Ad Hoc Networks 24 (2015) 85–102

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

GeoCover: An efficient sparse coverage protocol for RSU deployment over urban VANETs

Huang Cheng, Xin Fei *, Azzedine Boukerche, Mohammed Almulla

University of Ottawa, Canada

ARTICLE INFO

Article history:
Received 10 February 2014
Received in revised form 24 July 2014
Accepted 28 July 2014
Available online 7 August 2014

Keywords:
Road geometry
Sparse coverage
Resource constraint
RSU deployment

ABSTRACT

Vehicular ad hoc networks have emerged as a promising research area. Designing a realistic coverage protocol for RSU deployment in vehicular networks presents a challenge due to different service area, assorted mobility patterns, and resource constraints. In order to resolve these problems, this paper proposes a geometry-based sparse coverage protocol *GeoCover*, which aims to consider the geometrical attributes of road networks, movement patterns of vehicles and resource limitations. By taking the dimensions of road segments into account, *GeoCover* provides a buffering operation to suit different types of road topology. By discovering *hotspots* from trace files, *GeoCover* is able to depict the mobility patterns and to discover the most valuable road area to be covered. To solve the resource-constrained coverage problem, we provide two variants of sparse coverage which take into consideration budget constraints and quality constraints, respectively. The coverage problem is resolved by both genetic algorithm and greedy algorithm. The simulation results verify that our coverage protocol is reliable and scalable for urban vehicular networks.

1. Introduction

Vehicular Ad Hoc Networks (VANETs) have elicited great interest in both industry and academia. As important components of the Intelligent Transportation System (ITS), VANETs assist in improving road safety, traffic control and infotainment as well as commercial applications [1,2]. To differentiate from general Mobile Ad Hoc Networks (MANETs), we note some of the unique characteristics of VANETs, such as the restricted movement of vehicles, the rapidly shifting network topology, and the frequent hand-offs between on-board units (OBUs) and road-side units (RSUs) [3]. Therefore, VANETs raise several challenges with regard to coverage, data dissemination, packet routing, security, privacy, etc.

Coverage is one of the key performance matrix to evaluate the Quality of Service (QoS) [4] in the network. It represents how well services have been supplied in a network. In order to maintain the network connectivity RSUs are deployed to fulfil certain coverage quality. An access point (AP) deployment problem is usually modeled as an optimization problem under different constraints, such as severe resource limitations and assorted hostile environmental conditions [5].

The coverage problem in VANETs focus on covering the street area where the Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications occur. Due to the obstruction from buildings and complex topology, the feasible region for deployment is fragmentary. Most existing coverage based deployment protocols treat vehicle networks as ideal graphs of nodes and straight lines. Such simplifications misrepresent real-world road networks, and the geometrical characteristics of vehicle networks, such as shape, direction and area. In addition, deploying

* Corresponding author. Tel.: +1 613 869 2858; fax: +1 613 562 5664.
E-mail addresses: hchen069@uottawa.ca (H. Cheng), xfei@site.uottawa.ca (X. Fei), boukerch@uottawa.ca (A. Boukerche), almulla@sci.kuniv.edu.kw (M. Almulla).

http://dx.doi.org/10.1016/j.adhoc.2014.07.022
1570-8705/

RSUs beside roads rather than intersections can result in better quality of communication [6].

Sparse coverage is a classic coverage problem for driving-assistance and business promotion in VANETs [7]. It focuses on covering critical regions with high traffic flow or crowded vehicles. Since sparse coverage requires a lower deployment cost it is suitable for cost-efficient services under a tight resource budget.

Assorted mobility pattern of vehicles is another challenge in sparse coverage problems. Based on the consideration of vehicular movement, there are three types of coverage for VANETs. Spatial coverage prefers to deploy RSUs at locations with distinct spatial attributes, such as intersections and the midpoints of roads. Spatial coverage is easy to maintain but it fails to consider the mobility of vehicles. Temporal coverage is another way to deploy RSUs; it focuses on covering the V2I communications. However, the movement of vehicles follows the drivers' own sense so that it is hard to find a certain pattern to depict the mobility. Spatiotemporal coverage considers both spatial attributes and temporal characteristics. Existing research on spatiotemporal coverage either exploits the classic three traffic flow description [8] or infers hidden mobility patterns through historical information. Our *GeoCover* is a type of spatiotemporal coverage.

Aside from the geometrical attributes and vehicular mobility, sparse coverage suffers the challenge to consider two types of constraints: budgeted and quality. Budget constraint in coverage is often treated as the deployment cost of RSUs as well as the expense of management and scheduling. Therefore, budgeted coverage keeps the total cost of RSU deployment under a predefined budget while maximizing the quality of coverage. Quality constraint in coverage is a necessary standard for the RSU deployment to satisfy. It specifies in the lower bound of performance how well these RSUs are able to cover the service area. Thus, the qualified coverage guarantees at least a minimum level of coverage while minimizing the total cost of deployment.

In this paper, we propose a geometry-based sparse coverage protocol *GeoCover* over urban VANETs to solve the above-mentioned challenges. To match the geometrical attributes, the feasible deployment region is defined using our buffering operation based on the shape, area, and other features of road systems. To apply the mobility pattern of a certain area, *GeoCover* investigates the *hotspot* area, which is a region where most vehicles accumulate. The *hotspots* are discovered by proposed α-DBSCAN algorithm and measured by a new metric *coverage value*. To meet budget and quality requirements, sparse coverage is derived as Budgeted Sparse Coverage (BSC) and Qualified Sparse Coverage (QSC). We use both genetic algorithm and greedy algorithm to solve the geometry-based sparse coverage problem. Through the comparison with related coverage, the simulation results verify the effectiveness of our resource constrained coverage.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 presents the system model of *GeoCover* and the definitions of *coverage value* and *hotspot*. The description and derivation of sparse coverage is given in Section 4. Section 5 introduces the genetic algorithm and greedy algorithm for sparse coverage. Section 6 presents the simulation process and the evaluation results. Section 7 concludes our work.

2. Related work

Coverage is one of the key matrix used to evaluate the performance of network. Many researches have been done in order to optimize the deployment of wireless sensor network [9–11]. Recently the coverage problem in the VANET has drawn many attentions because the high mobility of vehicles make the optimal deployment of roadside units (RSU) harder than in other wireless network. Affordable and reliable RSU deployment plans over vehicular networks have been studied extensively. Generally, these plans are classified into spatial coverage, temporal coverage and spatiotemporal coverage. As application requirements evolve, there are more specific objectives for optimization in coverage problems and more reduction methods designed for the coverage model over VANETs.

Spatial coverage is based on the analysis of spatial attributes of a given road system. Dubey [12] indicate that the placement of RSUs in the centers of an intersection instead of at corners will cover more road area. In this way, the data transmission rate is increased by 15% in the service area. However, Kafsi et al. [13] note that even though most vehicles accumulate in congested intersections, the isolated vehicles are more likely to appear in the middle of road segments or the entering points of the domain. Therefore, placing RSUs in the middle of the road is a more efficient strategy for avoiding uncovered isolated vehicles.

Liu et al. [14] explore the hidden connectivity in urban vehicular networks by transferring the original road network into an "intersection graph". According to their research, the likelihood that the V2I communication occurs at the intersections is weighted by the length of the road segments. Also, most of the data disseminations are completed at the connected dominating set (CDS) on the intersection graph. Lee and Kim [15] also consider the intersections as potential deployment locations of RSUs. They propose a ranking scheme to order these locations based on the number of reports sent by taxis within the communication range of each RSU.

By evaluating the structural properties of road networks, some centrality-based coverage strategies come into being. By simply mapping the crossroads to graph vertices and treating the roads to graph edges, Crucitti et al. [16] compare five different centrality indices over real geographic networks: degree, closeness, betweenness, straightness and information. They indicate that centrality is applicable to measurements in which some nodes are more important (central) than others in a network. Based on the five centrality indices, Do et al. [17] integrate centrality with VANETs and the related connectivity analysis. They find that centrality metrics would not outperform density-based strategies, but that they can still offer some interesting functionalities, such as monitoring the traffic flow with betweenness centrality.

Kchiche and Kamoun [18] propose a greedy approach based on group centrality to select the best organization of RSUs. Group centrality aims to measure the centrality of a group rather than of individuals. For each centrality metric, the specific averaging or summing methods are exploited to obtain the group centrality. Kchiche and Kamoun succeed in achieving the best performance in terms of delay and overhead during V2V communication in their scenario. Through the simulations in [19], the authors further show that the use of centrality can optimize the performance of VANETs, particularly in low density areas and in cases of long-distance communication.

Spatial coverage considers only the spatial attributes of a road network, while failing to consider the mobility of vehicles. With a better understanding of the communication system in VANETs, temporal coverage has been researched widely so as to cover the communications between high-moving vehicles and fixed RSUs. Abdrabou and Zhuang [20] study the end-to-end packet delivery delay in multi-hop VANETs. Since the end-to-end delay occurs in the packet forwarding, the number of hops is a good metric for temporal coverage. Wu et al. [21] consider both the movement of vehicles and multi-hop forwarding. They developed an analytical model to study the spatial propagation of information in VANETs. This model can be used to develop a location where the average delay in the network is bound by a threshold.

Li et al. [22] consider the method to minimize the average number of hops from RSUs to gateways. The gateways are used in their scenario to connect RSUs to the Internet. In the coverage model, each RSU is connected to both vehicles and a center gateway, so that the optimal deployment of the gateway assists in improving the performance of communication. More specifically, they have obtained the results for optimal deployment of gateways in 1-D vehicular networks. They also proposed two algorithms for the placement of the center gateway in 2-D dense vehicular networks. Their final results show that when the average number of hops between gateways and RSUs is minimized, the average capacity of each RSU is also maximized.

Other types of temporal coverage focus on the contacts of OBUs and RSUs. Trullols et al. [23] seek to maximize the number of vehicles that make contact with the RSUs and formulate their problem as a Maximum Coverage Problem. Even though the formulated problem is NP-hard, the authors tackle it through heuristic algorithms with different levels of complexity. They further formulate the problem into another version, which is to guarantee that the majority of vehicular travel is covered by one or more RSUs for a sufficient amount of time. Lochert et al. [24] present a landmark-based aggregation scheme for saving travel time in road networks. They distribute information about the travel times between prominent points and landmarks and estimate the travel time savings achieved by a given vector of active RSU locations. These estimations are then used as fitness metrics in the genetic algorithm to make an application-centric optimization of RSU deployment. They show that optimal placement improves information dissemination over large distances, especially in a large-scale city model of VANETs.

Sun et al. [25] mention that the extra overhead time used for adjusting routes to update certificates could be improved by optimization of the RSU deployment. To ensure security and privacy preservation in VANETs, the authors minimize the time that vehicles reach each RSU in order to provide a short-time certificate. They model such an optimal deployment problem as a set-covering problem which is NP-hard, and solve it with a classical greedy algorithm. Kaur and Kaur [26] discovered that the optimistic deployment of RSUs takes too much time due to its sequential processing. Therefore, they provided a parallelization-based strategy to place RSUs by using fork and join algorithms. Their task duplication based scheduling makes full use of the idle time of processors to duplicate the tasks. By using Trivial Database (TDB), the authors then minimize the parallel time taken to deploy RSUs with high efficiency and maximum area coverage.

