector indicating which RSUs are opened. Let $\hat{y}^{(k)}$ denote the portion of $\hat{y}$ consisting of the facilities in $L_k$, i.e., $\hat{y}^{(k)} = (\hat{y}^{(k)}_i)_{v_i \in L_k}$.

**Lemma 3.** The solution $(\hat{x}^{(k)}, \hat{y}^{(k)})$ for cluster $N_k$ yields an assignment $\hat{x}^{(k)}$ to each single-node instance is not far away from the optimal solution. For convenience, we assume that the service demand $r_j = 1, \forall j$. It is straightforward to find the following result also holds for varying $r_j$. Let $U^* = F_L(x, y)$ and $B^* = F_B(x)$ be the optimal utility and the corresponding total benefit for (P2), respectively.

Recall that $L_k = \{v_i \in N_k : y_i < 1\}$, $(\hat{x}^{(k)}, y^{(k)})$ is the optimal solution to (P1) found by the greedy algorithm for the single-node instance corresponding to this cluster, and $O_k^*$ is the value of this solution.

**Lemma 1.** For each $e_k \in \mathcal{C}$, the optimal value

$$O_k^* \geq \sum_{j} b_k x_{ij} - \sum_{v_i \in L_k} f_i y_i,$$

and hence,

$$\sum_{e_k \in \mathcal{C}} O_k^* \geq \sum_{j} \sum_{i : y_i < 1} b_k x_{ij} - \sum_{i : y_i < 1} f_i y_i. \quad (12)$$

**Proof.** First, we propose a feasible solution $(\hat{x}, \hat{y})$, where $\hat{y}_i = y_i$, and $\hat{x}_i = \sum_{j} x_{ij}$ for all $v_i \in L_k$. Note that $\sum_{v_i \in L_k} \hat{x}_i = \sum_{v_i \in L_k} x_{ij} = R_k$. The facility cost of this solution is at most $\sum_{v_i \in L_k} \hat{y}_i = \sum_{v_i \in L_k} f_i y_i$. The service benefit is $\sum_{v_i \in L_k} f_k \hat{x}_i = \sum_{v_i \in L_k} f_k x_{ij}$. Combining this with the bound on the facility cost, we have that

$$O_k^* \geq \sum_{v_i \in L_k} f_k \hat{x}_i - \sum_{v_i \in L_k} f_i y_i \quad (13)$$

Since $N_k$ are disjoint, by summing up all clusters, we have

$$\sum_{e_k \in \mathcal{C}} O_k^* \geq \sum_{j} \sum_{i : y_i < 1} b_k x_{ij} - \sum_{i : y_i < 1} f_i y_i. \quad (14)$$

Lemma 1 shows that the sum of the optimal solutions of the clusters has a bounded gap to the global optimal solution.

**Lemma 2.** The cost of opening the (at most one) extra RSU in cluster $N_k$ is at most $2 \sum_{v_i \in N_k} f_i y_i$.

**Proof.** $\sum_{v_i \in N_k} y_i \geq \sum_{v_i \in N_k} x_{ik} \geq \frac{1}{2}$ since $N_k$ was established around $k$ in step S1, and no RSU is removed from $N_k$ in step S2. We open at most one extra RSU from $N_k$ according to the property of the greedy algorithm introduced in Section V-A. Since all RSUs have the same cost $f$, the cost of opening this facility is $f \leq f \cdot 2 \sum_{v_i \in N_k} y_i = 2 \sum_{v_i \in N_k} f_i y_i$. (This is the only place that we assume the RSU costs are all the same.)

Lemma 2 demonstrates that the cost of opening the extra RSU is also bounded.

Let $\hat{y}$ be the 0-1 vector indicating which RSUs are opened.

**Lemma 3.** The solution $(\hat{x}^{(k)}, \hat{y}^{(k)})$ for cluster $N_k$ yields an assignment $\hat{x}^{(k)}$ to each single-node instance is not far away from the optimal solution.

**Proof.** (See Appendix A.)

Lemma 3 shows that the optimal solutions of the clusters can be used to construct a solution to the original problem with a bounded benefit loss.

**Lemma 4.** The utility of opening RSUs $v_i$ with $y_i = 1$ and serving $e_j$ by such $v_i$ following the assignment $x_{ij}$, is at least $\sum_i \hat{z}_i - \sum_{j} i : y_i = 1 \alpha_j x_{ij}$.

