Combining Capital and Operating Expenditure Costs in Vehicular Roadside Unit Placement

COMBINING CAPITAL AND OPERATING EXPENDITURE COSTS IN VEHICULAR ROADSIDE UNIT PLACEMENT

BY

NABY NIKOOKARAN, M.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2017)	McMaster University
(Electrical & Computer Engineering)	Hamilton, Ontario, Canada

TITLE:	Combining Capital and Operating Expenditure Costs in Vehicular Roadside Unit Placement
AUTHOR:	Naby Nikookaran M.Sc., (Electrical and Computer Engineering) Isfahan University of Technology, Isfahan, Iran
SUPERVISOR:	Dr. Terence D. Todd
CO-SUPERVISOR:	Dr. George Karakostas
NUMBER OF PAGES:	xiv, 154

Abstract

Vehicular ad-hoc networks will be the next step towards intelligent transportation systems. Roadside infrastructure is a key component of these systems that will eventually support various applications such as road safety, transportation services, infotainment, and in-vehicle Internet access.

This thesis considers the problem of roadside unit (RSU) placement and configuration in vehicular networks. The goal is to select the RSU locations and configurations such that the sum of capital and operational expenditure costs is minimized. Historical vehicular traffic traces and a set of RSU candidate locations are used as inputs. First, the problem is formulated as an integer linear program (ILP), which provides a lower bound on the total cost, and can be found for moderate size problems. A practical algorithm called Minimum Cost Route Clustering (MCRC) is then introduced that solves a relaxed version of the ILP and uses a novel rounding procedure to obtain real RSU placements. The algorithm takes into account the energy costs incurred by transmitting vehicular requests when the latter are scheduled using a minimum energy online scheduler. Performance results are presented that show that the proposed algorithm performs well compared to the case where placements are done without considering both capital and operational expenditure costs.

The problem of capacity augmentation is then addressed, as a way of adjusting the initial RSU network design, and serves to counterbalance its failure to take causality into account. The objective is to find an RSU radio capacity assignment that minimizes the long-term operating expenditure costs subject to meeting packet deadline constraints, subject to a given packet loss rate target. A procedure, referred to as the Capacity Augmentation (CA) Algorithm, is proposed that iterates over the RSUs, selecting candidates for capacity augmentation based on their packet loss rate sensitivities. A variety of results is presented that characterize and compare the performance of the CA Algorithm using a greedy online packet scheduler. It is shown that the CA Algorithm is an efficient way to assign RSU radio capacity that achieves the desired performance objectives.

The thesis then considers the problem of RSU job scheduling when vehicle routes are unknown. The objective is to minimize the long-term RSU energy costs, subject to satisfying hard deadline constraints and a packet loss criterion. A scheduler referred to as the Route Coverage Expectation Scheduler (RCES) is proposed that uses the historical traffic traces of an urban road network to estimate vehicular motion and the energy costs needed for RSU-to-vehicle communications. The algorithm schedules job requests across multiple RSUs whenever possible, by assigning part of a request to the current RSU and by deferring the remainder to future RSUs. A variety of results is presented that show that the RCES Algorithm achieves a packet drop ratio similar to that achieved when routes are known, with only a modest increase in energy cost.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Professor Terence D. Todd, for his continuous support of my Ph.D. study and research, for his patience, motivation, and immense knowledge. His guidance helped me with my research and the writing of this thesis. His insight and the critical thought provoking discussions we had were a huge influence on my way of thinking and his attention to detail showed me how quality work is crafted. I would also like to thank Dr. George Karakostas as my co-supervisor for his continuous support of my Ph.D. study and research. His thoughtful advice and insightful criticism had greatly motivated me throughout my Ph.D. program.

I extend my warm thanks to the rest of my supervisory committee and the members of the examining committee for reading my thesis and for their valuable suggestions and comments. I thank my fellow members of the Wireless Networking Group for their support and comments. I would like to thank my parents for their continuous support throughout these years.

Finally, I would like to thank my wife for her patience, understanding and for putting up with late working nights. It would have been impossible without you.

Abbreviations

AHP	Analytic Hierarchy Process
AP	Affinity Propagation
BEH	Balloon Expansion Heuristic
BIP	Binary Integer Program
BRP	Bayesian Route Predictor
CA	Capacity Augmentation
CAPEX	Capital Expenditure
CCH	Control Channel
CEPT	European Conference of Postal and Telecommunications Administrations
CLB	Cooperative Load Balancing
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CWM	Cumulative Weight-based Method

DOT	Department of Transportation
DSRC	Dedicated Short Range Communication
EDF	Earlier-Deadline-First
ETSI	European Telecommunications Standards Institute
FCC	Federal Communications Commission
GMCF	Greedy Minimum Cost Flow
ILP	Integer Linear Program
IoT	Internet of Things
ITS	Intelligent Transportation Systems
LOS	Line-of-Sight
LP	Linear Program
MAC	Medium Access Control
MANET	Mobile Ad-hoc Network
MCCP	Minimum Capital Cost Placement
MCRC	Minimum Cost Route Clustering
ML	Machine Learning
OBU	On-Board Unit
OPEX	Operational Expenditure

PLPR	Primal Linear Programming Relaxation
PMCP	Probabilistic Maximum Coverage Problem
RCES	Route Coverage Expectation Scheduler
RSU	Road-Side Unit
SAM	Service Announcement Message
SoI	Site of Interest
SCH	Service Channel
SUMO	Simulation of Urban MObility
TIS	Topology-aware Intersections Selection
TTL	Time-to-Live
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Vehicle and Vehicle-to-Infrastructure
VANET	Vehicular Ad-hoc Network
WAVE	Wireless Access in Vehicular Environments
WSA	WAVE Service Advertisement
WHO	World Health Organization

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

Equipping vehicles with wireless communication capabilities is expected to be the next step towards Intelligent Transportation Systems (ITS). Vehicular ad-hoc networks (VANETs) will be essential components of this functionality that will help enable future road services. VANETs will eventually support various applications such as road safety, intelligent transportation, location-dependent advertisement, and in-vehicle Internet access. This area has attracted much attention from government, industry, and academia in recent years.

Roadside units (RSUs), which are often deployed at intersections, mainly extend the communication range of VANETs, run safety, and non-safety applications, and provide Internet connectivity to vehicular onboard units (OBUs) as an alternative to cellular-based access technologies. RSU deployment includes two cost components, the capital expenditure (CAPEX) and the operational expenditure (OPEX) costs.

Unlike the solutions focused on minimizing the number of deployed RSUs or minimizing the CAPEX, we consider both of these cost components. This thesis provides a methodology that combines the CAPEX and the OPEX costs in the deployment process.

1.2 Road-Side Unit Placement in VANETs

This thesis focuses on the problem of roadside unit (RSU) placement and configuration in vehicular networks. The goal is to select the RSU locations and configurations such that the sum of CAPEX and OPEX costs is minimized. Our methodology considers two phases. The first is the design of the network itself, which is an offline problem, and occurs before any RSUs are deployed. In the offline design, we take historical vehicular traffic traces including vehicular communication requests and the RSU candidate location information as inputs. The output of the offline phase is an RSU network design, i.e., the set of RSU placements to be made and their (fixed) configurations.

Once the offline RSU network design is completed, the RSUs are installed and subjected to online vehicular traffic flow job requests. When an RSU is deployed and in operation, it incurs long-term operating costs due to its energy use. The objective of the offline design is to choose a subset of the candidate locations such that the sum of CAPEX and OPEX costs is minimized and that vehicular traffic requirements are met. An integer linear program (ILP) is first formulated that computes a minimum total cost RSU placement as a lower bound. The ILP has a prohibitive solution time, even for moderate traffic size instances that makes it impractical for real network designs. A practical algorithm called Minimum Cost Route Clustering (MCRC) is then introduced that solves a relaxed version of the ILP and uses a novel rounding procedure to obtain real RSU placements. The algorithm takes into account the energy costs incurred by transmitting vehicular requests when the latter are scheduled using a minimum energy online scheduler. Performance results are presented that show that the proposed algorithm performs well compared to the case where placements are done without considering both capital and operational expenditure costs.

The problem of capacity augmentation is then addressed, as a way of adjusting the initial RSU network design, and serves to counterbalance its failure to take causality into account. The objective is to find an RSU radio capacity assignment that minimizes the long-term operating expenditure costs subject to meeting packet deadline constraints, subject to a given packet loss rate target. A procedure, referred to as the Capacity Augmentation (CA) Algorithm, is proposed that iterates over the RSUs, selecting candidates for capacity augmentation based on their packet loss rate sensitivities. A variety of results is presented that characterize and compare the performance of the CA Algorithm using a greedy online packet scheduler. It is shown that the CA Algorithm is an efficient way to assign RSU radio capacity that achieves the desired performance objectives.

The thesis then considers the problem of RSU job scheduling when vehicle routes are unknown. The objective is to minimize the long-term RSU energy costs, subject to satisfying hard deadline constraints and a packet loss criterion. A scheduler referred to as the Route Coverage Expectation Scheduler (RCES) is proposed that uses the historical traffic traces of an urban road network to estimate vehicular motion and the energy costs needed for RSU-to-vehicle communications. The algorithm schedules job requests across multiple RSUs whenever possible, by assigning part of a request to the current RSU and by deferring the remainder to future RSUs. A variety of results is presented that show that the RCES Algorithm achieves a packet drop ratio similar to that achieved when routes are known, with only a modest increase in energy cost.

1.3 Thesis Organization

In Chapter 2, the background information related to this thesis is introduced. The role and capabilities of RSUs in VANETs from a standards and application point of view are presented. The necessity of RSU infrastructure from different points of view is discussed. The chapter also provides a comprehensive review of the research efforts related to RSU deployment in VANETs, with emphasis on cost and energy minimization of the deployed network.

In Chapter 3, the problem of RSU placement and configuration in VANETs is presented. The optimization problem is formulated as an ILP, which provides a lower bound on the total cost, and can be found for moderate size problems. MCRC is then introduced, as a practical algorithm. The MCRC Algorithm is compared with RSU placements that minimize only CAPEX, referred to as Minimum Capital Cost Placement (MCCP). The performance of the MCRC Algorithm is investigated in different scenarios that show its advantage in terms of total cost and request drop ratio.

Chapter 4 investigates the problem of capacity augmentation in energy efficient RSU deployments. The CA Algorithm is presented that iterates over the RSUs, selecting candidates for capacity augmentation based on a combination of the RSU loss rate sensitivities and their capacity augmentation costs. A variety of results is presented that show that the CA Algorithm achieves a very significant decrease of the drop ratio with only a very moderate (if any) increase of the network total cost.

In Chapter 5, the problem of RSU job scheduling is discussed when vehicle routes are unknown. The RCES Algorithm is introduced that uses the topology of an urban road network and historical traffic traces to extract vehicular motion statistics. Results show that employing the RCES Algorithm, when vehicle routes are unknown, achieves a drop ratio similar to that achieved when these routes are known, with only a modest increase in energy cost.

The thesis is concluded in Chapter 6 with suggestions for possible future work.

Chapter 2

Background

2.1 Introduction

More than a billion vehicles are currently on the road, worldwide, and this number is expected to surpass 2 billion by the year 2035. This, coupled with new technological improvements in areas such as safety and self-driving, is initiating a new era in transportation systems. On the infrastructure side, cellular-based access technologies are suffering from high costs of service (as a result of high CAPEX and OPEX) and data traffic congestion due to mobile data traffic growth. Roadside infrastructures, based on Dedicated Short Range Communication (DSRC), may provide a more flexible solution to these problems. As a result, VANETs have attracted much attention from government, industry, and academia in recent years.

