

A Resource-Constrained Coverage Protocol Over Urban VANETs

by

Huang Cheng

Supervisor: Prof. Azzedine Boukerche

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the M.A.Sc. degree in
Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

© Huang Cheng, Ottawa, Canada, 2014

Abstract

Vehicular ad hoc networks have emerged as a promising area of research in academic fields. However, it presents a challenge to design a realistic coverage protocol for vehicular networks, due to the service requirements, assorted mobility patterns, resource constraints and irregularity of the service area. It is also a challenge to meet a high quality of coverage with a tight deployment budget. In order to resolve these problems, this thesis proposes a resource-constrained coverage protocol with statistical analysis, which aims to consider the application demands, mobility patterns of vehicles, resource limitations, and geometrical attributes of road networks.

We study two types of resource-constrained coverage: the continuous coverage model and the sparse coverage model. We then reduce each model to a Knapsack Constrained Steiner Tree problem and a Maximum Coverage Problem, respectively. Since the two reduced problems are NP-hard, we resolve each of them with the Lagrangian Decomposition approach and greedy algorithm. By taking the dimensions of road segments into account, our coverage protocol provides a buffering operation scheme to suit different types of road topology. By discovering *hotspots* from the historical trace files, the proposed protocol is able to depict the mobility patterns and to discover the most valuable regions of a road system. To solve the problems of resource constraint, we provide two variants of continuous coverage and sparse coverage by taking budget constraint and quality constraint into consideration. The comparison with other mature algorithms verifies that our coverage protocol is reliable, and suitable for urban vehicular networks.

Acknowledgements

Foremost, I would like to express my sincere gratitude to my supervisor, Prof. Azzedine Boukerche for his inspiring guidance, extraordinary patience and technical assistance throughout my entire masters degree. His insightful guidance and critical comments have played an essential role in my research work and the completion of this thesis. His financial support and sense of humour have not only made my life easier, but have also increased my motivation to go deeper in my academic research. I shall treasure for my whole life this experience as a student with such a distinguished supervisor.

My earnest thanks must also go to Dr. Mohammed Almulla and Dr. Xin Fei, two great mentors and friends over the past two years. Their rigorous scholarship, constructive advices and truthful encouragement were really helpful for my learning and life experience. This thesis could not have been finished without the help and support from them.

Special thanks go to my colleagues at PARADISE Research Laboratory for all the fun times and collaboration. The thorough discussions and great ambience in the lab facilitated my work daily.

Last but not least, my deepest gratitude to my parents and my girlfriend, who have stood by me throughout my degree. Their unconditional love, endless support and continuous encouragement serve as the most powerful drive for my achievements.

Publications Related to Thesis

Conference Paper:

- Huang Cheng, Xin Fei, Azzedine Boukerche, Abdelhamid Mammeri, and 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, pages 121128, New York, NY, USA, 2013. ACM.
- Huang Cheng, Xin Fei, Mohammed Almulla, and Azzedine Boukerche. A Knapsack Constrained Steiner Tree Model for Continuous Coverage over Urban VANETs. Proceedings of IEEE International Conference on Communications, *IEEE ICC'14*, June 2014. (accepted)
- Huang Cheng, Xin Fei, Azzedine Boukerche: A Genetic Algorithm-based Sparse Coverage Over Urban VANETs. NIDISC 2014. (accepted)

Journal Paper:

- Huang Cheng, Xin Fei, Azzedine Boukerche, and Mohammed Almulla. *GeoCover*: An Efficient Geometry-Based Sparse Coverage Protocol Over Urban VANETs. A Special Issue on Modeling and Performance Evaluation of Wireless Ad-Hoc Networks, 2014. (submitted)

Glossaries

VANETs	Vehicular Ad Hoc Networks	ITS	Intelligent Transportation System
MANETs	Mobile Ad Hoc Networks	WSNs	Wireless Sensor Networks
WLAN	Wireless Local Area Network	APs	Access Points
DSRC	Dedicated Short Range Communications	WAVE	Wireless Access in Vehicular Environments
OBU	On-Board Unit	RSU	Road-Side Unit
V2V	Vehicle-to-Vehicle	V2I	Vehicle-to-Infrastructure
BCC	Budgeted Continuous Coverage	QCC	Qualified Continuous Coverage
BSC	Budgeted Sparse Coverage	QSC	Qualified Sparse Coverage
KCST	Knapsack Constrained Steiner Tree	MCP	Maximum Coverage Problem
SMT	Steiner Minimum Tree	DBSCAN	Density-based Spatial Clustering of Applications with Noise
GA	Genetic Algorithm	KP	Knapsack Problem
LR	Lagrangian relaxation	LD	Lagrangian Decomposition
MST	Minimum Spanning Tree	DP	Dynamic Programming
ILP	Integer Linear Programming	KMB	Kou-Markowsky-Berman
NS2	Network Simulator	SUMO	Simulation of Urban Mobility
MCC	Maximum Continuous Coverage	SCP	Set Cover Problem

Contents

1	Introduction	1
1.1	Background	1
1.2	Problem Statement	3
1.3	Contribution	4
1.4	Thesis Organisation	5
2	Related Work	7
2.1	Spatial Coverage	8
2.1.1	Coverage on Special Positions	8
2.1.2	Coverage on Centrality	9
2.1.3	Coverage on Spatial Measurement	10
2.1.4	Mathematic Models in Spatial Coverage	15
2.1.5	Summary of Spatial Coverage	16
2.2	Temporal Coverage	17
2.2.1	Coverage on Delay	17
2.2.2	Coverage on Contacts	19
2.2.3	Coverage on Extra Time Overhead	20
2.2.4	Mathematic Models in Temporal Coverage	21
2.2.5	Summary of Temporal Coverage	24
2.3	Spatiotemporal Coverage	25
2.3.1	Coverage on Traffic Flow Theory	25
2.3.2	Coverage on Data Mining	27
2.3.3	Mathematic Models in Spatiotemporal Coverage	30
2.3.4	Summary of Spatiotemporal Coverage	31
2.4	Summary	31

3	System Model and Definitions	32
3.1	System Model	32
3.2	Definitions	34
3.3	Geometry-Based Deployment Model	37
3.4	Summary	39
4	Resource-Constrained Coverage Protocol	40
4.1	Continuous Coverage Model	40
4.1.1	Budgeted Continuous Coverage	42
4.1.2	Qualified Continuous Coverage	44
4.1.3	Lagrangian Decomposition Algorithm (BCC-LD)	45
4.2	Sparse Coverage Model	47
4.2.1	Budgeted Sparse Coverage	49
4.2.2	Qualified Sparse Coverage	50
4.2.3	Genetic Algorithm (BSC-genetic)	51
4.2.4	Greedy Algorithm (BSC-greedy)	54
4.3	Budget Estimation	57
4.4	Summary	59
5	Performance Evaluation	60
5.1	Methodology and Experimental Setup	60
5.2	Baseline Algorithms	62
5.3	Analysis for <i>Hotspot</i> Discovery	63
5.4	Analysis for Continuous Coverage Simulation	64
5.5	Analysis for Sparse Coverage Simulation	68
5.6	Summary	73
6	Conclusion and Future Work	74
6.1	Conclusion	74
6.2	Future Work	75

List of Tables

2.1	Coverage strategies in spatial coverage	11
2.2	Coverage algorithms in spatial coverage	12
2.3	Coverage strategies in temporal coverage	22
2.4	Coverage algorithms in temporal coverage	23
2.5	Coverage strategies in spatiotemporal coverage	27
2.6	Coverage algorithms in spatiotemporal coverage	28
5.1	Simulation parameters	62
5.2	Lagrangian Decomposition results	64

List of Figures

1.1	Coverage over VANETs	2
2.1	Classifications of coverage over VANETs	7
3.1	Process of resource-constrained coverage protocol	33
3.2	Assignment of <i>coverage value</i>	34
3.3	Illustration of buffering operation	37
3.4	Buffering operation on straight roads	38
3.5	Buffering operation on curved roads	38
4.1	Formation of continuous coverage model	41
4.2	Formation of sparse coverage model	48
4.3	Budget estimation model	58
5.1	Downtown map of Ottawa	61
5.2	Simulation scenario	61
5.3	<i>Hotspot</i> discovery analysis	63
5.4	Continuous coverage in terms of packet delivery rate	65
5.5	Continuous coverage in terms of packet loss	66
5.6	Continuous coverage in terms of average end-to-end delay	67
5.7	Sparse coverage in terms of packet delivery rate	69
5.8	Sparse coverage in terms of packet loss	70
5.9	Sparse coverage in terms of average end-to-end delay	71

Chapter 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 [23], [60]. In this chapter, we introduce architectures, technical specifications and three types of communication of vehicular networks. The coverage problems caused by application demands, mobility patterns and resource constraints will then be discussed. Finally, we describe our contributions and thesis organization.

1.1 Background

VANETs are a kind of Ad Hoc Network (MANET), in which the nodes are vehicles that follow some particular mobility patterns regulated by road topologies. The static hosts and mobile nodes in MANETs are characterized as road-side units (RSUs) and as on-board units (OBUs) in VANETs. Both types of nodes are Dedicated Short Range Communication (DSRC) devices. To differentiate from general MANETs, some of the unique characteristics of VANETs have been noted, such as the restricted movement of vehicles, the rapidly shifting network topology, and intermittent communications due to the fragmentation of the networks [28]. To suit these characteristics of VANETs, several technical specifications are applied based on the modification of IEEE 802.11 standard. Wireless Access in Vehicular Environments (WAVE), also known as IEEE802.11p, introduce a multichannel wireless standard designed for vehicular communications by combining the IEEE802.11e and IEEE 802.11a standards.

Rapid advances in Wireless Local Area Network (WLAN) technology allows assorted

deployment architectures for vehicular networks in highway, urban and city scenarios. The design of deployment architecture consists in the concepts of network operators, service providers, and a governmental authority. No matter what kind of architecture is applied, the major purpose is to improve the communications between the units in vehicular networks, which are classified into three categories: the vehicle-to-vehicle (V2V) communication which is a pure wireless communication with no infrastructure support; infrastructure-to-infrastructure communication where the RSUs communicate with each other in a wired backbone; and, the vehicle-to-infrastructure (V2I) communication that hybridizes the wireless and wired communications. In this way, a vehicle in the vehicular networks can communicate with service providers either in a single hop or a multi-hop fashion according to the geographic topology. Due to the complicated and hybrid nature of different types of different communications, VANETs raise several challenges with regard to data dissemination, packet routing, security and privacy, etc.

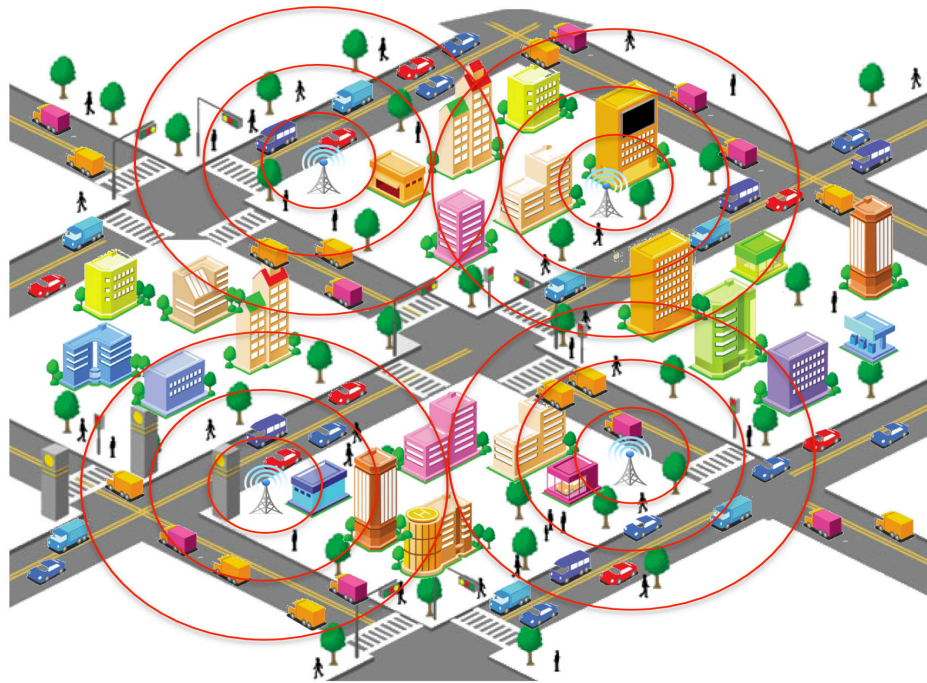


Figure 1.1: Coverage over VANETs

The coverage problems are some of the most important concerns in VANETs as Figure 1.1 shows. In traditional wireless sensor networks (WSNs), coverage is used to evaluate the Quality of Service (QoS) by deploying access points (APs) in feasible regions [49]. The AP deployment problem is modeled as an optimization problem under different

constraints, such as severe resource limitations and assorted hostile environmental conditions [12]. The coverage problem in VANETs is more complex than in WSNs. As a special type of AP, RSUs in VANETs only focus on the street area where the V2V and V2I communications occur. Furthermore, due to obstruction from buildings and complex topology, the feasible region for deployment is not only irregular but also fragmentary.

1.2 Problem Statement

The research is focused in the following four areas:

- Application Demand

Based on different application demands, coverage over VANETs can be classified as either continuous coverage or sparse coverage. For safety-related applications like accident avoidance and incident notification, a reliable continuous coverage for popular paths is required [57]. Even though continuous coverage provides sound coverage quality, it usually calls for a high deployment budget and an operational cost for no coverage holes on selected paths. Sparse coverage is designed for applications like driving-assistance and business promotion in VANETs [7]. This kind of coverage focuses on covering the critical regions with high traffic flow or crowded vehicles. Sparse coverage requires less deployment cost so that it is suitable for cost-efficient services under a tight resource budget. The matter of designing a strategy to handle the trade-off between these two kinds of application demands is a critical problem in VANETs.

- Mobility Pattern

Based on a consideration of vehicular movement, there are three types of coverage for VANETs. Spatial coverage has a tendency to deploy RSUs at locations with distinct spatial attributes, such as intersections and the midpoint of roads. Spatial coverage is easy to operate but 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 decision making so that it is hard to find a certain pattern to depict mobility. Spatiotemporal coverage considers both spatial attributes and temporal characteristics. Existing research on spatiotemporal coverage either exploits the classic traffic flow descriptions [4] or infers hidden mobility patterns through historical information.

Assorted data mining techniques like clustering are common methods used to infer the hidden mobility patterns and to discover data similarities with or without prior knowledge. Therefore, the ability to define a suitable discovery technique to depict the mobility pattern has led to a wide amount of academic concern.

- Resource Constraint and Quality Constraint

In order to meet different optimization objectives, both resource and quality constraints are often mentioned in coverage research. Generally, two variants are derived from coverage over VANETs: budgeted coverage and qualified coverage. 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 that the quality of coverage is of a good level of quality while minimizing the total cost of deployment. Modern coverage techniques should consider both the resource and quality constraints.

- Road Geometry

Most existing algorithms treat vehicle networks as an ideal graph of nodes and straight lines. Such simplifications misrepresent real-world road networks, like the geometrical characteristic of vehicle networks, such as shape, direction and area. In addition, deploying RSUs beside roads rather than at intersections could result in a better quality of communication [6]. Therefore, VANET operators should consider the geometrical characteristics of a road network.

1.3 Contribution

In this thesis, we design a resource-constrained coverage protocol over urban VANETs to solve the above-mentioned four problems. The main contributions of this thesis are as shown in the following:

- To meet different application demands, we propose both continuous coverage and sparse coverage in this thesis. We provide a budget-based method to help network

operators select suitable coverage. Our continuous coverage is modeled as the Knapsack Constrained Steiner Tree (KCST) problem and solved by Lagrangian Decomposition method. As for sparse coverage, we model it as Maximum Coverage Problem (MCP) and solve it with greedy algorithms.

- To discover the mobility pattern of a certain area from historical trace files, we propose two new definitions: *coverage value* and *hotspot*. The *coverage value* refers to the degree it is worth covering a region, while the *hotspot* defines the popular site where most vehicles accumulate. All *hotspots* are discovered by the α -DBSCAN algorithm and measured by the metric *coverage value*.
- To meet budget and quality requirements, continuous coverage is derived as Budgeted Continuous Coverage (BCC) and Qualified Continuous Coverage (QCC), while sparse coverage is derived as Budgeted Sparse Coverage (BSC) and Qualified Sparse Coverage (QSC). We also prove that the budgeted version and qualified version of each type of coverage is equivalent to each other.
- To match the geometrical attributes of road networks, we propose a buffering operation method to define feasible deployment regions. In this way, the deployment locations are distributed in a banded area along the shape of the road network.

1.4 Thesis Organisation

The remainder of this thesis is organized as follows:

- Chapter 2 elaborates upon an overview of coverage problems and corresponding algorithms in VANETs. We classify these coverage strategies into three categories and analyze the objectives, scenarios, mathematic models and algorithms in each category. For each category, we introduce the key mathematic models and algorithms in detail.
- Chapter 3 outlines the system model for the resource-constrained coverage protocol. To discover the hidden popular sites in a road network, the following terms are proposed: two new definitions, *hotspot* and *coverage value*. We also propose a buffering operation scheme to suit the geometric attributes of a road network.
- Chapter 4 introduces two types of resource-constrained coverage models: continuous coverage and sparse coverage. We map the two models to classical mathematic

problems KCST and MCP. Due to the NP-hardness of the reduced problems, we exploit the Lagrangian Decomposition approach and a greedy algorithm to solve them.

- Chapter 5 describes the simulation of our resource-constrained coverage protocol. We choose NS2 and SUMO as the network simulator and traffic simulator, respectively. Two baseline algorithms are introduced to verify the efficiency of the proposed coverage protocol. The simulation is based on a different transmission range, budget limitations, and different routing protocols.
- Chapter 6 concludes the thesis with a summary of the main contributions. Some potential extensions and several other research directions are also proposed a future work.

Chapter 2

Related Work

In this chapter, we review some related algorithms in the literature that handle the coverage problems for vehicular networks. We classify these coverage problems into three types categorized by different design patterns: spatial coverage, temporal coverage and spatiotemporal coverage. By comparing of existing coverage algorithms we analyze the objectives, scenarios, mathematic models, algorithms, etc. We also introduce in detail the most common algorithms used in the three categories. By reviewing and analyzing the corresponding coverage algorithms in literature, we analyze the structures, models and algorithms of each classification. Figure 2.1 shows the classifications of coverage over VANETs.

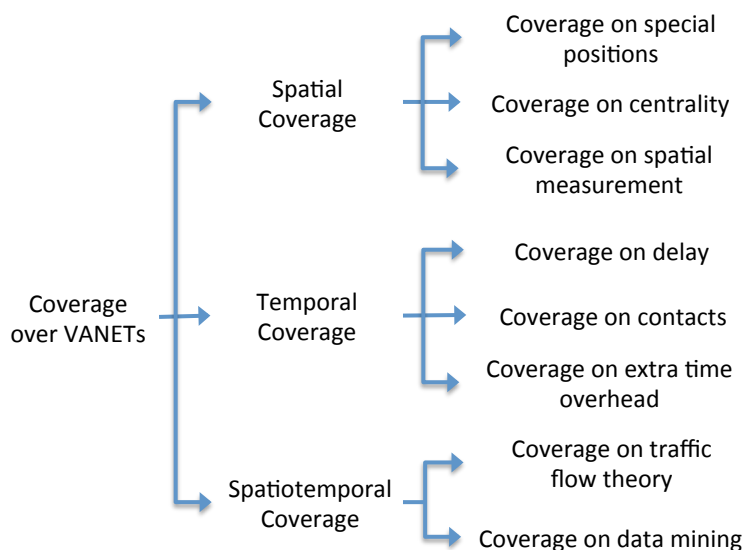


Figure 2.1: Classifications of coverage over VANETs

2.1 Spatial Coverage

Spatial coverage is based on the analysis of spatial attributes of a road system. According to different deploying methods, the spatial coverage mechanism may choose to deploy RSUs in special positions like the middle point of road segments, or it may choose the intersections with high centrality.

2.1.1 Coverage on Special Positions

In the road network, there are some special positions, which serve as natural deployment locations. The intersection and the middle point of a road segment are two popular locations at which to place RSUs.