Temporal coverage presents another way to deploy RSUs, which focuses on covering the V2I communications. However, the movement of vehicles follows the drivers' own decisions so that it is hard to find a specific pattern to depict mobility. Therefore, researchers have proposed three descriptions according to the traffic flow theory: the microscopic description, the kinetic description and the macroscopic description [8]. In the microscopic description, each vehicle is identified individually through speed, weather, habits, etc. Assorted factors make the implementation of a microscopic model difficult. The kinetic description is a global state description which shows statistical distribution on each lane for position and velocity. As for the macroscopic model, it is also a global state description incorporating locally averaged quantities in three major traffic parameters: density, velocity and flow.

In spatiotemporal coverage, choosing the most suitable traffic flow model is a very important step. Some researchers exploit the classic three traffic flow descriptions mentioned before, or infer their own mobility pattern through historical information. Xiong et al. [27] investigate a time-stable mobility pattern from realistic traces of buses, taxis and pedestrians. They observe the mobility pattern and characterize it with a graph model. After dividing a road system into several non-overlapping uniform zones, the authors calculate the time-stable transition probabilities between all zones. They then transfer the statistical mobility pattern to a time homogeneous Markov chain. The transferred gateway deployment problem is then reduced into an NP-hard vertex selection problem.

Zhu et al. [28] remark that the movement of vehicles shows a strong regularity by mining a large dataset of real vehicular GPS traces. They model the movement of vehicles by the Markov chain. In this way, the mobility pattern for vehicles is extracted from the historical vehicular trace files. Based on the assumption that the future movements of vehicles are able to be treated as a priori knowledge, the authors provide efficient algorithms to determine RSU deployment. If the future traces are unknown, they formulate a new objective, which is to maximize the expectation of the weighted sensing coverage, by taking the random vehicular mobility into account. They also show that their greedy algorithm provides a performance guarantee.

To improve the cooperative downloading of data among vehicles in an urban vehicular network, Fiore and Barcelo-Ordinas [29] devise a strategy for RSU deployment based on vehicular traffic flow analysis. Not all urban roads are identical and some of them are more congested or have higher speed limits than others. The authors take all this into account and thus transfer the road topology into a graph where vertices are intersections and edges are streets. Based on this graph, they evaluate the average time vehicles spend to travel to each edge and redeem the calculation results as traversing volumes. They then propose that placing RSUs at the crossing-volume area will maximize the potential for collaboration among vehicles as relay nodes.

3. System model and definitions

In this section, we discuss the system model for our geometry-based sparse coverage protocol *GeoCover* and give two new definitions for the terms *hotspot* and *coverage value*, in our effort to analyze the historical trace files.

3.1. System model

Sparse coverage over VANETs is designed for traffic monitoring and management, navigation of cooperative local services and advertisement delivery. To provide realistic coverage, the geometrical attributes, mobility patterns and resource constraints all merit concern. The *GeoCover* algorithm resolves the three issues simultaneously.

Fig. 1 shows the process of *GeoCover* algorithm. The algorithm start from obtaining the mobility pattern and assigning the coverage value for each cell. The valued cells are then grouped into *hotspots* by α-DBSCAN algorithm. Because the two-dimensional properties of a road network will impact the effects of deployment, we exploit a buffering operation method, as shown in Figs. 5 and 6, to mesh the buffer regions along the road segments. After the buffering operation, we resolve the BSC and the QSC based on different optimization objectives. The BSC problem and the QSC problem are modeled as the Budgeted Maximum Coverage (BMC) problem [30] and the Set Cover Problem (SCP) [31], respectively. It has been proven that the BMC is a maximum version of the SCP [32], so that the BSC problem and the QSC problem can be understood in terms of each other. Due to the NP-hardness of the BSC problem and the QSC problem, we first propose a genetic algorithm to resolve the problems, and then employ a greedy algorithm to provide the performance guarantee for our solution.

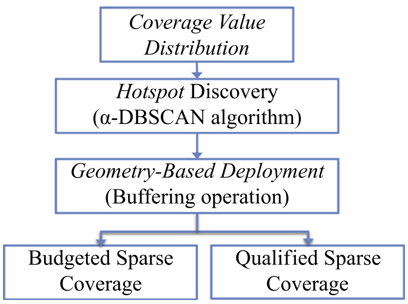


Fig. 1. Process of *GeoCover*.

3.2. Definition of coverage value

Definition 1. *Coverage value*: The value of a specific region that measures how much communication volume is covered.

Because the regions with great amounts of traffic volume will experience an increased likelihood of communication, we use *coverage value* to present the communication volume of a certain area. The region with a high *coverage value* is a valuable region and should be covered. Based on the probability distribution model proposed in [33] we apply a 2-D Gaussian distribution to vehicle speeds, vehicular density and traffic flow. At each position, we obtain the probability of parameters and then use the formula (1) to calculate the *coverage value*.

High vehicle speed will result in frequent hand-offs, so that speed is inversely proportional to *coverage value*. In an extreme situation, vehicles that are parked or stopped have the highest likelihood of accomplishing information exchange with RSUs. Vehicular density and traffic flow reflect the degree of vehicle accumulation, so that density and flow are directly proportional to *coverage value*.

$$coverage\ value = flow * density/speed \tag{1}$$

Fig. 2 shows the assignment of *coverage value* to a cell-based road system. The cells with higher traffic flow and density will be assigned a higher *coverage value*, which is represented by an increased color opacity in the figure.

3.3. Definition of hotspot

Definition 2. *Hotspot*: A group of cells in which the *coverage value* is larger than a threshold α.

Hotspot represents the area where most vehicles accumulate. In order to discover the *hotspots* we divide the area into fixed-sized cells and assign the corresponding *coverage value* to each cell. A density-based clustering algorithm,α-DBSCAN, is proposed to identify clusters with irregular and trivial geometry characteristics.

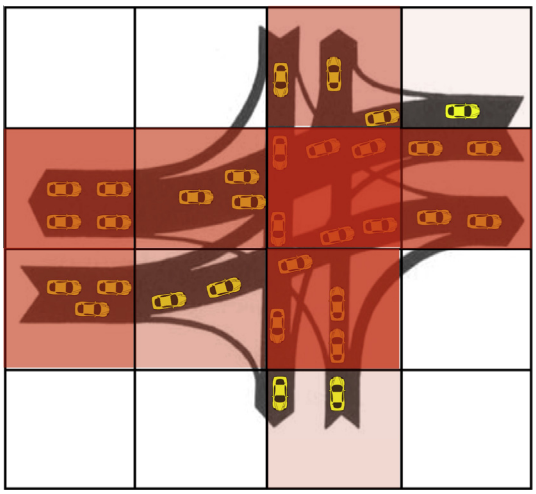


Fig. 2. Assignment of *coverage value*.

The α-DBSCAN algorithm is a revised version of classical density-based algorithm DBSCAN [34]. DBSCAN defines two parameters: ϵ (searching radius) and *minPts* (minimum points required to form a cluster). The algorithm starts from an arbitrary core point; and, it then absorbs all the neighbor points within distance ϵ as its cluster members. When the number of neighboring nodes reaches the minimum requirement (*minPts*), a cluster is formed. In α-DBSCAN, each cell is a point in the algorithm. The original parameter *minPts* is replaced by α, the threshold of the average *coverage value*. Since our *GeoCover* algorithm deploys RSUs beside roads to cover the street area only, the discovery of *hotspots* does not consider the obstacles of building in urban area.

Algorithm 1. α-DBSCAN – *hotspot* discovery algorithm

Input: A_{road}, ϵ(searching radius), α (*coverage value*)
Output: S

```
S ← ∅;
for each UNVISITED g in A_road do
    mark g as VISITED;
    N ←getNeighbours (g, ϵ);
    if V_N ≥ α then
        create new H ∈ S;
        H ← g;
        for each UNVISITED g′ in N do
            mark g′ as VISITED;
            N′ ←getNeighbours(g′, ϵ);
            if V_N′ ≥ α then
                N ← N ∪ N′;
            if g′ doesn't belong to any H′ ∈ S then
                H ← H ∪ g′;
```

Algorithm 1 shows the pseudocode of α-DBSCAN algorithm. Based on a different α, another parameter ϵ is able to be estimated by the k-dist graph [34], so that we treat the threshold α as the pacing factor. If the average *coverage value* of cell g's neighbors is less than α, this cell will be treated as noise and removed. Conversely, g is a core point and a new *hotspot* will be formed. The process is repeated until all of the cells have been visited and all density-reachable areas have been defined. The final output of α-DBSCAN is a set of *hotspots* $S = \{H_1, H_2, H_3, \ldots, H_n\}$.

4. Geometry-based sparse coverage

In this section, we discuss how to deploy RSUs based on *hotspots* and how to exploit geometrical attributes of road segments. We first propose the formation of a sparse coverage model. We then propose the buffering operation to define the candidate deployment locations. Finally, the two variants of the sparse coverage, BSC and QSC, are proposed to meet different optimization objectives.

4.1. Sparse coverage model

Fig. 3 is an example of sparse coverage model. Each cluster (*hotspot*) in Fig. 3(a) was discovered by the α-DBSCAN algorithm and contains the cells to be covered. Assuming the RSU's propagation model is a disk; RSUs are deployed on the buffered roadside area which is also divided into cells. By choosing different roadside cells to deploy RSUs we can cover the hotspots as much as we can and meet the budget or quality requirement. Fig. 3(b) is an example of the cell-based sparse coverage on a *hotspot*. In Fig. 3(b), there are 14 different cells $\{g_1, g_2, g_3, \ldots, g_{14}\}$. Each cell owns a *coverage value*. If, in a road network scenario, there are a total of 6 RSUs $\{A, B, C, D, E, F\}$, then a possible deployment is shown as the following:

$$\mathrm{RSU}_A = \{a * 2, ab, ad\}$$
$$\mathrm{RSU}_B = \{ab, bc * 2, bcd\}$$
$$\mathrm{RSU}_C = \{c, bc * 2, bcd\}$$
$$\mathrm{RSU}_D = \{df, de * 3, ad, bcd\}$$
$$\mathrm{RSU}_E = \{de * 2\}$$
$$\mathrm{RSU}_F = \{f * 3, df\}$$

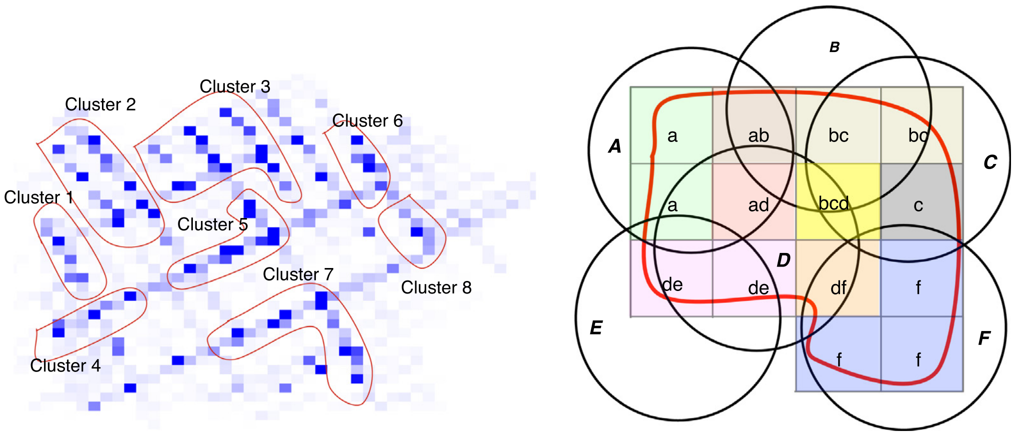
The label of a cell is a combination of all the RSUs that cover the cell. For example, cell *bcd* is covered by three RSUs: b, c and d. The road cells with same label belong to the same *set*: in the case where the two *de* cells belong to *set de*. As for the cells only covered by one RSU, they are regarded as their own *set*. Thus, there are a total of 9 *set* in Fig. 3(b) as shown by the color code. For each *set*, we define its weight w as the mean coverage value of all the road cells in the *set*.