This chapter presents an overview of previous work related to roadside infrastructures. First, we start with a brief discussion of VANET applications and their unique characteristics and challenges. Then, we overview DSRC features, including Wireless Access in Vehicular Environments (WAVE) standards. Following this, we discuss

VANET infrastructure from different perspectives and RSU deployment strategies.

2.2 Vehicular Ad-hoc Networks

The U.S. Department of Transportation (DOT) estimated that DSRC-based communications between vehicles could reduce up to 82% of all road crashes in the U.S. and about 40% of all crashes occurring at intersections. This can save thousands of lives and billions of dollars (Zheng *et al.*, 2015; Wu *et al.*, 2013). In addition to the waste of energy and the production of greenhouse gasses and other environmentally harmful pollutants, traffic congestion is an obstacle to economic growth (Force, 2012). Thus, the current transportation systems have room for significant improvements regarding safety and efficiency.

VANETs will be essential components of Intelligent Transportation Systems (ITS) that eventually support various applications such as road safety (e.g., collision detection, lane change warning, and cooperative merging), intelligent transportation (e.g., traffic signal control and intelligent traffic scheduling), location-based advertisement and services (e.g., point of interest and route optimization), and in-vehicle Internet access (Lu *et al.*, 2014).

VANET operation can include both vehicular onboard units (i.e., OBUs), and fixed roadside unit infrastructure (i.e., RSUs). The latter is typically installed along the roadside or at intersections where power grid connectivity is common (Al-Sultan *et al.*, 2014). There are three modes of communication between RSUs and OBUs: (i) vehicle-to-vehicle (V2V), where an OBU communicates with other OBUs, (ii) vehicle-to-infrastructure (V2I), where an OBU communicates directly with an RSU, and, (iii) combination of V2V and V2I, where an OBU communicates with an RSU in multi-hop fashion when direct communication is not possible (Lu *et al.*, 2014; Al-Sultan *et al.*, 2014; Cunha *et al.*, 2016).

VANETs have some unique characteristics that differentiate them from a Mobile Ad-hoc Network (MANET). These characteristics include (Al-Sultan *et al.*, 2014; Cunha *et al.*, 2016; Lu *et al.*, 2014; Wu *et al.*, 2013):

1. *Highly dynamic topology*: Vehicle speed, which on average is 50 km/h in urban areas and 100 km/h in highways, creates a highly dynamic environment, in which the lifetime of the link between vehicles is short, especially when vehicles are moving in opposite directions.

2. *Intermittent connectivity*: As a result of the highly dynamic topology, the link connectivity frequently changes. Thus, the link between vehicles can disconnect in the middle of the transmission or before they can start communicating. The transmission power can be increased to prolong the link lifetime, which can degrade the throughput.

3. *High computational ability with no power constraints*: Vehicles can be equipped with plenty of sensors and computational resources, such as processors, memory, and GPS. The vehicle battery can also provide continuous power to the OBUs.

4. *Predictable mobility*: Although vehicles have high mobility patterns, they are constrained by road topology, traffic signs (such as speed limit, traffic light), and other traffic conditions (such as other vehicles, weather, etc.). These constraints can be helpful to predict the position of vehicles.

5. *Propagation model*: The nature of the obstacles in areas where VANET operates, i.e., urban areas, rural areas, and highways, is different. Therefore, the propagation model that works in one environment is not necessarily accurate for others. For example, the free-space propagation model is usually used in highways, but, in urban areas, shadowing, multi-path, and fading effects are very common. Multi-path delay spread causes frequency selectivity, and the vehicle mobility causes Doppler effects and time-selective fading channels. In V2V communications, the line-of-sight (LOS) path can be easily blocked, which may lead to a significant attenuation and packet loss (e.g., buildings at intersections, trucks at highways).

2.3 Wireless Communication in Vehicular Ad-hoc Networks

Recognizing the importance of VANETs, the U.S. Federal Communications Commission (FCC) has allocated a 75 MHz bandwidth to DSRC for ITS radio services (U.S. Federal Communications Commission, 2004) (a 50 MHz bandwidth was also allocated by the European Conference of Postal and Telecommunications Administrations (CEPT) (Campolo and Molinaro, 2013)). DSRC includes the IEEE 802.11p (PHY and MAC layers) and IEEE 1609 (upper layers) standards, which refer to Wireless Access in Vehicular Environments (WAVE) (Lu *et al.*, 2014; Al-Sultan *et al.*, 2014). A similar access layer is defined by the European Telecommunications Standards Institute (ETSI) that also includes 802.11p (Campolo and Molinaro, 2013).

In the U.S., the allocated spectrum is divided into seven channels of 10 MHz

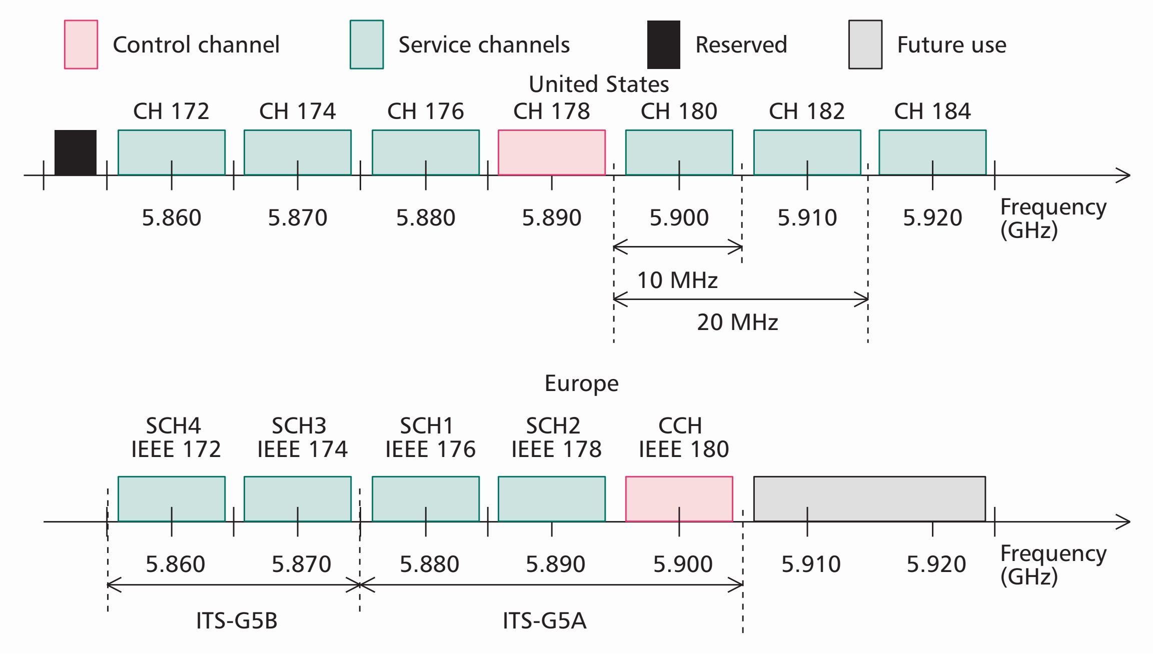


Figure 2.1: Frequency allocation in U.S. (top) and Europe (bottom) (Campolo and Molinaro, 2013).

of bandwidth. The channels are evenly numbered from 172 to 184. Channel 178 is the control channel (CCH), and the rest are service channels (SCHs). Among the six SCHs, channels 172 and 184 are reserved for public safety applications that involve the safety of life and property, including V2V collision avoidance and road intersection collision mitigation. Other channels can be used for safety and non-safety applications (Al-Sultan *et al.*, 2014; Campolo and Molinaro, 2013; Cunha *et al.*, 2016). In Europe, the allocated spectrum is divided into five channels: one CCH and four SCHs. The first two SCHs are road safety services, and the other two SCHs are for general-purpose ITS services. ETSI also allocated 20 MHz for future ITS applications (Campolo and Molinaro, 2013; Cunha *et al.*, 2016). Figure 2.1 shows the frequency allocation in U.S. and Europe.

Although WAVE does not explicitly restrict CCH to any particular traffic type (except for IP packets not allowed on CCH), CCH is considered for the exchange of management information only, including WAVE service advertisements (WSAs) in 1609 and service announcements (SAMs) in the ETSI architecture (Campolo and Molinaro, 2013). However, under some circumstances, CCH is also considered for safety communications (Al-Sultan *et al.*, 2014; Campolo and Molinaro, 2013). WSAs and SAMs are the messages that contain information about the services that are about to be offered and necessary parameters for the users to receive the services (e.g., the selected SCH) (Campolo and Molinaro, 2013).

WAVE allows multi-channel operations of single-radio and dual-radio devices. Single-radio devices can only listen to one frequency at a time, but they can switch between CCH and one of the SCHs to take advantage of provided services on the SCHs. Devices with dual-radio capability can tune to two different frequencies (e.g., CCH and one of the SCHs) simultaneously, but, this capability comes at the higher cost. In fact, single-radio transceivers are considered as a short-term solution for deploying OBUs and dual-radio transceivers as a mid- to long-term solution (Campolo and Molinaro, 2013).

DSRC/WAVE supports an environment in which vehicle speeds can be up to 200 km/h, and the communication range can reach up to 1000m, with a data rate of more than 27 Mbps (Al-Sultan *et al.*, 2014; Lu *et al.*, 2014; Cunha *et al.*, 2016; J. A. Volpe National Transportation Systems Center, 2008). The goal of DSRC/WAVE is to provide a highly reliable communication means with low latency to meet vehicle safety application requirements (Wu *et al.*, 2013; Cunha *et al.*, 2016).

2.4 Roadside Units in Vehicular Ad-hoc Networks

2.4.1 The Need for Roadside Infrastructure

The necessity of an infrastructure for VANETs has been discussed in the literature from different points of view, from connectivity to Internet access.

Connectivity

VANETs were originally developed for V2V communications to allow vehicles to communicate with each other in an ad-hoc manner, either directly or indirectly through the carry-and-forward mechanism with a proper routing protocol. However, the efficiency of V2V communication degrades as vehicle density decreases especially in areas like highways, rural areas, and during the off-peak hours in urban areas. In information dissemination scenarios, for example, vehicles may receive the information after a long delay and some vehicles may never receive it at all. With no infrastructure, the vehicular systems are only limited to a short communication range, with routing issues such as delay and incorrect routing. It has been shown that a minimum support of VANET infrastructure has a significant impact on the overall efficiency of the network (Kchiche and Kamoun, 2014; Silva *et al.*, 2016; Li *et al.*, 2015; Zou *et al.*, 2011; Liang *et al.*, 2012).

Medium Access Control (MAC)

The current medium access control (MAC) protocol in WAVE is based on carrier sense multiple access with collision avoidance (CSMA/CA) in which collisions may occur indefinitely due to the non-deterministic back-off mechanism, particularly in high-speed and high-density environments. Relying on the roadside infrastructure is one of the choices to implement a deterministic MAC protocol for vehicular networks, to guarantee real-time behavior and safety, as users are more likely to trust a network with roadside infrastructure management (Meireles *et al.*, 2016).

Security and Privacy

As in other types of networks, VANETs also face security and privacy challenges, which are intrinsic and unique due to their unique characteristics. Roadside infrastructure can also play an important role in assuring security and privacy preservation of vehicles (Cunha *et al.*, 2016; Sun *et al.*, 2010; Wang and Chang, 2011; Li *et al.*, 2013; Aslam *et al.*, 2016).