B. B. Dubey et al. [10] propose an approach that places the fixed RSUs in the centre of intersections. The authors made a calculation that demonstrates how if the RSUs are placed in the centres of intersections, coverage area is increased by nearly 15% more than for those RSUs placed at the corner of the intersections. Therefore, based on different deployment locations of RSUs, the transmission time of data packets in V2I communications will be different. Since the time the packets can be forwarded is within the period that vehicles are in the coverage range of RSUs, the quality of the V2I communication can be measured by the time period in which a vehicle enters the coverage area and evacuates the coverage area. If the sensing model of RSUs is in a form of disk, then the longest contacting time period requires vehicles to move from the border of the disk to the centre. Therefore, placing RSUs in the centre of intersections will help to cover the maximum amount of vehicles for the forwarding of data packets in a single hop, the maximum data dissemination capacity on the road and the maximum communication range.

J. Lee et al. [36] consider intersections as potential deployment locations of RSUs. Their objective is to reduce the disconnection intervals of RSUs and to minimize the overlap ratio of the coverage. They claim that the intersections with more vehicle reports are more important than the other junctions. Therefore, a ranking scheme is proposed to order these locations based on the number of reports sent by taxis within the communication range of each RSU. Their simulation also shows that the order scheme assists in the reducing of transmission range and the overlap ratio on the actual road network to save extra energy consumption.

M. Kafsi et al. [22] maintain that deploying RSUs in the middle of roads is more efficient to avoid uncovered and isolated vehicles. Their research is based on the phe-

nomenon that after most vehicles are stopped at an intersection, these vehicles are spaced apart and they are likely to be isolated. Especially when congestion occurs in the central part of a road system, the vehicles entering the network are more likely to be isolated. The authors notice that the most congested spots in a road system are found at the intersection and the most isolated vehicles are more likely to be in the middle of the road or at the entering points of a road network. Therefore, these authors exploit Poisson model to predict the isolated vehicles in the middle of road segments. Thus, in order to benefit these isolated vehicles, the RSUs should be placed in the middle of road segments rather than in the centre of intersections.

2.1.2 Coverage on Centrality

By evaluating the structural properties of the road network, some centrality-based coverage strategies have come into being. Centrality stresses that some places are more important than others because they are more central.

Existing approaches exploiting centrality all follow a primal representation of spatial attributes of road systems. In this system, notable and distinctive geographic entities (parking lots, settlements, public places of entertainment, intersections) are turned into nodes in a graph and the streets connecting these entities are treated as edges. By simply mapping the crossroads to graph vertices and treating the roads as graph edges, P. Crucitti et al. [8] study the centrality in networks of urban street patterns from different cities. Based on geographical space mapping, the authors compare five different centrality indices over real geographic networks: degree, closeness, betweenness, straightness and information. They indicate that centrality may be used to measure that some nodes are more important (central) than others in a network.

Based on the centrality indices, Y. Do et al. [9] integrate the centrality with Social Network Analysis to identify the most important actors in social networks, or the most important regions in VANETs. Their study focuses on undirected graphs. They assume that the communication between vehicles and RSUs is two directional; thus, the undirected edges could be treated as two-way directed edges. Based on the centrality metrics, the authors propose that by quantifying the importance of single entities in a network, it is easy to discover the centre when looking up the index of these entity individuals. Apart from that, through the aggregation of entities, it is possible to find the relationship between a group-level index and a given measure. The authors claim that centrality metrics would not outperform density-based strategies. But, they can still offer some

interesting functions, such as the monitoring of traffic flow.

A. Kchiche et al. [27] point out that centrality characteristics of an environment achieve the best performance in deploying RSUs. The authors first propose the concept of availability. Availability is a measurement used to express the possibility of a vehicle to meet a RSU. The considerations for availability factor in social networks, connections and conductors. Social considerations imply that cities and essential city infrastructure are highly correlated with the social factor. To measure the social consideration, closeness centrality is taken into account to discover the relation of a node to all other nodes of a road network. Connection consideration is based on the fact that if there are more roads connected to a junction, more vehicles will use this junction. Therefore, connection consideration refers to the degree of centrality. Conductor behavior refers to the fact that drivers prefer to choose the shortest path to reach their destinations. In this way, the betweenness centrality is able to symbolize the junctions with the shortest routes. Through the simulations, the authors further show that the use of centrality can optimize the performance of VANETs, especially in low density areas and in cases of long-distance communication.

Maximizing the availability rate improves the coverage on the service area. However, due to the lack of consideration of distribution of availability, it fails to guarantee a fair service for all vehicles. Thus, in order to reduce the redundancy of coverage area of RSU, a group-based deployment strategy was proposed by A. Kchiche et al. [26] to guarantee a minimum connection time for each vehicle. This idea resides in the notion of group centrality. The authors 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 that of individuals. For each centrality metric, specific averaging or summing up methods are exploited to obtain the group centrality. A. Kchiche et al. succeed to achieve the best performance in terms of delay and overhead during V2V communication in their scenario.

2.1.3 Coverage on Spatial Measurement

Except for the special positions and centrality-based deployment, some researchers exploit the spatial measurement to deploy the RSUs. The total coverage area of RSUs and the spacing distance between RSUs in the path are popular measurements used in research.

P. Lin et al. [39] formulate the RSU deployment problem as a constrained optimization

Table 2.1: Coverage strategies in spatial coverage

Paper	Category	Primary Objective	Secondary Objective	Constraints	Scenario
B. B. Dubey et al. [10]	special positions	improve dissemination capacity	packet dissemination latency	transmission range	urban
J. Lee et al. [36]	special positions	improve connectivity	reduce disconnection interval	number of RSUs / coverage overlap	city
M. Kafsi et al. [22]	special positions	improve connectivity	N/A	vehicle density / traffic light	urban
Y. Do et al. [9]	centrality	improve connectivity	compare density and centrality	density / centrality	rural / urban / city
A. Kchiche et al. [27]	centrality	increase availability and accessibility	improve Access Regularity	service-access delay	low density scenario
A. Kchiche et al. [26]	centrality	increase availability	reduce coverage redundancy	minimum connection time	urban
P. Lin et al. [39]	spatial measurement	improve coverage ratio	minimize the deployment cost	cover all service areas	urban
K. Liu et al. [40]	spatial measurement	improve information dissemination and retrieval	N/A	network size	urban
Z. Zheng et al. [61]	spatial measurement	provide intermittent connectivity	N/A	interconnection gap	urban
H. Cheng et al. [6]	spatial measurement	improve coverage ratio	N/A	topology of road network	urban
S. Sou et al. [51]	spatial measurement	improve connectivity	improve power saving	deployment cost / traffic density	highway
B. Aslam et al. [3]	spatial measurement	minimize the reporting time	N/A	density / speed / likelihood of event	urban

Table 2.2: Coverage algorithms in spatial coverage

Paper	Assumption	Mathematical Model	Algorithm	Mobility Model	Metrics
B. B. Dubey et al. [10]	transmission range is fixed maximum	N/A	N/A	N/A	packet delivery rate
J. Lee et al. [36]	overlap ratio is constant	N/A	N/A	N/A	connectivity
M. Kafsi et al. [22]	vehicles arrival with Poisson distribution	percolation theory	N/A	car following model	connectivity
Y. Do et al. [9]	communication is nondirectional	graph theory	greedy algorithm	MMTS	multi-hop dissemination
A. Kchiche et al. [27]	drivers choose the shortest paths	centrality analysis	heuristic algorithm	car following model	end-to-end delay
A. Kchiche et al. [26]	averaging scheme is reasonable	group centrality	heuristic algorithm	car following model	end-to-end delay
P. Lin et al. [39]	roads are parallel to the corresponding axes	binary integer programming	branch and bound	N/A	coverage ratio
K. Liu et al. [40]	road map updating is a pre-knowledge	connected dominating set	greedy algorithm	ETH traces	connectivity
Z. Zheng et al. [61]	drivers choose the shortest paths	multicut / set cover	greedy algorithm	random waypoint	avaliability
H. Cheng et al. [6]	RSUs are deployed in road buffers	N/A	genetic algorithm	car following model	packet drop rate
S. Sou et al. [51]	RSU has a pre-assigned coordinate ID	power saving model	probabilistic method	Monte-Carlo simulator	connectivity / power-saving index
B. Aslam et al. [3]	vehicles enter with Poisson distribution	binary integer programming	balloon expansion heuristic	ETH traces	reporting time

problem, in which a minimum number of RSUs are used to cover all service areas. To simplify the deployment problem, the authors first transfer the road map into a new graph with multiple nodes. Each node in the graph denotes a candidate position used to deploy RSUs and the locations that should be covered. The transmission range of RSUs determines the distance between these nodes. To optimize the graph and to reduce the workload, the authors prune the graph by removing the locations, which may not need any coverage. By noticing that some locations are not suitable for deployment, two mask matrices are introduced. Based on the new graph, the authors formulate the deployment problem as a binary integer programming problem. Since the transferred problem is NP-hard, the authors use the branch and bound method [58] which could effectively reduce the complexity. To verify the effectiveness of the proposed method, the authors design a simple square scenario without any movement of vehicles.

K. Liu et al. [40] explore the hidden connectivity in urban vehicular networks by transferring the original road network into “intersection graph”. Their research is based on two observations: the street map of the urban road network restricts the mobility pattern of vehicles, and the communication signal of vehicles is directional to the length and orientations of roads. Therefore, the main idea of K. Liu et al. is to embed the traffic flow into the static street map, so that the spatial attributes of a road network reflects the positions of RSU deployment. To solve this problem, the authors design a map transformation by calculating the likelihood that the V2I communication happens at the intersections. The length of the road segments weights such likelihood. Also, most of the data dissemination is completed at the connected dominating set (CDS) on the intersection graph. The use of CDS guarantees that all streets in a road network are either appear in the CDS or directly intersect with at least one road segment within the CDS. The authors then exploit a greedy algorithm to iteratively pick sets based on the ratio of their weight in order to find the minimum weight of CDS. Their mobility files are obtained from ETH traces [44].

Z. Zheng et al. [61] propose a scalable RSU deployment strategy called α -coverage and guarantee the worst-case tracking delay. Their main purpose is to guarantee that there is at least one contact between vehicles and RSUs when vehicles move within α meter. However, since the selection of routes is infinite, it is impossible to determine in polynomial time whether a deployment provides α -coverage. Therefore, such a problem is NP-hard. To resolve this problem, the authors simplify the definition of α -coverage in two ways. First, the movement of vehicles is restricted from one intersection to another, so that the routes of vehicles can be defined as a sequence of junctions. Second, drivers

will always choose the shortest paths from the starting point to the destination. Based on these two simplifications, Z. Zheng et al. solve the α -coverage problem with two greedy algorithms: Multicut and Set Cover. They then prove that the results of their greedy algorithms are guaranteed to be two times worse than the optimal solution.

H. Cheng et al. [6] propose an area-based RSU deployment approach based on the shape of road systems. The authors claim that it is impractical to deploy RSUs at the center part of junctions even though through this manner of deployment the impact of buildings and of the candidate positions of RSUs would be less. Most modern vehicular communication frameworks are built on existing traffic networks so that the APs are only placed beside streets rather than on the surface of the road. Furthermore, it is improper to treat roads as straight lines. Modern road systems involve all kinds of road types and surfaces. Each type has a unique shape, priority, geometry, etc. Without considering the two-dimensional property of roads, the theoretical deployment strategy may fail in assorted real-world road networks. Thus, the geometric characteristics and area of road segments should also be considered. Therefore, the authors define the feasible placement region with the help of the buffering operation. For a certain number of RSUs, the genetic algorithm (GA) is then employed to solve the maximum area coverage problem. To facilitate the evolutionary process, internal-angle-side calibration is employed to adjust the positions of RSUs at corners.

S. Sou et al. [51] designed a power-saving model for the active RSU deployment under a connectivity constraint. In the power-saving model, an RSU is either in active mode or power-saving mode. Therefore, the objective of their research is to design a trade-off between the number of active nodes and the number of power-saving nodes while achieving the guaranteed connectivity. In order to measure how far a message can be delivered through 1-hop forwarding, the authors propose *connectivity index* as the metric. After some manipulations, the authors prove that the *connectivity index* could be obtained in linear time. Based on the *connectivity index* derivation and a pre-defined connectivity, two simple ways are proposed for designers to decide upon the status of RSUs. The first idea assigns an ID to each RSU based on the coordinate sequence. Whenever the status of one RSU is decided, the next RSU within the connectivity distance will be set correspondingly. The ID-based method, however, requires the manipulation of users. Therefore, the authors propose another idea, the probabilistic method, to help RSUs determine their power-saving mode automatically in a distributed manner.

B. Aslam et al. [3] investigate the RSU deployment problem along highways. In their optimization scheme, three factors are taken into consideration: the vehicle speed

which follows the Poisson distribution, the vehicular density which is constant and the likelihood of the occurrence of an accident. In order to simplify the calculation of the average reporting time, the reporting time is measured by the time duration that ranges from the occurrence of an event to the time when information is collated by corresponding RSUs. To resolve this optimization problem, the authors present a so-called balloon optimization method. In balloon optimization, RSUs are considered as balloons whose boundaries are represented by the coverage area of RSUs. Initially, RSUs are deployed uniformly along the road segment. As the algorithm goes on, the balloons will expand to both sides until there is no more space for expansion of balloons.

2.1.4 Mathematic Models in Spatial Coverage

Most spatial coverage models are based on the graph theory and spatial analysis. Among these models, centrality-based analysis and connected dominating set model are commonly used in research. We introduce the two important mathematic models below.

In a social network, each node is treated as an actor. Through centrality analysis, five different measurements are used to define the “most important” actors in the road network. Degree centrality is the simplest type of centrality, which defines central actors as having the highest degrees of centrality in the road network. The node with the highest degree will have the most tiers in a graph. Thus, we treat such a node as the most active node. The degree centrality index C_D for node n_i is defined as follows:

$$C_D(n_i) = \sum_j x_{ij} = \sum_i x_{ji}$$

where $x_{ij} = 1$ if there is an edge connecting nodes n_i and n_j .

Another type of centrality is based on closeness. Closeness centrality measures the distance from one actor to another. If an actor has a high degree of closeness centrality, it means such a node can quickly interact with the other nodes. Define $d(n_i, n_j)$ as the number of edges connecting nodes n_i and n_j . Assuming there are a total of g different nodes in a network, the total distance of actor n_i to the others is $\sum_{j=1, j \neq i}^g d(n_i, n_j)$. Therefore, the closeness index C_C for node n_i is as follows:

$$C_C(n_i) = \left[\sum_{j=1, j \neq i}^g d(n_i, n_j) \right]^{-1}$$

Betweenness centrality is a state in which nodes are on many of the shortest paths of any pair of nodes in a network. This is a very suitable metric for VANETs since

most drivers prefer to drive through the shortest path in a road network. Betweenness centrality is based on the assumption that all the shortest routes are equally likely to be selected by vehicles. The following betweenness centrality index C_B for node n_i is used:

$$C_B(n_i) = \sum_{n_s \neq n_t \neq n_i} \frac{g_{st}(n_i)}{g_{st}}$$

where g_{st} refers to the number of shortest paths linking node s and node t . The symbol $g_{st}(n_i)$ means the shortest path connecting nodes s and t , which contains node n_i .

Two other types of centrality are not as popular as the previously mentioned three ones. Flow betweenness is one such type: sometimes communication does not travel through the shortest paths in a road network and we still want to count the longer routes. Information centrality quantifies the ability of a network to respond to its deactivation. If a node can still maintain network connectivity when one edge is removed, we add a score of 1 to this node. Then the highest score of any node in this network corresponds with its information centrality.

Another important mathematic model exploited by researchers in spatial coverage is the connected dominating set model [35]. In graph theory, a connected dominating set of an undirected graph G is a set of vertices D with two properties. The first property is that every node in D can be connected to another node in D . This means that the subgraph with the node set D is connected. The second property is that every vertex in G is either directly connected to a node in D or belongs to D . Generally, in a spatial coverage strategy the intersections are treated as nodes. The connected dominating set with the minimum number of cardinality is often set as the deployment locations for RSUs.

2.1.5 Summary of Spatial Coverage

Through the comparison of different research on spatial coverage, we can conclude some characteristics for this type of coverage over VANETs. From Table 2.1, it can be found that most spatial coverage concerns seek to improve the connectivity and to increase the coverage ratios on the regions of interest. This is because the connectivity and coverage ratio are natural traits of spatial coverage, which is used to cover the sphere of activity of vehicles. Based on this primary objective, some researchers also advocate secondary objectives, such as improving power saving and reducing the coverage redundancy in the scheduling phase. The major constraints of spatial coverage are the deployment cost

and topology. Some researchers quote the ideas of temporal coverage and spatiotemporal coverage. Thus, their secondary objective and constraints touch upon the delay requirements and statistical analysis. Since their major design pattern covers the spatial characteristics of the road network, we still relegate this research to spatial coverage.

From Table 2.2, we can find that the assumptions of spatial coverage algorithms are based on either the distribution of vehicles or the habits of drivers. Because spatial coverage does not analyze the mobility patterns of the historical traces, a common way to simplify their research is to assume that vehicles enter with Poisson distribution. In some path-based coverage problems, drivers are also assumed to choose the shortest paths. As for the mathematical models used in these coverage algorithms, centrality analysis and other types of graph theory are popular. Since intersections are the most interesting positions in the road network, vertex-based graph analysis is commonly used; some types of the analysis are: the connected dominating set, the centrality analysis, the set-cover calculation and so on. Due to the NP-hardness of spatial coverage problems, some approximation algorithms are exploited. Heuristic greedy algorithms and genetic algorithm are the most common methods used to solve the NP-hard problems. To simulate realistic scenarios and the movement of vehicles, SUMO car following model [33] is commonly used in this research.

2.2 Temporal Coverage

Spatial coverage is attuned to the spatial attributes of a road network, but it fails to consider the mobility of vehicles. With a better understanding of the communication system in VANETs, temporal coverage has been widely researched to cover communication between quick-moving vehicles and fixed RSUs.

2.2.1 Coverage on Delay

Delay is one of the most important metrics used in vehicular networks. Both end-to-end delay and the average packet delay merit consideration when studying the matter of temporal coverage. In the existing research, the number of hops in the packet relayed is often used to measure the degree of delay.

A. Abdrabou et al. [2] study the end-to-end packet delivery delay in multi-hop VANETs. The authors consider a disrupted V2I communication scenario where the direct path from a vehicle to the target RSU is unlikely to exist. Since the end-to-end

delay is due to contention, the number of hops is a good metric for temporal coverage in their scenario. The authors propose an analytical framework to determine the maximum separation distance between RSUs in order to guarantee a required V2I packet delivery rate. Based on this estimated distance, the authors then deploy the RSUs in corresponding locations to cover a straight road segment while satisfying the required number of transmission hops for low density VANETs. The end-to-end packet delivery delay is also verified by parameters like vehicle density, speed and different transmission ranges. The simulation results demonstrate that the end-to-end delay is not influenced by the density, speed and transmission ranges.

P. Li et al. [37] consider the method used 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 centre 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. For a dense network, the authors only deploy one gateway in the optimal location. For a large network, the service area is partitioned into several small dense clusters and each cluster is optimally placed in one gateway. The authors propose two algorithms for the placement of the centre gateway in 2-D dense vehicular networks after the clustering-based partition. 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.

F. Malandrino et al. [42] find that existing coverage over VANETs only considers the V2V or V2I communications. Therefore, they put forward a new coverage, which combines the two types of communications. The authors consider three data transfer paradigms: direct transfers, connected forwarding and carry-and-forward transfers. Their objective is to minimize the number of hops in the communication while maximizing the system throughput. The authors formulate a max-flow problem that accounts for several practical aspects, including channel contention and the data transfer paradigm. Based on a dynamic network topology graph, the vehicular mobility trace is modeled as the possibility of contact occurring events between RSUs and downloaders. By setting two virtual vertices, the source node and the sink node, the network topology is represented by a weighted and directed graph and the RSU deployment problem can be transferred to the max-flow problem. The simulation with the real-world map and realistic traffic volumes shows that the traffic relaying will result in a higher throughput.