Therefore, a sparse coverage is a problem to cover all the *set*, $S = \{S_1, S_2, \ldots, S_m\}$ using a set of RSU $U = \{U_1, U_2, \ldots, U_n\}$ Each section S_j is associated with a weight w and each RSU U_i is associated with a cost c, the expense used to deploy the RSU.

4.2. Buffering operation

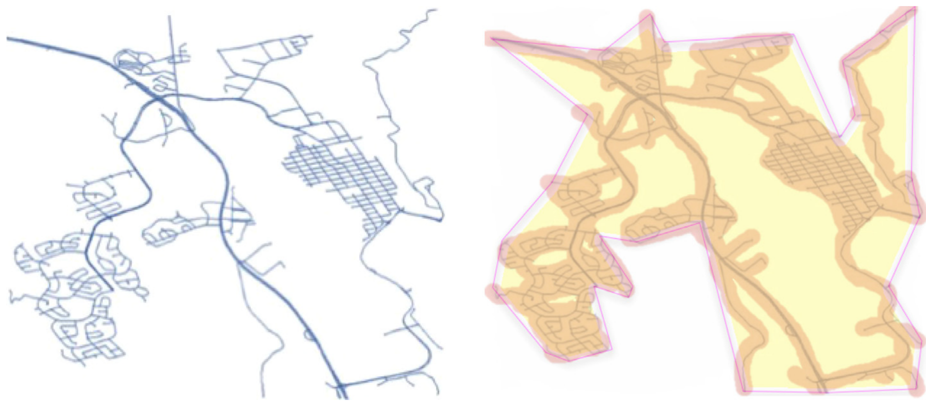
After the design of the sparse coverage model, the buffering operation is exploited to match the geometrical attributes of road segments. Because the area around road segments is the primary zone in which to place RSUs, the definition of a feasible region is required in order to define the extracted roadside area. Fig. 4(a) represents the vehicle networks in the Yukon Territory of Canada obtained from ArcGIS [35]. We can see in the figure that real-world road networks consists of all kinds of crossings, turns, forks, curves, etc. Even though these elements are of various shapes and areas, the buffering operation is still able to pick up the feasible region according to different road geometrical characteristics [36].

Fig. 5 provides a sketch of the buffering operation on straight road segments. The coordinates of roads are known; thus we simply add buffering lines, the shadow parts shown in Fig. 5, on both sides of the road segment along the edges of the street. Then, by defining the width of the buffering line, the feasible region is marked. The width of the buffering-line region is adjusted according to different RSU transmission ranges. Since the candidate deployment locations should be as close as possible to the road area and avoid occupy the road space, we set $width_{buffer} = width_{road}$ in our simulation. By adding buffer



(a) *hotspot* Discovery (b) Sparse Coverage Model

Fig. 3. Formation of sparse coverage model.



(a) Yukon Territory (before operation) (b) Yukon Territory (after operation)

Fig. 4. Illustration of buffering operation.

regions along both sides of the roads, we can divide the buffers into cells and treat each cell as a candidate deployment location.

Even if the road segment is in the shape of a curve, the buffering operation also works by breaking a curve into a series of line segments. Fig. 6 shows the buffering operation on curved road segments. This idea is the same as those used for the approaches in OpenStreetMap [37] when circular arcs are represented.

With the help of the buffering operation, we extract the feasible deployment region from the original road networks. Fig. 4(b) shows the Yukon Territory after the buffering operation. The shadow area in Fig. 4(b) represents the feasible region.

4.3. Budgeted Sparse Coverage

In the field of sparse coverage, a common requirement for deployment is to maximize the total deployment value of infrastructures under a predefined budget on the available number of RSUs. We define this problem as Budgeted

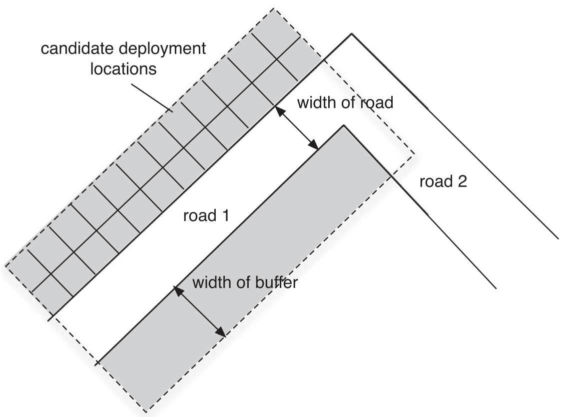


Fig. 5. Buffering operation on straight roads.

Sparse Coverage (BSC) and the definition is shown as follows.

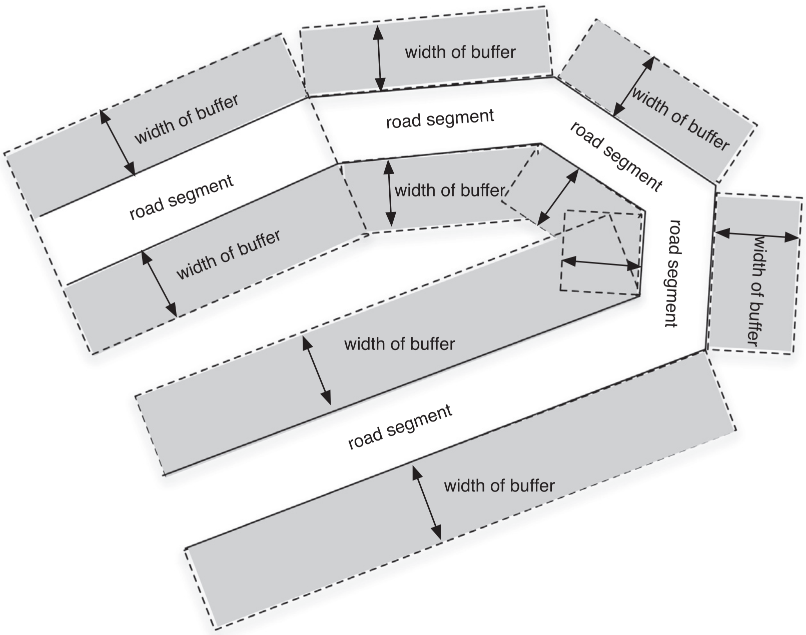


Fig. 6. Buffering operation on curved roads.

Definition 3. *Budgeted Sparse Coverage:* A deployment of RSUs provides Budgeted Sparse Coverage to a road network R, if the average weight of areas cover by selected RSUs is maximized; and, the total deployment cost is no larger than a given budget B.

The ILP formulation of the optimization of the BSC problem is shown as follows:

$$\begin{aligned} \text{maximize} \quad & \mathrm{BSC}(x) = \sum_{j=1}^{m} y_j \cdot w_j \\ \text{subject to} \quad & \sum_{i=1}^{n} x_i \cdot c_i \leqslant B \\ & \sum_{S_j \in U_i} x_i \geqslant y_i \\ & x_i, y_j \in \{0,1\}, \quad \forall 1 \leqslant i \leqslant n, \quad 1 \leqslant j \leqslant m \end{aligned} \tag{2}$$

where x_i and y_j represent the 0–1 selection of RSUs and *sections*. If $x_i = 1$ then the corresponding location is selected to deploy RSUs. If $y_j = 1$ then the corresponding *section* is covered by at least one RSU.

We reduce the BSC problem to the BMC problem [30] by considering the *sections* as a domain of elements and RSUs as a subset of *sections*. Since the BMC problem is a well-known NP-hard problem, our BSC problem is also NP-hard. To solve such an NP-hard area coverage problem, we use a natural framework for the *Greedy Cover* algorithm, as shown in Algorithm 3.

4.4. Qualified Sparse Coverage

Budget constraints aside, people are sometimes more concerned about the quality of the RSU deployment rather than the cost of the infrastructures. Thus, it raises the question of how to minimize the total expense of RSUs while guaranteeing quality and value. We refer to this problem as Qualified Sparse Coverage (QSC), and the formal definition is as follows.

Definition 4. *Qualified Sparse Coverage:* A deployment of RSUs provides Qualified Sparse Coverage to a road network R, if the total deployment cost of selected candidate RSUs is minimized; and, the average weight of covered areas meets the lowest *coverage value* threshold Q.

The ILP formulation of the QSC problem is shown as follows:

$$\begin{aligned} \text{minimize} \quad & \mathrm{QSC}(x) = \sum_{i=1}^{n} x_i \cdot c_i \\ \text{subject to} \quad & \sum_{j=1}^{m} y_j \cdot w_j \geqslant Q \\ & \sum_{S_j \in U_i} x_i \geqslant y_i \\ & x_i, y_j \in \{0,1\}, \quad \forall 1 \leqslant i \leqslant n, \quad 1 \leqslant j \leqslant m \end{aligned} \tag{3}$$

where x_i and y_j represent the 0–1 selection of RSUs and *sections*. If $x_i = 1$ then the corresponding location is selected to deploy RSUs. If $y_j = 1$ then the corresponding *section* is covered by at least one RSU.

We reduce the QSC problem to the SCP [31] by considering the *sections* as the universe of elements and RSUs as a set of subsets of *sections*. Since SCP is a classical NP-hard problem, our QSC problem is also NP-hard. Based on the study of Khuller et al. [30], the unit cost version of the MCP is a straightforward reduction from the SCP. Therefore, the QSC problem is a minimum variant of the BSC problem. Both of the NP-hard problems can be approximately solved by the *Greedy Cover* algorithm.

5. Algorithms and analysis

In this section, we propose two types of algorithms to resolve the *GeoCover* model. The first algorithm is the genetic algorithm, which aims to search for the best solution globally in the solution space. Because it is difficult to guarantee the performance of the genetic algorithm, we provide a greedy algorithm, BSC_{unit}, to solve the coverage problem. BSC_{unit} is able to provide $1 - 1/e$ and $\ln n + 1$ approximations for the BSC problem and the QSC problem, respectively.

5.1. Genetic algorithm (GeoCover-genetic)

Due to the NP-hard characteristics of the *GeoCover* problem, we use the genetic algorithm (GA) to find an optimal RSU placement solution with a fixed RSU transmission range. Algorithm 1 is the algorithmic description of GA.

Algorithm 2. Genetic algorithm

```
Input: candidate locations, threshold
Output: optimal chromosome
Encoding and Initialization
    Encoding chromosome;
    Set area, transmission range and number of RSUs;
    Initial population randomly;
while fitness ≤ threshold do
    Selection
        Rank chromosomes with fitness function;
        Place optimal population into next generation;
    Reproduction
        Crossover to generate new offsprings;
        Mutate to generate new gene;
return optimal chromosome
```

5.1.1. Encoding and initialization

The final output of BSC is the position of all RSUs, so we encode the coordinates of each RSU as the gene. In equation $gene = (l_x, l_y), l_x$ and l_y are 2-dimensional coordinates of an AP. Because each solution is a set of RSU coordinates, we encode the deployment solution as the chromosome. In formula $chromosome = (g_1, g_2, g_3, \ldots, g_n), g$ is the gene. If the number of RSUs is set to n, each chromosome consists of n different genes.

We generate a group of chromosomes, known as population, to elect the optimal solution in GA. At the beginning, we generate m different chromosomes in an initial population through normal distribution. Each generation has a population, which is defined as $P = (c_1, c_2, c_3, \ldots, c_m)$.