Safety and Emergency Services

V2V based communications are crucial to spreading the warning messages to the surrounding vehicles, e.g., safety messages for lane changing, emergency braking, sudden hard braking, and maintaining braking distance (Patil and Gokhale, 2013; Al-Sultan *et al.*, 2014; Lu *et al.*, 2014). Regarding traffic safety, the advantage of having an infrastructure in vehicular networks is twofold. First, the Internet access that is provided through infrastructure can be used by vehicles to communicate with the emergency services immediately. According to the statistics, many deaths occur between the time of the accident and the time of arriving medical assistance on the road. Within the first hour of a car crash, called the golden hour, medical intervention by a specialized team can save lives with the high probability. Hence, any technology that reduces the notification time of an accident will increase the survival chance of injured people. Second, the safety messages can be rebroadcast to more vehicles in less time, especially in low-density areas (Barrachina *et al.*, 2013; Liu *et al.*, 2014, 2016).

Intelligent Transportation System (ITS) related Services

VANETs can support a variety of ITS services such as traffic congestion control, traffic light control, differentiated road pricing and tolling, and disseminating information about road congestion, pavement condition, route status, travel time estimation, air pollution levels, etc. (Rashidi *et al.*, 2012; Abdrabou and Zhuang, 2011; Xiong *et al.*, 2013; Patil and Gokhale, 2013). In a typical vehicular network, vehicles can be equipped with various sensors to collect real-time traffic and environmental data. They can also read any sensor that is installed along the road. This process can be either periodic or event-triggered. All sensor readings will be tagged with a time-stamp and the geographical coordinates, which can be obtained, e.g., from a GPS, and will be reported to a remote control center through an infrastructure. The processed data can be accessed by vehicles through the same infrastructure (Xiong *et al.*, 2013; Rashidi *et al.*, 2012; Abdrabou and Zhuang, 2011).

Information and Entertainment (Infotainment)

VANETs are primarily focused on travel safety and efficiency. However, in order to accelerate the market penetration of DSRC-equipped devices, they also support value-added services such as infotainment (e.g., email, news, streaming, etc.) and commercial applications (e.g., location-based advertising, etc.) (Trullols *et al.*, 2010; Abdrabou and Zhuang, 2011; Xiong *et al.*, 2013; Campolo *et al.*, 2011; Patil and

Gokhale, 2013). Drivers and passengers spend on average 541 and 274 hours per person per year in vehicles in America and Europe, respectively. As the number of mobile Internet applications and social network services increases, accessing to the rich-media contents on the Internet at anytime from anywhere becomes one of the must-have features of VANET (Liu *et al.*, 2013).

2.4.2 Roadside Units as Infrastructure

The full benefits of DSRC technology can only be realized when it is widely adopted by the market. However, justifying the benefits of RSU infrastructure is not easy, especially when existing infrastructures provide both safety and traffic efficiency, as well as Internet access for other ITS applications (Tonguz and Viriyasitavat, 2013). In the case of a car crash, for example, a notification system that is triggered by an event, e.g., airbag release, sends emergency messages to nearby emergency responders. But, in the future, applications may participate in accident prevention by warning the driver or by automatically braking. V2X communications can also warn approaching vehicles of accident locations. Emergency responders can strategically prepare themselves for the accident scene if emergency video streaming is available (Cunha *et al.*, 2016).

Cellular networks are considered to be economically efficient to support V2I communications since they have been widely deployed (Zheng *et al.*, 2015; Kumrai *et al.*, 2014). Using cellular networks to access the Internet inside fast moving vehicles suffers from low bandwidth, high cost, and long delay. RSU infrastructure is another solution that can provide mobile vehicles with high-speed and low-cost Internet access services (Liu *et al.*, 2013). As reported in (Xiong *et al.*, 2013), the feasibility of

RSU-based Internet access for non-interactive applications is confirmed through various experiments. With the emergence of Internet of Things (IoT), requests for higher bandwidth will increase significantly, which makes it almost infeasible for cellular networks to handle vehicular data traffic. In fact, it has been proposed in the literature that RSU infrastructure could be used to offload cellular network data (Bazzi *et al.*, 2015). On the other hand, the ITS services widely rely on the information that vehicles collect and upload to ITS servers, which is currently done through cellular networks (Bazzi *et al.*, 2015). Since vehicles voluntarily participate in sensing and collecting data for ITS purposes, providing free Internet access to vehicles will give them incentives to collect information (Kumrai *et al.*, 2014).

The main functionalities of the RSU infrastructure in VANETs include extending the communication range of the ad-hoc network, running safety and non-safety applications, and providing Internet connectivity for OBUs (Al-Sultan *et al.*, 2014). More specifically, RSUs can allow access to Internet gateways that provide a variety of mobile services as an alternative to cellular-based access technologies (Lu *et al.*, 2014; Liu *et al.*, 2013).

2.5 Roadside Unit Placement in Vehicular Ad-hoc Networks

2.5.1 Assumptions and Challenges

Deployed RSUs have a significant impact on the reliability of VANETs and information exchange (Mehar *et al.*, 2015). In fact, deploying more RSUs can guarantee a better connectivity, coverage and performance (Campolo *et al.*, 2011; Patra *et al.*, 2014). However, installing a sufficient number of RSUs to provide full coverage such that every vehicle can always be connected to at least one RSU leads to large installation and maintenance costs (Abdrabou and Zhuang, 2011; Xiong *et al.*, 2013; Li *et al.*, 2015). A simple RSU, for example, has a CAPEX of $13,000$ to $15,000$ USD per unit, and OPEX of up to $2,400$ USD per unit per year (Tonguz and Viriyasitavat, 2013). The gaps between the coverage areas force OBUs to buffer data until they meet an RSU, which can be a viable solution to delay tolerant applications, but not for safety-critical applications (Rashidi *et al.*, 2012). Ideally, RSUs should be widely deployed to provide continuous coverage or connectivity, however, ensuring such deployment is not possible during the initial stages of VANET deployment because of its prohibitive cost and the lack of market penetration of VANET enabled vehicles (Campolo *et al.*, 2011; Aslam and Zou, 2011; Aslam *et al.*, 2012; Liu *et al.*, 2016). Therefore, it is always preferable to deploy a minimum number of RSUs (Liu *et al.*, 2014). In fact, authorities may limit this number or the maximum budget for deploying RSUs, especially in areas with sparse population (Barrachina *et al.*, 2013). Even in urban areas, covering the whole area may be inefficient and uneconomical (Xie *et al.*, 2013). For example, the interference generated by the dense deployment of RSUs may decrease the performance of message propagation. Also, the redundant deployment of RSUs without proper placement may contribute a little to the performance improvement, while it imposes a huge deployment cost (Tao *et al.*, 2014).

The trade-off between the performance of VANETs and the deployment cost motivated two types of deployment strategies. The first approach focuses on the quality of deployment outcome such as improving the connectivity, coverage, delay, etc., with a limited budget. This is a budget-constrained strategy. The second strategy is the quality-constrained approach, in which the deployment cost is minimized subject to a minimum required quality of service. Some applications require a minimum quality of service to prove their effectiveness, which cannot be compromised by limiting the cost. Therefore, the latter approach seems more appropriate, since it can give the authorities a sense of deployment cost that can guarantee a given minimum essential quality of service.

The effectiveness of VANETs mostly depends on the location of RSUs and their density (Aslam *et al.*, 2012). There are many factors that influence RSU placement decisions, such as the topological and topographical characteristics of road networks, the temporal and spatial traffic characteristics of the roads, present and projected traffic patterns, the availability of the communication and power supply networks along the roads, the variety of services that are emerging and their communication profile, and the requirements of the road operator (Patil and Gokhale, 2013; Rizk *et al.*, 2014).

Of course, VANETs will take some time until they become ubiquitous; during the early stages of deployment, a small fraction of vehicles will be equipped with DSRC devices, and the RSU coverage will be spotty or limited to main streets in urban areas. But eventually, VANETs will enable a broad range of applications (Malandrino *et al.*, 2012; Aslam *et al.*, 2016). Nevertheless, it will still be impractical to densely cover remote and rural areas with RSUs (Abdrabou *et al.*, 2013; Wang *et al.*, 2016).

Another challenge faced when deploying RSUs is the temporal and spatial fluctuations of the vehicular traffic. The population of vehicles on the roads has a non-uniform distribution, both in time (time of day, the day of the week, holidays, etc.) and space (urban and suburban areas, around the highway exits and in between, etc.) (Wu *et al.*, 2012; Liya *et al.*, 2013). Regarding the spatial fluctuations, RSUs are mostly proposed to be installed at locations with high vehicle density, so that more vehicles can benefit, although, if the main goal of deploying RSUs is to reduce the information dissemination delay, deploying them at less dense locations may be more beneficial (Mehar *et al.*, 2015). To tackle the temporal fluctuations, RSUs can be deployed according to the projected traffic peak, the average traffic level, or separately, for each different time period (Wang *et al.*, 2014; Vageesh *et al.*, 2014; Kim *et al.*, 2016). When RSUs are deployed, they continuously work to provide services (Tao *et al.*, 2014); hence, RSUs can be put to sleep whenever it is possible to save energy (Vageesh *et al.*, 2014) or they can be powered by renewable energy sources to save energy costs (Vageesh *et al.*, 2014).

The RSU deployment problem usually starts with a known set of candidate locations. Based on the objective function, a subset of locations will be selected. Potentially, any point on the map can be considered as a candidate location, which can translate to a set of infinite candidate locations. This raises some practical issues, since not every location can accommodate all requirements of RSU installation. The output of the deployment algorithm, for example, could be on private land, or there could be obstructions such as hills or buildings that block the RSUs (Patil and Gokhale, 2013). The candidate locations can be predetermined based on the availability of power sources and backhaul connections, e.g., traffic light poles. However, in general, intersections and road segments can be considered as candidate locations for RSU deployment. Since more candidate locations increase the complexity of RSU placement algorithms, some previous work (e.g., (Trullols *et al.*, 2010; Lee and Kim, 2010; Wang and Chang, 2011; Aslam *et al.*, 2012; Lin, 2012; Chi *et al.*, 2013b,a; Wang *et al.*, 2014; Yan *et al.*, 2014)) limits their candidate locations only to intersections. Whether intersections are better candidates than road segments or not depends on the application. For example, for collection and dissemination of small messages, intersections are better locations than road segments (Trullols *et al.*, 2010; Campolo *et al.*, 2011). However, data dissemination in long road segments may suffer from long delays, which may violate the requirements of the application. To solve this issue, a long road segment can be divided into small segments with their breaking points as RSU candidate locations (Liu *et al.*, 2014; Liang *et al.*, 2012).

RSUs are considered to be interconnected (Aslam *et al.*, 2012), either through wired lines or wireless communication, but not necessarily all RSUs are gateways to the Internet (Li *et al.*, 2015). Obviously, RSUs are connected through direct wireless links only if they are within each other's transmission range. Two RSUs can also be indirectly connected if there is a flow of vehicles that can carry-and-forward the messages from one RSU to another (Chi *et al.*, 2013a,b). To disseminate information when a few RSUs are deployed, V2V communications can be used to extend the RSU coverage. However, the gain that can be achieved through V2V communications strictly depends on the particular cooperation paradigm, which makes it difficult to evaluate in the general case (Trullols *et al.*, 2010; Xie *et al.*, 2013).