Y. Liang et al. [38] formulate the placement of RSUs as an optimization problem and

solve this with integer linear programming. The authors set two constraints to deploy the RSUs in the candidate intersections. The first constraint is that the RSUs should cover at least a minimum percentage of the road areas in a road network. The second constraint is an upper limit, which requires that the number of hops in the packet forwarding should not exceed a certain upper bound. To discover the best configuration for the deployment of RSUs, the authors introduce an incidence matrix, which is a five-dimensional matrix. The incidence matrix consists of the following elements: the coordinates of each RSU in the road network, the given antenna type, the power level and the distance from a RSU to intersections with frequent accidents. Based on these constraints and optimization objectives, the deployment problem is described as an integer linear programming. To simulate this problem, the Poisson distribution is exploited to generate the random traffic scenarios based on a real-world map; and the CPLEX solver [1] is used to obtain the optimization deployment results

2.2.2 Coverage on Contacts

One type of temporal coverage algorithm focuses on the contacts of OBUs and RSUs. Contact occurs the moment vehicles enter the coverage area of RSUs and the V2I communication is about to happen.

O. Trullols et al. [55] seek to maximize the number of vehicles that make contact with the RSUs. They assume that the packet is small enough to be transmitted by only one contact and the V2I contact time impacts the dissemination process. Through the experiments on different location types, the authors select intersections as the optimal candidate deployment position. In order to select the best intersections from a road topology, O. Trullols et al. 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. To simulate the reality of VANETs, the authors exploit the vehicular movement files from VanetMobiSim [18]. They further formulate the problem into another version, which is used to guarantee that the majority of vehicle travel is covered by one or more RSUs for a sufficient amount of time.

C. Lochert et al. [41] present a landmark-based aggregation scheme for saving travel time in road networks. Their main objective is to make full use of strictly limited network bandwidth and to realize the minimal initial deployment. The aggregation scheme is based on a hierarchical method, in which the degree of coarseness of information is increased as the distance of the region enlarges. Unlike existing hierarchical aggregation

schemes focusing on combining data with geographic information of regions, C. Lochert et al. use the travel times as the measurement to scale the coarseness of information in a road network. The authors distribute information about the travel times between prominent points (landmarks) and estimate the travel timesaving 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 the optimal placement improves information dissemination over large distances, especially in a large-scale city model of VANETs.

2.2.3 Coverage on Extra Time Overhead

Except for the delay and contact times, some extra time overhead is also researched in the temporal coverage. Both the time needed for certificate updates in secure authentication and the time needed for algorithm processing are researched thoroughly.

Y. Sun et al. [52] mention that the deployment of RSUs is critical for vehicles to process their short-time certificates. Even though vehicles do not need to store or check a large certificate, the time used for the updating of a certificate is critical due to the limited coverage area of RSUs and intermittent V2I communication. Besides, since the short-time certificate is an important factor for security and privacy, it is important to design a deployment strategy to minimize the driving time for vehicles to communicate with any RSU and the extra overhead for adjusting routes. In their research, an incremental RSU deployment scheme is proposed to achieve the cost-efficient deployment by minimizing the time taken for vehicles to reach each RSU in order to provide a short-time certificate. The small driving time is suitable for privacy sensitive users to change certificates frequently; and, the driving interference is also reduced. Y. Sun et al. model such an optimal deployment problem as a set-covering problem which is NP-hard, and solve it with a classical greedy algorithm. Their simulation is based on the real-world map from TIGER database [5].

S. Wang et al. [56] consider the problem of deploying as few RSUs as possible to meet requirements for a short time update for certificates without expiration. Since privacy conservation is an important issue in VANETs, the frequent update of certificate leads to a more secure authentication. Therefore, the deployment of RSUs, which are responsible for the update of certificates, acts as an important role in the process of updating certificates before expiration. The authors prove that such a RSU deployment problem is NP-hard and propose three RSU allocation methods to resolve it. The first method

is called *the most driving routes first method*. This method is based on the idea that the driving route in which more vehicles pass is more likely to become a location where the certificate is updated. In this way, deploying RSUs on these intersections will benefit secure authentication. The second method is called *the most satisfied intersection pair first method*. This method is based on the idea that if more vehicles travel through a pair of intersections, the intersection between the source and destination will experience more certificate updates. The third method is called *the critical intersection first method*. The degree of *critical intersection* is calculated by the sum of driving time along the roads through this intersection.

R. Kaur et al. [25] study the temporal coverage in another method. The authors discover that the optimistic deployment of RSUs takes too much time due to the sequential processing. It is shown that the task duplication scheduling requires less time to provide higher efficiency. Therefore, based on this idea, the authors provide 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.

2.2.4 Mathematic Models in Temporal Coverage

The temporal coverage models have different objectives from the spatial coverage ones. Most temporal coverage considers the chance of communication and the number of hops between vehicles and RSUs. Therefore, most temporal coverage algorithms choose to cover the communication flows or to increase contacts. We introduce two important temporal coverage models to resolve the two problems.

Max flow problem [35] is a popular optimization problem used in the coverage problem. Researchers use max-flow algorithms to find a feasible flow, which makes full use of link capacity from a single-source, single-sink flow network. Given a directed network $G(V, E)$ where V is the node set and E is the arc set, the existing V2I communication flow, a pair of source nodes s and destination node d , the max-flow problem may be understood as finding the maximum flow supported by the network between pair s and d . Define x_{ij} as the flow over arc (i, j) , f as the flow from source node s to source node d , and C_{ij} as the capacity of arc (i, j) . The maximum flow problem can be formulated

Table 2.3: Coverage strategies in temporal coverage

Paper	Category	Primary Objective	Secondary Objective	Constraints	Scenario
A. Abdrabou et al. [2]	delay	improve multi-hop connectivity	minimize RSU deployment cost	density / transmission range / speed	urban
P. Li et al. [37]	delay	minimize average number of hops	minimize power consumption	number of gateways	highway / urban
F. Malandrino et al. [42]	delay	maximize system throughput	N/A	flow conservation / number of RSUs / channel access	urban
Y. Liang et al. [38]	delay	minimize deployment cost	optimize configurations	transmit power level / antenna type / connectivity	urban
O. Trullols et al. [55]	contacts	maximize the number of contacts	maximize the contact time	network topologies / number of RSUs	urban
C. Lochert et al. [41]	contacts	minimize overall bandwidth	N/A	connectivity / information propagation speed	city
Y. Sun et al. [52]	extra time overhead	improve availability	reduce overhead of certificate update	driving time / certificate update time	urban
S. Wang et al. [56]	extra time overhead	certificates updated before expiration	N/A	driving time / traffic routes	large city
R. Kaur et al. [25]	minimize the parallel time for deployment process	improve coverage ratio	N/A	deployment cost	urban

Table 2.4: Coverage algorithms in temporal coverage

Paper	Assumption	Mathematical Model	Algorithm	Mobility Model	Metrics
A. Abdrabou et al. [2]	vehicles follow Poisson distribution / distance headway follows exponential distribution	bandwidth theory	N/A	user-defined	end-to-end delay / packet delivery rate
P. Li et al. [37]	routing protocol always chooses the lowest number of hops	N/A	greedy algorithm	N/A	average number of hops
F. Malandrino et al. [42]	any node has one interface only	max-flow problem	N/A	multi-agent microscopic traffic	average throughput
Y. Liang et al. [38]	transmission power of RSUs is always lower than that of OBU's / only one transmission power level for all OBU's	integer linear program	CPLEX solver	ETH traces	deployment cost
O. Trullols et al. [55]	packet is small enough to be relayed at once	maximum coverage problem	heuristic algorithm	ETH traces	coverage ratio / contact time
C. Lochert et al. [41]	mean travel time savings service as benefit	hierarchical aggregation	genetic algorithm	VISSIM	travel times
Y. Sun et al. [52]	digital maps are available for OBU's	set-covering problem	greedy algorithm	TIGER traces	number of RSUs
S. Wang et al. [56]	driving time on driving route is known in advance	minimum-hitting-set problem	greedy algorithm	user-defined	average driving time
R. Kaur et al. [25]	all processing is available in n-processor parallel computer	N/A	N/A	N/A	processing time

as follows:

$$\begin{aligned}
& \text{maximize} && f \\
& \text{subject to} && \sum_{(i,j) \in E} x_{ij} - \sum_{(j,i) \in E} x_{ji} = \begin{cases} f & i = s \\ 0 & i = V - \{s, d\} \\ -f & i = d \end{cases} \\
& && \sum_{(k,l) \in C_{ij}} x_{kl} \leq u_{ij}, \forall (i, j) \in E \\
& && x_{i,j} \geq 0, \forall (i, j) \in E
\end{aligned} \tag{2.1}$$

where u_{ij} ($0 \leq u_{ij} \leq 1$) is the normalized remaining capacity or bandwidth for arc (i, j) . The second constraint specifies the resource requirements of each arc according to the resource-sharing graph.

Another important temporal coverage model is the minimum-hitting-set model [35]. The minimum-hitting-set problem is a classical question in combinatorial problems. Given a set of elements U and a set S of n sets whose union equals to U , the target of this problem is to identify the smallest subset of S whose union equals U . Define x_s as a binary indication where $x_s = 1$ means the set s is selected in the solution of minimum-hitting-set problem. The minimum-hitting-set problem can be formulated as an integer linear program as follows:

$$\begin{aligned}
& \text{maximize} && \sum_{s \in S} x_s \\
& \text{subject to} && \sum_{e \in s} x_s \geq 1, \forall e \in U \\
& && x_s \in \{0, 1\}, \forall s \in S
\end{aligned} \tag{2.2}$$

where the first constraint is to cover every element of U and the second constraint points out that every set is either in the final solution or not.

2.2.5 Summary of Temporal Coverage

Table 2.3 presents the comparisons of coverage strategies in temporal coverage. It can be found that most researchers map the metric delay to the number of hops; the more hops used for packet relay, the longer time it takes for the sink node to receive the packet. Therefore, most primary objectives in this category are based on the number of hops. Except for the delay in the packet transmission, the time used to update certificates is also counted in the temporal coverage. Compared with the spatial coverage strategies,

the constraints used in the temporal coverage are more close to the time-related factors, such as the propagation speed, driving time and so forth.

Table 2.4 shows the coverage algorithms used in the temporal coverage category. The assumptions used in temporal coverage can be categorized into two types. The first type of assumptions targets the physical status of nodes in the road network. Thus, the node is either restricted to having only one interface or only one transmission power level. The second kind of assumption is the pre-knowledge of the network. Under these assumptions, the digital maps are available for all vehicles, or the driving time on the driving route is known for OBUs in advance. Unlike the spatial coverage algorithms, temporal coverage algorithms exploit the maximum coverage problem, the set-covering problem, the minimum-hitting-set problem, etc. As for the specific algorithms, most researchers prefer to use the heuristic greedy algorithms, which provide performance guarantees. The mobility models in temporal coverage strategies are also very important. Unlike the existing microscopic mobility models used by spatial coverage researchers, the authors of temporal coverage algorithms prefer to define their own mobility model by analyzing the historical trace files.

2.3 Spatiotemporal Coverage

Temporal coverage is another way to deploy RSUs, which focuses on covering the V2I communications. However, the movement of a vehicle follows the driver's own sense so that it is hard to find a certain pattern to depict mobility. Therefore, researchers either exploit the traffic flow theory to describe the movement of vehicles or mine their own mobility patterns from historical trace files. This kind of coverage is called spatiotemporal coverage.

2.3.1 Coverage on Traffic Flow Theory

Traditional traffic flow theory consists of three descriptions: microscopic description, kinetic description and macroscopic description [4]. In microscopic description, each vehicle is identified individually through speed, weather, driver's habits and so on. Assorted factors make the implementation of a microscopic model hard. Kinetic description is a global state description, which shows statistical distribution on each lane for the position and velocity. As for the macroscopic model, this also provides a global state description by showing locally averaged quantities on three major traffic parameters: density, veloc-

ity and flow.

To improve the cooperative download of data among vehicles in the urban vehicular network, M. Fiore et al. [16] devise a strategy for RSU deployment based on vehicular traffic flow analysis. Since 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 when they 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 travelling on each edge and redeem the calculation results as traversing volumes. They then propose that placing RSUs at the crossing-volume area is able to maximize the potential for collaboration among vehicles as relay nodes. Apart from the cross volume-based RSU deployment, the authors also propose a density-based RSU placement technique to maximize the direct data transfers in V2I communications. Since the downloaders are unknown to the vehicular network designer, it makes sense to deploy RSUs at the most congested intersections based on the large-scale microscopic-level traces.

Rather than cooperation-based coverage, I. Filippini et al. [15] propose a non-cooperative RSU deployment strategy. Unlike traditional strategies acted on by only one operator, the authors study a dynamic scenario where many operators compete to deploy RSUs to cover the maximum number of vehicles. They model this strategy as a game theory model where multiple operators perform their deployment decisions concurrently. Based on the game theory, they propose two types of placement: simultaneous and leader-follower deployment. In simultaneous RSU deployment, the two operators are of equal status. Such as problem is modeled as a strategic game and the strategy degree is measured by the efficiency of the Nash Equilibria (NE) of the game. In leader-follower deployment, one operator acts as the market leader whose priority is higher than the follower. Such a problem is modeled as extensive-form game and efficiency of the sub-game perfect NE serves as the measurement.

T. Wang et al. [57] formulate the continuous coverage based on statistical mobility models. The basis of their work is the definition of popular sites by M. Kim et al. [30]. These authors discovered that the speed and pause time follow a log-normal distribution, weighted by the duration of pause. By aggregating user destinations, they treat the most popular destinations where people spend the most time as the hot spot of a wireless network. T. Wang et al. assume the movement of users start from one virtual popular site through a sequence of consecutive edges. Based on two objectives, maximizing the continuous coverage and minimizing the deployment cost, the authors formulate identical greedy algorithms with a guaranteed approximation. Through the analysis of

two different topologies, tree topology and general topology, the authors transfer the road network into corresponding graphs to find the routes with the highest profit density; this in turn refers to the ratio of deployment profit and road length. The authors also prove the performance guarantee of their greedy algorithms.

Table 2.5: Coverage strategies in spatiotemporal coverage

Paper	Category	Primary Objective	Secondary Objective	Constraints	Scenario
M. Fiore et al. [16]	traffic flow theory	exploit RSU inactivity periods to transmit	N/A	delivery probability	urban
I. Filippini et al. [15]	traffic flow theory	efficiency of the Nash Equilibria	qualified Price of Anarchy	nominal infrastructure capacity / interference / vehicle flows	urban
T. Wang et al. [57]	traffic flow theory	maximize continuous coverage ratio	minimize deployment cost	length of covered paths	urban
Y. Xiong et al. [59]	data mining	improve connectivity	minimize deployment cost	network topology	urban
Y. Zhu et al. [62]	data mining	maximize the expected sensing coverage	minimize deployment cost	delay / probability of successful transmission	urban
X. Fei et al. [13]	data mining	maximize link information gains	maximize origin-destination flow	uncertainty	city

2.3.2 Coverage on Data Mining

In spatiotemporal coverage, choosing the suitable traffic flow model is a very important step. Some researchers maintain that the classic three traffic flow descriptions fail to consider the features of a certain scenario. Therefore, many researchers begin to mine their own mobility pattern through historical information.

Y. Xiong et al. [59] 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. By claiming that the movement of nodes in vehicular networks is relative with social networks, the authors oppose the random movement model used in existing works. Based on the realistic vehicle trace generated by MMTS [45], the authors discover that during some certain time slices the transitions of vehicles are relatively fixed

Table 2.6: Coverage algorithms in spatiotemporal coverage

Paper	Assumption	Mathematical Model	Algorithm	Mobility Model	Metrics
M. Fiore et al. [16]	vehicles are active downloaders of the same contents	mixed-integer quadratic programming	greedy algorithm	ETH traces	download rate
I. Filippini et al. [15]	revenue of an operator is proportional to the traffic load	strategic game	game theoretic tool	ns-3 constant velocity model	transmitted contents
T. Wang et al. [57]	number of RSUs is proportional to the length of the covered path	0-1 knapsack problem	greedy algorithm	microscopic description	probability achieves / number of RSUs
Y. Xiong et al. [59]	each zone can be covered by the communication range of an RSU	time-stable mobility pattern / minimum vertex coverage problem	heuristic greedy algorithm	MMTS	achieved meeting probability
Y. Zhu et al. [62]	RSU communication capacity is large enough to transfer all reports	Markov chain / budgeted maximum coverage problem	greedy algorithm	historical vehicular GPS traces	number of RSUs
X. Fei et al. [13]	time-dependent user equilibrium	stochastic programming model / origin-destination flow model	greedy algorithm	mesoscopic traffic simulator	origin-destination flow coverage / information gains

from one zone to another. Therefore, Y. Xiong et al. divide a road system into several non-overlapping uniform zones and calculate the time-stable transition probabilities between each two zones. The authors then transfer the statistical mobility pattern to a time homogeneous Markov chain. To represent the distribution of transition probability between different state spaces, the transition matrix is exploited. Since the transferred gateway deployment problem can be reduced into a vertex selection problem, which is NP-hard, the authors use a heuristic algorithm *RoadGate* to greedily search current optimal positions.

By mining a large data set of realistic vehicular trace files, Y. Zhu et al. [62] maintain that the movement of vehicles shows a strong regularity. The authors claim that vehicles are moving across the road network with their own will and the probability that vehicles will traverse a road is also different. Y. Zhu et al. 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. The authors not only use the sensing reports for statistical analysis, but the sensing weight matrix is also introduced to measure different degree of importance for reports. Based on the assumption that the future movement of vehicles can be treated as a priori knowledge, the authors provide an efficient greedy algorithm to exhaustively search among the optimal pre-determined positions of RSUs. 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 performance guarantee.

X. Fei et al. [13] maintain that vehicular networks often suffer from different kinds of uncertain changes, so it makes sense to consider the cost of potential damage when deploying access points (APs). At the first stage of the proposed method, the normal incidents, which are modeled as a scenario tree, are considered in the RSU deployment. Based on the analysis of historical origin-destination (OD) flows, the authors apply a stochastic programming model to make the decisions of RSU deployment to maximize the OD flows. The traffic flows are assumed to be time-dependent user equilibriums, which means the travel time experienced by a single vehicle on any path is equal for all users. At the second stage, since in the uncertainty level of situation is considered, a time-dependent assignment matrix is exploited from a simulation-based dynamic traffic assignment model. The proposed methodology is tested on a real-world highway scenario, where the mesoscopic traffic simulator is used to generate the mobility patterns of vehicles.

2.3.3 Mathematic Models in Spatiotemporal Coverage

Spatiotemporal coverage models can be divided into the data-mining phase and NP-hard deployment component. For each phase, the researchers in spatiotemporal coverage propose corresponding mathematic models. We introduce two important models used to solve the data mining and deployment problems in this category of coverage.

Markov chain model [46] is a mathematical system, which quantifies the transitions from one state to another on a state space through observation. In Markov chain model, the next state only depends on the current state rather than the preceding events. A Markov chain is defined as a sequence of random variables $\{X_i\}$. The union of all kinds of values belonging to these variables is called state space. The value of X_n is the state of the system at time n . Since the conditional probability distribution of X_{n+1} to the previous events is the function of X_n , the probability of system state at time $n + 1$ is defined as follows:

$$P(X_{n+1} = x|X_0, X_1, \dots, X_n) = P(X_{n+1} = x|X_n)$$

where x is a random state in the Markov process. This equation can be treated as the property of Markov chain model.

Minimum vertex cover problem [35] is a classical NP-hard optimization problem in graph theory. Since most deployment positions of RSUs are intersections in the road network, the minimum vertex cover problem is commonly used to find a set of nodes so that each edge of the graph is incident to at least one node in the set. Given a graph $G = (V, E)$ where V is the node set and E is the edge set in the graph, $c(v)$ is defined as the cost or weight associated with node v , the minimum vertex cover problem can be formulated as the following integer linear program:

$$\begin{aligned} & \text{maximize} && \sum_{v \in V} c(v)x_v \\ & \text{subject to} && x_u + x_v \geq 1, \forall \{u, v\} \in E \\ & && x_v \in \{0, 1\}, \forall v \in V \end{aligned} \tag{2.3}$$

where x_v is a binary variable which indicates the node v is selected in the solution when $x_v = 1$. The first constraint of this formulation means the problem should cover every edge of the graph. The second constraint means every node is either in the vertex-cover-set or not.