5.1.2. Selection, reproduction and termination

In the selection loop of GA, chromosomes with higher priority will be selected into the next generation. The fitness function is used to rank chromosomes at each generation. The fitness function is as follows:

$$R = \frac{\sum_{i=1}^{G_{road}} A_i}{G_{road}}, \quad \begin{cases} A_i = 1 | q_i \subseteq C_{\text{RSU}} \\ A_i = 0 | q_i \not\subset C_{\text{RSU}} \end{cases}$$

We set the coverage ratio R as the rank criterion. To facilitate computation, we divided the area of feasible regions and roads into 1 m $\times$ 1 m cells. Each cell is represented by q_i. If a cell is within the transmission range of at least one AP, we claim that the cell is covered. C_{RSU} is the set of cells covered by RSUs and G_{road} is the number of cells in the road area.

In the reproduction stage, we used the "cut and splice" approach to generate new offspring. We selected the same crossover point c on two parent chromosomes; and, we cut each chromosome into two parts at the position of c. We then exchanged the second part of two parents so that two new children chromosomes were generated. The "cut and splice" approach is as follows:

$$child^{1'} = \{\{g_1^1, g_2^1, \ldots, g_c^1\}\{g_{c+1}^2, g_{c+2}^2, \ldots, g_n^2\}\}$$
$$child^{2'} = \{\{g_1^2, g_2^2, \ldots, g_c^2\}\{g_{c+1}^1, g_{c+2}^1, \ldots, g_n^1\}\}$$

Reproduction from crossover may only result in a locally optimal solution, since the genes only come from parents. Therefore, we used mutation to produce new information for genes. Mutation happens with a predefined probability $p(0 \leqslant p \leqslant 1)$, which is very small, so that it will not develop into an intolerable influence. Because a gene is denoted by two-dimensional coordinates, we chose to add a random offset ε to the coordinate value. The choice of ε should be applied very carefully in order to avoid genetic drift. Therefore, we have set the offset value to the same size as the cell size so that every time the mutation happens the position of an RSU only moves to its neighbor cell. The mutation function is shown below:

$$gene' = \{l_x + \varepsilon, l_y + \varepsilon\}$$

When the fitness of a chromosome exceeds threshold (i.e. 90%), we claim that the optimal solution for node distribution has been found. The algorithm will then terminate. To guarantee the termination of evolution, we also bound the number of generations by 1000. The algorithm will terminate and out the best result if the generation is over 1000 (see Table 1).

5.1.3. Time complexity

The selection of population size and mutation probability may actually lead GA to converge towards local optimal positioning or even genetic drift. But it is impractical to define the upper and lower bounds, for those parameters and genetic algorithms do not scale well with complexity. Theoretically speaking, the time complexity of our GA is $O(gen * (sel + cro + mut))$. The number of generations is denoted by gen, which is a constant. If $T(R)$ denotes the computational cost of the fitness function, $|P|$ represents the size of the population, and $T(sorting)$ is the complexity

Table 1
Genetic algorithm parameters.

Parameter	Value
GA population size	100
GA crossover	Single point
GA mutation offset	2
GA generation	1000
GA fitness threshold	90%
Cost per RSU	1

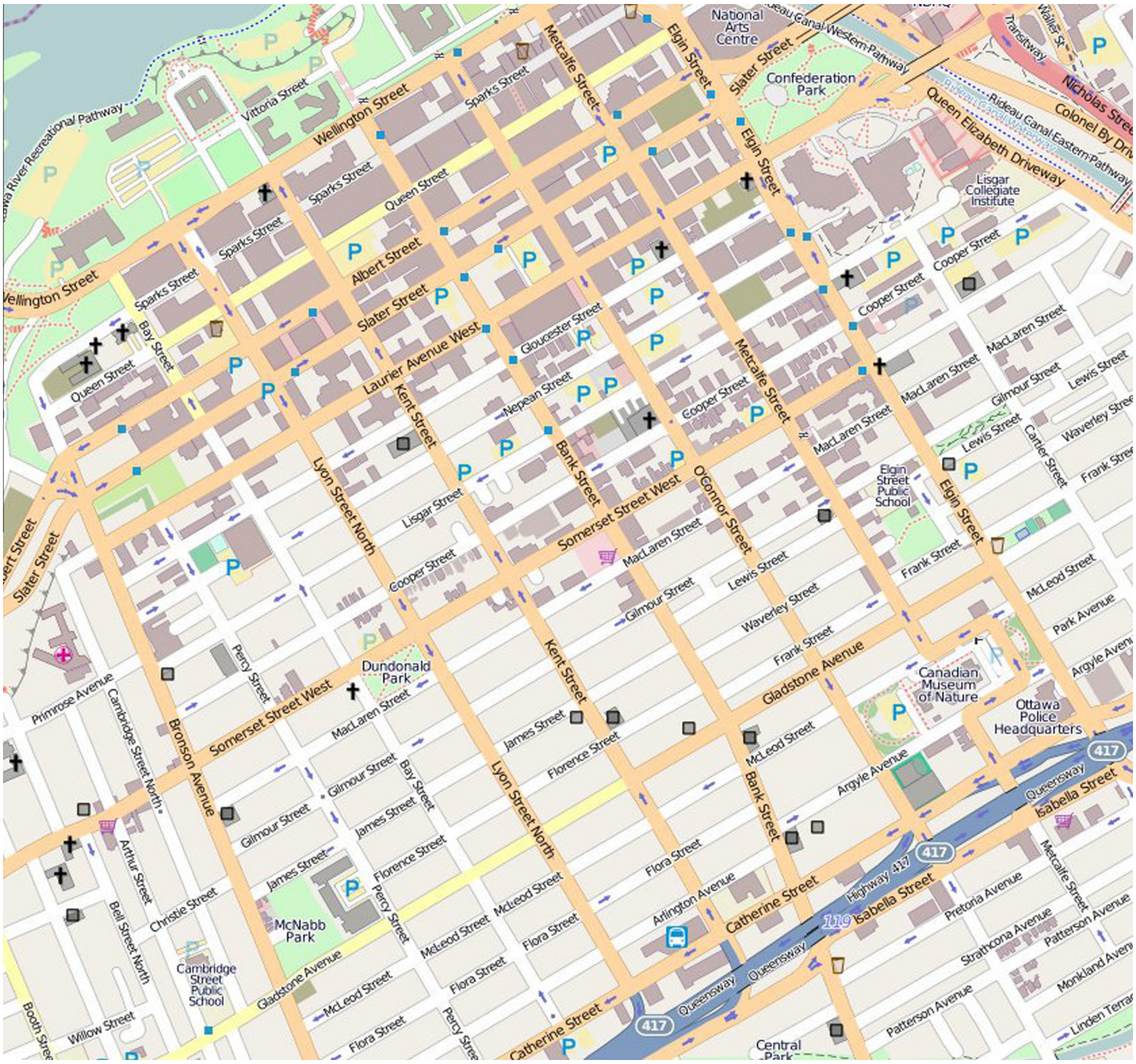


Fig. 7. Downtown map of Ottawa.

of sorting; thus, *sel*, which is set as the time complexity of selection, should be $max(T(R) * |P|, T(sorting))$. The time complexity of crossover is denoted by *cro*, which is $O(|P|^2)$ while *mut* is the computational cost of mutation, which is $O(1)$ (the product of probability p and single mutation $O(1)$). Because the complexity of GA is hard to determine through pure theory, the running time of GA in a practical simulation is more meaningful for analysis.

5.2. Greedy algorithm (GeoCover-greedy)

Although GA is able to provide the best global solution by heuristically searching the solution space, it is hard to guarantee an approximation for an optimal deployment solution. We design a greedy algorithm to solve the geometry-based sparse coverage problem instead. Algorithm 3, *Greedy Cover*, is the basic framework for the greedy algorithm on the general coverage problem.

Algorithm 3. *Greedy Cover* – Greedy algorithm

```
Input: S = {S_j}, U = {U_i}
repeat
    select U_i to cover a set of UNVISITED S_j with maximum profit;
    mark the covered S' = {S_j} as VISITED;
until done;
//S is the set to be covered,
//U is the set used to cover S
```

Based on the *Greedy Cover* framework, we propose a unit-profit-version greedy algorithm, BSC_{unit}, to obtain an approximate optimal solution in polynomial time. Algorithm 4 describes the details of the BSC_{unit} algorithm. In our greedy algorithm BSC_{unit}, each set U_i has a unit cost and the goal is to find a subset of U so that the total weight of covered S is maximized. Assuming there are only K RSUs available for deployment under the budget B, we can use the enumeration technique to select subsets of U with

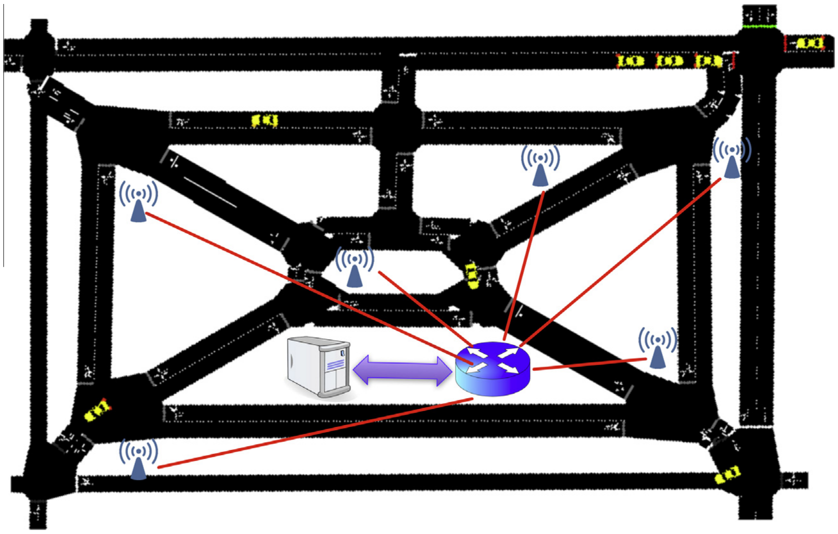


Fig. 8. Simulation scenario.

the cardinality of K. Let $w(U')$ be the total weight of all elements covered by RSUs in U' and $c(U')$ be the total deployment cost of all RSUs in U'. The output of BSC_{unit} is the candidate deployment solution D with maximum weight. Since every time we choose the best candidate deployment location based on a ranking scheme, the total time complexity of *Greedy Cover* framework is $O(KlgK)$.

Because the BSC problem and the QSC problem are interchangeable, both coverage problems can be solved by the *Greedy Cover* framework, so it is known as the BSC_{unit} algorithm. However, due to the different optimization objectives, BSC_{unit} results in different approximations in the BSC problem and the QSC problem.