There are three types of RSU deployment strategies. The first category does not take into account the vehicle mobility information, and, similar to the base station deployment of cellular networks, the densest locations are targeted. On the other side of the spectrum is the second category, which uses the full historical trajectory of vehicles, i.e., realistic vehicular mobility traces. Since the full knowledge of all vehicle trajectories may lead to a large computational complexity, the third category of deployment strategies relies on traffic patterns, such as the turning probabilities of vehicles at intersections and the migration ratios of vehicles between urban cells, which may be available or can be extracted from real traces (Silva *et al.*, 2016; Xie *et al.*, 2013). Of course, the assumption in the second and the third approaches is that the mobility pattern is similar for long periods, which is true for real-world traffic traces, and, therefore, a deployment strategy based on a history log of vehicle movements will work well in the future (Lee and Kim, 2010).

RSU infrastructure can provide high-speed Internet access to vehicles. However, the connection time between vehicles and RSUs is short, because of their speeds. As a result, a vehicle may not be able to download an entire media file from a single RSU, and therefore, large files are always downloaded through file fragmentation. The location of deployed RSUs, the travel path of vehicles, and the road traffic conditions can affect file downloading in VANETs (Liu *et al.*, 2013). In fact, tracking vehicles is crucial in VANETs for communication protocols as well as applications and services that can benefit from this type of information (Cunha *et al.*, 2016), e.g., energy-aware scheduling algorithms (Hammad *et al.*, 2013; Khezrian *et al.*, 2015). Tracking requires a mechanism to identify the path of a vehicle in the network and predict the next positions, if necessary. Of course, any approach for this purpose should consider the privacy protection of users (Cunha *et al.*, 2016).

2.5.2 Static vs. Mobile RSU Deployment

RSUs are fixed entities installed at intersections and along the road. However, in the initial stages of DSRC deployment, several factors such as cost, complexity, existing systems, and lack of cooperation between government and private sectors, hinders the deployment of RSUs (Tonguz and Viriyasitavat, 2013). As a solution to the low market penetration of RSUs, especially in the early phases of RSU deployment, some work proposes that cars and buses can serve as RSUs.

Reference (Campolo *et al.*, 2011) conducted an experiment to investigate the benefit of mobile RSUs, which concludes that static RSUs are preferable to provide connectivity to vehicles. The mobile RSUs, i.e., special vehicles such as police cars, buses, and trams can offer connectivity services to nearby vehicles, can lead to easy, low-cost, fast, and low power deployment, at the expense of an intermittent connectivity. There should be a large number of moving RSUs to provide the same level of connectivity as static RSUs. Another observation the authors made is that a hybrid scenario, where both static and mobile RSUs provide connectivity services to vehicles, achieves a better performance compared to the scenario with only static RSUs. Reference (Malandrino *et al.*, 2012) investigates the possibility of exploiting parked vehicles to extend the RSU service coverage. The authors focus on content downloading and aim to maximize the freshness of the content as well as the efficiency of the radio resource utilization.

Reference (Tonguz and Viriyasitavat, 2013), on the other hand, focuses only on cars and proposes that certain DSRC-equipped vehicles can play the role of temporary RSUs, i.e., relaying messages to nearby vehicles and acting as a communication bridge for other vehicles. The authors propose a biologically inspired self-organizing network approach. Then, a specific safety application is used to illustrate the feasibility of the proposed solution, which tries to address some key questions, such as the method of selecting vehicles as temporary RSUs, the tasks of temporary RSUs, and the duration of serving as an RSU.

Reference (Kim *et al.*, 2016) considers an RSU deployment strategy with static and mobile RSUs so that their spatiotemporal coverage is maximized under a budget constraint. Three types of RSUs are considered. The first is a static RSU that is deployed at a fixed location. The second is a mobile but uncontrollable RSU that is deployed on public transportation vehicles such as buses or light rails, whose routes are known in advance. The third is a mobile and fully controllable RSU that is deployed on a local government vehicle. The deployment cost of each type is assumed to be fixed and known in advance.

Mobile RSUs may have less deployment cost, but they increase the complexity of the problem. However, similar to sleep scheduling of static RSUs, mobile RSUs can play an important role in tackling temporal fluctuations of the vehicular traffic.

2.5.3 Budget-Constrained RSU Deployment

Deploying RSUs is mainly considered as a way to improve the connectivity problem of V2V communications in VANET (Tonguz and Viriyasitavat, 2013; Mehar *et al.*, 2015; Zou *et al.*, 2011; Patil and Gokhale, 2013; Brahim *et al.*, 2014; Tao *et al.*, 2014), as well as a means to provide Internet access to a variety of ITS based applications and infotainment applications. The solutions that have been introduced in the literature to solve the RSU placement problem mostly focus on the former, i.e., connectivity improvement of V2V communication. Therefore, this group of work uses objectives such as throughput, delay, coverage, vehicle contact with an RSU, and contact time with an RSU, subject to a constraint on the number of available RSUs or the maximum permitted CAPEX cost.

Reference (Wu *et al.*, 2012) considers the RSU placement problem over a highway scenario, where vehicles can communicate with RSUs directly or through multi-hop relaying. The road has multiple lanes and is divided into segments of RSU coverage range. RSUs and vehicles are assumed to have the same transmission range. The objective of this work is to determine the road segments for RSU installation so that the aggregated uplink throughput is maximized subject to a limited deployment budget. Assuming that all RSUs are identical and of the same cost, the total number of deployed RSUs is constrained. As the achievable throughput can be degraded as the number of hops increases, the hop count in each multi-hop relaying path is also limited. By taking the effects of interference, vehicle population, and their speed into account, the achievable data rate and the lifetime of the link are then derived.

Reference (Aslam and Zou, 2011) considers the RSU placement problem over a highway. The highway is divided into equal-length segments of twice the RSU coverage range. For a given number of RSUs, the objective is to find segments to install RSUs, so that the average time of reporting an event to the nearby RSU is minimized. For simplicity, the density and the speed of vehicles are considered constant, the vehicles arrive according to a Poisson process, and two simple event distributions (flat and step) over the road are considered. The authors introduce a heuristic algorithm, called Balloon optimization. Reference (Aslam *et al.*, 2012) focuses on urban areas. Similar to (Aslam and Zou, 2011), the objective is to install a given number of RSUs, so that the average reporting time of an event to an RSU is minimized and a given fraction of roads are covered by RSUs. In this work, only the major roads that carry the majority of the traffic load are considered. Two simple distributions are considered for event/incident distribution. In the first model, roads have different, but flat event distributions. In the second model, not only roads have different distributions, but also events happen more frequently around intersections than in the middle of the road. Each intersection is an RSU candidate location. Vehicles follow a Poisson distribution as they enter each sub-segment of the roads. The possibility of taking a specific route is based on the fraction of the vehicles traveling through that route. The average reporting time over a single path is then relaxed to the average reporting time over the entire region. Therefore, the relaxed problem minimizes the total reporting time over the entire region. The problem is formulated as a binary integer program (BIP) and solved by a branch-and-bound algorithm. Also, a heuristic algorithm is developed, called balloon expansion heuristic (BEH); it iteratively removes RSU candidate locations that do not satisfy certain criteria, one by one, until the objective is achieved.

For a given number of RSUs, transmission range, and overlap ratio, Reference (Lee and Kim, 2010) solves the RSU placement problem by maximizing the possibility that a vehicle can access an RSU. In this scheme, all intersections are considered as initial RSU candidate locations. A history log of vehicles on a real city is used, based on the assumption of daily similarity of mobility patterns. The locations of the vehicles, regardless of their timestamps, are marked as a dot on a map. A vehicle is considered connected to an RSU if it is inside the coverage range. Iteratively, the RSU with the maximum number of dots in its coverage area, which is far enough from previously opened RSUs (according to the allowed overlap ratio) will be opened. This procedure continues until the number of opened RSUs reaches the target value.

Reference (Trullols *et al.*, 2010) formulates the deployment of a given number of RSUs in an urban area as a maximum coverage problem. In an environment for disseminating information, the goal is to find a placement strategy so that the dissemination of information is maximized. The authors argue that the intersections are better locations than road segments for deploying RSUs, by comparing the results of placing a single RSU at the middle of a road segment first and then at an intersection ending the same road. Therefore, they only consider the intersections as RSU candidate locations. The authors divide the problem into two cases. First, it is assumed that the information has a small size and a vehicle just needs to get in contact with an RSU at least once. Therefore, the number of vehicles that come in contact with RSUs will be maximized. In the second case, the contact time of vehicles with RSUs also matters. In this case, both the number of served vehicles and the vehicle-to-RSU contact time will be maximized. For the first case, several algorithms are introduced where one does not require the global knowledge of the road topology, and another deals with the case that the identity of vehicles is unknown. In the second case, the algorithms are adjusted to tackle the additional requirement of the problem, which guarantees a minimum value of contact time for each vehicle.

To protect the privacy of drivers during the planning stage, Reference (Silva *et al.*, 2015) introduces a probabilistic maximum coverage problem (PMCP) for allocating RSUs using only partial mobility information, instead of full knowledge of vehicle trajectories. Partial mobility information consists of the turning probability of vehicles at intersections, or in general term, the probability of migrating vehicles from an urban cell to its adjacent cell. Similar to the work in (Trullols *et al.*, 2010), it is assumed that the information that is about to be collected or disseminated, is small and self-contained. Therefore, the objective is to select a given number of urban cells to install RSUs so that the number of distinct vehicles that come in contact with

RSUs is maximized. Placing RSUs inside the cells is not discussed in this work. A heuristic algorithm based on the work of (Trullols *et al.*, 2010) is introduced, which starts with the cell with the highest vehicle concentration and then iteratively projects the flow of uncovered vehicles between the cells and selects the cells with the highest projection of the vehicle flow, until the given number of RSUs is opened. Reference (Silva *et al.*, 2016) improves the work of (Silva *et al.*, 2015) significantly. Unlike the work of (Silva *et al.*, 2015) that uses only the migration ratios of 4 out of 8 adjacent cells to project the flow of uncovered vehicles, Reference (Silva *et al.*, 2016) considers the migration ratios between all pairs of urban cells.

Reference (Wang *et al.*, 2014) formulates the RSU placement problem as a mobility clustering problem by adopting an affinity propagation (AP) algorithm (Frey and Dueck, 2007). In this work, some of the intersections are considered as RSU candidate locations. For every particular time period, such as rush hours and off-peak hours, the AP algorithm is used to find cluster centers, and, in the end, the union of the cluster centers are the final solution to the RSU deployment problem. The intuition behind this approach is to find intersections with higher influence on adjacent intersections, e.g., intersections with more vehicles passing by, congested intersections with low average velocity, etc. The AP algorithm can also determine the optimal number of clusters and consequently the number of deployed RSUs. However, when the number of deployed RSUs exceeds the budget, the algorithm needs to be re-executed with a different preference initialization.

Reference (Patil and Gokhale, 2013) introduces an RSU placement algorithm based on Voronoi diagrams. For a given number of RSUs, the area covered by RSUs is maximized subject to a maximum allowable delay bound. In this approach, a region is partitioned into polygons centered around RSUs, and the maximum tolerable delay in disseminating data packets from RSUs to vehicles using V2I and V2V communications is used as a criterion to form the polygons. Also, by assigning each vehicle to only one RSU, i.e., the nearest RSU, the number of vehicles served by each RSU is maximized. First, the RSUs will be placed at some initial positions, either randomly or uniformly. Then, the maximum delay that an RSU-generated packet experiences is used to create the RSU neighborhood map in the Voronoi diagram, where the extended range of RSUs defines the contours of the polygons. The neighborhood map depends on the initial placement of the RSUs, which may lead to some overlapped areas as well as some uncovered areas.