2.3.4 Summary of Spatiotemporal Coverage

Table 2.5 compares the coverage strategies in spatiotemporal coverage. Unlike spatial coverage and temporal coverage, the objective of spatiotemporal coverage is very simple: maximize the coverage ratio in critical areas. The key difference and strengths of spatiotemporal coverage depends of the particular analysis on historical trace files. Just as the Table 2.6 shows, the authors of spatiotemporal coverage algorithms exploit three classical traffic flow theories to discover the most critical regions in the road network. Because the picked regions are extracted from the spatial area and the trace file is an accumulation of vehicle movement in time, this type of coverage is called spatiotemporal coverage and it covers both the spatial area and also the temporal attributes to some extent.

Some spatiotemporal coverage algorithms use data mining techniques to analyze the most valuable regions in a road network. Table 2.6 presents the data mining techniques such as the user-defined mobility graph, time-stable mobility pattern, Markov chain and so on. Based on these mining methods, the coverage problem enters the RSU deployment phase which is NP-hard. In this way, the corresponding mathematic models, knapsack problem, vertex coverage problem and budgeted maximum coverage problems are exploited to model the deployment problems. Due to the performance guarantee provided by most heuristic greedy algorithms, researchers prefer to use these approximation-based algorithms to solve the NP-hard problems.

2.4 Summary

In this chapter, we present an overview of coverage problems in VANETs. Based on different design patterns, we classify these coverage strategies as spatial coverage, temporal coverage and spatiotemporal coverage. For each category of coverage, we provide the detailed classifications according to different objectives. To compare different coverage algorithms in different categories, we analyze the objectives, scenarios, assumptions, mathematic models, algorithms and so on. In a word, spatial coverage considers the spatial attributes of a road network, but it fails to consider the mobility of vehicles. With a better understanding of communication system in VANETs, temporal coverage has been researched widely to cover the communications between high-moving vehicles and fixed RSUs. However, the movement of vehicles follows drivers' own sense so that it is hard to find a certain pattern to depict mobility.

Chapter 3

System Model and Definitions

In this chapter, we discuss the system model for resource-constrained coverage protocol 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

Resource-constrained coverage protocol aims to cover only the most valuable regions due to budget concerns. Therefore, the first step is to define regions worth being covered. We maintain that the regions where most vehicles accumulate are valuable regions. These regions are called *hotspots*. To discover *hotspots* in a road network, we need a metric to measure them. We call this metric *coverage value*. Based on the discovery of *hotspots*, we exploit the geometry-based deployment model to pick candidate locations in which to deploy RSUs. The system model will then select the suitable type of coverage, continuous coverage or sparse coverage, based on the budget estimation. Figure 3.1 uses the flow chart to present the process of resource-constrained coverage protocol.

After obtaining the *coverage value* distribution and *hotspot* discovery, resource-constrained coverage protocol will enter the budget estimation step. Budget estimation selects different coverage models according to different deployment budgets and service demands. For adequate resources, continuous coverage is provided as a KCST model for safety-related applications. However, if the deployment budget is not large enough for minimum continuous coverage, sparse coverage will be designed as an MCP model for driving-assistance and business-promotion services. We use the Lagrangian Decomposition approach and a greedy algorithm to solve continuous coverage and sparse coverage,

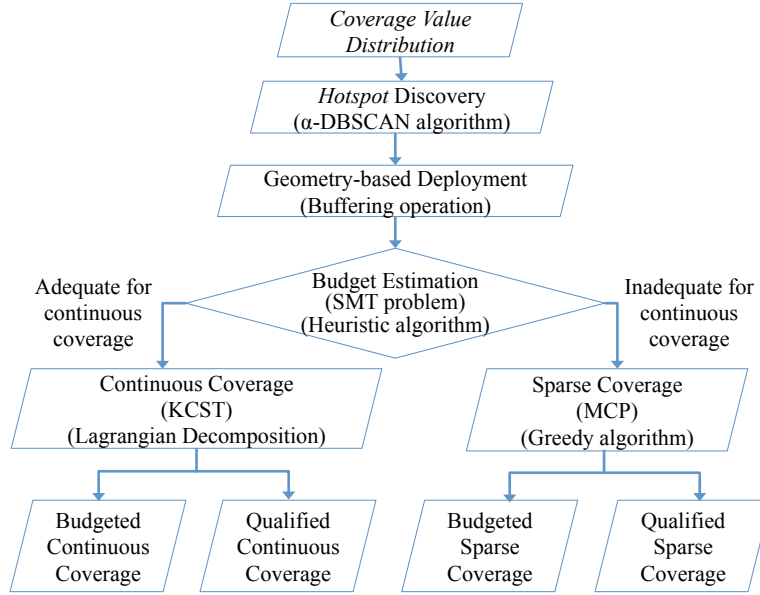


Figure 3.1: Process of resource-constrained coverage protocol

respectively. The budget estimation is reduced to a Steiner Minimum Tree (SMT) problem, which is solved by a heuristic algorithm.

To obtain the mobility pattern of a certain area, we define those regions worth covering in a road system. We maintain that the regions where most vehicles accumulate are the valuable regions. These regions are called *hotspots*. To discover the *hotspots* in a road network, we need a metric to measure it. We call this new metric *coverage value*. After obtaining the *coverage value* distribution and *hotspot* discovery, the system model enters into a geometry-based deployment step. Because the two-dimensional properties of a road network will impact the effects of deployment, we exploit the buffering operation to mesh the buffer regions along the road segments. In this way, the geometrical characteristics of a road network are considered in deployment.

After the buffering operation, the system model will decide to provide the BSC or the QSC based on different optimization objectives. The BSC problem and the QSC problem are modeled as the Budgeted Maximum Coverage (BMC) problem [29] and the Set Cover Problem (SCP) [14], respectively. It has been proven that the BMC is a maximum version of the SCP [20], 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 them. To provide the performance guarantee for our solution, we further employ a greedy algorithm to solve

the BSC and QSC problem.

3.2 Definitions

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

$$\text{coverage value} = \text{flow} * \text{density} / \text{speed} \quad (3.1)$$

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. We maintain that the region with a high *coverage value* is a valuable region and should be covered. To calculate the *coverage value*, we analyze the effects of three macroscopic mobility characteristics, speed, density and traffic flow, in a grid-based road system.

High vehicle speed will result in frequent handoffs, 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*.

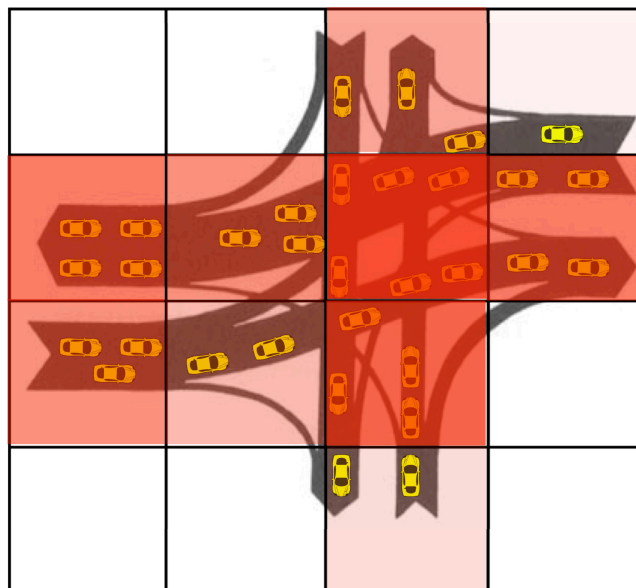


Figure 3.2: Assignment of *coverage value*

We apply a 2-D Gaussian distribution to vehicle speed, vehicular density, and traffic flow according to the probability distribution model proposed in [30]. At each position, we can then calculate the probability of the three parameters: *flow*, *density* and *speed*; and, we can then use formula 3.1 to calculate the *coverage value*. Figure 3.2 shows the assignment of *coverage value* to a grid-based road system. The grids with higher traffic flow and density will be assigned a higher *coverage value*, which is represented by an increased colour opacity in the figure.

Definition 2. *Hotspot*: The region in which the *coverage value* is larger than a threshold α .

Hotspot represents the popular site where most vehicles accumulate. To discover the *hotspots*, we need to aggregate vehicle statistical information to determine the most popular area in which the regional value surpasses the threshold. One simple approach is to apply a type of distribution to describe *coverage value*, such as the log-normal distribution [30]. However, the underlying assumption is unjustified and objective. Instead, in this thesis we divide the area into fixed-sized grids and assign the corresponding *coverage value* to each grid.

To discover the *hotspots* from the distribution of *coverage value*, a density-based clustering algorithm is used to automatically identify clusters with irregular and trivial geometry characteristics. We have developed the α -DBSCAN algorithm by revising the classical density-based algorithm DBSCAN [11]. DBSCAN considers two parameters: ϵ (searching radius) and *minPts* (minimum points required to form a cluster). It starts with an arbitrary core point; and, it then absorbs all the neighbour points within distance ϵ as members of the cluster, based on the distance measurement. When the number of neighbouring nodes reaches the minimum requirement (*minPts*), a cluster is formed. In α -DBSCAN, each grid is a point in the algorithm. The original parameter *minPts* is replaced by α , the threshold of the average *coverage value*.

Algorithm 1 shows the pseudocode of α -DBSCAN algorithm. Based on a different α , another parameter ϵ can be estimated by the k-dist graph [11], so that we treat the threshold α as the pacing factor. If the average *coverage value* of grid g 's neighbours is less than α , this grid 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 grids have been visited and all density-reachable areas have been defined. The final output of α -DBSCAN is a set of *hotspots* $S = \{H_1, H_2, H_3, \dots, H_n\}$.

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

Input: A_{road} , ϵ , α

Output: S

```

1  $S \leftarrow \emptyset$ ;
2 for each UNVISITED  $g$  in  $A_{road}$  do
3   mark  $g$  as VISITED;
4    $N \leftarrow \text{getNeighbours}(g, \epsilon)$ ;
5   if  $V_N \geq \alpha$  then
6     create new  $H \in S$ ;
7      $H \leftarrow g$ ;
8     for each UNVISITED  $g'$  in  $N$  do
9       mark  $g'$  as VISITED;
10       $N' \leftarrow \text{getNeighbours}(g', \epsilon)$ ;
11      if  $V_{N'} \geq \alpha$  then
12         $N \leftarrow N \cup N'$ ;
13      if  $g'$  doesn't belong to any  $H' \in S$  then
14         $H \leftarrow H \cup g'$ ;

```

3.3 Geometry-Based Deployment Model

After the *hotspot* discovery phase, the buffering operation is exploited to match the geometrical attributes of road segments. Because the area around road segments is the primary zone for RSU placement, the definition of a feasible region is required in order to define the extracted roadside area. Figure 3.3(a) represents the vehicle networks in the Yukon Territory of Canada obtained from ArcGIS [47]. We can see in the figure that real-world road networks consist 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 road characteristics [48].

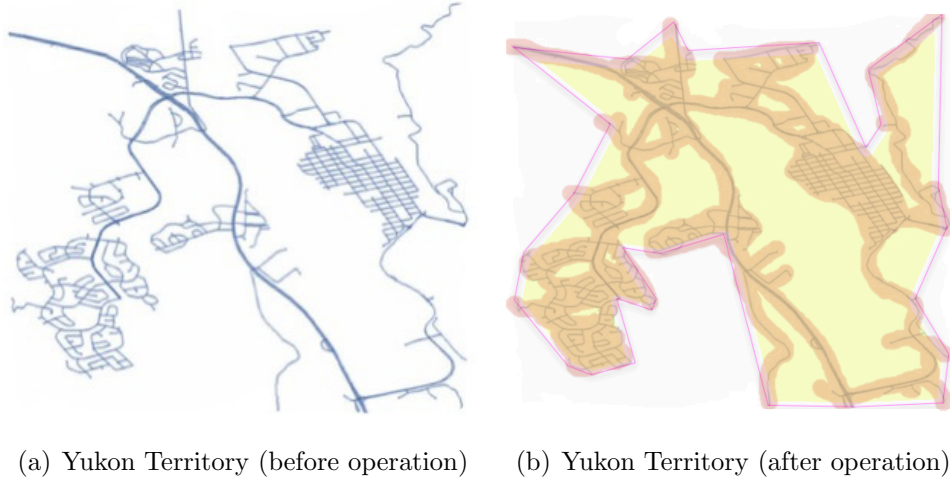


Figure 3.3: Illustration of buffering operation

Figure 3.4 provides a sketch of the buffering operation on straight road segments. The coordinates of roads are known; thus, we simply add buffering lines, the shaded parts shown in Figure 3.4, on both sides of the road segment along the edges of the street. By defining the width of the buffering line, the feasible region is then marked. The width of the buffering-line region is adjusted according to different RSU transmission ranges. To simplify the selection of the buffer width, it is set to the same width as the road: $width_{buffer} = width_{road}$. By adding buffer regions along both sides of the roads, we can divide the buffers into grids and treat each grid 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. Figure 3.5 shows the buffering operation on curved road segments. This idea is the same as those used for the approaches

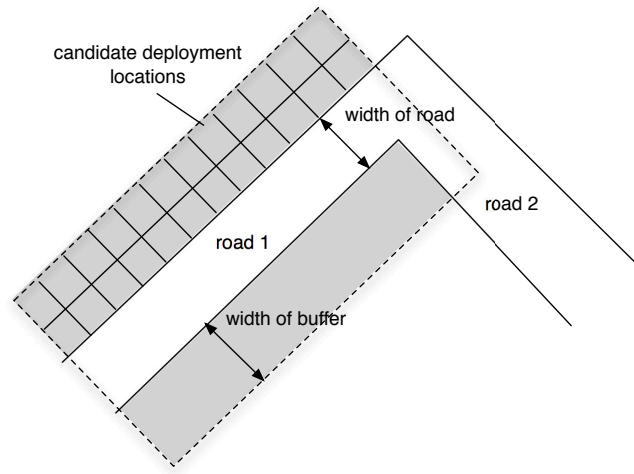


Figure 3.4: Buffering operation on straight roads

in OpenStreetMap [17] when circular arcs are represented.

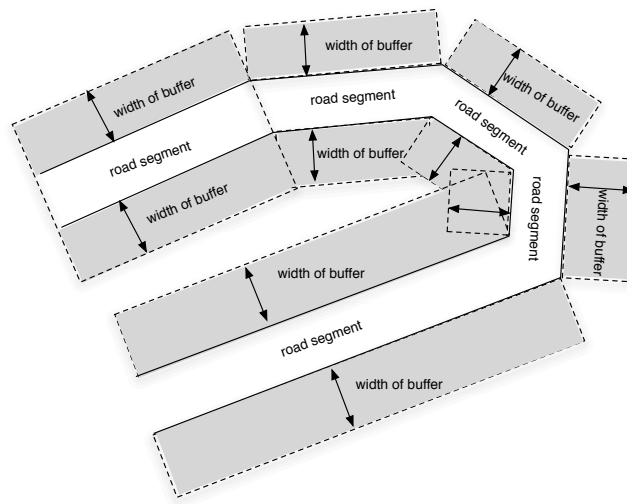


Figure 3.5: Buffering operation on curved roads

With the help of the buffering operation, we extract the feasible deployment region from the original road networks. Figure 3.3(b) shows the Yukon Territory after the buffering operation. The shadow area in Figure 3.3(b) represents the feasible region.

3.4 Summary

In this section, we discuss the system model for resource-constrained coverage protocol and two new definitions. The system model analyzes the budget of deployment and then selects continuous coverage or sparse coverage to meet the budget. Both types of coverage focus on covering *hotspots*, where most vehicles accumulate. To evaluate the *hotspot*, we propose a new metric, *coverage value*, to measure the communication volume of the service area. Based on the *hotspot* discovery, we propose a geometry-based deployment model to select realistic candidate positions to deploy RSUs.

Chapter 4

Resource-Constrained Coverage Protocol

In this chapter, we propose two types of resource-constrained coverage protocols: continuous coverage model and sparse coverage model. For each coverage model, we provide two variants, budgeted coverage and qualified coverage, to meet resource constraints. It is also proven that the budgeted coverage and the qualified coverage in each coverage model are equivalent to each other. We also propose a budget estimation scheme to help select the most suitable coverage model based on resource constraints. Several coverage algorithms are designed to resolve these coverage problems with performance guarantees.

4.1 Continuous Coverage Model

Active road safety applications play important roles in decreasing the probability of traffic accidents and reducing the loss of life of the occupants of vehicles. This kind of service usually needs to satisfy stringent performance requirements of interactive and time-sensitive mobile applications. As general types of intermittent coverage can not satisfy these requirements, we provide continuous coverage for road networks, in which we aim at covering the whole path that the vehicles primarily take. A *hotspot* is able to be treated as an accumulating site for all vehicles in the road network. We build our continuous coverage upon the assumption that the majority of vehicles move from one hotspot to another. By considering the road system as a graph, each *hotspot* is understood as a terminal point in the graph where vehicles accumulate from incident paths. Therefore, we build the system model as shown in Figure 4.1.

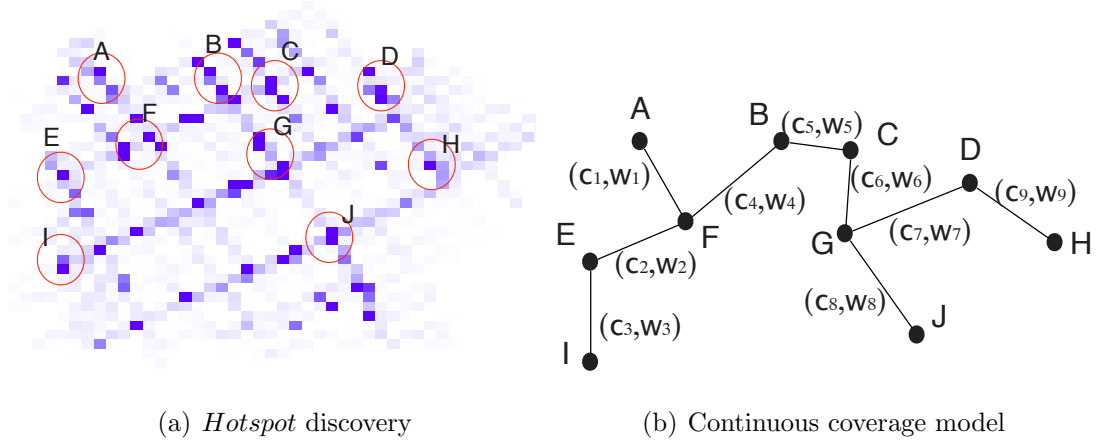


Figure 4.1: Formation of continuous coverage model

Figure 4.1(a) shows the result of *hotspot* discovery, in which the sites with a great number of vehicles will be discovered. Based on the assumption that the majority of vehicles move from one *hotspot* to another, we set the objective of our continuous coverage, which is to find the most suitable subset of paths that will connect all *hotspots*. Figure 4.1(b) shows our continuous coverage model, in which every edge represents the selected path between the two *hotspots*. Each path is a series of connected road segments in a real road system, with the deployment cost and corresponding deployment weight. In practice, most *hotspots* will appear as intersections. However, some *hotspots* may also locate in the middle of a road segment, such as the parking space and waiting zone of a market place. In this case, we separate the road into two edges, and these *hotspots* are treated as new vertices in our model.

In our continuous coverage model, a road system R is treated as an undirected graph $G = (V, E)$. E denotes road segments and each edge $e \in E$ is associated with an RSU deployment cost c_e ($c_e \geq 0$) and a deployment weight w_e ($w_e \geq 0$). The nodes V represent the union of intersections and *hotspots*: i.e., the vertices set *hotspots* H is a subset of V . Because the length of a path is directly proportional to the deployment cost, we simply represent the cost (c) of each path with its length. To select the most valuable paths to be covered, we set the profit (w) of a path as its average *coverage value*. Our objective is to select a subset of paths connecting all *hotspots* with the maximum profit and also maintain the cost as small as possible.

The continuous coverage is modeled to a Knapsack Constrained Steiner Tree (KCST) problem [50]. The KCST problem is a natural combination of two subproblems: Steiner Minimum Tree (SMT) problem [32] and Knapsack Problem KP [54]. Without considering deployment weight, our target is to find the edges that will incur a minimal cost to connect

terminals with additional intersection nodes. We can reduce such a problem to a typical SMT problem, which is a well-known NP-hard problem. If we do not consider the 'Steiner Tree' constraint, the problem is a typical KP. Even though the KCST problem is NP-hard, we exploit the Lagrangian Decomposition approach to resolve it.