**Proof.** (See Appendix B.)

Lemma 4 uses the complementary slackness to connect the utility from those fully opened RSUs, i.e., $y_i = 1$, with the dual objective function. Combining the above conclusions together, we get the following theorem.

**Theorem 1.** Under URDA, the utility of the solution returned is at least $4 \cdot U^* - 3 \cdot B^*$, which means the gap between the returned solution and the optimal solution is less than $3 \cdot (B^* - U^*)$.

**Proof.** (See Appendix C.)

As shown in Lemma 1 and Lemma 3, the solutions of all single-node instances (P1) connect the optimal solution of (P2) and the returned feasible solution of (P0) with bounded gaps, respectively. Lemma 1 shows that the total utility of the optimal solution for all the single-node problems has a bounded gap to the optimal solution of (P2), while Lemma 3 shows that the optimal solution of each single-node problem (P1) can be used to construct a feasible solution to the original problem (P0) while losing a bounded term. When constructing the feasible solution of (P0) from that of (P1), the extra cost from fully opening the extra RSU should be considered. Lemma 2 shows that the corresponding cost is bounded. As shown in Lemma 4, when opening the RSUs with $y_i = 1$, the corresponding utility is proved to be a bounded value using complementary slackness. Finally, combining the above facts, Theorem 1 shows that the obtained feasible solution has a bounded gap to the global optimal solution.

Given the fact that the $U^*$ is not necessarily to be a positive value in practice, we cannot bound the returned solution with a constant. However, as shown in Theorem 1, if the deployment cost (i.e., $B^* - U^*$) is small enough, then the returned solution is guaranteed to be close to the optimal value. This is because the algorithm will be less affected by the cost and prefer to achieve a higher benefit from the service.
The vehicle density probability distribution

Fig. 4: The vehicle density probability distribution

Furthermore, Theorem 1 provides the guidance for the deployment strategy not only about how to achieve a satisfactory performance, but also on whether it is worthwhile to deploy the vehicular systems in an area. More precisely, it presents the algorithm performance bound after properly selecting the benefit function and evaluating the cost. When the total utility is low in an area, it reflects the low return from the deployment in such an area, and also the algorithm cannot guarantee a promising bound.

VII. PERFORMANCE EVALUATION

To validate the effectiveness of the proposed RSU deployment strategy, extensive simulations have been conducted by using Matlab. The comparisons with the existing methods are provided in terms of the deployment benefit and utility, under different RSU costs and service requirements.

A. Simulation Setup

An 8 × 8 Manhattan grid model is adopted in the simulation. To obtain the vehicle density distribution, the real trace collected from about 2,300 taxis in Shanghai in 2007 [42]–[44] were analyzed. First, we selected 300 observation spots in Hongkou District in Shanghai, and recorded the number of vehicles passing by each observation spot during midday. Then, the distribution of vehicle density is estimated using the polynomial fitting method, and scaled up considering the relationship between the number of taxis in 2007 and that of the vehicles at present. The trace statistics and the estimation result are shown in Fig. 4, which is used for the generation of the vehicle densities of the road segments in the simulation. Finally, according to the vehicle densities of the road segments, the positions of vehicles were randomly generated as in [38]. Note that, we cannot directly apply the taxi trace for simulation because the taxis are only a portion of all vehicles. For the accuracy of the evaluation, the estimation of the density distribution of all vehicles is necessary. In the simulation, the messages are transmitted from the RSU to vehicles randomly located in its designated service area. The message delivery delays from RSU to vehicles are recorded and categorized by road segments. The expected delivery delay requirement is 60 seconds which is reasonable to ensure the feasibility of the studied problem. Each intersection is connected with its neighbor intersections by a 2,000–meter road segment, as the arterials in a city. If not stated otherwise, \( f_i \), the cost of deploying an RSU, is 300 and the benefit function is \( f(d) = \max(0, 1 - d/60) \). The unit of \( d \) is seconds. Benefit \( f(d) \) and deployment cost \( f_i \) have been normalized into a unified monetary unit. The capacity of RSU and the service tasks of each road segment varies between [50, 70] and [1, 40] per time unit, respectively.