In Reference (Rizk *et al.*, 2014), the RSU placement problem is considered as a maximum coverage problem in urban and rural areas. In this work, any site of interest (SoI) can be a candidate location for RSU placement, including intersections, road curves, or any other points that can benefit from installing RSUs. It assumes that the SoIs are initially prioritized and sorted according to their importance and operator preferences. Then, an overlap-based greedy algorithm is introduced, which iteratively selects the next SoI and adds it to the final list if its overlapping ratio with the RSUs already in the final list does not exceed a threshold. This process continues until it reaches the operator stopping criteria. Reference (Makkawi *et al.*, 2015) continues the previous work, by introducing a cumulative weight-based method (CWM) for placing RSUs in urban, rural and mountain areas. Again, it is assumed that a sorted list of SoIs based on operator preferences is given to the RSU placement algorithm. CWM starts by calculating the cumulative weight of SoIs, which is a summation of the weights of the SoI itself and its neighbors. Any two SoIs, whose distance is less than two times the RSU coverage radius, are considered neighbors. SoIs are sorted based on their initial weights in descending order. SoIs with the same initial weight are sorted based on their cumulative weights in descending order. Finally, redundant SoIs are removed. For any two SoIs that have a distance of less than the RSU coverage radius, the SoI that comes later in the sorted list is removed. This process continues until no such removals are possible.

Reference (Cheng *et al.*, 2015) introduces geometry-based sparse coverage protocols for deploying RSUs over an urban area. By providing a new definition for coverage value, the authors aim to capture the spatiotemporal features of the vehicular traffic, i.e., vehicular density, traffic flow and vehicle speed, as a measurement of the importance of covering each urban cell. Then, a revised version of a classical density-based algorithm groups some of the fixed-sized urban cells into clusters, called hot-spots, and removes the rest of the cells. Each hotspot contains the cells that have to be covered. After defining the RSU candidate locations based on the shape and other features of the road network, two formulations of the coverage problem are proposed. In the first formulation, the weighted mean coverage is maximized subject to a budget constraint, while in the second, the deployment cost is minimized subject to a quality constraint. These formulations are then modeled as the budgeted maximum coverage problem and the set cover problem, respectively. Since these problems are symmetrical, they are solved by the same greedy algorithm proposed by the authors. The authors also use a genetic algorithm to solve the RSU placement problem.

For more of these approaches, see References (Campolo *et al.*, 2011; Barrachina *et al.*, 2013; Cavalcante *et al.*, 2012; Xie *et al.*, 2013; Jiang *et al.*, 2014; Brahim *et al.*, 2014; Eftekhari *et al.*, 2015).

2.5.4 Minimizing the Number of Deployed RSUs

Minimizing the number of deployed RSUs subject to performance requirements such as network connectivity and road coverage has been studied in the literature. This is typically done under the assumption that all RSUs are identical, with no attention to each RSU configuration.

Highway Scenario

Reference (Abdrabou and Zhuang, 2011) considers the RSU placement problem on a one-dimensional road network. By maximizing the separation distance between adjacent RSUs on a low-density VANET subject to a maximum vehicle-to-RSU data packet delivery delay with a predetermined delay violation probability, the minimum number of RSUs required to cover the road segment can be estimated. The authors restrict their focus to highways or rural areas with low vehicle density and high speed, where vehicle-to-vehicle and vehicle-to-RSU connectivity are disrupted. The proposed approach is not applicable to cases where no packet relaying is possible, or a multi-hop path from a vehicle to an RSU can be found with a high probability. Vehicles are distributed as Poisson points over the road segment (spatial Poisson distribution), and they travel with two possible speeds between which they alternate. All RSUs have one channel, which allows only one direct communication between each RSU and one of the vehicles inside its coverage area.

Reference (Abdrabou *et al.*, 2013) looks at a different case of the work presented in (Abdrabou and Zhuang, 2011). Vehicles store and carry their packets until they meet an RSU, but, vehicles moving in the opposite direction will receive copies of those packets and will carry them to the nearest RSU, if they encounter one before they leave the road. For simplicity, only the vehicles moving in one of the directions are assumed to be data packet generators, while the vehicles moving in the opposite direction are considered packet carriers. All vehicles moving in the same direction are assumed to have the same constant speed, and the vehicles moving in the opposite direction have a different constant speed. The main objective is to find the maximum separation distance between adjacent RSUs such that a certain required vehicle-to-RSU packet delivery delay is probabilistically satisfied for the first arrived copies of the majority of vehicles packets, and only a small fraction of packets are delivered late or lost.

Reference (Liya *et al.*, 2013) designs a randomized algorithm to find an approximate distance between two consecutive RSUs in a highway that can guarantee the message delivery from any accident site with a given probability and within a certain time. Then, by assuming an equal distance deployment of RSUs, the minimum number of RSUs can be obtained by maximizing the distance between RSUs subject to a given delivery probability. Vehicles are assumed to be traveling on a road consisting of multiple lanes with speeds within a certain range. Vehicles can access an RSU directly or through multi-hop relaying. Vehicle density is also assumed to be constant along the road segment. The randomized algorithm initially starts with two times the transmission range of an RSU, and iteratively increases the distance between two RSUs until the connectivity requirements of the system cannot be satisfied.

Reference (Liu *et al.*, 2016) analyzes the delay of transmitting alert messages along a highway when the clusters of vehicles are disconnected, and the messages should be relayed by other vehicles until they meet an RSU. The relationship between system parameters such as traffic flow density, transmission range, and delay, is derived.

Then, for any given delay bound and by assuming that RSUs can be uniformly distributed along the highway, the optimal number of RSUs is calculated. Vehicles are grouped into clusters, where cluster members can communicate with each other within two hops. Clusters can communicate with each other through different or common gateways, which are cluster nodes located at the end sides of their clusters, or directly through cluster heads, which are cluster nodes reachable by their cluster members within one hop. If adjacent clusters cannot communicate with each other, the messages should be carried by vehicles until they reach the end of the highway or meet an RSU.

Reference (Patra *et al.*, 2014) proposes an analytic hierarchy process (AHP) to solve the RSU placement problem on a highway-like road, and uses expected RSU-to-RSU delay as a performance metric to compare the performance of AHP with uniform and hotspot placement strategies. The goal is to put RSUs as far away from each other as possible, to minimize the number of deployed RSUs, and consequently, to minimize the total cost of deployment. In this approach, a highway-like road with multiple lanes and intersections is divided into equal-length segments, each of which is characterized by its vehicle density, vehicle speed, and event generation rate. The vehicle population follows a Pareto distribution for various densities, and vehicle speeds follow a truncated exponential distribution. The size of the road segments is twice the transmission radius of RSUs and vehicles, and at most one RSU can be placed at the center of each road segment. A message generated by an RSU will be delivered to the next RSU using the carry-and-forward model with the help of vehicles moving in that direction as relay stations. AHP decomposes the problem into a hierarchy of goal, criteria, and alternatives. Then, it chooses the appropriate segment for RSU deployment, according to system parameters such as vehicle density, vehicle speed, and event generation rate, and uses alternative segments for evaluation. AHP iteratively selects the next best segment to place RSU until the expected RSU-to-RSU delay is achieved.

To achieve the minimum number of deployed RSUs in a sparse highway, Reference (Wang *et al.*, 2016) also tries to maximize the distance between RSUs. The authors develop a mathematical model to describe the relationship between the average delivery delay of road condition messages that are randomly generated on a bidirectional road segment and the deployment distance between two neighboring RSUs. This model also takes into account the vehicle speed, the vehicle density, and the likelihood of an incident. A highway with two lanes in opposite directions is considered, where traffic load and connectivity is low. It is assumed that the vehicles moving in the same direction have an inter-vehicle spacing with exponential distribution and a random speed with truncated normal distribution. The information about the random incident that occurs between RSUs, will be collected by the first vehicle arriving at the incident location. The collected information will be delivered to both RSUs directly or indirectly via multi-hop relaying. The average information delivery delay is defined as the average time between the incident and receiving the incident information by both RSUs. The RSUs are considered disconnected; hence, the calculation is based on the last RSU that receives the information. It is argued that in a sparse scenario, the delay of the direct transmission and the delay caused by contention at the MAC layer are much smaller than the delay caused by carry-and-forward of the message, and are simply ignored.

Urban Scenario

In an urban scenario, uniform RSU placement is not always possible. Therefore, their locations are selected from intersections and streets. Nevertheless, the goal is still to install the minimum number of RSUs at a subset of these candidate locations.

Reference (Liu *et al.*, 2014) considers the RSU placement problem in an urban area with the objective of finding the minimum number of RSUs to deploy so that the alert messages would be propagated to RSUs within a delay bound. An alert message can be delivered to an RSU through direct communication or through a carry-and-forward model. For simplicity, it is assumed that each road segment has a constant vehicle density for the entire segment and that the forwarding period is constant for all the forwarded messages. Therefore, if the vehicle density is not enough, the alert message cannot be forwarded, and it will be carried until the vehicle reaches an RSU. Primarily, intersections are considered to be candidate locations for deploying RSUs, although any long road whose transmission delay is larger than the delay bound, is segmented and its cut-off points are considered as intersections as well. If an alert message from any point of a road can be successfully delivered to an RSU within the given delay bound, the road is considered to be covered by that RSU. The goal is to cover all the roads with a minimum number of RSUs, and is formulated as a classic set cover problem. The problem is solved by integer linear programming (ILP) and a heuristic greedy algorithm.

Reference (Chi *et al.*, 2013b) introduces an intersection-priority-based RSU placement methodology to gather the traffic data from intersections. The authors seek the optimal number and positions of RSUs so that all intersections are covered, and RSUs are connected to each other, while the number of RSUs is minimized. An intersection is considered covered by an RSU if it is within the transmission range of that RSU. The intersection priority concept is introduced as a measure of intersection importance in the process of deploying RSUs. It is a weighted summation of traffic factors such as vehicle density, intersection popularity, etc. Three heuristic algorithms are then proposed: greedy, dynamic, and hybrid. Initially, all intersections are considered as candidate locations for RSU deployment. The greedy algorithm iteratively places an RSU at the intersection with the highest priority and removes all intersections within the transmission range of the RSU from the list of RSU candidate locations until all intersections are covered. In the dynamic approach, however, RSUs are evenly distributed so that the overlapped area will be minimized. Finally, the hybrid approach combines the two approaches.

Reference (Chi *et al.*, 2013a) provides an RSU placement algorithm based on the concept of intersection-connectivity between intersections. This is defined for two intersections based on the number of vehicles passing through both, i.e., by the probability that the traffic information from one intersection can be carried-and-forwarded to the other intersection by those vehicles. The connectivity between each intersection pair is approximated by the average traffic volume on the path between the two intersections. If there are more than one path, the path with the highest connectivity is considered. The proposed RSU placement algorithm starts with the hybrid algorithm introduced in (Chi *et al.*, 2013b) to find the initial RSU candidate locations according to the intersection priorities, where all intersections are covered, and the overlapped coverage is minimized. Then, the RSU connectivity network is constructed. In this network, two RSUs are connected either directly, if they are within transmission range, or indirectly, through the intersection-connectivity of the intersections where RSUs are installed. Then, iteratively, the RSU with the minimal effect on the RSU connectivity and coverage will be removed, if the remaining RSU connectivity and coverage is still bigger than a given threshold.