Based on the *hotspot* discovery and continuous coverage model, we propose continuous coverage of different objectives according to two different realistic requirements: Budgeted Continuous Coverage (BCC) Problem and Qualified Continuous Coverage (QCC) Problem. BCC considers providing maximum *coverage value* for continuous coverage under a predefined budget. QCC considers the minimum deployment cost and maintaining the lower bound threshold of *coverage value* at the same time. We prove that the two types of continuous coverage are symmetric. To solve the two problems, we exploit Lagrangian Decomposition approaches and corresponding Subgradient algorithm.

4.1.1 Budgeted Continuous Coverage

In the field of continuous coverage, a common requirement for deployment is to maximize the total deployment value of infrastructures under a predefined budget. We describe this problem as Budgeted Continuous Coverage (BCC).

Definition 3. *Budgeted Continuous Coverage:* A deployment of RSUs provides budgeted continuous coverage to a road network R , if the deployed paths form a Steiner tree that spans all the *hotspots* with a maximum total *coverage value* under a given budget B .

The ILP formulation of BCC is shown in the following:

$$\begin{aligned}
 & \text{maximize} && BCC(x) = \sum_{e \in E} x_e \cdot w_e \\
 & \text{subject to} && \sum_{e \in E} x_e \cdot c_e \leq B \\
 & && x \text{ represents a Steiner tree} \\
 & && x_e \in \{0, 1\}, \forall e \in E
 \end{aligned} \tag{4.1}$$

The BCC problem is a KCST problem, which is a natural combination of the SMT and the 0-1 KP problems. We apply Lagrangian relaxation (LR) to reduce the difficult original problem with intertwined constraints to a simpler problem [31]. In LR theory, the problematic constraints are added to the objective function (i.e. dualized) with a penalty term (lagrangian multiplier); this is proportional to the amount of violation of

the dualized constraints. Lagrangian Decomposition (LD) is a special case of LR, in which the NP-hard problem is decomposed into two or more subproblems, which are easier to solve. Since specialized algorithms can efficiently solve the two subproblems, SMT and KP, it is suitable to exploit the LD approach in our problem.

To use the LD approach, we duplicate the 0-1 variable $y_e = x_e, \forall e \in E$ in (4.1), leading to the formulation (4.2):

$$\begin{aligned}
& \text{maximize} && BCC(x) = \sum_{e \in E} x_e \cdot w_e \\
& \text{subject to} && \sum_{e \in E} x_e \cdot c_e \leq B \\
& && y_e = x_e, \forall e \in E \\
& && y \text{ represents a Steiner tree} \\
& && x_e, y_e \in \{0, 1\}, \forall e \in E
\end{aligned} \tag{4.2}$$

By relaxing the SMT constraint in a Lagrangian fashion, we use Lagrangian multipliers $\lambda_e \in R, \forall e \in E$ to absorb the equivalent constraint into the objective function. By doing so, we obtain the LD version of the original problem (4.2); this is denoted by $BCC_{LD}(\lambda)$ as equation (4.3) shows:

$$\begin{aligned}
& \text{maximize} && BCC_{LD}(\lambda) = \sum_{e \in E} x_e \cdot w_e + \sum_{e \in E} \lambda_e (x_e - y_e) \\
& \text{subject to} && \sum_{e \in E} y_e \cdot c_e \leq B \\
& && y \text{ represents a Steiner tree} \\
& && x_e, y_e \in \{0, 1\}, \forall e \in E
\end{aligned} \tag{4.3}$$

$BCC_{LD}(\lambda)$ could then be further decomposed into two subproblems: $BCC_{KP}(\lambda)$ and $BCC_{SMT}(\lambda)$. Each of these could be solved by the KP_{DP} algorithm and SMT_{pru} algorithm respectively.

$$(BCC_{KP}) \max \{(w + \lambda)^T x \mid c^T x \leq B, x \in \{0, 1\}^E\} \tag{4.4}$$

$$(BCC_{SMT}) \min \{\lambda^T y \mid y \text{ is a Steiner tree}, y \in \{0, 1\}^E\} \tag{4.5}$$

Based on the LD theory, for any choice of Lagrangian multipliers λ , the optimal solution value to BCC_{LD} is always at least as large as the optimal solution value of the original BCC problem [31]: i.e., BCC_{LD} provides a valid upper bound. To obtain the

tightest upper bound $\nu(BCC_{LD})$, we solve the Lagrangian dual problem (4.6) with the Subgradient algorithm BCC_{sub} as shown in Section 4.1.3.

$$\forall \lambda, \nu(BCC_{LD}) = \min \nu(BCC_{KP}(\lambda) + BCC_{SMT}(\lambda)) \quad (4.6)$$

4.1.2 Qualified Continuous Coverage

Besides the budget limitations, sometimes people are concerned about the quality of the RSU deployment rather than the cost of the infrastructures. We refer to such a problem as Qualified Continuous Coverage (QCC) problem.

Definition 4. *Qualified Continuous Coverage:* A deployment of RSUs provides QCC to a road network R , if the deployed paths form a Steiner tree that spans all the *hotspots* with a minimum length while the total *coverage value* must extend beyond the quality threshold Q .

The ILP formulation of QCC is as shown in the following:

$$\begin{aligned} & \text{minimize} && QCC(x) = \sum_{e \in E} x_e \cdot c_e \\ & \text{subject to} && \sum_{e \in E} x_e \cdot w_e \geq Q \\ & && x \text{ represents a Steiner tree} \\ & && x_e \in \{0, 1\}, \forall e \in E \end{aligned} \quad (4.7)$$

We can observe that formulations (4.1) and (4.7) are symmetrical; this means the QCC problem could be treated as a minimization variant of BCC. Therefore, the QCC problem could also be reduced to the KCST problem. After the process of Lagrangian Decomposition, the original QCC problem (4.7) could be partitioned into two subproblems: QCC_{SMT} and QCC_{KP} .

$$(QCC_{SMT}) \min \{(c - \lambda)^T x \mid x \text{ is a Steiner tree}, x \in \{0, 1\}^E\} \quad (4.8)$$

$$(QCC_{KP}) \min \{\lambda^T y \mid w^T y \geq Q, y \in \{0, 1\}^E\} \quad (4.9)$$

QCC_{SMT} is similar to the BCC_{SMT} subproblem. Both problems can be solved by SMT_{pru} algorithm. The QCC_{KP} subproblem is a Dual-Knapsack problem from the BCC_{KP} subproblem, thus both the two subproblems can be solved using the KP_{DP} algorithm. Likewise, as the Lagrangian multiplier λ changes, the optimal solution value to the LD version of the QCC problem, QCC_{LD} , is always at least as low as the optimal

solution value of the original QCC problem. The tightest lower bound $\nu(QCC_{LD})$ is also obtained by the BCC_{sub} algorithm mentioned in Section 4.1.3.

$$\forall \lambda, \nu(QCC_{LD}) = \max \nu(QCC_{SMT}(\lambda) + QCC_{DP}(\lambda)) \quad (4.10)$$

4.1.3 Lagrangian Decomposition Algorithm (BCC-LD)

To solve the dual problems BCC_{LD} (4.6) and QCC_{LD} (4.10), we employ a Subgradient algorithm BCC_{sub} to approach the optimal solution [19]. Algorithm 2 is the pseudocode of BCC_{sub} algorithm. As for the subproblems within the BCC_{LD} and QCC_{LD} , we use KP_{DP} algorithm and the SMT_{pru} algorithm to solve them, respectively. The BCC_{heu} algorithm is a subalgorithm of BCC_{sub} used to obtain a tighter lower bound to speed up the convergence.

Algorithm 2: BCC_{sub} – Subgradient algorithm

Input: $\Omega, \epsilon, \lambda^0 = \{\lambda_e^0\} \forall e \in E$

Output: $x = \{x_e\}$

```

1  $i \leftarrow 0, \gamma_0 \leftarrow 2, Z^{H*} \leftarrow +\infty;$ 
2 repeat
3    $i \leftarrow i + 1;$ 
4    $Z^{H_i} \leftarrow KP_{DP}(\lambda^i) + SMT_{pru}(\lambda^i);$ 
5   if  $Z^{H_i} \geq Z^{H_{i-\Omega+1}}$  then
6      $\gamma_i \leftarrow \frac{1}{2}\gamma_{i-1}, \lambda_e^i \leftarrow \lambda_e^{i-\Omega+1}, \forall e \in E;$ 
7   if  $Z^{H_i} \leq Z^{H*}$  then
8      $Z^{H*} \leftarrow Z^{H_i};$ 
9    $Z^L \leftarrow BCC_{heu}(y);$ 
10   $s \leftarrow x^T - y^T;$ 
11   $\lambda_e^{i+1} = \lambda_e^i + \gamma_i(Z^{H*} - Z^L)s/\|s^2\|;$ 
12 until  $\gamma_i \leq \epsilon$  or  $x_e = y_e, \forall e \in E;$ 

```

In BCC_{sub} , we define i as an iteration counter. γ_i is a parameter to be adjusted by the algorithm. ϵ is a threshold, which is close to 0, and Ω is a fixed iteration number. Z^H and Z^L denote upper and lower bounds, respectively. Z^{H*} means $\nu(BCC)$, the best upper bound. Each Lagrangian multiplier λ_e is initially set as 0. Z^{H_i} which denotes the upper bound value on the i -th iteration; λ^i and λ_e^i denote the multipliers generated on

the i -th iteration; and, x_e^i and y_e^i denote the solutions generated on the i -th iteration. At each iteration, the multipliers λ_e are updated by moving a specified step-size along the subgradient direction $x_e - y_e$. Z^L is also obtained by Lagrangian heuristic algorithm BCC_{heu} . If the algorithm fails to improve the upper bound Z^{H_i} after Ω iterations, the multipliers are reset to $\lambda^{r-\Omega+1}$. BCC_{heu} subsequently resumes its search for an improved bound with a halved γ_r .

KP_{DP} is an efficient and exact algorithm for the BCC_{KP} problem based on Dynamic Programming (DP). The time complexity of BCC_{KP} is $O(|E|B)$ and space complexity is $O(|E|B)$ [54], where $|E|$ represents the number of edges. Algorithm 3 is the pseudocode of KP_{DP} algorithm. KP_{DP} recursively solves the subproblem: at stage e the total expense is C_e and the current quality is W_e , whether or not the adding of a new edge will result in a higher quality W_{e+1} while the total expense $C_{e+1} \leq B$. If there are only $|E|$ candidate edges and the array $T[|E|, B]$ is used to store entry items $I[e, c]$, the algorithm will find the optimal solution.

Algorithm 3: KP_{DP} – Dynamic Programming algorithm

Input: $\{c_e\}, \{w_e + \lambda_e\}, |E|, B$

Output: $T[|E|, B]$

1 **for** c from 0 to B **do**

2 $T[0, c] \leftarrow 0;$

3 **for** e from 1 to $|E|$ **do**

4 **for** c from 0 to B **do**

5 **if** $c \geq c_e$ **then**

6 $T[e, c] \leftarrow \max\{T[e - 1, c - c_e] + w_e + \lambda_e, T[e - 1, c]\};$

7 **else**

8 $T[e, c] = T[e - 1, c];$

To solve the SMT subproblem, we develop a heuristic algorithm SMT_{pru} based on the Minimum Spanning Tree (MST). Algorithm 4 represents the details of SMT_{pru} where a minimum cost spanning tree is computed first, and the unnecessary nodes and edges are subsequently pruned [53]. Compared with a distance network-based heuristic, the pruned-MST heuristic is adequate for negative weights and even forests. This is the reason for which we chose pruned-MST heuristic to solve BCC_{SMT} subproblem. The time complexity of SMT_{pru} is $O(n^2)$ and the approximate ratio of the algorithm to the

optimal solution is proven to be $|V| - |H| + 1$ [53].

Algorithm 4: SMT_{pru} – Pruned MST algorithm

Input: $G(V, E)$, $H \subseteq V$, $c = \{\lambda_e\}$

Output: T

- 1 $T \leftarrow \emptyset$;
 - 2 Construct a minimum spanning tree T of G ;
 - 3 **forall the** $\{(s, t) | s, t \notin H, s \text{ or } t \in \text{leaf}(T)\}$ **do**
 - 4 $T \leftarrow T \setminus \{(s, t)\}$;
-

BCC_{heu} is a Lagrangian heuristic responsible for generating feasible solutions to derive better lower bounds. Algorithm 5 describes the details of BCC_{heu} algorithm. After obtaining a new solution, BCC_{heu} will search for a better set of candidate edges to improve the lower bound under the budget. If it fails, there will be no change to the solution.

Algorithm 5: BCC_{heu} – Heuristic Local Search algorithm

Input: $G = (V, E)$, $y = \{y_e\}, \forall e \in E$

Output: $y = \{y_e\}$

- 1 **if** $\sum_{e \in E} y_e \cdot w_e \leq B$ **and** $\sum_{e \in E} y_e \cdot w_e$ **then**
 - 2 $y' \leftarrow \text{LocalSearch}(x)$ **if** $\sum_{e \in E} y'_e \cdot w_e \geq \sum_{e \in E} y_e \cdot w_e$ **then**
 - 3 $y \leftarrow y'$;
 - 4 $Z^L \leftarrow \sum_{e \in E} y'_e \cdot w_e$;
-

4.2 Sparse Coverage Model

Even though continuous coverage provides persistent monitoring on safety-related applications, its deployment cost is prohibitive for a tight budget. Therefore, when the deployment budget is not enough to provide minimum continuous coverage, we formulate sparse coverage to provide traffic monitoring and management, navigation cooperative local services and advertisement delivery over *hotspots*. Traffic efficiency and management are typical driving-assistance services, such as speed management, navigation cooperative local services and so forth. These applications focus on locally based services in

areas with high traffic flow or crowded vehicles. 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.

The sparse coverage algorithm is based on the *hotspots* in a road system. However, the method of deploying RSUs with the help of *hotspot* must be illustrated with a suitable coverage model. The sparse coverage model is shown in Figure 4.2. It indicates that for each cluster (*hotspot*) discovered in Figure 4.2(a), we divide the area into fixed grids to be covered. In the sparse coverage model, *hotspots* serve as the main regions for coverage with arbitrary shapes. Therefore, we relax the α value used in *hotspot* discovery process to obtain *hotspots* with arbitrary shapes.

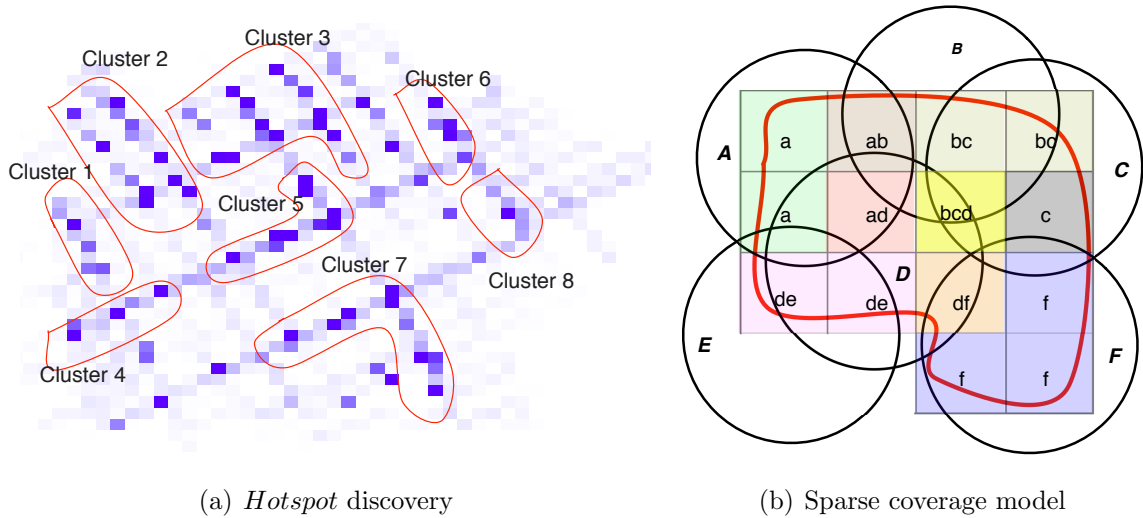


Figure 4.2: Formation of sparse coverage model

The grid-based method can be a natural choice for RSU deployment in which the infinitely available space is mapped into finite grid cells. It should be pointed out that the size of buffer grids has nothing to do with the size of the road area. Figure 4.2(b) is an example of the grid-based sparse coverage on a *hotspot* of arbitrary shape. In Figure 4.2(b), there are 14 different grids $\{g_1, g_2, g_3, \dots, g_{14}\}$. Each grid owns a *coverage value* so that we also have a *coverage value* set $\{v_1, v_2, v_3, \dots, v_{14}\}$. If, in a road network scenario, there are a total of 6 buffer grids $\{A, B, C, D, E, F\}$ for RSU deployment and the sensing model is set as a disk, then the coverage result of each RSU is shown in the

following:

$$\begin{aligned}
 RSU_A &= \{a * 2, ab, ad\} \\
 RSU_B &= \{ab, bc * 2, bcd\} \\
 RSU_C &= \{c, bc * 2, bcd\} \\
 RSU_D &= \{df, de * 3, ad, bcd\} \\
 RSU_E &= \{de * 2\} \\
 RSU_F &= \{f * 3, df\}
 \end{aligned}$$

The label of a grid is a combination of all the RSUs that cover this grid. For example, grid bcd is covered by three RSUs: b , c and d . The road grids of the same label belong to a specific *section*: in the case where the two de grids belong to *section de*. As for the grids only covered by one RSU, they are regarded as their own *section*. Thus, there are a total of 9 *sections* in Figure 4.2(b) as shown by the colour label; and, each *section* is a set of grids covered by the same set of RSUs. We define the *coverage value* of a *section* as the mean arithmetical value of the combined grids. Therefore, a common sparse coverage is based on a collection of *sections* $S = \{S_1, S_2, \dots, S_m\}$ and a set of RSUs $U = \{U_1, U_2, \dots, U_n\}$; and, U_i is a subset of S for every $i \leq n$. Each section S_j is associated with a weight w , the *coverage value* of that region. Each RSU U_i is associated with a cost c , the expense used to deploy the RSU.

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

Definition 5. *Budgeted Sparse Coverage:* A deployment of RSUs provides Budgeted Sparse Coverage to a road network R , if the selected subset of RSUs maximizes the sum of the weight of covered sections under the constraint that the sum of these deployment costs is no larger than a given budget B .

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

$$\begin{aligned}
& \text{maximize} && BSC(x) = \sum_{j=1}^m y_j \cdot w_j \\
& \text{subject to} && \sum_{i=1}^n x_i \cdot c_i \leq B \\
& && \sum_{S_j \in U_i} x_i \geq y_j \\
& && x_i, y_j \in \{0, 1\}, 1 \leq i \leq n, 1 \leq j \leq m
\end{aligned} \tag{4.11}$$

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 [29] 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 7.

4.2.2 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 6. *Qualified Sparse Coverage:* A deployment of RSUs provides Qualified Sparse Coverage to a road network R , if the selected subset of candidate RSUs minimizes the cost of RSU deployment and meets the lowest *coverage value* threshold Q at the same time.

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

$$\begin{aligned}
& \text{minimize} && QSC(x) = \sum_{i=1}^n x_i \cdot c_i \\
& \text{subject to} && \sum_{j=1}^m y_j \cdot w_j \geq Q \\
& && \sum_{S_j \in U_i} x_i \geq y_j \\
& && x_i, y_j \in \{0, 1\}, 1 \leq i \leq n, 1 \leq j \leq m
\end{aligned} \tag{4.12}$$

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 [14] 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 S. Khuller et al. [29], 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. The *Greedy Cover* algorithm can approximately solve both of the NP-hard problems.

To solve the BSC and QSC problems, we propose two types of algorithms to resolve the sparse coverage 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.

4.2.3 Genetic Algorithm (BSC-genetic)

Due to the NP-hard characteristics of the sparse coverage 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.

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

Algorithm 6: Genetic Algorithm

Input: candidate locations, *threshold***Output:** optimal chromosome

- 1 Encoding and Initialization
 - 2 Encoding chromosome;
 - 3 Set area, transmission range and number of RSUs;
 - 4 Initial population randomly;
 - 5 **while** *fitness* \leq *threshold* **do**
 - 6 | Selection
 - 7 | Rank chromosomes with fitness function;
 - 8 | Place optimal population into next generation;
 - 9 | Reproduction
 - 10 | Crossover to generate new offsprings;
 - 11 | Mutate to generate new gene;
 - 12 **return** optimal chromosome
-

solution as the chromosome. In formula $chromosome = (g_1, g_2, g_3, \dots, g_n)$, 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 $P = (c_1, c_2, c_3, \dots, c_m)$.