B. Deployment Algorithm Comparison

The performance comparisons among the greedy algorithm improved from [45], the LP-based algorithm improved from [35], [36], the full deployment algorithm and the proposed URDA are presented in Fig. 6. Each benchmark algorithm considers both the RSU deployment and the task assignment. The greedy algorithm aims at maximizing the utility increment in each RSU selection step, while still satisfies the capacity constraint. In each iteration, if there is only one unselected RSU that can serve a certain road segment, then this RSU is selected and the service task of that road segment is assigned. Otherwise, the anticipated utility increase by deploying one of the unselected RSUs will be calculated. For example, assuming \( v_i \) is an unselected RSU, we assign the service tasks to \( v_i \) in the descending order of \( b_{ij} \) where \( e_j \in C_i \) and \( r_j \) has not been fully assigned yet, until either the capacity of \( v_i \) is totally occupied or all the possible service tasks are assigned. Then, the RSU with the highest utility increase will be selected and the corresponding task assignment is applied. The iteration ends when all the tasks are assigned. In the LP-based algorithm improved from [36], a simple deterministic rounding scheme is applied. Specifically, the solutions to the relaxed problem (P2) are obtained and the RSUs are selected according to the descending order of \( y_i \) until the total capacity is sufficient to serve all tasks. Then, the task assignment is optimized with the selected RSU by solving (P0) given \( y_i \). To achieve the best service performance, the full deployment algorithm ignores the effect of cost and focuses on maximizing the service benefit by setting \( y_i = 1, \forall i \). The service assignment is obtained by solving (P0) with \( y_i = 1, \forall i \), accordingly.

The delivery delays to the vehicles in each road segment and those to the entire network are averaged first. Fig. 6(a) shows the maximum average delivery delay among road segments and the average delivery delay of the entire network. URDA always has a lower average delay compared with the Greedy algorithm and the LP-based algorithm. When compared with the full deployment algorithm which achieves the best possible performance, the average delay of URDA is only about 0.2 second more than the full deployment algorithm while the number of the RSUs deployed using URDA is 36% less. Fig. 6(b) illustrates that the benefit and utility of RSU deployment strategy. The upper bound of benefit is obtained by solving (9) under \( y_i = 1, \forall i \) and the utility upper bound is obtained without integer constraints, respectively. URDA has a higher benefit and utility compared with the Greedy algorithm and LP-based algorithm. When compared with the full deployment algorithm, although the benefit of URDA is slightly lower because of deploying much fewer RSUs, the utility of URDA is much higher than the full deployment algorithm. As shown in Fig. 5(a), the cumulative distribution function (CDF) of the delivery delay of URDA is quite close to the full deployment algorithm. Furthermore, under the URDA deployment and assignment strategy, most of the tasks are
Comparison of varying Benefit and utility

Comparison of estimation error

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/JIOT.2018.2882436, IEEE Internet of Things Journal

![Graph 1](image1.png)  ![Graph 2](image2.png)  ![Graph 3](image3.png)  ![Graph 4](image4.png)

(a) Comparison of deployment algorithm  (b) Comparison of varying $f_i$  (c) Comparison of service type  (d) Comparison of estimation error

Fig. 5: CDF of delivery delay

![Graph 5](image5.png)  ![Graph 6](image6.png)

(a) Maximum and average delay  (b) Benefit and utility

Fig. 6: Algorithm comparison

![Graph 7](image7.png)  ![Graph 8](image8.png)

(a) Number of deployed RSUs  (b) Benefit and utility

Fig. 7: Influence of deployment cost $f_i$

Fig. 8: Influence of benefit function

In the real world, different types of message dissemination services may obtain different benefits from the same message delivery delay meeting the QoS requirement. For example, the value of the parking lot availability information decreases rapidly with the growth of the delivery delay since the space may be seized by other vehicles during the message delivery. However, the value of the shopping mall sales advertisement decreases slowly as the information does not change frequently. To illustrate the influence of the benefit function to the deployment strategy, we use three types of benefit functions including a concave function $f_1(d) = 1 - (d/60)^2$, a linear function $f_2(d) = 1 - d/60$ and a convex function $f_3(d) = 2 - \sqrt{5 - (d/50)^2}$ for comparison. Generally speaking, applications more delay-sensitive will choose a convex benefit function over a concave benefit function.

As shown in Fig. 8, the strategy performance highly depends on the benefit function. According to Fig. 8(a), the strategy based on $f_3(d)$ deploys the most RSUs because the benefit decreases rapidly with the growth of the delay in $f_3(d)$. To achieve a higher total utility, URDA gives priority to reducing the delay of the covered area rather than covering more road segments. In addition to the number of deployed RSUs, the benefit/utility evaluation results under different functions are quite diverse. As shown in Fig. 8(b), although convex benefit function based strategy deploys more RSUs, the benefit is still lower than the former two strategies. Fig. 5(c) illustrates the delay performance of $f_3(d)$ based strategy is relatively worse than the latter two while its total benefit/utility is higher. Even smaller $f_i$ is, a better performance the strategy gets thanks to the less influence of the deployment cost, which also verifies the analytical results in Section VI.