RSUs play a major role in providing services to vehicles while preserving their privacy. Reference (Wang and Chang, 2011) considers the RSU placement problem with the objective of minimizing the number of deployed RSUs in the city, so that the issued certificates can be updated before they expire on all driving routes. The driving route between an origin and a destination is determined by the navigation system based on the status of each road. It is assumed that the driving time on each route is known in advance. It is also assumed that the RSUs can only be deployed at intersections. Hence, the driving time of each road should be less than or equal to the length of the valid certificate interval. Three heuristic algorithms are introduced. *The most driving routes first method* sorts the intersections in descending order of the number of driving routes passing through the intersections, and places RSUs at the intersections one by one, until the certificate can be updated in time on all driving routes. *The most satisfied intersection pairs first method* iteratively places the next RSU at the intersection with the maximum number of new origin-destination pairs that can benefit from the newly deployed RSU in terms of successful certificate update, until the certificates on all driving routes can be updated successfully. *The critical intersections first method* places RSUs at the critical intersections first, and then, continues with the most satisfied intersection pairs first method, until the certificate on all driving routes can be updated successfully. An intersection that connects two consecutive roads with the total driving time of more than the valid certificate interval is considered critical.

Reference (Kumrai *et al.*, 2014) formulates the RSU placement problem as a two-objective optimization problem that minimizes both the number of deployed RSUs and the fraction of areas with no coverage. A genetic-based heuristic algorithm is then introduced that seeks the Pareto-optimal RSU positions.

Considering RSUs as a means of providing high-speed Internet access to vehicles, Reference (Liu *et al.*, 2013) proposes an RSU deployment strategy for file-downloading that guarantees the file downloading success ratio and delay requirements with the lowest RSU deployment cost. Assuming the randomness of RSU deployment and road traffic conditions, a vehicle state, i.e., being inside the coverage area of an RSU or non-coverage area, is modeled as a time continuous homogeneous two-state Markov chain. Then, the relationship between the density of deployed RSUs and the probability of successfully downloading a file within a satisfying delay is derived. The RSU deployment problem maximizes the distance between deployed RSUs, subject to a satisfactory file downloading success ratio; its solution gives a mean non-coverage length for different values of mean coverage length. To deploy RSUs in an urban area, the urban road network is first modeled as a weighted graph, where intersections and streets are nodes and edges, respectively. The weight of an edge is the average passing time of the corresponding street. Then, a heuristic algorithm based on the depth-first traversal of the graph is designed, which traverses its edges and alternatively generates random coverage and non-coverage lengths with the mean values calculated from the solution of the RSU deployment problem.

To disseminate sensitive information, such as safety information, Reference (Yan *et al.*, 2014) aims at finding the intersections to install access points so that all the vehicles driving in the area will pass by at least one of the selected intersections. The authors introduce the topology-aware intersections selection (TIS) problem, whose objective is to select a minimum number of intersections so that the selected intersections intersect every possible path taken by drivers. Each eligible path includes at least two intersections. If the geographic area can be presented as a planar graph, the solution to the TIS problem is proved to be the solution to the vertex cover problem. A class of heuristics is then proposed. They transform a planar graph into a bipartite graph, solves the vertex cover problem by converting it to a maximum matching problem, and obtains the exact solution by applying an existing algorithm, such as Hopcroft-Karp. Then, the graph is converted back to the original planar graph, and the solution is adjusted accordingly. RSUs will be deployed at the selected intersections.

Reference (Xiong *et al.*, 2013) proposes an RSU deployment strategy in which the number of deployed RSUs is minimized while guaranteeing a predetermined vehicle-to-RSU contact probability. The contact probability is defined as the probability of a vehicle entering the communication range of at least one RSU within a given traveled distance after entering the region. The authors argue that there is a time-stable statistical mobility pattern in actual vehicular traces. By dividing the area into a set of non-overlapped uniform zones, the transition probabilities of vehicles between zones within time units of 20 minutes are estimated from an actual trace. A mobility graph is then formed with the zones as the nodes and the transition probabilities as the weights of the edges between neighboring zones. The mobility graph is then simplified by removing the useless vertices and the edges with very small transition probabilities. For simplicity, it is assumed that each zone can be covered by the transmission range of an RSU. Then, the RSU placement problem is transformed into the minimum vertex subset selection problem, and a heuristic greedy algorithm, called RoadGate, is introduced. RoadGate greedily searches for the vertex that can maximize the number of vertices that can reach this vertex with a minimum predefined probability. At each iteration, the vertices that benefit from the selected vertex will be removed, and the next vertex will be selected from the remaining vertices.

Reference (Hu *et al.*, 2016) considers the RSU rental problem for data dissemination to vehicles. In the RSU rental problem, RSUs are already deployed, but the provider wants to rent the minimum number of the RSUs to cover all of the vehicles, while the probability of each vehicle successfully receiving data is no less than a threshold. A graph model is first formed to describe the moving pattern of vehicles. Vehicles and deployed RSUs are vertices of this graph. Each vehicle is only connected to the RSUs that it passes by, and the weights of these edges are equal to the probability of receiving data from these RSUs successfully. It is assumed that the probability values can be derived from the history records. A greedy heuristic algorithm is then proposed, which iteratively selects the RSU that has the maximum effect on both the coverage of more new vehicles and the probability increase of receiving data successfully, until all vehicles are covered with a minimum required probability for receiving data successfully.

2.5.5 RSU Placement and RSU Configuration

Similar to cellular network design, the RSU deployment problem is a heterogeneous network design. That is, RSUs can have different settings and configurations, such as coverage range, power level, power source, antenna type, channel capacity, etc. These have to be determined during the planning phase of the network. Different frameworks and approaches have been introduced in the literature to tackle these issues; from single to multiple configurations for each candidate location.

To collect/disseminate information using V2I communication, Reference (Lin, 2012) studies the RSU placement problem with the objective of minimizing the deployment cost, subject to covering all intersections. The problem is formulated as a binary integer programming problem and is solved by a branch-and-bound algorithm. All intersections are considered as RSU candidate locations. If an intersection is not suitable for placing an RSU, a large installation cost is used in the objective function. Each candidate location is assumed to have a different deployment cost based on the location and the backhaul connection (i.e., one configuration for each location). All intersections within the transmission range of an RSU are considered covered, and each intersection should be covered by at least one RSU.

Reference (Liang *et al.*, 2012) formulates the problem of RSU placement and selecting their configurations, i.e., power level and antenna type, as an integer linear program (ILP). The remaining configuration settings, such as backhaul connectivity type can be optimized for each RSU individually, based on available resources and the overall cost. The total deployment cost is minimized subject to covering a minimum desired percentage of streets with limited multi-hop packet relaying. Setting the maximum multi-hop relaying to zero results in a purely I2V network. For simplicity, all intersections are assumed to be RSU candidate locations. Long streets are divided into short segments, whose end points are also considered RSU candidate locations. Multiple temporal traffic realizations are considered that can represent different times of the day or week and have different coverage requirements. A set of intersections with high accident rate is defined with a 100% coverage requirement to incorporate spatial traffic conditions.

Reference (Li *et al.*, 2015) considers the RSU placement problem where emergency messages are disseminated to all vehicles on the road through RSUs if vehicles are within RSU transmission range, or by a carry-and-forward method otherwise. The transmission time of a message is considered negligible. A set of candidate locations is given with two types of RSUs, at most one of which can be installed at each location. The first type is connected to an information center via wire with a larger communication range, but with a higher cost. The second type is connected to other RSUs through wireless communication as an extension of the first type and only disseminates the messages it receives from other RSUs; hence, it should be placed within the transmission range of at least another RSU. For a given delay bound, the objective is to find an optimal placement of the RSUs such that the total deployment cost is minimized and the emergency message can be received by all vehicles within the delay bound. First, a greedy algorithm is introduced that iteratively selects an RSU with the minimum per newly covered road segment cost. If the new RSU is outside the transmission range of already placed RSUs, its type is limited to the first type. A two-stage algorithm is then proposed, which places the first type of RSU during the first stage, by employing the greedy algorithm. When the percentage of the covered roads reaches a threshold value, the second stage of the algorithm starts by placing the second type of RSU until all road segments are covered.

Reference (Song *et al.*, 2015) proposes a hierarchical RSU network architecture composed of three layers/types of RSU with different coverage radii. The problem is formulated as an integer linear program (ILP), which minimizes the total deployment cost subject to satisfying the coverage and minimum data rate requirements, and takes radiation constraints into consideration. A set of candidate locations, where uninterrupted power can be provided, is given. A set of test points is defined to measure the achievable data rate and radiation intensity. Vehicles are assumed to be uniformly distributed with uniform traffic demand. Each test point is assigned to exactly one RSU. A ray-tracing technique is adopted for RSU coverage evaluation, and since the focus is on the long-term planning and design, small-scale fading is not considered. In addition to solving the ILP directly, a heuristic algorithm is also introduced, which starts by sorting the RSU types in descending order of their coverage. Using the second type of RSU, test points will be covered, and other types are only used to improve the coverage.

To overcome the shortcomings of VANETs, Reference (Mehar *et al.*, 2015) proposes an RSU placement strategy to improve the network connectivity for delay-sensitive applications. The goal is to select the best RSU locations so that the combination of the total RSU deployment cost and the total message delivery delay to the nearest RSU is minimized, and the message delivery delay from any road segment to the closest RSU is less than a given threshold. Different candidate locations can have different RSU installation costs. A heuristic algorithm is introduced to solve the optimization problem. Traffic information, such as density and speed, is used to find the RSU candidate locations by calculating the connectivity probability. All roads are divided into equal-length segments with the length of twice the RSU transmission range. The midpoints of the road segments with low connectivity probability are considered as candidate locations, because, in road segments with a high probability of connectivity, V2V communication can be used to exchange information. Dijkstra's algorithm is used to find the delivery delay from any point on a road segment to the closest RSU, and a genetic algorithm is used to find the final deployment positions. The vehicle density changes over time and from one road segment to another.

2.5.6 RSU Placement and OPEX

Deploying RSUs is a costly process, but, costs are not limited to CAPEX only. The operating cost of RSUs is high because of their continuous activity, providing services (Tao *et al.*, 2014). Some previous work deals with this issue.

Reference (Zhang *et al.*, 2015) studies the energy-efficient RSU placement problem on a winding road by minimizing the RSU transmission power, so that all vehicles on the road are covered. The road is approximated with a predefined number of continuous ellipses to represent the possible locations of vehicles, and a given number of RSUs with the same transmission power should be deployed to cover the road. The problem is formulated as a non-convex problem, which is then converted to solving a convex problem using linear approximation, an S-procedure, and semi-definite relaxation.