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:

$$R = \frac{\sum_{i=1}^{i \leq |G_{road}|} \{q_i \mid q_i \subseteq \sum A_{RSU} \text{ for all } q_i \subseteq G_{road}\}}{G_{road}}$$

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

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:

$$\begin{aligned} child^{1'} &= \{ \{g_1^1, g_2^1, \dots, g_c^1\} \{g_{c+1}^2, g_{c+2}^2, \dots, g_n^2\} \} \\ child^{2'} &= \{ \{g_1^2, g_2^2, \dots, g_c^2\} \{g_{c+1}^1, g_{c+2}^1, \dots, g_n^1\} \} \end{aligned}$$

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 $p(0 \leq p \leq 1)$, which is very small, so that it will not develop into an intolerable influence. Because a gene is denoted by a two-dimensional coordinate, 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 grid size so that every time the mutation happens the position of an RSU only moves to its neighbour grid. The mutation function is shown below:

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

To guarantee the termination of evolution, we bound the number of generations by 1000. In addition, if the fitness of some chromosomes exceeds 99%, we claim that the optimal solution for node distribution has already been found. The algorithm will then also terminate.

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 $O(gen * (sel + cro + mut))$. Gen is the number of generations, 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 of sorting; thus, sel , which is set

as the time complexity of selection, should be $\max(T(R) * |P|, T(\text{sorting}))$. Cro is the time complexity of crossover 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.

4.2.4 Greedy Algorithm (BSC-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 sparse coverage problem instead. Algorithm 7, *Greedy Cover*, is the basic framework for the greedy algorithm on the general coverage problem.

Algorithm 7: *Greedy Cover* – Greedy algorithm

Input: $S = \{S_j\}$, $U = \{U_i\}$

```

1 repeat
2   | select  $U_i$  to cover a set of UNVISITED  $S_j$  with maximum profit;
3   | mark the covered  $S' = \{S_j\}$  as VISITED;
4 until done;
```

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 8 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 *sections* 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 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.

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.

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

Algorithm 8: BSC_{unit} – Greedy algorithm

Input: $S = \{S_j\}, U = \{U_i\}, K, B$ **Output:** D

```

1  $D_1 \leftarrow \operatorname{argmax}\{w(U') \mid U' \subseteq U, |U'| < K, c(U') \leq B\};$ 
2  $D_2 \leftarrow \emptyset, D \leftarrow \emptyset;$ 
3 forall the  $\{U' \mid U' \subseteq U, |U'| = K, c(U') \leq B\}$  do
4    $S \leftarrow U \setminus U';$ 
5   repeat
6      $\text{select } U_i \leftarrow \operatorname{argmax}\{\frac{w(U_i)}{c_i} \mid U_i \subseteq S\};$ 
7     if  $c(U') + c_i \leq B$  then
8        $U' \leftarrow U' \cup U_i;$ 
9        $S \leftarrow S \setminus U_i;$ 
10    until  $S \leftarrow \emptyset;$ 
11    if  $w(U') > w(D_2)$  then
12       $D_2 \leftarrow U';$ 
13 return  $\operatorname{argmax}\{w(D) \mid D \in \{D_1, D_2\}\}$ 

```

Since the proof of BSC_{unit} algorithm is similar to the proof of greedy algorithms used in the BMC problem [29] and MCP [21], 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 $U'_i = \sum_{j=1}^i U_j$ 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. $w(U_{i+1}) \geq w(\widehat{U}_i)/K$.

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

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

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

the induction steps:

$$\begin{aligned}
w(\widehat{U}_{i+1}) &\leq w(\widehat{U}_i) - w(U_{i+1}) \\
&\leq w(\widehat{U}_i)(1 - 1/K) \text{ (using Lemma 1)} \\
&\leq (1 - 1/K)^{i+1} \cdot w(OPT)
\end{aligned} \tag{4.13}$$

Now we start to prove Theorem 1.

Proof: It follows from Lemma 2 that

$$\begin{aligned}
w(\widehat{U}_{i+1}) &\leq (1 - 1/K)^{i+1} \cdot w(OPT) \\
&\leq w(OPT)/e
\end{aligned} \tag{4.14}$$

Therefore,

$$\begin{aligned}
w(U'_i) &= w(OPT) - w(\widehat{U}_i) \\
&\geq w(OPT) - w(OPT)/e \\
&= (1 - 1/e)w(OPT)
\end{aligned} \tag{4.15}$$

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.

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 [21], we simply declare the process of the proof. Let OPT 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 $U'_i = \sum_{j=1}^i U_j$ 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 OPT RSUs. Therefore, on average, any RSU in the optimal solution is able to cover at least $(S - U'_i)/OPT$ uncovered *sections*. We can infer the following equations:

$$U'_{i+1} - U'_i \geq S - U'_i \tag{4.16}$$

In this way,

$$\begin{aligned}
S - U'_{i+1} &\leq (S - U'_i)(1 - 1/OPT) \\
&\leq S(1 - 1/OPT)^{i+1} \\
&\leq S \cdot e^{-(i+1)/opt}
\end{aligned} \tag{4.17}$$

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.

$$\begin{aligned}
S - U'_{K+1} &< OPT \\
&\leq S - U'_K
\end{aligned} \tag{4.18}$$

Therefore,

$$\begin{aligned}
K &\leq i + OPT, OPT \\
&\leq S \cdot e^{-K/OPT} \\
&\leq OPT(1 + \ln S/OPT) \\
&\leq OPT(1 + \ln r)
\end{aligned} \tag{4.19}$$

4.3 Budget Estimation

Resource-constrained coverage protocol provides a selection scheme that falls between the continuous coverage and sparse coverage protocols. The metric for selecting which type of coverage to use is based on budget estimation model. We maintain that the budget estimation model is similar to the continuous coverage model, which develops a KCST based on *hotspots*, without considering the weight of *coverage value*. The budget estimation model is a SMT model as Figure 4.3 shows.

The objective of the budget estimation model is to find the shortest paths connecting these *hotspots*. Such a model can be reduced to a classical SMT model. However, the SMT model is a NP-hard problem. We exploit Kou-Markowsky-Berman (KMB) algorithm [32], which is a proven efficient heuristic with near-optimal solutions and an approximation ratio to solve it. The KMB algorithm is a 2-factor approximation algorithm for the SMT problem in a graph.

Based on the KMB algorithm, we introduce SMT_{KMB} algorithm. Algorithm 9 represents the pseudocode of SMT_{KMB} algorithm. For a connected undirected road graph $G(V, E)$ and a set of *hotspots* $H \subseteq V$, each edge $e \in E$ is associated with a cost c_e . Algorithm 9 is able to output a Steiner tree spanning all the *hotspots* in such a road

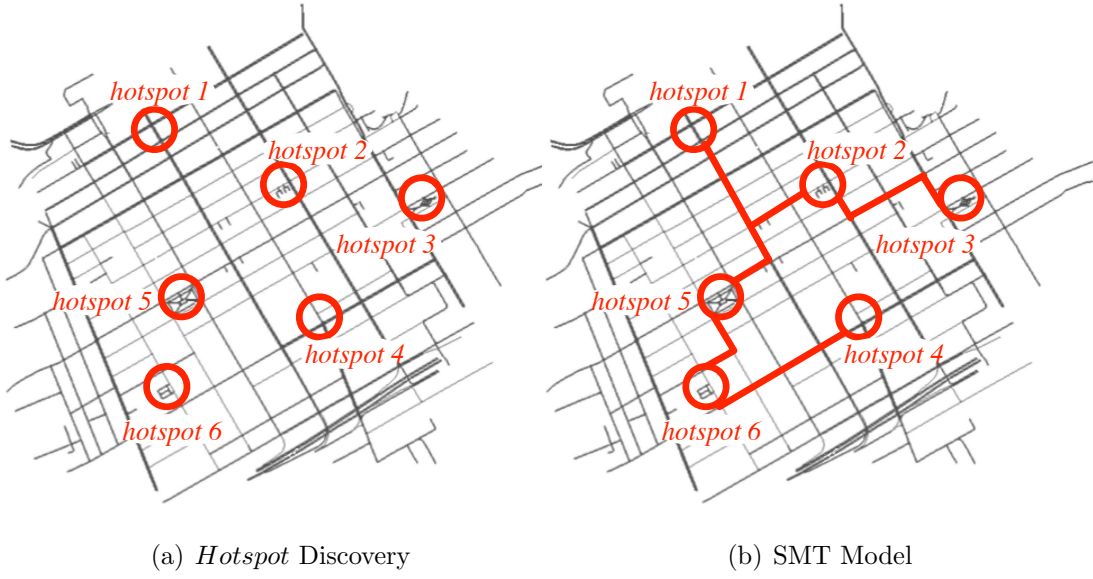


Figure 4.3: Budget estimation model

graph. The implementation of Algorithm 9 is also straightforward. The first step is to find the complete minimum-distance graph G' upon *hotspot* terminals and the corresponding spanning tree T . The next step is to translate the tree T to a graph and to remove the cycles. The complexity of SMT_{KMB} algorithm is $O(|H||V|^2)$. Let $\nu(\cdot)$ denote the optimal solution value of problem (\cdot) , $SMT_{KMB} \leq 2 * \nu(SMT)$; and, the proof is presented in [32].

Algorithm 9: SMT_{KMB} – Heuristic algorithm

Input: $G(V, E)$, $H \subseteq V$, $c = \{\lambda_e\}$

Output: T

- 1 $T \leftarrow \emptyset$, $G' \leftarrow \emptyset$;
 - 2 **forall** the $\{(s, t) | s, t \in H\}$ **do**
 - 3 $G' \leftarrow G' \cup$ min-cost path $P(s, t)$;
 - 4 $T \leftarrow$ minimum spanning tree of G' ;
 - 5 **forall** the $e \in T$ **do**
 - 6 substituting e with a corresponding path in G ;
 - 7 remove cycles in T ;
-

4.4 Summary

In this chapter, we formulate two types of resource-constrained coverage protocols over VANETs which satisfy the resource budget and quality requirements. For a sufficient budget, we propose a continuous coverage as a KCST problem. However, if resources are inadequate for minimum continuous coverage, we provide a sparse coverage and reduce it to a MCP. Due to the NP-hardness of the two coverage models, we resolve them with Lagrangian Decomposition and a greedy algorithm, respectively. Both the two coverage models are based on *hotspots*, where most vehicles accumulate. To help network operators make decisions, we provide a budget estimation scheme to select the suitable type of coverage based on the resource constraint.

Chapter 5

Performance Evaluation

In this chapter, we present the methodology and experimental setup for the evaluation of the algorithms proposed in the resource-constrained coverage protocol. We then provide the experimental results for *hotspot* discovery and Lagrangian Decomposition to verify the effectiveness of our coverage algorithms. Based on the comparison with baseline algorithms, the performance of BCC and BSC algorithms are proven to be suitable and stable in vehicular networks.

5.1 Methodology and Experimental Setup

Our simulation is based on the Network Simulator (NS2) [43] and the Simulation of Urban Mobility (SUMO) [33]. SUMO is responsible for generating the mobility models of vehicles in a road network. NS2 exploits the mobility model files to simulate the V2I communication with a given protocol stack. To emulate the real scenario, we captured the real road networks of Ottawa’s downtown area. Figure 5.1 provides the map of our simulation environment. The map data was obtained from OpenStreetMap, in which road segments are represented as 2-D polygons with various shapes. The simulation scenario is a $2300m \times 2100m$ map. This map consists of a total of 377 intersections and 776 road segments.

Figure 5.2 illustrates the scenario of my simulation. In this simulation, vehicles send packets to the RSUs through a wireless channel. The RSUs then forward packets to the sink base station. After all packets are gathered in the center server, the feedback is sent to vehicles through the RSUs. We use the car-following model [34] to imitate the real movement of vehicles. The car-following model is implemented in SUMO to describe the

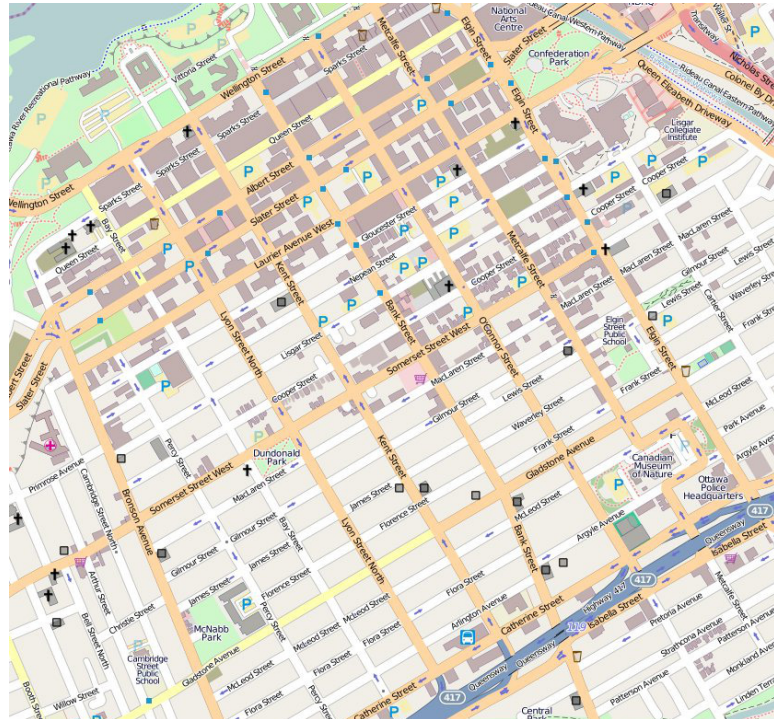


Figure 5.1: Downtown map of Ottawa

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.

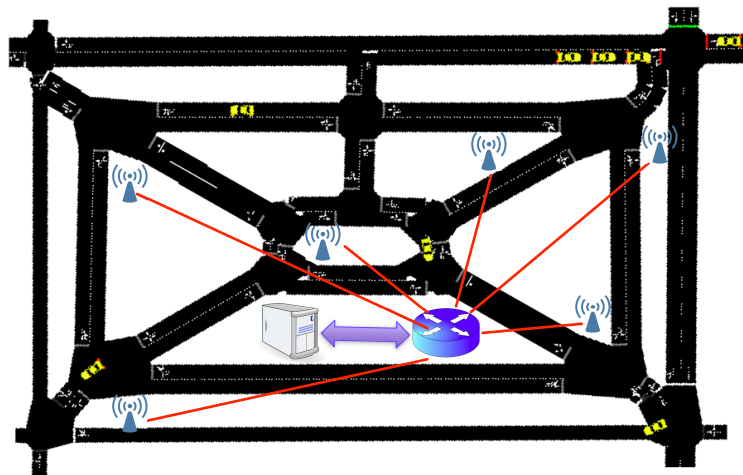


Figure 5.2: Simulation scenario

Table 5.1 shows the detailed simulation parameters. Due to the high mobility of

Table 5.1: Simulation parameters

Simulator	NS-2.35
Mobility Model	Car-Following Model
Area of Map	2300m \times 2100m
Number of Vehicles	100
Vehicle Speeds	0~20 meter/s
PHY / MAC	IEEE 802.11p
Routing Protocol	GPSR/AODV
Transport Protocol	UDP
Network Traffic	CBR (160 bytes, 50 pps)
Simulation Time	500 seconds

vehicles in VANETs, we used IEEE802.11p at the physical layer and at the MAC layer. We analyzed 50 groups of vehicle mobility files, and each file records the running of 100 vehicles in 500 seconds. 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 algorithms with AODV and GPSR protocols.

5.2 Baseline Algorithms

In order to evaluate algorithms in the resource-constrained coverage protocol, Maximum Continuous Coverage (MCC) algorithm [57] and α -coverage algorithm [61] are introduced as the baseline algorithms.

Maximum Continuous Coverage (MCC). As a continuous coverage problem, MCC treats the most popular destinations where people spend the most time as the popular site of a wireless network. The objective of MCC is to select a subset of paths that maximize the sum of path probabilities under the constraint that the sum of these paths cost is no larger than a given constant. Based on two objectives, maximizing the continuous coverage and minimizing the deployment cost, the authors formulate identical greedy algorithms with a guaranteed approximation. The authors transfer the road network into corresponding graphs to find the routes with the highest profit density; this in turn refers to the ratio of deployment profit and road length.

α -coverage. Acting as a sparse deployment idea, α -coverage also focuses on the

same metric that suggests using fewer RSUs to provide better coverage performance. This 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 sparse coverage strategy is in the category of spatial coverage, the key point of our idea is to maximize the contact time between an RSU and vehicles, which is similar to α -coverage. Therefore, we think that the comparison with α -coverage is a good way to justify the effectiveness of geometry-based sparse coverage protocol.

5.3 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 Figure 5.3.

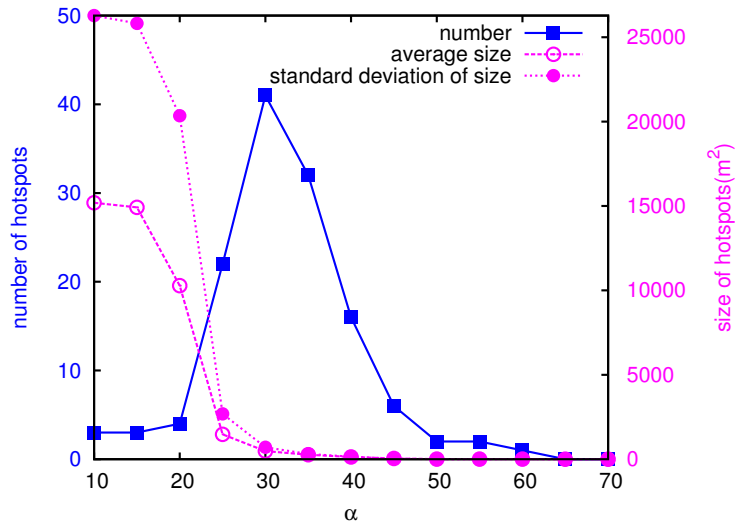


Figure 5.3: *Hotspot* discovery analysis

Through Figure 5.3, 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. When α reaches 65, no *hotspot* can be discovered.

Unlike the number of *hotspots*, the average size of *hotspots* and the standard deviation of sizes continually 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 experiment documented below, we chose to use 20 *hotspots*, as α equals 25.

5.4 Analysis for Continuous Coverage Simulation

We discuss the performance of Lagrangian Decomposition and continuous coverage simulations in this section. We first describe the performance of Lagrangian Decomposition method based on BCC_{sub} algorithm. Since BCC and QCC problems are symmetrical, we only show the result of BCC_{sub} algorithm under different budgets. Table 5.2 shows the final value of the gap for different budget constraints using the Subgradient algorithm. Since the total length of covered paths is directly proportional to the total cost of RSU deployment, we use the total length of paths as the budget constraint.

Table 5.2: Lagrangian Decomposition results

budget (m)	iteration	<i>coverage value</i>	gap
1000	40	52636	1.89774%
2000	30	63690	1.0514%
3000	31	69810	1.9361%
4000	31	75688	1.8770%
5000	31	81641	1.4173%
6000	32	89279	0.0292%
7000	34	96494	0.1678%
8000	36	105474	0.0569%
9000	38	115242	0.1216%

According to Table 5.2, the BCC_{sub} algorithm converges well between 30 and 40 iterations regardless of the budget. This means that the proposed algorithm shows stable running time for different budget thresholds. The gap between upper bound and lower bound is also very tiny, which means the final result is very close to the optimal

solution. As the budget for RSU deployment increases, more popular paths become available for selection in a road network. In this way, the generated Steiner tree is closer to the optimal solution. The corresponding gap between upper bound and lower bound will also be very small.