D. Different Types of Services

To demonstrate the influence of the deployment cost to URDA, $f_i$ is set to 100, 500, 1000, 2000, 3000, and 10000, for comparison. As shown in Fig. 7(a), the number of the deployed RSUs decreases with the growth of $f_i$. When the cost is low, URDA prefers to deploy more RSUs to achieve a better performance. However, when the cost is relatively high, URDA tries to reduce the number of the deployed RSUs while satisfies the requirement. It is found that the number of the RSUs will not keep decreasing with the increase of $f_i$ due to the delay requirement and capacity restriction. When the fewer RSUs are deployed due to the high cost, the total benefit also decreases slightly as shown in Fig. 7(b), while the required expected delivery delay is always met. However, the utility decreases significantly with the increase of $f_i$, and even below zero when $f_i$ is greater than 2000. To present the results clearly, the CDFs of the delivery delay of $f_i = 100, 500, 2000, 3000, 10000$ are selected for comparison. As illustrated in Fig. 5(b), the delay performances under different $f_i$ slightly vary. Generally, the
for the similar delay performance, the benefit/utility with \( f_2(d) \) is higher than that with \( f_3(d) \). This is because the service type significantly affects the evaluation of the benefit. Specifically, the same delay will lead to quite different benefits in different types of services. Thus, it is crucial to choose a proper benefit function when determining the RSU deployment.

### E. Time-varying Traffic

In practice, the vehicle density of the road segment may change with time. As a consequence, the estimation of the vehicle traffic, based on which the algorithm is operated inevitably encounters some errors. To investigate the influence of these estimation inaccuracies, we compare the performance with and without vehicle density variations in Fig. 9. The estimation error rates are set to \( \pm 10\% \), \( \pm 20\% \), and \( \pm 30\% \), respectively, which means the vehicle density of each road segment used in each simulation randomly varies in the given range. As shown in Fig. 9(a), even under the estimation error, the maximum average delays are still within the required range and the average delays oscillate slightly. From the above figures, there is no obvious trend that the delivery delay will suffer from the growth of the estimation variation. Thus, it is natural to conclude that the delay performance will not significantly affect the time-varying estimation error and URDA still sounds in such cases. It is noted that the traffic statistics used in the algorithm are not restricted to traffic statistics used in the algorithm are not restricted to vehicle traffic, based on which the algorithm is operated.

**APPENDIX A**

**THE PROOF OF LEMMA 3**

**Proof.** It is straightforward to see that constraint (9d) is satisfied for \( v_i \in L_k \), since \( g^{(k)} \) is a \( \{0, 1\} \)-vector. The facility cost \( \sum_{v_i \in L_k} f_i y_i^{(k)} \) is at most \( \sum_{v_i \in L_k} f_i y_i^{(k)} + 2 \sum_{v_i \in N_k} f_i y_i \) since every RSU except the extra one is either fully open or not open in the solution \( x^{(k)}, y^{(k)} \) and according to Lemma 2, the cost of opening the extra RSU is at most \( 2 \sum_{v_i \in N_k} f_i y_i \).

The service benefit of the single-node solution is the benefit of serving all the tasks \( R_k = \sum_j v_i \in L_k x_{ij} \) from the facilities in \( L_k \) to the center \( e_k \). Now we want to move the tasks, \( \sum_{v_i \in L_k} x_{ij} \), of road segment \( e_j \) from \( e_k \) back to \( e_j \). As a consequence, an additional benefit loss \( \sum_j v_i \in L_k (b_k - b_i) x_{ij} \) is incurred. Specifically, we set \( \hat{x}_{ij}^{(k)} = v_i \in L_k \) arbitrarily such that, (1) \( \sum_{v_i \in L_k} x_{ij}^{(k)} = \sum_{v_i \in L_k} x_{ij} \) for each road segment \( e_j \), and (2) \( \sum_j x_{ij}^{(k)} = x_i^{(k)} \) for each RSU \( v_i \in L_k \). This satisfies constraints (9b), (9c)—if \( x_i^{(k)} > 0 \) then \( x_i^{(k)} > 0 \), so \( b_i^{(k)} = 1 \), and \( \sum_j x_{ij}^{(k)} = x_i^{(k)} \leq u_i = u_i y_i^{(k)} \). The service benefit is

\[
\sum_j \sum_{v_i \in L_k} b_i x_{ij}^{(k)} \geq \sum_{v_i \in L_k} \sum_{j} b_i x_{ij}^{(k)} - \sum_j \sum_{v_i \in L_k} (b_k - b_i) x_{ij}^{(k)} \\
\geq \sum_{v_i \in L_k} b_i x_{ij}^{(k)} - \sum_j \sum_{v_i \in L_k} b_k x_{ij}.
\]