Reference (Vageesh *et al.*, 2014) considers the energy-efficient RSU placement strategy in which RSUs will be scheduled into a sleep mode for a fixed interval to save energy. The energy consumption of the RSUs is only limited to the uplink direction, i.e., the energy consumed to receive data packets, a common case for data collection. Vehicles are assumed to be distributed according to a Poisson distribution along a one-dimensional two-way road. Also, vehicles arrive at the road according to a Poisson process, and the average vehicle density is constant. A packet generated by a vehicle can be communicated to an RSU either directly or by multi-hop relaying with a limited hop count due to a delay bound. A set of candidate locations is considered, where the distance between every two adjacent locations is at least twice the transmission range of an RSU. All RSUs are powered by the grid and solar energy sources, although the former is only used when there is not enough battery energy available. According to historical traffic data and its variation over time, a static schedule is computed to put RSUs into sleep mode. Time is divided into equal-length time slots each of which captures a particular road traffic condition. For every time slot, a subset of candidate locations for RSU deployment is determined, so that a minimum required packet delivery ratio can be achieved. Any candidate location where an RSU is required for at least one time-slot is finally selected for deployment. The problem is formulated as a multi-objective optimization problem, which minimizes (i) the number of deployed RSUs, (ii) the OPEX incurred by the energy consumption of RSUs, and (iii) the fraction of grid energy consumed to the total energy consumption of the RSUs. Minimizing the OPEX is considered a primary objective, and a heuristic algorithm is used to solve the optimization problem for every time slot. Solving the optimization problem for all time slots gives the locations to deploy RSUs and their sleep schedules.

Chapter 3

RSU Placement and Configuration in VANET

3.1 Introduction

Vehicular ad hoc network (VANET) performance is clearly dependent on the RSU deployment strategy and must take into account the services that will eventually be supported. Most first generation VANETs will primarily focus on enabling vehicular safety applications; however, in time, they will also support ITS-related services as well as Internet access for rich-media streaming applications. The bandwidth requirements for collecting and disseminating safety messages is very small compared to that available, although this type of application can have very stringent delay response requirements. On the other hand, many non-safety applications may be delay tolerant (e.g., on the order of one or two minutes), but, they may require much higher bandwidth for relatively longer durations (Malandrino *et al.*, 2014). It is clear that a network that has been tailored to accommodate one type of application may obviously perform poorly when subjected to a range of different services. For this reason, considering the diversity of applications and their performance requirements is of utmost importance in the network planning phase.

Planning any infrastructure requires an upfront investment, i.e., capital expenditure (CAPEX) costs. Afterwards, the designed network is subject to various maintenance and operating costs, i.e., the operational expenditure (OPEX) components. Obviously, deploying more RSUs leads to higher CAPEX costs, however, it is not always true that a deployment with a higher number of RSUs will lead to increased OPEX costs. In fact, there can be a trade-off between these two cost components. For example, decreasing the number of deployed RSUs and then increasing their coverage range may reduce CAPEX costs, but, this may clearly increase longer-term OPEX costs (Zhang *et al.*, 2015). The OPEX costs of deployed RSUs can clearly be very high, since they will constantly accumulate while services are being provided (Tao *et al.*, 2014). Conventional RSU deployment strategies typically assume that CAPEX is the only prohibitive cost component in RSU deployment. For this reason, minimizing the number of RSUs is usually the approach taken.

In this chapter, we consider the problem of RSU placement that minimizes the sum of capital expenditure and operating expenditure costs. Our methodology considers two phases of the RSU facility location problem. The first is the design of the network itself, which is an offline problem, and occurs before any RSUs are deployed. In the offline design, we take historical vehicular traffic traces and the RSU candidate location information as inputs. The sample functions include the associated vehicular traffic communication requests, which may be delay tolerant, i.e., each having an associated time deadline. The road description input also identifies candidate locations where RSUs can be placed, and each candidate location has an associated installation (CAPEX) cost. The output of the offline phase is an RSU network design, i.e., the set of RSU placements to be made and their (fixed) configurations. This is done before any RSU is ever deployed.

Once the offline RSU network design is completed, the RSUs are installed and subjected to online vehicular traffic flow job requests. In this case, the vehicular traffic demands must be processed by the system in a causal fashion, as would be the case in a deployed network. Once an RSU is deployed and in operation, it incurs long-term operating (OPEX) costs due to its energy use. The objective of the offline design is to choose a subset of the candidate locations such that the sum of CAPEX and OPEX costs are minimized such that vehicular traffic requirements are met.

The main contributions of this work are summarized below.

1. To the best of our knowledge, this is the first work that focuses on minimizing the combined capital and operating expenditure costs. The total RSU cost includes both that of RSU installation, i.e., CAPEX, and long-term energy operating, i.e., OPEX, components. This combination affects both the initial placement costs of the RSUs, and their associated long-term operating costs.

2. An integer linear program (ILP) is formulated that computes a minimum total cost RSU placement. The ILP has a prohibitive solution time, even for moderate traffic size instances that make it impractical for real network designs.

3. A novel RSU placement algorithm is introduced, referred to as Minimum Cost Route Clustering (MCRC). MCRC is obtained by solving the LP relaxation of the ILP, and using a rounding procedure to obtain RSU placements based

on the approximation algorithm of (Levi *et al.*, 2012) for Capacitated Facility Location problems. MCRC is efficient and can be used for large scale problems.

4. Results are presented that show that a conventional RSU placement that minimizes the total number of deployed RSUs, for example, may result in significantly higher operating costs in the long-term. It is therefore natural to study the placement of RSUs that minimize the combined CAPEX and OPEX costs. A variety of performance results are presented that show that the MCRC Algorithm outperforms RSU placements that directly solve the ILP, but minimize only CAPEX expenditures. The results demonstrate the inherent inefficiency introduced by considering only CAPEX costs.

The remainder of the chapter is organized as follows. Section 3.2 briefly overviews the related work, which includes RSU placement strategies that focus on minimizing the cost of deployed RSUs. In Section 3.3, a detailed description of our system model is presented. An integer linear program (ILP) formulation of the RSU placement and configuration problem is introduced in Section 3.4. Then, in Section 3.5, an LP-based heuristic algorithm referred to as Minimum Cost Route Clustering (MCRC) is introduced. Performance results are presented and discussed in Section 3.6. Finally, this chapter is concluded in Section 3.7.

3.2 Related Work

Several solutions have been introduced in the literature for RSU deployment, which can be categorized into two groups. The first category is budgeted-constrained optimization problem in which one or two features of the network are maximized or minimized, e.g., throughput (Wu *et al.*, 2012), the number of vehicles that get in contact with at least one vehicle (Trullols *et al.*, 2010; Silva *et al.*, 2015, 2016), coverage (Patil and Gokhale, 2013; Rizk *et al.*, 2014; Makkawi *et al.*, 2015; Cheng *et al.*, 2015), delay (Aslam and Zou, 2011; Aslam *et al.*, 2012). In this category, either the total number of RSUs that is allowed to be deployed or the total available budget for RSU deployment is limited to what authorities determined. Of course, the former assumes that all RSUs are identical.

In the second category, the RSU deployment problem is formulated as a quality-constrained optimization problem in which a minimum level of quality is determined by the network operator and the total cost of deployment is minimized. This approach seems to be more appropriate since the network operator can estimate the deployment cost for a satisfactory level of service. In this category, also, the objective function is either the minimization of the number of deployed RSUs (e.g., see References (Abdrabou and Zhuang, 2011; Abdrabou *et al.*, 2013; Liya *et al.*, 2013; Liu *et al.*, 2016; Patra *et al.*, 2014; Wang *et al.*, 2016) for highway scenario and References (Liu *et al.*, 2014; Chi *et al.*, 2013b,a; Wang and Chang, 2011; Kumrai *et al.*, 2014; Liu *et al.*, 2013; Yan *et al.*, 2014; Xiong *et al.*, 2013; Hu *et al.*, 2016) for urban scenario) or the minimization of the total CAPEX (e.g., see References (Lin, 2012; Liang *et al.*, 2012; Li *et al.*, 2015; Song *et al.*, 2015; Mehar *et al.*, 2015)).

Assuming identical RSUs is not necessarily realistic since not all locations have the same characteristics such as tempo-spatial traffic pattern, grid power availability, and backhaul connection options. Therefore, deploying identical RSUs may not be the right solution for general scenarios. Also, considering RSU configuration without paying attention to OPEX may lead to a placement with large operating costs.

Reference (Zhang *et al.*, 2015), for example, uses transmission power, which controls RSU radio coverage range, to represent OPEX costs. The objective is to minimize the transmission power while achieving full road coverage with minimum OPEX. In this study, all RSUs are considered identical and the proposed algorithm takes the number of RSUs as an input parameter. The problem is therefore solved for different numbers of RSUs to obtain the value that leads to minimum OPEX cost. Reference (Vageesh *et al.*, 2014), also assumes that RSUs are identical. A multi-objective optimization is used to minimize both the number of RSUs and the OPEX. However, since it focuses only on vehicle data collection, the energy consumption of each RSU happens during data reception, standby, and awakening periods. By breaking down the problem into a set of independent time slots, the minimum number of RSUs that satisfies the packet delivery ratio is determined. The union of the solutions for all time slots determines the RSU deployment. Each RSU goes to sleep during the time slots when it was not part of the solution. All RSUs use their battery energy, charged by a solar panel, and only use grid power if there is not enough battery power to operate.

The current approaches proposed in the literature either focus on RSU configuration with no attention to OPEX or focus on OPEX with no attention to RSU configuration. To the best of our knowledge, our work is the first that proposes a framework to jointly select RSU configurations and their deployment locations, while minimizing the sum of CAPEX and OPEX costs. This is done by incorporating energy aware scheduling into the design process.

3.3 System Model

The problem that is addressed is that of RSU *facility placement*. Our results, therefore, involve two phases. The first is the *offline design* phase, during which RSU placements are made that determine the designed network of RSUs. The second is an *online performance* assessment of the designed network that quantifies the quality of the offline design. These are discussed in more detail, as follows.

- *Offline Design*: In the offline design, historical vehicular traffic traces and RSU candidate location information are used as inputs. Since this process is offline, the traffic traces used are therefore completely known to the offline design algorithm, and the packet scheduling that occurs can also have complete knowledge of the offline design traffic traces. The output of this process is a network design, i.e., a set of RSU placements and their chosen configurations, taken from the candidate location inputs.

- *Online Performance*: Once the offline design is completed, the RSUs are installed and are subjected to vehicular traffic data experiments. In this phase, vehicular traffic demands must be processed by the system in a causal fashion, as would be the case in a deployed network. These experiments are therefore performed using traffic input traces that are different from those used in the offline design phase. In this case however, the inputs are provided to the system in real time and packet transmissions must be scheduled in a causal fashion, based solely on past and current inputs.

The overall objective of the design is to create offline RSU placements and configurations so that the network can properly schedule online vehicular demands, and such that the total of the RSU opening and service costs, discussed below, are minimized. In online operation, vehicles are assumed to travel along a given road network and generate requests for service that are communicated on an uplink channel to the next RSU that is encountered. The responses to these requests are then scheduled and served by one or more RSUs over time slotted downlink channels. Each request has an associated time deadline.

When an RSU is installed in the offline design phase, we pay an *opening cost*, and to serve a vehicle by an opened RSU in the online phase, we pay a *service cost.* These are defined as follows:

Opening cost: The opening cost of an RSU is determined by its location and its configuration settings (such as its backhaul connection type, power source, channel capacity, coverage range, antenna type, etc.). A location-based RSU cost analysis was done in (J. A. Volpe National Transportation Systems Center, 2008). Our model can accommodate non-homogeneous RSUs that are operated with different costs (e.g., operated by the wired electrical power grid or by solar power (J. A. Volpe National Transportation Systems Center, 2008)), in addition to limited but different coverage range. This limitation on the maximum coverage range is sometimes used to control radio interference levels (U.S. Federal Communications Commission, 2004; Al-Sultan *et al.*, 2014; J. A. Volpe National Transportation Systems Center, 2008).