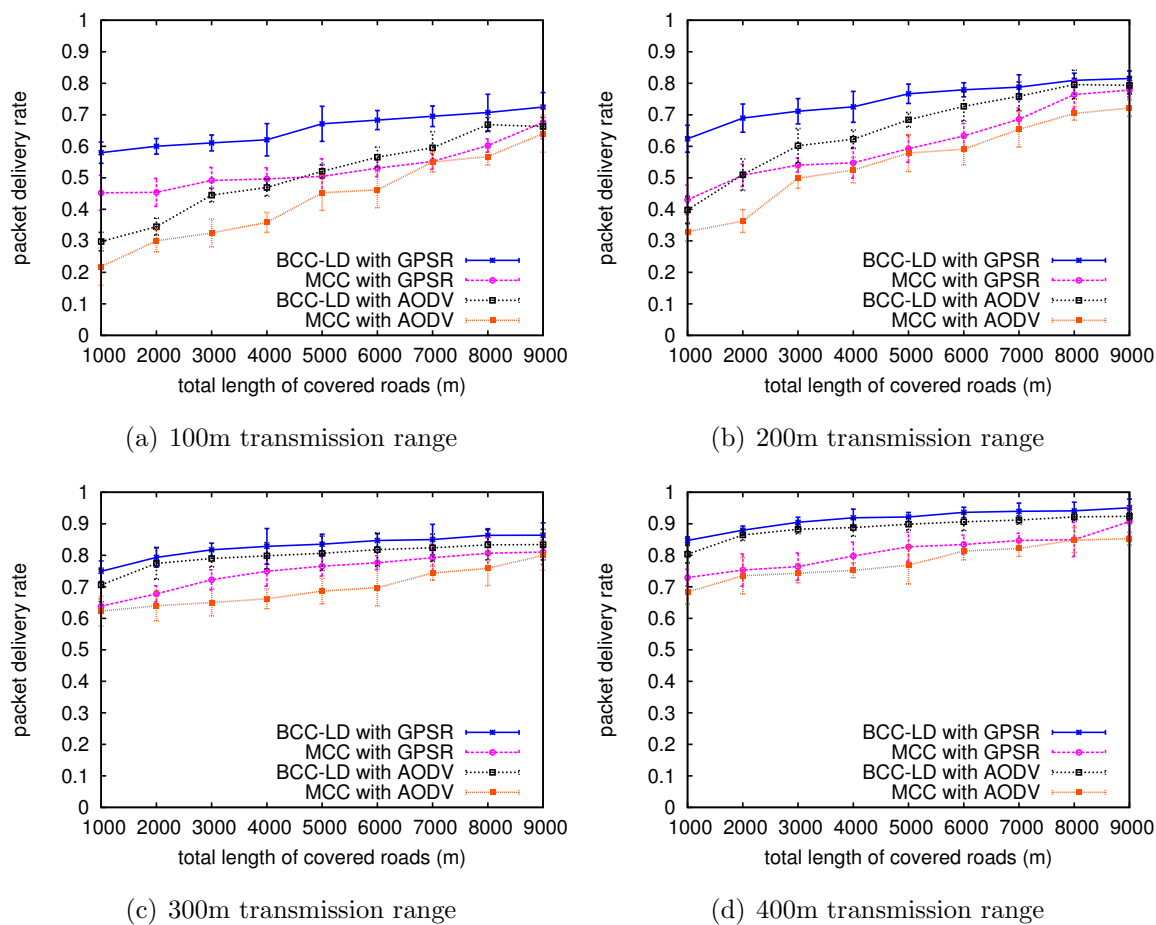


Figure 5.4: Continuous coverage in terms of packet delivery rate

Figure 5.4 evaluates the packet delivery rate of BCC-LD algorithm and MCC algorithm with the GPSR and AODV routing protocols. The simulation results of continuous coverage are similar to the results of sparse coverage. As the RSUs and transmission range increase, the packet delivery rates of the two continuous coverage algorithms grow. The more the RSUs are deployed, the more movements of vehicles can be covered. Therefore, the packet delivery rate appears to be higher and more stable. When the transmission range is small, most of the packet loss is caused by a lack of routing for packets in need of forwarding; this is because both the BCC and MCC provide a relatively low coverage

ratio. However, as the transmission range and the total length of covered paths increase, the packet delivery rates of both continuous coverage algorithms reaches a peak. Under this situation, most of the vehicles have already been covered by RSUs but there are still some packet loss caused by the DROP_IFQ_QFULL problem; this means the queue between Link layer and Mac layer is full so that new packets are dropped. Such a problem happens when the packet flow is larger than the capability of RSUs in some regions.

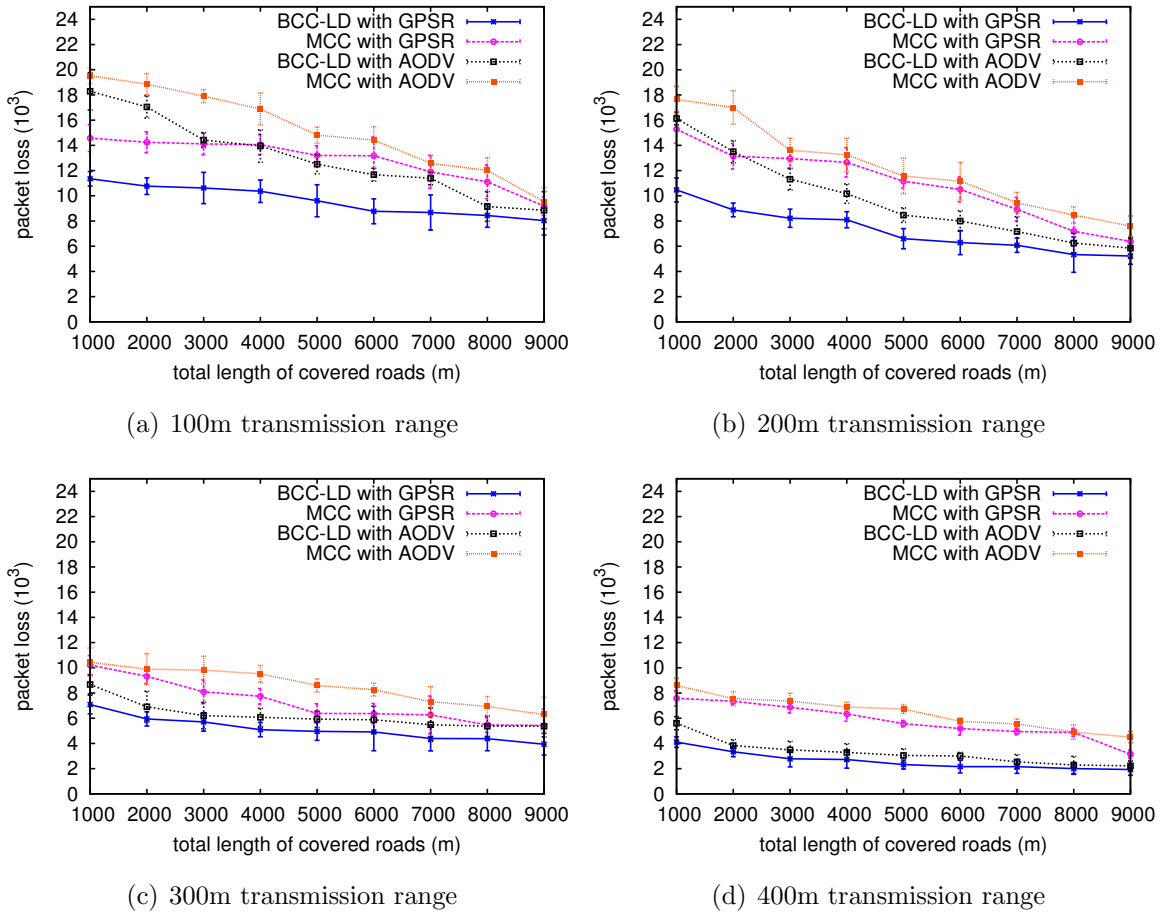


Figure 5.5: Continuous coverage in terms of packet loss

Figure 5.5 shows the performance of BCC-LD algorithm and MCC algorithm in terms of packet loss. It can be found that BCC-LD always performs better than MCC regardless of budget and RSU transmission range. *Coverage value* is a reasonable compromise based on vehicle speed, traffic flow and density. It represents the hidden value of paths better than single traffic flow used in MCC. The *hotspot* scheme also ensures that the most popular accumulating sites of a road system will be covered by RSUs, so that the

budgetary increases does not impact the packet drop rate of a fixed transmission range. Besides, the SMT model used in BCC is also more suitable for the vehicular network scenarios. Since the Steiner tree is the best resource-saving model for connecting *hotspots* in a road network, more vehicle movement can be covered with the RSUs by BCC-LD algorithm when the budget is small. In this way, the packet loss is smaller for BCC than for MCC. But, as the budget increases, the predominance of SMT model is not that obvious, so that the difference in terms of packet loss between BCC and MCC is smaller.

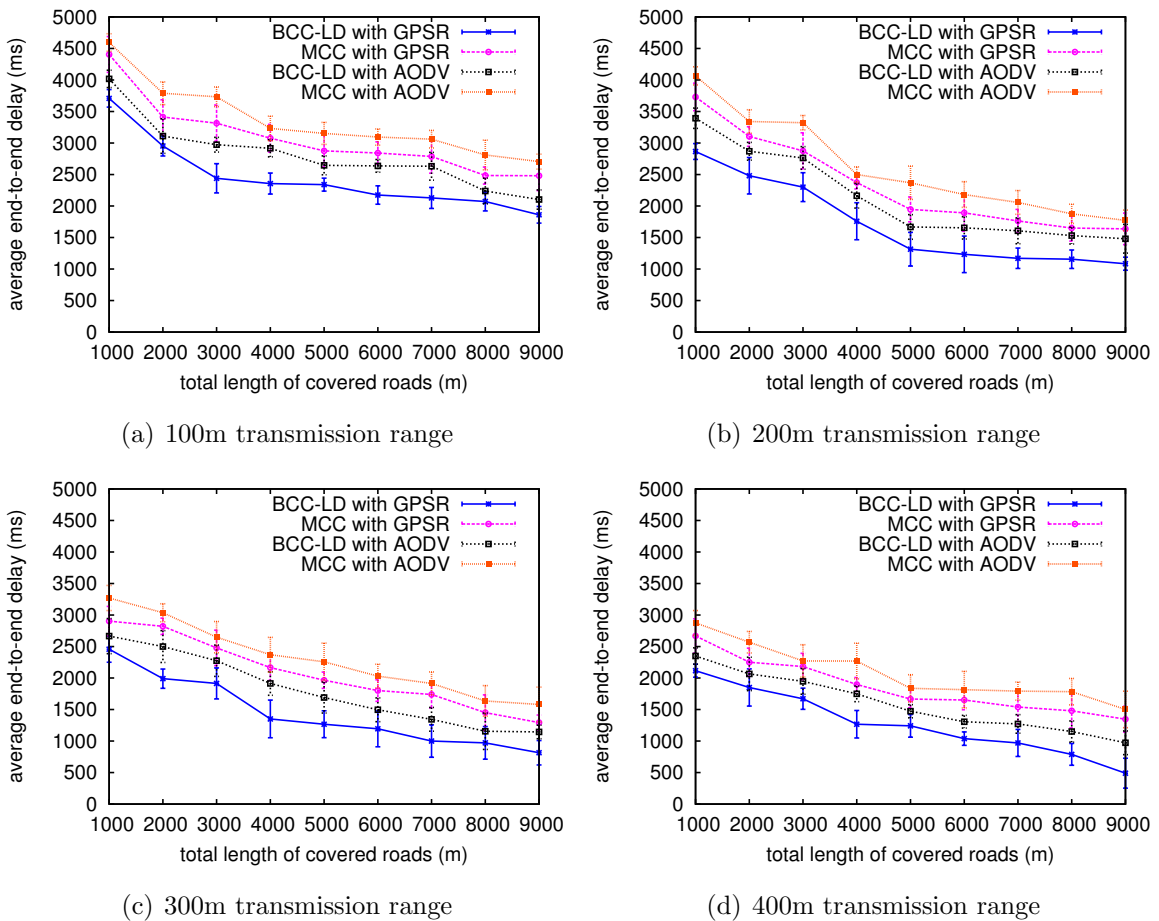


Figure 5.6: Continuous coverage in terms of average end-to-end delay

Figure 5.6 presents the simulation results of two continuous coverage algorithms in terms of average end-to-end delay. The scalability of the two continuous coverage algorithms under GPSR and AODV is also the same as for the simulation of sparse coverage algorithms. Compared with BCC-LD algorithm, the MCC algorithm is more sensitive to the increase of RSUs in terms of end-to-end delay. Generally, the BCC-LD algorithm

performs better than the MCC algorithm. In some cases, there is no difference between BCC-LD algorithm with AODV routing protocol and MCC algorithm with GPSR routing protocol. This is because AODV establishes communication paths with the RREQ message, while the high-speed mobility of vehicles makes the topology information hard to propagate on time. The different paths between source nodes and sink nodes are established, so that routing information can not maintain the latest topology structure to forward packets in a small number of hops. Unlike AODV, GPSR uses geographic information to determine the next hop. In this way, the dynamic topology does not impact the transmission of packets greatly. It can be found from Figure 5.6 that the average-end-to-end delay is much smaller for algorithms under GPSR than AODV.

To sum up, the BCC is more stable and scalable than MCC in our simulation. The packet delivery rate in the BCC-LD algorithm is higher than for the MCC 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.

5.5 Analysis for Sparse Coverage Simulation

To analyze the performance of BSC algorithm in the simulation, we compare the BSC-greedy algorithm with α -coverage algorithm. Since BSC and QSC problems are symmetrical, we only show the result of BSC-greedy algorithm for reference. We select packet delivery rate, packet loss and average end-to-end delay as metrics to measure the quality of communication. The packet delivery rate is a metric calculated by dividing the number of packets received by the target RSUs with 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 the source node to the destination node. To research the effect of our geographic RSU deployment, the simulation was designed to compare two routing protocols: AODV and GPSR. AODV is an on-demand routing protocol for ad hoc networks that uses the shortest path algorithm, while GPSR is a responsive routing protocol that use the proposed location information of vehicles. By comparing the two routing protocols, we can find that our sparse cov-

erage suits different routing schemes and that the quality of V2I communication mainly depends on the RSU deployment rather than on geographic routing.

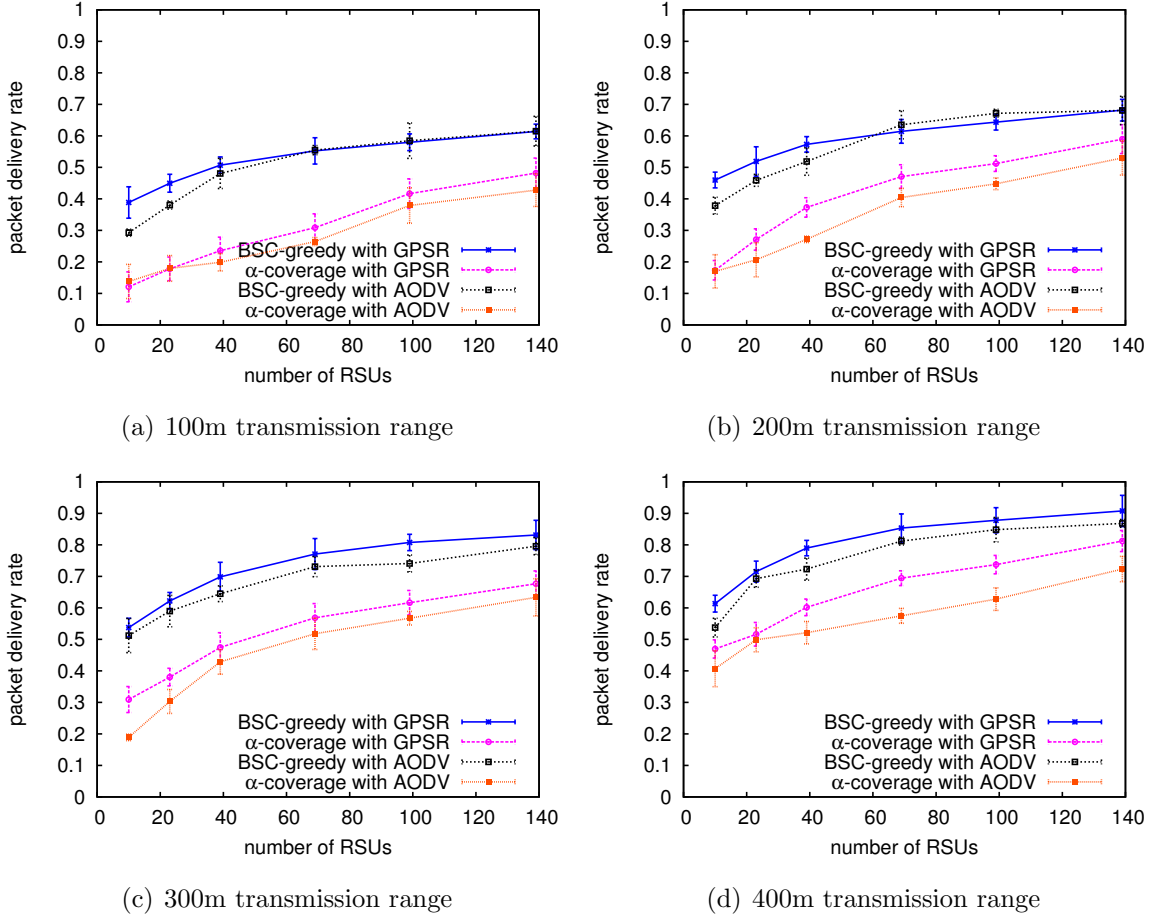


Figure 5.7: Sparse coverage in terms of packet delivery rate

Figure 5.7 represents the packet delivery rate results of two sparse coverage algorithms based on the GPSR and AODV routing protocols. As shown in the figures, the packet delivery rate grows regardless of the transmission range, the type of coverage, and the selection of routing protocols. A common phenomenon is that even though the distance differs between two sparse coverage algorithms as the transmission range increases, the packet delivery rates of both algorithms approach the same trend as the number of RSUs increases. The reason for this is that when there is a small number of RSUs, neither of the two coverage algorithms provides 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.

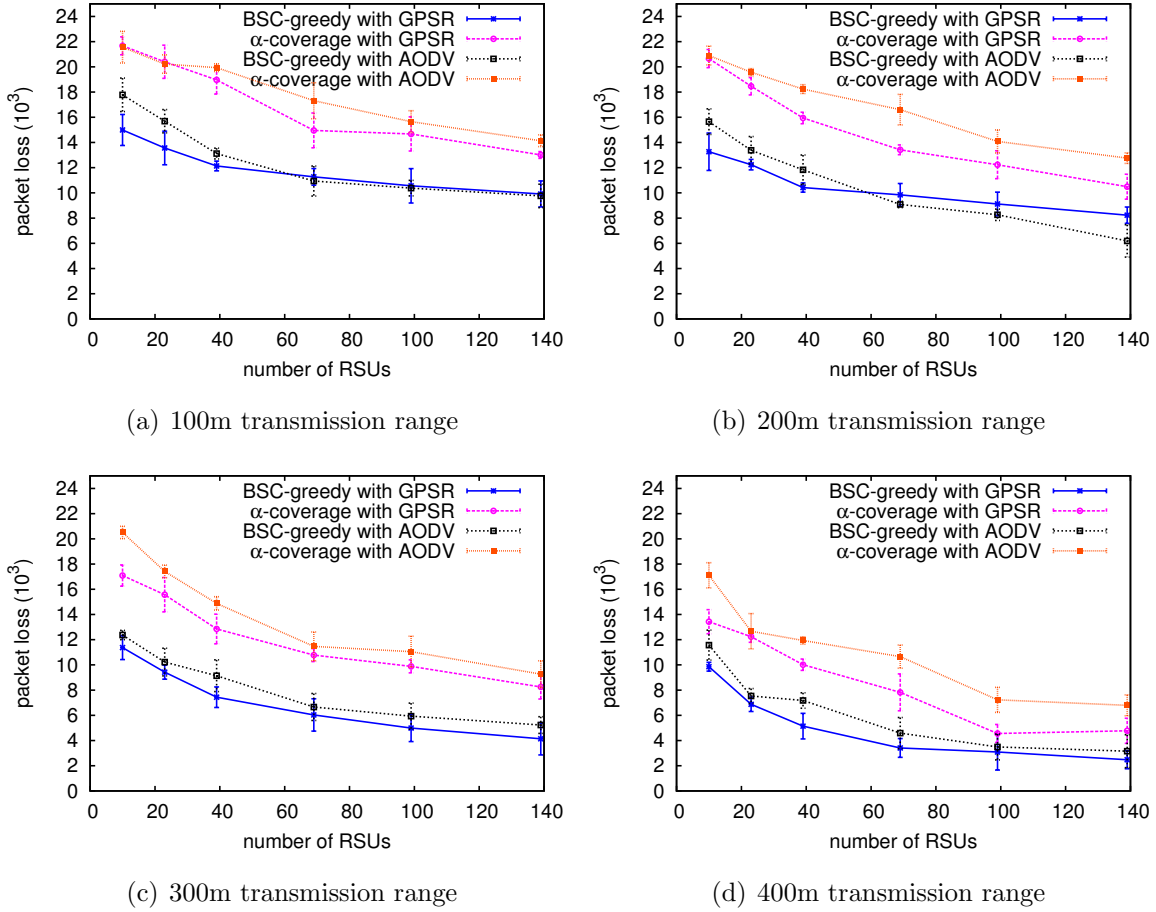


Figure 5.8: Sparse coverage in terms of packet loss

Figure 5.8 shows the packet loss results of two sparse coverage algorithms based on both GPSR and AODV routing protocols. From these figures, we find that both sparse 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 BSC-greedy algorithms always perform better than α -coverage even when the number of RSUs is as low as 10. This is because BSC 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 BSC 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 BSC outperforms the α -coverage. The superiority of the *hotspot* technique is also reflected in the trend of packet loss. Because BSC always picks the most popular sites to cover, the communication quality is as stable as the number of RSUs and the transmission range changes.