Algorithm 4. BSC_{unit} – Greedy algorithm

Input: $S = \{S_j\}, U = \{U_i\}, K, B$
Output: D
$D_1 \leftarrow argmax\{w(U')|U' \subseteq U, |U'| < K, c(U') \leq B\}$;
$D_2 \leftarrow \emptyset, D \leftarrow \emptyset$;
forall the $\{U'|U' \subseteq U, |U'| = K, c(U') \leq B\}$ **do**
    $S \leftarrow U \backslash U'$;
    repeat
        select $U_i \leftarrow argmax\{\frac{w(U_i)}{c_i}|U_i \subseteq S\}$;
        if $c(U') + c_i \leq B$ **then**
            $U' \leftarrow U' \cup U_i$;
        $S \leftarrow S \backslash U_i$;
    until $S \leftarrow \emptyset$;
    if $w(U') > w(D_2)$ **then**
        $D_2 \leftarrow U'$;
return $argmax\{w(D)|D \in \{D_1, D_2\}\}$

Theorem 1. *BSC_{unit} achieves an approximation factor of $1 - 1/e$ for the BSC problem.*

Since the proof of BSC_{unit} algorithm is similar to the proof of greedy algorithms used in the BMC problem [30] and MCP [38], we simply declare the process of the proof. Let *OPT* denote the *sections* covered by optimal solution D for the BSC problem, U_i denote the new *sections* added at i-th iteration, and $U'_i = \sum_{j=1}^{i} U_i$ and $\widehat{U}_i = OPT - U'_i$. It is obvious that $U_0 = 0, U'_i$ is the *sections* already covered by the algorithm at iteration i and $\widehat{U}_0 = OPT$.

We first prove the following two lemmas:

Lemma 1. $w(U_{i+1}) \geqslant w(\widehat{U}_i)/K$.

Proof. At each iteration, BSC_{unit} selects the new RSUs with the maximum unit weight of the others. Since the optimal solution uses K RSUs to cover OPT *subsections*, some RSUs must cover at least $1/K$ fraction of the OPT *subsections*. Therefore, the newly added RSU must cover at least $1/K$ of the remaining *subsections* from OPT, which means $w(U_{i+1}) \geqslant w(\widehat{U}_i)/K$. $\square$

Lemma 2. $w(\widehat{U}_{i+1}) \leqslant (1 - 1/K)^{i+1} \cdot w(OPT)$.

Proof. We prove Lemma 2 through induction. The base case is true when $i = 0$. We then set the induction hypothesis that $w(\widehat{U}_i) \leqslant (1 - 1/K)^i \cdot w(OPT)$. Finally, we prove the induction steps:

Table 2
Simulation parameters.

Simulator	NS-2.35/SUMO
Mobility model	Car-following model
Area of map	2300 m × 2100 m
Number of vehicles	100
Vehicle speeds	0–20 m/s
PHY/MAC	IEEE 802.11p
Network protocol	Mobile IP V4
Routing protocol	AODV/GPSR
Transport protocol	UDP
Network traffic	CBR (160 bytes, 50 pps)
Simulation time	500 s

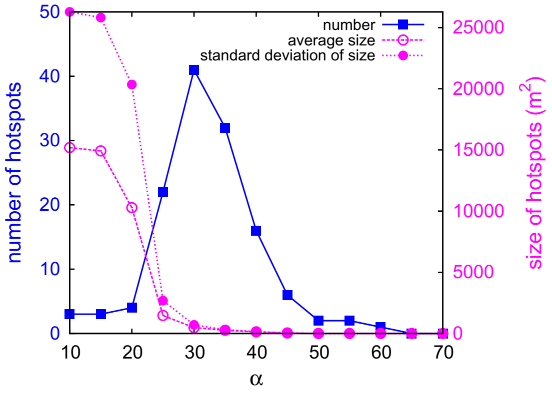


Fig. 9. *hotspot* Discovery analysis.

Table 3
BSC vs. QSC.

BSC – number of RSUs	QSC – coverage value
20	13.567
40	23.948
60	31.752
80	39.681
100	45.225
120	50.513
140	53.588

$$w(\widehat{U}_{i+1}) \leqslant w(\widehat{U}_i) - w(U_{i+1})$$
$$\leqslant w(\widehat{U}_i)(1-1/K) \text{ (using Lemma 1)}$$
$$\leqslant (1-1/K)^{i+1} \cdot w(OPT) \quad \square \tag{4}$$

Now we start to prove Theorem 1.

Proof. It follows from Lemma 2 that

$$w(\widehat{U}_{i+1}) \leqslant (1-1/K)^{i+1} \cdot w(OPT) \leqslant w(OPT)/e \tag{5}$$

Therefore,

$$w(U'_i) = w(OPT) - w(\widehat{U}_i) \geqslant w(OPT) - w(OPT)/e$$
$$= (1-1/e)w(OPT) \tag{6}$$

In the case of BSC, the BSC_{unit} algorithm is complete when precisely K RSUs have been selected and budget B has been met. As for QSC, BSC_{unit} ends when the total quality of covered *sections* reaches Q. Even though the basic algorithm framework is the same for the BSC problem and the QSC problem, the approximated ratio for the optimal solution is different for each problem. $\square$

Theorem 2. *The BSC_{unit} algorithm achieves an approximate factor of $1 + \ln n$ for the QSC problem.*

Since the proof of the BSC_{unit} algorithm is similar to the proof of the greedy algorithms used in the SCP problem [38], we simply declare the process of the proof. Let *OP*

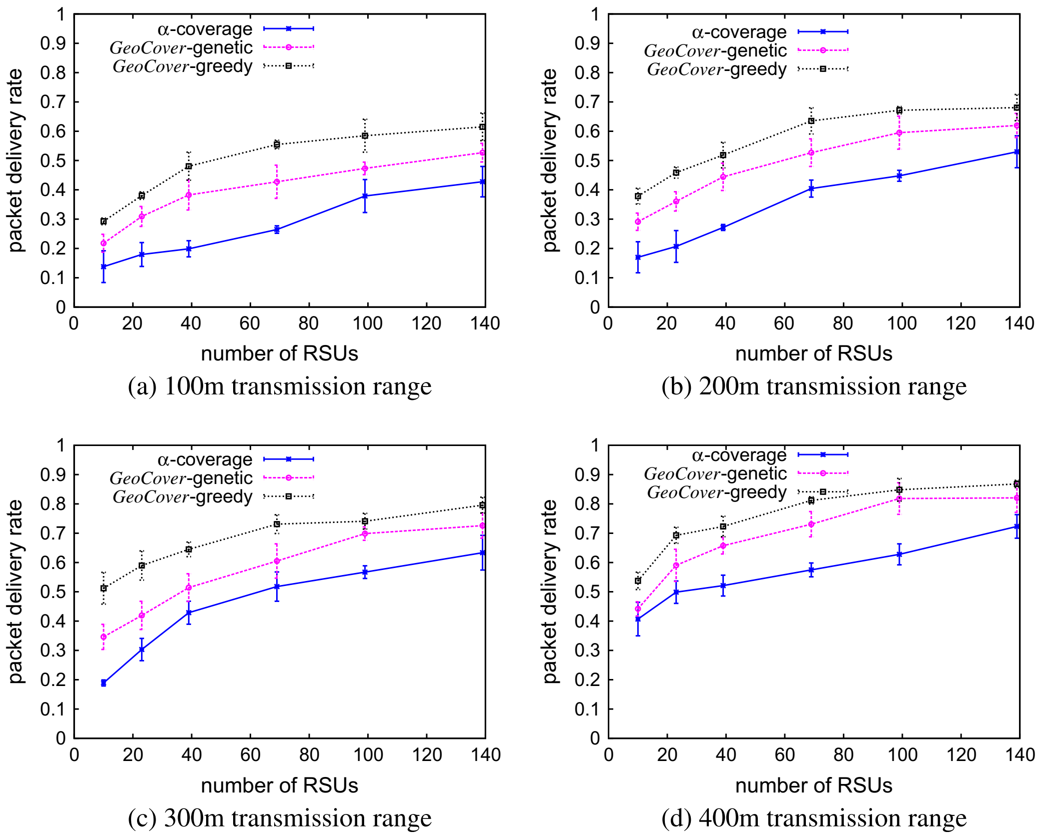
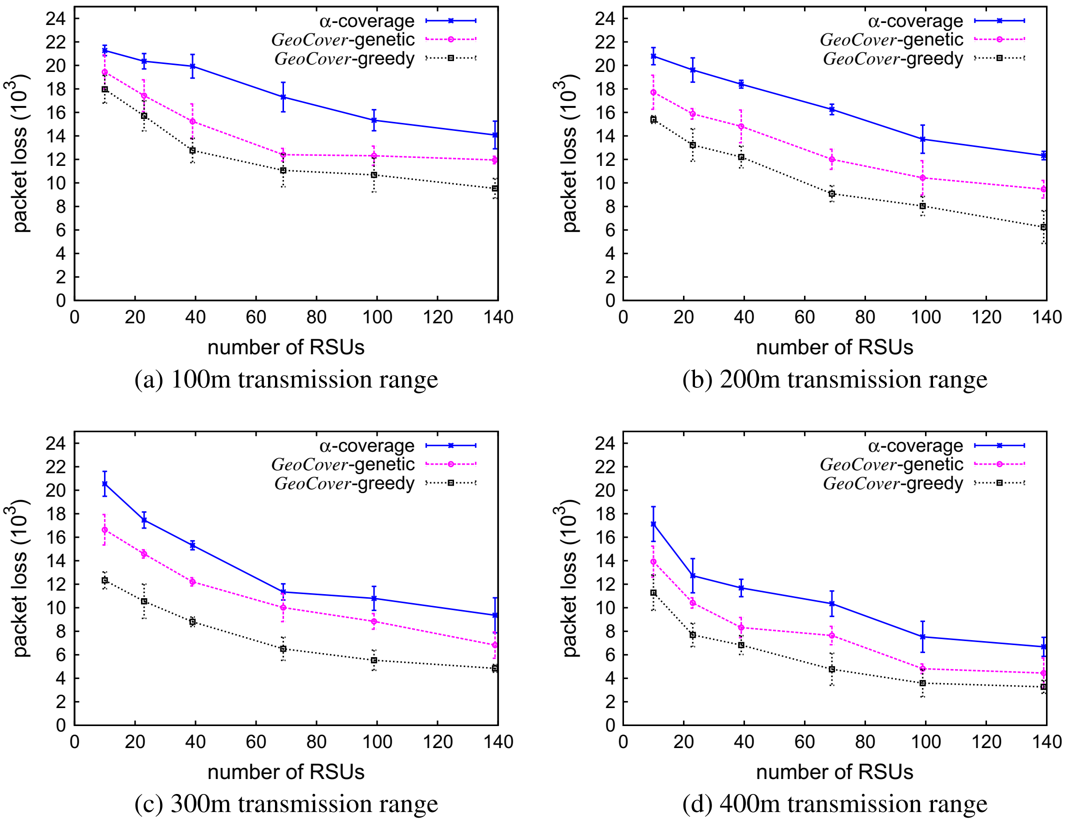


Fig. 10. Simulation results with AODV in terms of packet delivery rate.



(a) 100m transmission range (b) 200m transmission range

(c) 300m transmission range (d) 400m transmission range

Fig. 11. Simulation results with AODV in terms of packet loss.

denote the optimal solution of the QSC problem, which is the number of RSUs. Let U_i denote the new *sections* added at i-th iteration, and $U'_i = \sum_{j=1}^{i} U_i$and S represent the union of optimal covered *sections*. It is obvious that $U_0 = 0$ and U'_i are the *sections* already covered by the algorithm at iteration i.