(15)

We have \( O_k^* = \sum_{v_i \in L_k} (b_i x_{ij}^{(k)} - f_i y_i^{(k)}) \). Combining the bound of the service benefit and facility cost together, we obtained the desired results.

**REFERENCES**

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/JIOT.2018.2882436, IEEE Internet of Things Journal.
APPENDIX B
THE PROOF OF LEMMA 4

Proof. By the complementary slackness, each facility $v_i$ with $z_i > 0$ has $y_i = 1$. For each such facility we have that

$$
\sum_j \alpha_j x_{ij} = \sum_j -b_{ij} x_{ij} + \sum_j \beta_{ij} x_{ij} + \sum_j \gamma_i x_{ij}
$$

$(x_{ij} > 0 \Rightarrow \alpha_j = -b_{ij} + \beta_{ij} + \gamma_i)$

$$
= \sum_j -b_{ij} x_{ij} + \sum_j \beta_{ij} y_i + u_i \gamma_i y_i
$$

$(\beta_{ij} > 0 \Rightarrow x_{ij} = y_i, \gamma_i > 0 \Rightarrow \sum_j x_{ij} = u_i y_i)$

$$
= \sum_j -b_{ij} x_{ij} + f_i + z_i.
$$

$(y_i > 0 \Rightarrow \sum_j \beta_{ij} = f_i + z_i - u_i \gamma_i)$

Then, we have

$$
z_i - \sum_j \alpha_j x_{ij} = \sum_j b_{ij} x_{ij} - f_i.
$$

(17)

By summing over all $i$ with $y_i = 1$, we complete the proof.

APPENDIX C
THE PROOF OF THEOREM 1

Proof. The total utility is bounded at least $4 \cdot U^* - 3 \cdot B^*$ with a feasible solution $(\hat{x}, \hat{y})$ which is constructed as follows. First, we set $\hat{x}_{ij} = x_{ij}$ and $\hat{y}_i = y_i$ for each RSU $v_i$ that $y_i = 1$. This satisfies constraints (9a)–(9d) for $i$ such that $y_i = 1$. By Lemma 4,

$$
\sum_j \sum_{i: y_i = 1} b_{ij} \hat{x}_{ij} - \sum_{i: y_i = 1} f_i \hat{y}_i = \sum_i z_i - \sum_j \sum_{i: y_i = 1} \alpha_j x_{ij}.
$$

(18)

Second, we set $\hat{x}_{ij} = \hat{x}_{ij}^{(k)}$ for $v_i \in L_k$ where $(\hat{x}_{ij}^{(k)}, \hat{y}_{ij}^{(k)})$ is the partial solution for the corresponding cluster given by Lemma 3. The remaining $\hat{x}_{ij}$ is then set to be 0. It is apparent that $(\hat{x}, \hat{y})$ is a feasible solution to (P2). Since the clusters $N_k$ are disjoint, from part (iii) of Lemma 3 and Lemma 1, we have that

$$
\sum_j \sum_{i: y_i < 1} b_{ij} \hat{x}_{ij} - \sum_{i: y_i < 1} f_i \hat{y}_i
$$

$$
\geq \sum_{k \in C} O_k^* - 2 \sum_{i: y_i < 1} f_i \hat{y}_i - \sum_j \sum_{i: y_i < 1} b_{ik(i)} x_{ij}
$$