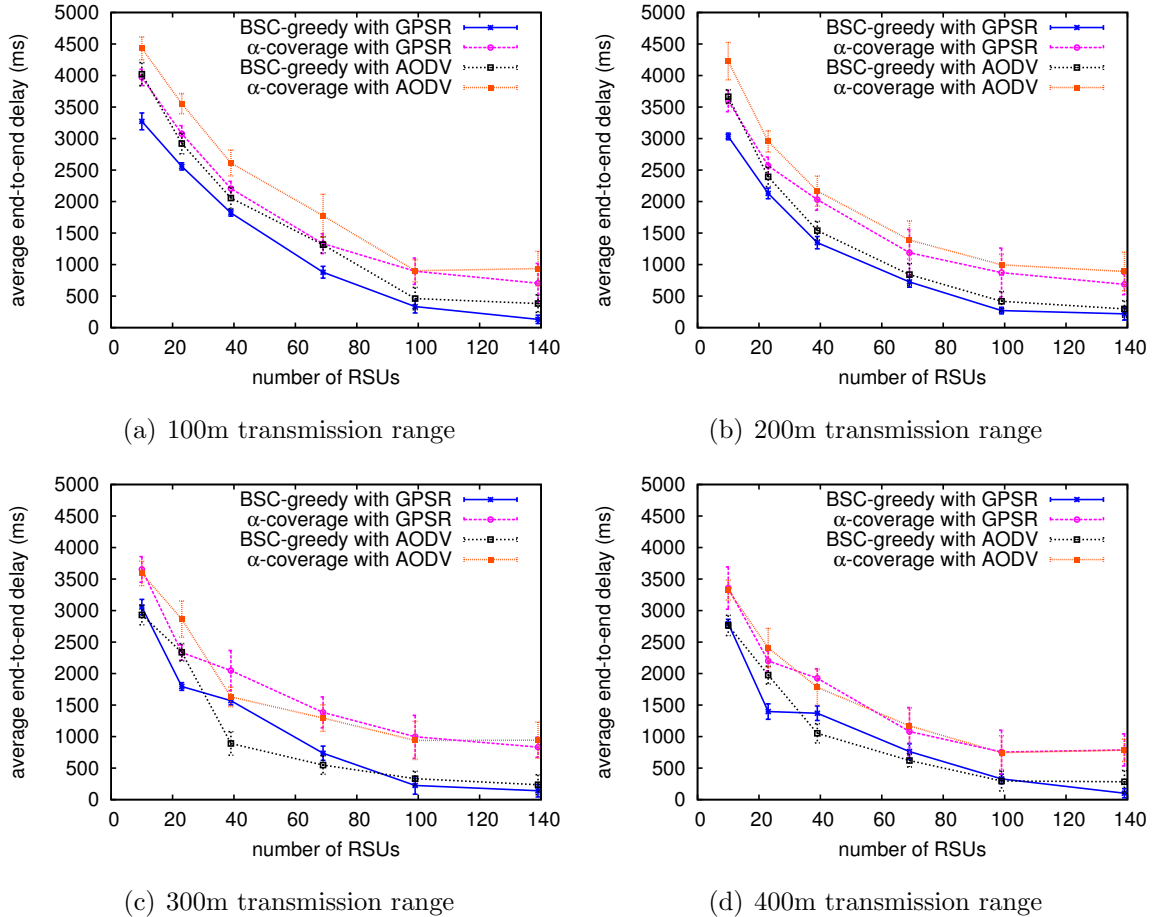


Figure 5.9: Sparse coverage in terms of average end-to-end delay

Figure 5.9 shows the average end-to-end delay of two sparse coverage algorithms based on the GPSR and AODV routing protocols. The end-to-end delay of two sparse coverage algorithms decreases as the number of RSUs increases. The larger the transmission range is, the smaller the end-to-end delay becomes. Even though BSC-greedy algorithm has similar trends with α -coverage, there is still some difference between the two sparse coverage algorithms. When the number of deployed RSUs is very low, both sparse algorithms can reach the highest coverage ratio. 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 level of redundancy in the coverage area occurs. The BSC-greedy algorithm can completely cover the *hotspots* and the marginal regions with the increasing number of RSUs, so that the effect of coverage reaches the peak at the same time.

However, since the α -coverage only deploys RSUs based on spatial attributes, this lack of consideration for vehicle movement could result in useless deployment and a waste of coverage. Thus, the average end-to-end delay of the BSC-greedy algorithm is more stable and scalable as the number of RSUs increases. Besides, the BSC-greedy algorithm provides a good performance guarantee for RSU deployment; this results in a better prediction of the coverage quality than the α -coverage algorithm within a 95% confidence interval.

By comparing Figures 5.7 and 5.8 with Figure 5.9, the difference between 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 in communication quality is caused by the difference in the two routing schemes; in wireless communication, the packet loss mainly occurs due to the end of TTL (Time to Live). If a routing protocol takes more time to find the source-destination path in the routing phase, the lifetime of packets will be shortened and the opportunity to drop the packets will become larger. In our scenarios, AODV takes a great deal of time to maintain the positions of mobile nodes in the routing table, which results in 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 level of 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 level of 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 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 [24]. 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.

5.6 Summary

In this chapter, we first present the methodology and experimental setup for the evaluation of algorithms proposed in the resource-constrained coverage protocol. We use NS2 and SUMO as the network simulator and traffic simulator, respectively. To verify the performance of BCC and BSC, we chose MCC and α -coverage as the baseline algorithms. The simulation results show that our BCC and BSC are suitable and scalable for VANETs under different routing protocols. The comparison of GPSR and AODV also proves that the geometric information-based routing protocol is more stable in high-speed movement scenarios.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we focus on solving four coverage problems in urban vehicular networks: application demands, vehicle mobility patterns, resource constraints, and road geometry. We propose a resource constrained coverage protocol and two types of coverage algorithms to resolve these problems. To meet the demands of applications, we propose continuous coverage and sparse coverage for different network designs. By reducing the BSC and QSC models to the MCP and SCT problems respectively, we designed two approximation algorithms, the Subgradient algorithm and the greedy algorithm, to maximize the quality of coverage while keeping the cost under budget. We also propose a budget estimation scheme to select the most suitable coverage model for a road network.

The mobility pattern of vehicles in a road network is captured by *hotspot* discovery approach used to discover the most popular regions in a road network. The experiments prove that the new metric *coverage value* is a reasonable compromise based on vehicle speed, traffic flow and density. For the resource constraint issues, we formulate two variants of coverage algorithms to suit different objectives: budget and quality. To design a practical RSU deployment based on road geometry, a buffering operation approach was designed in our resource constrained coverage model.

Simulation results reveal that the proposed schemes are stable and scalable in terms of packet delivery rate, packet loss and average end-to-end delay. The quality of coverage is significantly improved due to the effective analysis of mobility pattern and coverage model. The comparisons with MCC and α -coverage prove that our BCC and BSC algorithms perform better than baseline algorithms in urban vehicular scenarios.

6.2 Future Work

This thesis has achieved solid improvements in resource constrained coverage problems by providing effective solutions. In our future work, we plan to extend our research to some open issues and into several other research directions.

- Connectivity and Scheduling Issues in VANETs

RSU deployment problems in VANETs not only concern the quality of coverage. There are other issues that merit consideration, such as connectivity and scheduling. 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.

- Data Mining in Historical Trace Files

Vehicle movement in an urban area provides a huge source of data for data mining for the purpose of building an intelligent sensing system. The *hotspot* discovery approach designed in this thesis is based on such an idea. We use clustering method to mine hidden popular sites in a road network. By using more data mining tools and methods, we can predict the movement of vehicles, discover the potential popular sites of a road network, and even evaluate the trends of a city.

- Derived Applications in VANETs

Based on the efficient coverage algorithms, many derived applications can be proposed to provide assorted services. For example, RSUs can report on the position of vehicles, so that the management system can track the movement of vehicles and discover the vehicles that are in need of assistance. Also, business service providers can provide business information and useful alarms to the drivers.

Bibliography

- [1] IBM ILOG CPLEX Optimizer.
url<http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/>,
Last 2010.
- [2] A. Abdrabou and Weihua Zhuang. Probabilistic delay control and road side unit placement for vehicular ad hoc networks with disrupted connectivity. *Selected Areas in Communications, IEEE Journal on*, 29(1):129–139, 2011.
- [3] B. Aslam, F. Amjad, and C.C. Zou. Optimal roadside units placement in urban areas for vehicular networks. In *Computers and Communications (ISCC), 2012 IEEE Symposium on*, pages 000423–000429, 2012.
- [4] N. Bellomo and M. Delitala. On the mathematical theory of vehicular traffic flow i: Fluid dynamic and kinetic modelling. In *Mathematical Models and Methods in Applied Sciences*, pages 1801–1843, 2002.
- [5] Ayse Can. Tiger/line files in teaching gis. *International Journal of Geographical Information Systems*, 7(6):561–572, 1993. Erratum: *Int. J. of Geographical Information Systems* 8(4): 403.
- [6] Huang Cheng, Xin Fei, Azzedine Boukerche, Abdelhamid Mammeri, and 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, pages 121–128, New York, NY, USA, 2013. ACM.
- [7] Helal Chowdhury, Janne J. Lehtomäki, Juha-Pekka Mäkelä, and Sastri Kota. Data downloading on the sparse coverage-based wireless networks. *J. Electrical and Computer Engineering*, 2010, 2010.

- [8] Paolo Crucitti, Vito Latora, and Sergio Porta. Centrality measures in spatial networks of urban streets. *Physical Review E*, 73(3):036125, 2006.
- [9] Yannick Do, S Buchegger, T Alpcan, and JP Hubaux. Centrality analysis in vehicular ad-hoc networks. *EPFL/T-Labs, Tech. Rep*, 2008.
- [10] Brij Bihari Dubey. Effect of position of fixed infrastructure on data dissemination in vanets. *International Journal of Research and Reviews in Computer Science (IJRRCS)*, 2(2), 2011.
- [11] Martin Ester, Hans peter Kriegel, Jrg S, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.
- [12] GaoJun Fan and ShiYao Jin. Coverage problem in wireless sensor network: A survey. *Journal of Networks*, 5(9):1033–1040, 2010.
- [13] Xiang Fei, Hani S. Mahmassani, and Pamela Murray-Tuite. Vehicular network sensor placement optimization under uncertainty. *Transportation Research Part C: Emerging Technologies*, 29(0):14 – 31, 2013.
- [14] Uriel Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45(4):634–652, July 1998.
- [15] I. Filippini, F. Malandrino, G. Dan, M. Cesana, C. Casetti, and I. Marsh. Non-cooperative rsu deployment in vehicular networks. In *Wireless On-demand Network Systems and Services (WONS), 2012 9th Annual Conference on*, pages 79–82, 2012.
- [16] M. Fiore and J.M. Barcelo-Ordinas. Cooperative download in urban vehicular networks. In *Mobile Adhoc and Sensor Systems, 2009. MASS '09. IEEE 6th International Conference on*, pages 20–29, 2009.
- [17] M. Haklay and P. Weber. Openstreetmap: User-generated street maps. *Pervasive Computing, IEEE*, 7(4):12–18, 2008.
- [18] J. Härri, F. Filali, C. Bonnet, and Marco Fiore. Vanetmobisim: Generating realistic mobility patterns for vanets. In *Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks, VANET '06*, pages 96–97, NY, USA, 2006. ACM.

- [19] Michael Held, Philip Wolfe, and Harlan P. Crowder. Validation of subgradient optimization. *Mathematical Programming*, 6(1):62–88, 1974.
- [20] Dorit S. Hochbaum, editor. *Approximation Algorithms for NP-hard Problems*. PWS Publishing Co., Boston, MA, USA, 1997.
- [21] Dorit S. Hochbaum. Approximation algorithms for np-hard problems. chapter Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems, 94–143. PWS Publishing Co., Boston, MA, USA, 1997.
- [22] Mohamed Kafsi, Panos Papadimitratos, Olivier Dousse, Tansu Alpcan, and Jean-Pierre Hubaux. Vanet connectivity analysis. In *IEEE WORKSHOP ON AUTOMOTIVE NETWORKING AND APPLICATIONS*. IEEE Computer Society, 2008.
- [23] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *Communications Surveys Tutorials, IEEE*, 13(4):584–616, 2011.
- [24] Brad Karp and Hsiang-Tsung Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254. ACM, 2000.
- [25] Ramneek Kaur, , and Ravreet Kaur. Scalable TDB based RSUs deployment in VANETs. *International Journal of Innovation and Applied Studies*, 3(4):1025–1032, August 2013.
- [26] A. Kchiche and F. Kamoun. Access-points deployment for vehicular networks based on group centrality. In *New Technologies, Mobility and Security (NTMS), 2009 3rd International Conference on*, pages 1–6, 2009.
- [27] A. Kchiche and F. Kamoun. Centrality-based access-points deployment for vehicular networks. In *Telecommunications (ICT), 2010 IEEE 17th International Conference on*, pages 700–706, 2010.
- [28] Monika Khatri. An insight overview of issues and challenges in vehicular adhoc network. *Journal of Global Research in Computer Science*, 2(12):47–50, 2012.
- [29] Samir Khuller, Anna Moss, and Joseph (Seffi) Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39 – 45, 1999.

- [30] M. Kim, D. Kotz, and S. Kim. Extracting a mobility model from real user traces. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–13, 2006.
- [31] Krzysztof C. Kiwiel, Torbjörn Larsson, and Per Olov Lindberg. Lagrangian relaxation via ballstep subgradient methods. *Math. Oper. Res.*, 32(3):669–686, 2007.
- [32] L. Kou, G. Markowsky, and L. Berman. A fast algorithm for steiner trees. *Acta Informatica*, 15(2):141–145, 1981.
- [33] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5:128–138, 2012.
- [34] Stefan Krau. *Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics*. PhD thesis, Universität zu Köln, 1998.
- [35] E. Lawler. *Combinatorial optimization - networks and matroids*. Holt, Rinehart and Winston, New York, 1976.
- [36] Junghoon Lee and 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*, pages 196–202, Berlin, Heidelberg, 2010. Springer-Verlag.
- [37] Pan Li, XiaoXia Huang, Yuguang Fang, and Phone Lin. Optimal placement of gateways in vehicular networks. *Vehicular Technology, IEEE Transactions on*, 56(6):3421–3430, 2007.
- [38] Yingsi Liang, Hui Liu, and D. Rajan. Optimal placement and configuration of roadside units in vehicular networks. In *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th*, pages 1–6, 2012.
- [39] Po-Chiang Lin. Optimal roadside unit deployment in vehicle-to-infrastructure communications. In *ITS Telecommunications (ITST), 2012 12th International Conference on*, pages 796–800, 2012.
- [40] Kebin Liu, Minglu Li, Yunhao Liu, Xiang-Yang Li, Minglu Li, and Huadong Ma. Exploring the hidden connectivity in urban vehicular networks. In *Network Protocols (ICNP), 2010 18th IEEE International Conference on*, pages 243–252, 2010.

- [41] Christian Lochert, Björn Scheuermann, Christian Wewetzer, Andreas Luebke, and 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, pages 58–65, NY, USA, 2008. ACM.
- [42] F. Malandrino, C. Casetti, C. Chiasserini, and M. Fiore. Content downloading in vehicular networks: What really matters. In *INFOCOM, 2011 Proceedings IEEE*, pages 426–430, 2011.
- [43] S. Mccanne, S. Floyd, and K. Fall. ns2 (network simulator 2). <http://www-nrg.ee.lbl.gov/ns/>.
- [44] Valery Naumov, Rainer Baumann, and Thomas Gross. An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '06, pages 108–119, New York, NY, USA, 2006. ACM.
- [45] Valery Naumov, Rainer Baumann, and Thomas Gross. An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '06, pages 108–119, New York, NY, USA, 2006. ACM.
- [46] James R. Norris. *Markov chains*. Cambridge series in statistical and probabilistic mathematics. Cambridge University Press, 1998.
- [47] Yukon Department of Highways and Public Works. National road network road segments - yukon territory, canada. September 2008.
- [48] Kohei Okamoto, Kei-ichi Okunuki, and Toshibumi Takai. Sketch map analysis using gis buffer operation. In *Spatial Cognition IV. Reasoning, Action, Interaction*, pages 227–244. Springer, 2005.
- [49] H. M. Ammari R. Mulligan. Coverage in wireless sensor networks: A survey, network protocols and algorithms. *Network Protocols and Algorithms*, 2(2):27–53, 2010.
- [50] Moshe B. Rosenwein and Richard T. Wong. A constrained steiner tree problem. *European Journal of Operational Research*, 81(2):430 – 439, 1995.
- [51] Sok-Ian Sou. A power-saving model for roadside unit deployment in vehicular networks. *Communications Letters, IEEE*, 14(7):623–625, 2010.

- [52] Yipin Sun, Xiaodong Lin, Rongxing Lu, Xuemin Shen, and Jinshu Su. Roadside units deployment for efficient short-time certificate updating in vanets. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1–5, 2010.
- [53] Hiromitsu Takahashi and Akira Matsuyama. An approximate solution for the steiner problem in graphs. *Math. Japonica*, 24(6):573–577, 1980.
- [54] P. Toth. Dynamic programming algorithms for the zero-one knapsack problem. *Computing*, 25(1):29–45, 1980.
- [55] O. Trullols, M. Fiore, C. Casetti, C.F. Chiasserini, and J.M. Barcelo Ordinas. Planning roadside infrastructure for information dissemination in intelligent transportation systems. *Computer Communications*, 33(4):432 – 442, 2010.
- [56] Sheng-Wei Wang and Meng-Yi Chang. Roadside units allocation algorithms for certificate update in vanet environments. In *Communications (APCC), 2011 17th Asia-Pacific Conference on*, pages 472–477, 2011.
- [57] Tian Wang, Weijia Jia, Guoliang Xing, and Minming Li. Exploiting statistical mobility models for efficient wi-fi deployment. *Vehicular Technology, IEEE Transactions on*, 62(1):360–373, 2013.
- [58] L. A. Wolsey. *Integer programming*. Wiley-Interscience, New York, NY, USA, 1998.
- [59] Wendong Wang Yongping Xiong, Jian Ma and Dengbiao Tu. Roadgate: Mobility-centric roadside units deployment for vehicular networks. *International Journal of Distributed Sensor Networks*, 2013:47–50, 2013.
- [60] Sherali Zeadally, Ray Hunt, Yuh-Shyan Chen, Angela Irwin, and Aamir Hassan. Vehicular ad hoc networks (vanets): status, results, and challenges. *Telecommunication Systems*, 50(4):217–241, 2012.
- [61] Zizhan Zheng, P. Sinha, and S. Kumar. Sparse wifi deployment for vehicular internet access with bounded interconnection gap. *Networking, IEEE/ACM Transactions on*, 20(3):956–969, 2012.
- [62] Yanmin Zhu, Youchen Bao, and Bo Li. On maximizing delay-constrained coverage of urban vehicular networks. *Selected Areas in Communications, IEEE Journal on*, 30(4):804–817, 2012.