Proof. At stage i, the uncovered *sections* are $S - U'_i$, which could be covered by optimal OP RSUs. Therefore, on average, any RSU in the optimal solution is able to cover at least $(S - U'_i)/OP$ uncovered *sections*. We can infer the following equations:

$$S - U'_i \leqslant U'_{i+1} - U'_i \tag{7}$$

In this way,

$$\begin{aligned} S - U'_{i+1} &\leqslant (S - U'_i)(1 - 1/OP) \leqslant S(1 - 1/OP)^{i+1} \\ &\leqslant S \cdot e^{-(i+1)/OP} \end{aligned} \tag{8}$$

At each iteration, we will add a new RSU to the final solution. When the algorithm reaches the stage of the optimal solution, we let the K represent the number of RSUs in the optimal solution of our algorithm.

$$S - U'_{K+1} < OP \leqslant S - U'_K \tag{9}$$

Therefore,

$$\begin{aligned} K &\leqslant i + OP, OP \leqslant S \cdot e^{-K/OP} \leqslant OP(1 + \ln S/OP) \\ &\leqslant OP(1 + \ln r) \qquad \square \end{aligned} \tag{10}$$

6. Performance evaluation

In this section, we first present the methodology and experimental setup for the evaluation of the proposed *GeoCover* algorithm, and then we give the experimental results and corresponding analysis.

6.1. Methodology and experimental setup

We carried out our simulations on the Network Simulator (NS2) [39] and the Simulation of Urban Mobility (SUMO) [40]. SUMO is responsible for generating the mobility trace files in a road network and NS2 exploits the mobility model files to simulate the V2I communication with a given protocol stack. To emulate the real scenario, we captured a 2300 m × 2100 m real road network of Ottawa's downtown area. It consists of a total of 377 intersections and 776 road segments. Fig. 7 shows the outlook of our simulation environment on Goggle Maps. The geometry data of map were obtained from OpenStreetMap, in which road segments are represented as 2-D polygons with various shapes.

Fig. 8 is a part of the map shown in SUMO with the lanes and traffic lights. Vehicles send packets to the RSUs through a wireless channel. And then RSUs forward packets to the base station. After all the packets are gathered in the center server, the feedback will be sent to vehicles through the RSUs. We use the car-following model [41] to imitate the real movement of vehicles. The car-following

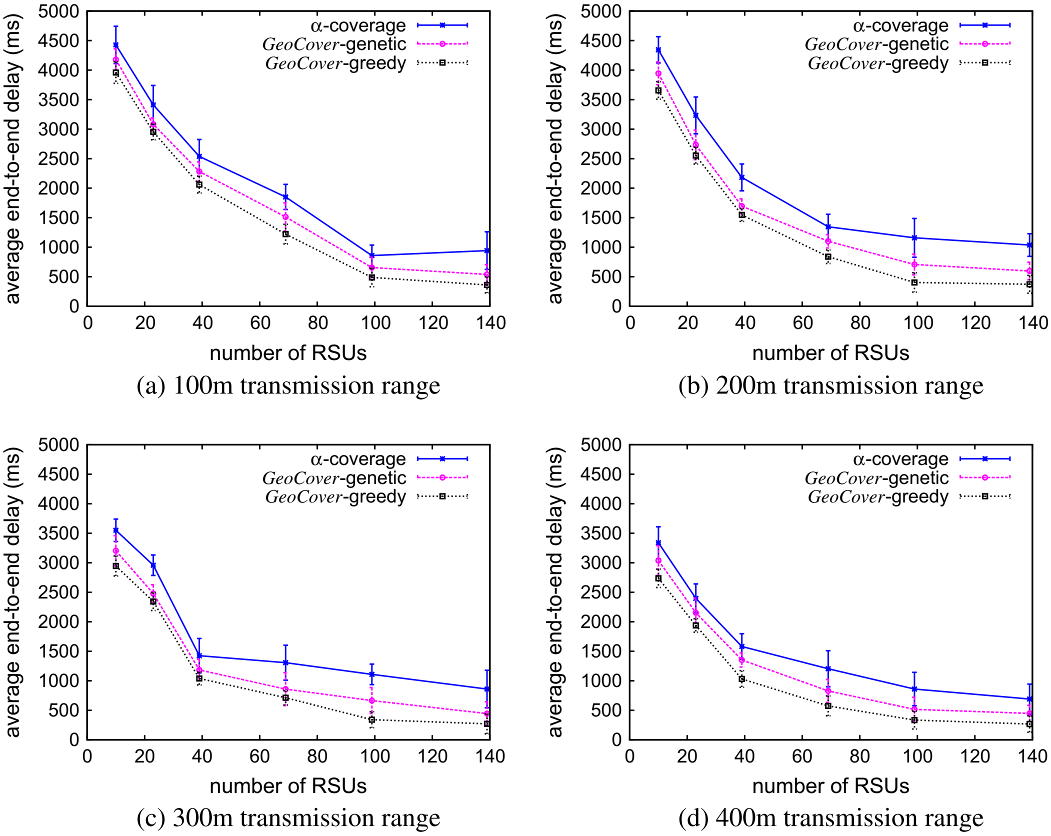


Fig. 12. Simulation results with AODV in terms of average end-to-end delay.

model is implemented in SUMO to describe the acceleration of a vehicle using the properties of the car in front of it. The speed of each vehicle is limited by the real speed restriction of the corresponding road segment.

Table 2 shows the detailed simulation parameters. Due to the high mobility of vehicles in VANETs, we used IEEE802.11p at the physical layer and MAC layer and mobile IP at the network layer. We analyzed 50 mobility files, and each file records the running of 100 vehicles in 500 s. We then used NS2 to simulate the V2I communications between vehicles and RSUs based on 10 different mobility scenarios. To compare the communication quality based on different protocols, we simulated our coverage algorithms and the baseline algorithm with AODV and GPSR protocols.

6.2. Baseline algorithm

In order to evaluate our *GeoCover* algorithms, α-coverage algorithm [42] is introduced as the baseline algorithm. The main purpose of α-coverage is to guarantee that there is at least one point of contact between vehicles and the RSUs when vehicles move within α meter. In our algorithms, the number of RSUs is limited by a budget.

Acting as a sparse deployment idea, α-coverage also focuses on the same metric that suggests using fewer RSUs to provide better coverage performance. It is the first reason why we chose α-coverage as the baseline algorithm. Furthermore, the RSUs in α-coverage are deployed in the center of junctions and the key point we want to prove is that the placement of RSUs beside roads may be more efficient than at intersections. In addition, although the geometry-based coverage strategy is based on spatial coverage, the key point of our idea is to maximize the contact time between an RSU and vehicles which are similar to α-coverage. Therefore, we think that the comparison with α-coverage is better to justify the effectiveness of *GeoCover* algorithms.

6.3. Analysis for hotspot discovery

To analyze the performance of the *hotspot* discovery algorithm, we compare the number, average size and mean square deviation of sizes of *hotspots* with the increase of threshold α. The result is shown in Fig. 9.

Through Fig. 9, we find that since the high value of α impedes the creation of *hotspots*, the number of *hotspots* increases rapidly as α grows from 10 to 30. The total number, however, declines sharply when α increases from 30 to 70. This is because more areas fail to reach the threshold as α becomes high enough. When α reaches 65, there is no *hotspot* that can be discovered.

Unlike the number of *hotspots*, the average size of *hotspots* and the standard deviation of sizes continually

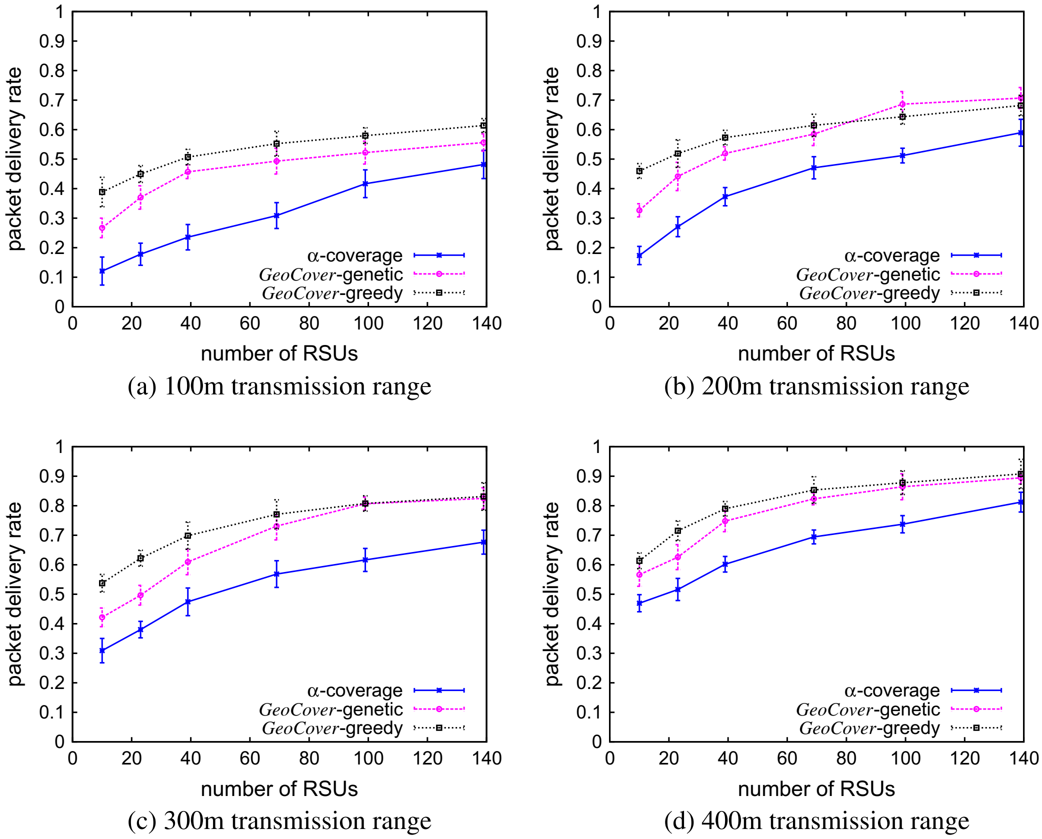


Fig. 13. Simulation results with GPSR in terms of packet delivery rate.

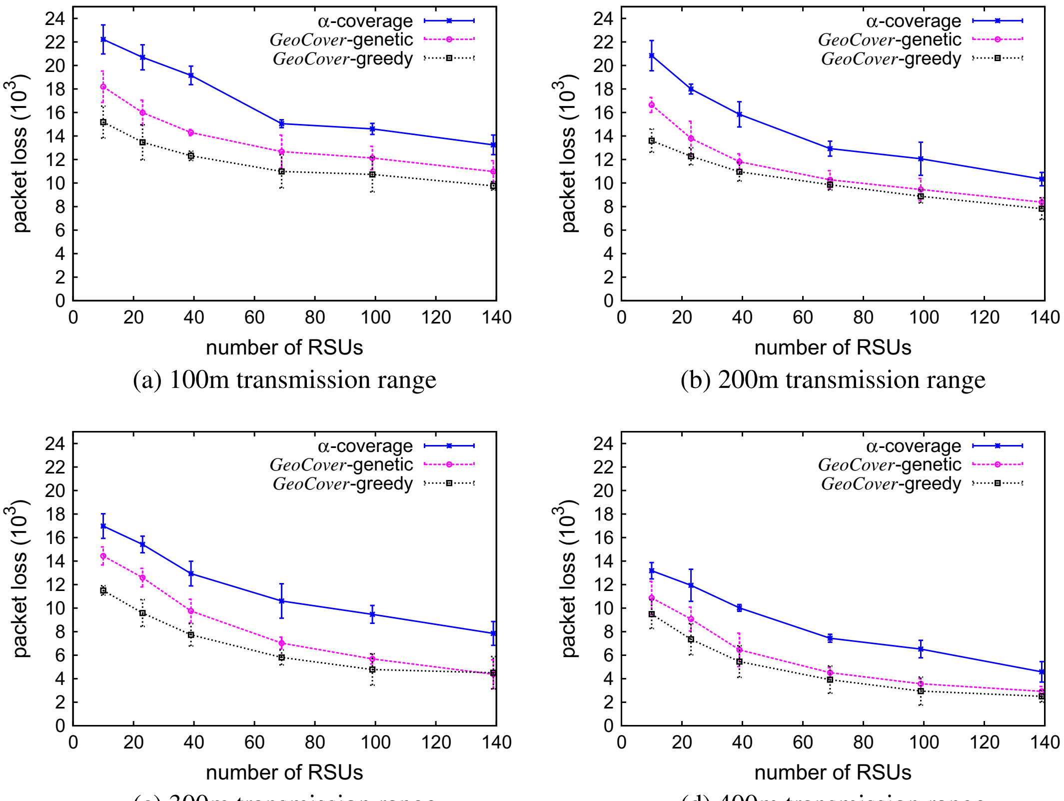
decrease. The reason for this is that the sizes of *hotspots* reflect the density of α. When α value is small, many low-density regions are included in a large cluster so that the deviation of different *hotspots* is also large. However, as the threshold increases the *hotspots* become purer and smaller, as does the deviation in sizes of *hotspots*. Empirically, we prefer the *hotspot* that is similar in size to an RSU's signal coverage and deviation in sizes that are small enough. Therefore, in the following experiment, we choose to use 20 *hotspots* as α equals 25.