(19)

$$
\geq -3 \sum_{i: y_i < 1} f_i \hat{y}_i.
$$

Combining (18) and (19), we obtain that

$$
\text{Utility} \geq \left( \sum_i z_i - \sum_j \sum_{i: y_i = 1} \alpha_j x_{ij} \right) - 3 \sum_{i: y_i < 1} f_i \hat{y}_i
$$

$$
= \left( \sum_i z_i - \sum_j \sum_{i: y_i = 1} \alpha_j x_{ij} \right) - 3 \sum_{i: y_i < 1} f_i \hat{y}_i
$$

$$
+ \sum_j \sum_{i: y_i < 1} \alpha_j x_{ij}
$$

$$
\geq U^* + 3 \left( \sum_j \sum_{i: y_i = 1} b_{ij} x_{ij} - \sum_{i: y_i = 1} f_i \hat{y}_i \right)
$$

$$
- 2 \sum_j \sum_{i: y_i < 1} b_{ij} x_{ij} - 3 \sum_{i: y_i = 1} \sum_j b_{ij} x_{ij}
$$

$$
\geq 4 \cdot U^* - 3 \sum_j \sum_{i: y_i = 1} b_{ij} x_{ij}
$$

$$
= 4 \cdot U^* - 3 \cdot B^*.
$$

Hence, the gap between the returned solution and the optimal solution is less than

$$
U^* - \left( 4 \cdot U^* - 3 \cdot B^* \right) = 3 \cdot (B^* - U^*).
$$

(21)

ACKNOWLEDGMENT

This work was supported in part by the National R&D Key Program of Ministry of Science and Technology of China YS2017YFGH001543, the Natural Science Foundation of China (NSFC) under grant 61473251, the Natural Sciences and Engineering Research Council of Canada (NSERC), the Canada Foundation for Innovation (CFI) and the B.C. Knowledge Development Fund (BCKDF).

REFERENCES


Lin Cai (S’00-M’06-SM’10) received her M.A.Sc. and PhD degrees in electrical and computer engineering from the University of Waterloo, Waterloo, Canada, in 2002 and 2005, respectively. Since 2005, she has been with the Department of Electrical & Computer Engineering at the University of Victoria, and she is currently a Professor. Her research interests span several areas in communications and networking, with a focus on network protocol and architecture design supporting emerging multimedia traffic and Internet of Things.

She was a recipient of the NSERC Discovery Accelerator Supplement (DAS) Grants in 2010 and 2015, respectively, and the Best Paper Awards of IEEE ICC 2008 and IEEE WCNC 2011. She has founded and chaired IEEE Victoria Section Vehicular Technology and Communications Joint Societies Chapter. She has been elected to serve the IEEE Vehicular Technology Society Board of Governors, 2019–2021. She has served as a member of the Steering Committee of the IEEE Transactions on Big Data, an Associate Editor of the IEEE Internet of Things Journal, IEEE Transactions on Wireless Communications, IEEE Transactions on Vehicular Technology, EURASIP Journal on Wireless Communications and Networking, International Journal of Sensor Networks, and Journal of Communications and Networks (JCN), and as the Distinguished Lecturer of the IEEE VTS Society. She has served as a TPC symposium co-chair for IEEE Globecom'10 and Globecom'13. She is a registered professional engineer of British Columbia, Canada.

Jianping Pan (S’96–M’98–SM’08) is currently a professor of computer science at the University of Victoria, Victoria, British Columbia, Canada. He received his Bachelor’s and PhD degrees in computer science from Southeast University, Nanjing, Jiangsu, China, and he did his postdoctoral research at the University of Waterloo, Waterloo, Ontario, Canada. He also worked at Fujitsu Labs and NTT Labs. His area of specialization is computer networks and distributed systems, and his current research interests include protocols for advanced networking, performance analysis of networked systems, and applied network security. He received the IEICE Best Paper Award in 2009, the Telecommunications Advancement Foundation’s Telesys Award in 2010, the WCSP 2011 Best Paper Award, the IEEE Globecom 2011 Best Paper Award, the JSPS Invitation Fellowship in 2012, the IEEE ICC 2013 Best Paper Award, and the NSERC DAS Award in 2016, and has been serving on the technical program committees of major computer communications and networking conferences including IEEE INFOCOM, ICC, Globecom, WCNC and CCNC. He is the Ad Hoc and Sensor Networking Symposium Co-Chair of IEEE Globecom 2012 and an Associate Editor of IEEE Transactions on Vehicular Technology. He is a senior member of the ACM and a senior member of the IEEE.

Yuming Bo received his Ph.D. degree from Nanjing University of Science and Technology, China. He worked as an assistant professor, associate professor and professor, respectively, in School of Automation at Nanjing University of Science and Technology, China. He is a member of the Chinese Association of Automation and Vice Chairman of Jiangsu Branch. He is a standing council member of China Command and Control Society. His research interests are focused on filtering and system optimization. He was granted a second-level prize of Natural Science from the Ministry of Education of China in 2005 and a second-level prize of technology promotion from Shandong Province in 2012.