6.4. Analysis for sparse coverage simulation

Since BSC and QSC are symmetrical we can resolve both problem using the same algorithm. As shown in Table 3, if we use the number of RSUs as the input for BSC (e.x. 20), there is a corresponding coverage value (e.x. 13567). Vice verse, if we take the coverage value as the input for QSC (e.x. 13567) we can have the corresponding number of RSUs (e.x. 20) as result. Therefore, in the follow simulation result, we use only the result of BSC for the illustration.

To analyze the performance of *GeoCover* algorithms, we compare the two types of *GeoCover* algorithms with α-coverage algorithm. We select packet delivery rate, packet loss and average end-to-end delay as metrics to measure the quality of communication under three coverage algorithms: *GeoCover*-genetic, *GeoCover*-greedy, and α-coverage. The packet delivery rate is a metric calculated by dividing the number of packets received by the target RSUs through the number of packets originating from vehicles. Packet loss refers to the number of packets dropped in transmissions, which is used to measure the ability of a network to relay. Average end-to-end delay refers to the time taken for a packet to be transmitted across a network from source node to destination node. To research the effect of our geographic RSU deployment, the simulation is designed to compare two routing protocols: AODV and GPSR. AODV is an on-demand routing protocol for ad hoc networks using the shortest path algorithm, while GPSR is a responsive routing protocol using the proposed location information of vehicles. By comparing the two routing protocols, we can find that our *GeoCover* suits different routing schemes and the quality of V2I communication mainly depends on the RSU deployment rather than on geographic routing.

Fig. 10 represents the packet delivery rate results of three coverage algorithms simulated on the AODV routing protocol. As shown in the figures, the packet delivery rate grows regardless of the transmission range and the type of coverage. A common phenomenon is that even though the distance differs between three coverage algorithms as the transmission range rises, their packet delivery rates approach the same trend as the number of RSUs increases. The reason for this is that when there is a small number of



(a) 100m transmission range (b) 200m transmission range

(c) 300m transmission range (d) 400m transmission range

Fig. 14. Simulation results with GPSR in terms of packet loss.

RSUs, none of the three coverage algorithms provide enough opportunity for the vehicles to enter the transmission range of the RSUs. In this situation, packet loss is mainly due to a failure to find the routing. However, as the number of RSUs increases, the coverage area and density improve. In this way, the communication quality peaks and the corresponding packet delivery rate reaches the top level.

Fig. 11 shows the packet loss results of three coverage algorithms running on the AODV routing protocol. From the figures, we find that all three coverage algorithms tend to be stable as the number of RSUs reaches 100. At this time, the majority of lost packets are caused by an overflow of queues in each RSU. Therefore, the increase of RSUs will no longer influence the packet loss. However, the two types of *GeoCover* algorithms always perform better than α-coverage even when the number of RSUs is as low as 10. This is because *GeoCover* is based on covering *hotspots*, where the most vehicles accumulate, while the α-coverage is based on spatial coverage of roads. Thus, α-coverage only considers the intersections of road networks to provide length-bounded coverage, while *GeoCover* chooses the most critical regions to be covered. Therefore, when both types of coverage fail to completely cover the network, the selection of covered regions by *GeoCover* outperforms the α-coverage. The superiority of the use of *hotspot* is also reflected in the trend of the packet loss. Because *GeoCover* always picks the most valuable regions to cover, the communication quality is better than α-coverage as the number of RSUs and the transmission range changes.

Fig. 12 shows the average end-to-end delay of three coverage algorithms based on the AODV routing protocol. The performance of all three coverage algorithms go worse as the number of RSUs increases. The larger the transmission range is, the smaller the end-to-end delay becomes. Even though both types of *GeoCover* algorithms have similar trends in Fig. 12, there is still some difference between the *GeoCover*-genetic algorithm and the *GeoCover*-greedy algorithm. When the number of deployed RSUs is very low, both *GeoCover* algorithms can only cover the center part of *hotspots*. In this way, the delay from source nodes to destination nodes is high due to the loss of hops in some regions without coverage. As the transmission range increases and the number of RSUs rises, the two *GeoCover* algorithms can completely cover the *hotspots* and the marginal regions, so that the effect of coverage reaches the peak at the same time for both algorithms. However, since the genetic algorithm heuristically searches for the best deployed positions by crossover and mutation, this randomness could result in a locally optimal solution and instability. Thus, the *GeoCover*-greedy algorithm provides a good performance guarantee for the RSU deployment, which results in a better prediction of the coverage quality than the *GeoCover*-genetic algorithm within a 95% confidence interval.

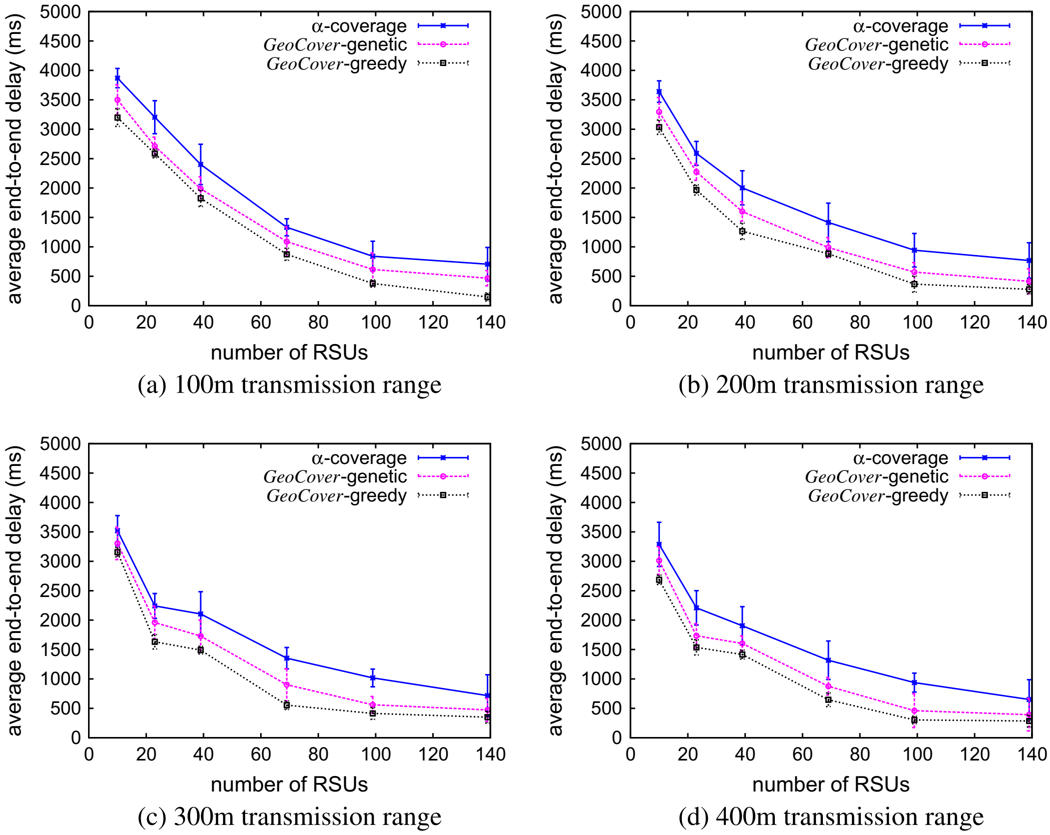


Fig. 15. Simulation results with GPSR in terms of average end-to-end delay.

Figs. 13–15 evaluate the packet delivery rate, packet loss and average end-to-end delay of the three coverage algorithms with the GPSR routing protocol. The simulation results with GPSR are similar to the results with the AODV protocol. As the RSUs and transmission range increase, the packet delivery rates of all three coverage algorithms grow. Meanwhile, the packet loss and average end-to-end delay decrease. The more RSUs are deployed, the more movements of vehicles are able to be covered. Therefore, the packet loss appears to be smaller and more stable. Since the higher coverage ratio means more choice in the information transmission and less hops in the packet forwarding, the end-to-end delay declines as the RSUs increase. The scalability of the three coverage algorithms under GPSR is also the same as for the simulation under AODV. Compared with *GeoCover* algorithms, the α-coverage algorithm is more sensitive to the increase of RSUs in terms of packet delivery rate and packet loss. Generally, the *GeoCover*-greedy algorithm performs better than the *GeoCover*-genetic algorithm. In some cases, the *GeoCover*-genetic algorithm can provide higher packet delivery rates than the *GeoCover*-greedy algorithm as shown in Fig. 13(b). This is due to the randomness of heuristic search in genetic algorithm.

By comparing Figs. 13–15 with Figs. 10–12, the difference of GPSR and AODV in our simulations is also obvious. The packet delivery rate in the GPSR simulation result is better than for the AODV simulation when the other conditions are the same. This difference of communication quality is caused by the difference of two routing schemes; in wireless communication, the packet loss occurs due to the end of TTL (Time to Live), broadcast storms or collisions. If a routing protocol takes more time to find the source–destination path in the routing phase, the life-time of packets will be shortened and the opportunity to drop the packets will be enlarged. In our scenarios, AODV takes much time to maintain the positions of mobile nodes in the routing table, which results in a worse performance than the position-based routing protocol GPSR.

More specifically, once AODV notices the failure of the communication link, this protocol will keep the packets in the buffer queue and then wait for the availability of the route. When the connectivity is stable, this technique can increase the packet delivery rate in some cases. However, in a vehicular network where vehicles move with high speed and the topology is continuously changing, the connectivity will become unstable due to the unavailable direct or indirect re-delivery. In this situation, the kept packets in the buffer queue will wait too long to be delivered and such an average end-to-end delay will aggregate the fracture of the routing connection. However, in the event of a link retransmission failure, GPSR applies a different method by removing the routing entry of the broken link before the packets in the buffer are queued [43]. GPSR then uses the greedy algorithm to find the next hop to forward packets by finding the geographically closest node to the sink node. The technique used in GPSR is more suitable for vehicular networks with high mobility and unstable topology.

To sum up, the *GeoCover* algorithms outperform the α-coverage algorithm in our simulation. In a comparison of both types of *GeoCover* algorithms, the *GeoCover*-greedy algorithm is more stable and scalable than the *GeoCover*-genetic algorithm. The packet delivery rate in the *GeoCover*-greedy algorithm is also higher than the *GeoCover*-genetic algorithm, even though the two algorithms perform at the same level of end-to-end delay when the transmission range and number of RSUs are large enough. By comparing the AODV and GPSR, we find the GPSR more suitable for vehicular networks with high mobility. However, the difference between AODV and GPSR is not very obvious in our simulation, which means our coverage algorithms are suitable for different routing schemes to provide a convincing quality of communication.

7. Conclusion

In this paper, we focus on solving three coverage problems in urban vehicular networks: road geometry, vehicular traffic distribution and resource constraints. We propose a geometry-based sparse coverage protocol *GeoCover* and two types of coverage algorithms to handle the problems. To design a practical RSU deployment based on road geometry, we propose a buffering operation in our coverage scheme. To capture the pattern of vehicular traffic distribution, we put forward the *hotspot* discovery approach to discover the most valuable regions in a road network. To solve the resource constraint, we formulate two variants of our sparse coverage algorithm to suit different objectives: budget and quality. By reducing the BSC and QSC to the MCP and SCT problems respectively, we design two approximation algorithms, the genetic algorithm and the greedy algorithm, to maximize the quality of coverage while keeping the cost under budget. Our simulations in NS2 prove that the proposed algorithms perform with better scalability and stability compared with α-coverage.

In our future work, we plan to extend our research to the connectivity and scheduling issues in VANETs. Connectivity measures how reliable the data dissemination of time-critical information will be in VANETs. It is an important metric in ad hoc networks. Scheduling means the control of RSU status in a vehicular network. Since the RSUs can be either active or reactive, the redundancy of energy and coverage can be saved effectively.

References

[1] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, T. Weil, Vehicular networking: a survey and tutorial on requirements, architectures, challenges, standards and solutions, Commun. Surv. Tutorials, IEEE 13 (4) (2011) 584–616.

[2] Sherali Zeadally, Ray Hunt, Yuh-Shyan Chen, Angela Irwin, Aamir Hassan, Vehicular ad hoc networks (VANETs): status, results, and challenges, Telecommun. Syst. 50 (4) (2012) 217–241.

[3] Monika Khatri, An insight overview of issues and challenges in vehicular adhoc network, J. Global Res. Comput. Sci. 2 (12) (2012) 47–50.

[4] H.M. Ammari, R. Mulligan, Coverage in wireless sensor networks: a survey,network protocols and algorithms, Netw. Protocols Algorithms 2 (2) (2010) 27–53.

[5] GaoJun Fan, ShiYao Jin, Coverage problem in wireless sensor network: a survey, J. Netw. 5 (9) (2010) 1033–1040.

[6] Huang Cheng, Xin Fei, Azzedine Boukerche, Abdelhamid Mammeri, Mohammed Almulla, A geometry-based coverage strategy over urban VANETs, in: Proceedings of the 10th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks, PE-WASUN '13, AMC, New York, NY, USA, 2013, pp. 121–128.

[7] Helal Chowdhury, Janne J. Lehtomäki, Juha-Pekka Mäkelä, Sastri Kota, Data downloading on the sparse coverage-based wireless networks, J. Electr. Comput. Eng. 2010 (2010).

[8] [8] N. Bellomo, M. Delitala, On the mathematical theory of vehicular traffic flow i: Fluid dynamic and kinetic modelling, Math. Models Methods Appl. Sci. (2002) 1801–1843.

[9] Xin Fei, Azzedine Boukerche, Regina Borges de Araujo, Irregular sensing range detection model for coverage based protocols in wireless sensor networks, in: GLOBECOM, IEEE, 2009, pp. 1–6.

[10] Azzedine Boukerche, Xin Fei, A coverage-preserving scheme for wireless sensor network with irregular sensing range, Ad Hoc Netw. 5 (8) (2007) 1303–1316.

[11] Azzedine Boukerche, Xin Fei, Regina Borges de Araujo, An optimal coverage-preserving scheme for wireless sensor networks based on local information exchange, Comput. Commun. 30 (14–15) (2007) 2708–2720.

[12] Brij Bihari Dubey, Effect of position of fixed infrastructure on data dissemination in VANETs, Int. J. Res. Rev. Comput. Sci. (IJRRCS) 2 (2) (2011).

[13] Mohamed Kafsi, Panos Papadimitratos, Olivier Dousse, Tansu Alpcan, Jean-Pierre Hubaux, VANET connectivity analysis, in: IEEE Workshop on Automotive Networking and Applications, IEEE Computer Society, 2008.

[14] Kebin Liu, Minglu Li, Yunhao Liu, Xiang-Yang Li, Minglu Li, Huadong Ma, Exploring the hidden connectivity in urban vehicular networks, in: 2010 18th IEEE International Conference on Network Protocols (ICNP), 2010, pp. 243–252.

[15] Junghoon Lee, Cheol Min Kim, A roadside unit placement scheme for vehicular telematics networks, in: Proceedings of the 2010 International Conference on Advances in Computer Science and Information Technology, AST/UCMA/ISA/ACN'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 196–202.

[16] Paolo Crucitti, Vito Latora, Sergio Porta, Centrality measures in spatial networks of urban streets, Phys. Rev. E 73 (3) (2006) 036125.

[17] Yannick Do, S. Buchegger, T. Alpcan, J.P. Hubaux, Centrality analysis in vehicular ad-hoc networks, EPFL/T-Labs, Tech. Rep., 2008.

[18] A. Kchiche, F. Kamoun, Access-points deployment for vehicular networks based on group centrality, in: 2009 3rd International Conference on New Technologies, Mobility and Security (NTMS), 2009, pp. 1–6.

[19] A. Kchiche, F. Kamoun, Centrality-based access-points deployment for vehicular networks, in: 2010 IEEE 17th International Conference on Telecommunications (ICT), 2010, pp. 700–706.

[20] A. Abdrabou, Weihua Zhuang, Probabilistic delay control and road side unit placement for vehicular ad hoc networks with disrupted connectivity, IEEE J. Selected Areas Commun. 29 (1) (2011) 129–139.

[21] Hao Wu, R.M. Fujimoto, G.F. Riley, M. Hunter, Spatial propagation of information in vehicular networks, IEEE Trans. Veh. Technol. 58 (1) (2009) 420–431.

[22] Pan Li, XiaoXia Huang, Yuguang Fang, Phone Lin, Optimal placement of gateways in vehicular networks, IEEE Trans. Veh. Technol. 56 (6) (2007) 3421–3430.

[23] O. Trullols, M. Fiore, C. Casetti, C.F. Chiasserini, J.M. Barcelo Ordinas, Planning roadside infrastructure for information dissemination in intelligent transportation systems, Comput. Commun. 33 (4) (2010) 432–442.

[24] Christian Lochert, Björn Scheuermann, Christian Wewetzer, Andreas Luebke, Martin Mauve, Data aggregation and roadside unit placement for a VANET traffic information system, in: Proceedings of the Fifth ACM International Workshop on VehiculAr Inter-NETworking, VANET '08, ACM, New York, NY, USA, 2008, pp. 58–65.

[25] Yipin Sun, Xiaodong Lin, Rongxing Lu, Xuemin Shen, Jinshu Su, Roadside units deployment for efficient short-time certificate updating in VANETs, in: 2010 IEEE International Conference on Communications (ICC), 2010, pp. 1–5.

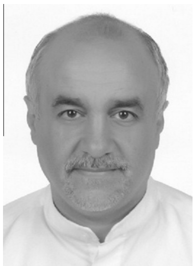
[26] Ramneek Kaur, Ravreet Kaur, Scalable TDB based RSUs deployment in VANETs, Int. J. Innovation Appl. Stud. 3 (4) (2013) 1025–1032.

[27] Wendong Wang Yongping Xiong, Jian Ma, Dengbiao Tu, Roadgate: mobility-centric roadside units deployment for vehicular networks, Int. J. Distrib. Sens. Netw. 2013 (2013) 47–50.
[28] Yanmin Zhu, Youchen Bao, Bo Li, On maximizing delay-constrained coverage of urban vehicular networks, IEEE J. Selected Areas Commun. 30 (4) (2012) 804–817.
[29] M. Fiore, J.M. Barcelo-Ordinas, Cooperative download in urban vehicular networks, in: IEEE 6th International Conference on Mobile Adhoc and Sensor Systems, 2009. MASS '09, 2009, pp. 20–29.
[30] Samir Khuller, Anna Moss, Joseph (Seffi) Naor, The budgeted maximum coverage problem, Inf. Process. Lett. 70 (1) (1999) 39–45.
[31] Uriel Feige, A threshold of ln n for approximating set cover, J. ACM 45 (4) (1998) 634–652.
[32] Dorit S. Hochbaum (Ed.), Approximation Algorithms for NP-hard Problems, PWS Publishing Co., Boston, MA, USA, 1997.
[33] M. Kim, D. Kotz, S. Kim, Extracting a mobility model from real user traces, in: Proceedings of the 25th IEEE International Conference on Computer Communications, INFOCOM 2006, 2006, pp. 1–13.
[34] Martin Ester, Hans peter Kriegel, S. Jg, Xiaowei Xu, A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, AAAI Press, 1996. pp. 226–231.
[35] Yukon Department of Highways and Public Works. National Road Network Road Segments – Yukon Territory, Canada, September 2008.
[36] Kohei Okamoto, Kei-ichi Okunuki, Toshibumi Takai, Sketch map analysis using GIS buffer operation, in: Spatial Cognition IV. Reasoning, Action, Interaction, Springer, 2005, pp. 227–244.
[37] M. Haklay, P. Weber, Openstreetmap: user-generated street maps, IEEE Pervasive Comput. 7 (4) (2008) 12–18.
[38] Dorit S. Hochbaum, Approximation algorithms for NP-hard problems, in: Approximating Covering and Packing Problems: Set Cover, Vertex Cover, Independent Set, and Related Problems, PWS Publishing Co., Boston, MA, USA, 1997, pp. 94–143.
[39] S. Mccanne, S. Floyd, K. Fall, ns2 (Network Simulator 2). <http://www-nrg.ee.lbl.gov/ns/>.
[40] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, Laura Bieker, Recent development and applications of SUMO – simulation of urban mobility, Int. J. Adv. Syst. Measur. 5 (3&4) (2012) 128–138.
[41] Stefan Krauß, Microscopic modeling of traffic flow: investigation of collision free vehicle dynamics, PhD Thesis, Universität zu Köln, 1998.
[42] Zizhan Zheng, P. Sinha, S. Kumar, Sparse WiFi deployment for vehicular internet access with bounded interconnection gap, IEEE/ACM Trans. Netw. 20 (3) (2012) 956–969.
[43] Brad Karp, Hsiang-Tsung Kung, Gpsr: Greedy perimeter stateless routing for wireless networks, in: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, ACM, 2000, pp. 243–254.

Huang Cheng is currently a master student with the Department of Electrical Engineering and Computer Science, University of Ottawa. His current research interests are in coverage problems over vehicular networks.

Xin Fei is currently a Post-Doctoral with the Department of Electrical Engineering and Computer Science, University of Ottawa. His current research interests are in coverage optimization and hand-over problem of wireless sensor network and vehicular ad hoc network.

Azzedine Boukerche is a full professor and holds a Canada Research Chair Tier 1 position. His interests are vehicular networks, sensor networks and wireless and mobile computing.



Mohammed Almulla is a Visiting professor at the University of Ottawa. He finished all his degrees B.Sc. (1986), M.Sc. (1990), and Ph.D. (1995) from McGill University in Montreal – Canada.His main interests are distributed and mobile computing, Web-services and video streaming on vehicular networks.