Service cost: It is assumed that the RSUs use power control when communicating with the vehicles, i.e., they adapt their transmit power in order to maintain a constant bit rate (Hammad *et al.*, 2013; Khezrian *et al.*, 2015). This is in contrast to the use of rate adaptation, but even in this latter case, an RSU will experience different energy expenditures on links with different path loss. The lower bit rates on poorer links will result in longer packet transmit times (Hammad *et al.*, 2015). The energy cost of this communication thus depends on the radio link propagation conditions. The total operating cost depends on the *planning time horizon*, i.e., the time period over which the RSU cost is amortized, which may be as long as one or more decades. We assume that the vehicle traffic load input trace is statistically representative of the traffic flow and we can normalize the operating cost to the long-term planning time horizon.

Once an RSU has been deployed, it remains in continuous operation serving vehicular requests. Each vehicle request has a release date, i.e., the time when the request is generated, and a due date, i.e., the deadline of the associated RSU response. A request that is un-served or is served beyond its due date is counted as a *dropped request*. We assume that a vehicle generating a request, communicates its size, release, and due dates to the first RSU it encounters, and, therefore, the system is aware of these parameters for scheduling purposes. This formulation is very general in that it can be used to model a wide range and mix of application quality-of-service requirements. For example, if traffic with real-time or time-critical constraints is to be considered, the dropping rate and job deadlines can be adjusted accordingly. In a similar way, delay tolerant traffic can be modelled using appropriate settings. The results given in Section 3.6 take this latter approach. The job deadline and loss parameters may also determine the level of network radio coverage permitted in the RSU placement. For example, tight delay constraints will tend to dictate that contiguous coverage is required throughout the network. Conversely, in data dissemination types of applications, there may be considerable delay tolerance, which will permit partial radio coverage.

It is assumed that the route of a vehicle is known, and that each vehicle communicates to the system (through the first RSU encountered) its current location, final destination, and intended route (Zou *et al.*, 2011; Ali *et al.*, 2014b). This is a reasonable assumption, since drivers tend to follow their habits and traffic information in planning their daily route to work, home or other destinations (Cascetta, 2009). This assumption is also consistent with the driver-less car functionality that is beginning to appear. During our experiments, we will assume that the vehicle traffic and request flows are stable, i.e., the traffic flow and requests are characterized by a constant arrival rate (that would typically be chosen to accommodate worst case traffic conditions), which can be seen as the arrival rate at traffic equilibrium (Cascetta, 2009).

3.4 RSU Placement and Configuration Problem

The system model is more formally defined as follows. Let $\mathcal{N} = \{1, \ldots, N\}$ be the set of RSU candidate locations, and $\mathcal{V} = \{1, \ldots, V\}$ be the set of vehicles serviced by the installed RSUs, each with a set of requests $\mathcal{R}_v$, and $|\mathcal{R}_v| = R_v$. Let $\mathcal{R} = \cup_{v \in \mathcal{V}} \mathcal{R}_v = \{1, \ldots, R\}$ be the set of all requests. Request r has an associated download size in time slots, denoted by ℓ_r. With a slight abuse of notation, we will refer to an RSU installed at location n as 'RSU n'. We define decision variables Y_n, so that $Y_n = 1$ if RSU n is installed, and $Y_n = 0$ otherwise. The cost of opening an RSU at location n is f_n. Let $\mathcal{T} = \{1, \ldots, T\}$ be the set of time slots; within a time slot, RSU n has the capacity to transmit to at most u_n vehicles, and a vehicle can communicate with at most one RSU. Note that f_n and u_n depend only on the location n, i.e., RSUs installed in different locations are allowed to be of different types with different opening costs.

We define the decision variables, X_{ntr}, such that $X_{ntr} = 1$ if RSU n serves request r of vehicle v during time slot t, and $X_{ntr} = 0$ otherwise. The energy cost for servicing this request is denoted by c_{ntr}. We define $P_{n,v}(t)$ as the communication cost between RSU n and vehicle v during time slot t. $P_{n,v}(t)$ depends on the RSU-vehicle distance (and other propagation effects) in time slot t. This is done by first computing the transmit power needed to overcome the path loss from RSU n to vehicle v at time t, such that a target SNR is achieved that supports the chosen data rate. This power is added to the quiescent radio power consumption, and the total energy is computed by multiplying by the time slot duration (Hammad *et al.*, 2013). When vehicle v is within the coverage area of RSU n during time-slot t, $c_{ntr} = P_{n,v}(t)$ if the request r is serviced after its release date and before its deadline, and $c_{ntr} = \infty$ otherwise.

In order to enforce the servicing of all requests, if possible, we define the non-serviced portion of request r by variable Z_r, and give it a large cost, D_r. That is, if $Z_r > 0$, then request r is *dropped* and incurs a very large cost $D_r Z_r$. As a result, in the optimization defined below, the scheduler will never drop a request, unless there is a capacity constraint violation. Since this part of the objective function is an artifice to ensure service, it will not be included in the total cost we present in the results obtained.

Given the above definitions and for a given input traffic design trace, we formulate the optimum cost as an integer linear program. This provides a lower bound on the total cost and is used in Section 3.5 to obtain a practical RSU placement algorithm using a novel rounding procedure. The optimization is given as follows and discussed below.

$$\min_{X,Y,Z} \sum_{n\in\mathcal{N}} f_n Y_n + \sum_{n\in\mathcal{N}} \sum_{t\in\mathcal{T}} \sum_{r\in\mathcal{R}} c_{ntr} X_{ntr} + \sum_{r\in\mathcal{R}} D_r Z_r \tag{ILP}$$

subject to:

$$Z_r + \sum_{n\in\mathcal{N}} \sum_{t\in\mathcal{T}} X_{ntr} = \ell_r \qquad \forall r \in \mathcal{R} \tag{3.1}$$

$$X_{ntr} \leq Y_n \qquad \forall n \in \mathcal{N}, t \in \mathcal{T}, r \in \mathcal{R} \tag{3.2}$$

$$\sum_{r\in\mathcal{R}} X_{ntr} \leq u_n Y_n \qquad \forall n \in \mathcal{N}, t \in \mathcal{T} \tag{3.3}$$

$$\sum_{n\in\mathcal{N}} \sum_{r\in\mathcal{R}_v} X_{ntr} \leq 1 \qquad \forall t \in \mathcal{T}, v \in \mathcal{V} \tag{3.4}$$

$$Y_n, X_{ntr} \in \{0, 1\} \qquad \forall n \in \mathcal{N}, t \in \mathcal{T}, r \in \mathcal{R} \tag{3.5}$$

$$Z_r \in \{0, 1, \ldots, \ell_r\} \qquad \forall r \in \mathcal{R} \tag{3.6}$$

The objective function in (ILP) consists of three terms. The first is the total CAPEX cost that sums the individual capital costs of each placed RSU, i.e., the cost is f_n when RSU n is placed ($Y_n = 1$), and zero otherwise. In general, f_n consists of hardware and installation costs of RSU n, which may be site dependent. The prior includes the chosen configuration of the RSU, i.e., items such as the radio configuration (u_n, etc.), the antenna type, power option (grid/solar, etc.), and backhaul connection type (J. A. Volpe National Transportation Systems Center, 2008). The second term is the OPEX costs associated with operating the RSU. RSU n incurs an energy cost of c_{ntr} for transmitting request packet r in time slot t.

The final term in ILP is used to minimize the un-served fraction of requests, Z_r

for request r using a large penalty D_r, as discussed previously.

Constraint (3.1) guarantees that requests are satisfied and Constraint (3.3) enforces the capacity constraint for the RSUs. Constraint (3.4) implies that only one request of vehicle v can be serviced during time slot t. Note that (3.3) and (3.5) imply Constraint (3.2), but the latter is crucial for our rounding heuristic, strengthening the LP relaxation presented below. It is also clear from the above formulation, that vehicle job requests are splittable, in that they may be serviced across multiple RSUs.

Solving (ILP) is NP-complete, since if restricted to a single time-slot and capacities of 1, it becomes the classic minimum facility location problem, which is NP-complete (Vazirani, 2003). Therefore we turn to approximation algorithms.

3.5 Minimum Cost Route Clustering Algorithm

Our proposed heuristic is based on the following primal LP relaxation of (ILP). Unlike (ILP), it can be solved in polynomial time complexity but does not give integral solutions for the decision variables. This issue is addressed by using the rounding procedure discussed below.

$$\min_{X,Y,Z} \sum_{n\in\mathcal{N}} f_n Y_n + \sum_{n\in\mathcal{N}} \sum_{t\in\mathcal{T}} \sum_{r\in\mathcal{R}} c_{ntr} X_{ntr} + \sum_{r\in\mathcal{R}} D_r Z_r \quad \text{s.t.} \tag{PLPR}$$

$$Z_r + \sum_{n\in\mathcal{N}} \sum_{t\in\mathcal{T}} X_{ntr} \geq \ell_r \qquad \forall r \in \mathcal{R} \tag{3.7}$$

$$\text{Constraints } (3.2) - (3.4)$$

$$Y_n \leq 1 \qquad \forall n \in \mathcal{N} \tag{3.8}$$

$$Z_r \leq \ell_r \qquad \forall r \in \mathcal{R} \tag{3.9}$$

$$Y_n, X_{ntr}, Z_r \geq 0 \qquad \forall n \in \mathcal{N}, t \in \mathcal{T}, r \in \mathcal{R} \tag{3.10}$$

Rounding the solution of (PLPR) to an integral one is non-trivial, since the integrality gap for this relaxation is infinite (Kolliopoulos and Moysoglou, 2015). Levi et. al. (Levi *et al.*, 2012) introduced an LP-based approximation algorithm for the capacitated facility location problem, in which the service has no time constraints, the service cost is time-independent, and there is no capacity associated with clients. It consists of a two-phase clustering procedure, followed by a rounding algorithm, and has a provable approximation factor of 5 when the connection costs are in a metric space. Unfortunately, our model is more complicated than the problem in (Levi *et al.*, 2012), and, moreover, our operating costs do not come from a metric space; therefore, the known approximation factor guarantees for facility location problems do not necessarily apply in our case. Accordingly, we develop a novel heuristic referred to as the Minimum Cost Route Clustering (MCRC) algorithm, which operates in two steps. In the first, all (partially) opened RSUs from the solution of (PLPR) are partitioned into clusters. In the second step, the rounding algorithm installs all fully opened RSUs in each cluster, and continues installing fractionally open RSUs, until it opens enough RSUs to satisfy all service requirements for that cluster.

The algorithm starts with the fractional solution of (PLPR). This solution consists of (partially) opened RSUs and (fractional) request assignments to the (partially) opened RSUs. In the next step, our algorithm moves the fractional requests of vehicles from one RSU to another, so it can fully open some RSUs and fully close the rest, thus producing an integer solution. It is obvious that the displacement of requests increases the assignment costs and, although we cannot guarantee an upper bound for this increase, as done in (Levi *et al.*, 2012), our simulation results show that the extra cost of assignment displacements is low.

As argued in (Levi *et al.*, 2012), moving assignments too much leads to prohibitively expensive results. For this reason, a clustering step is used before the rounding procedure. It divides the problem into subproblems, and the rounding of their fractional solutions is done separately for each. The clustering step imposes an extra cost, which is due to the aggregated effect of rounding the fractional solutions in each sub-problem.

Algorithm 1 shows the details of our algorithm. Let (X, Y) be the optimal solution to (PLPR) (assuming that all requests are feasible and $Z = 0$), and α_r the optimal dual variables for relaxed constraints (3.7). The two steps of Algorithm 1 are denoted as *Clustering* and *Rounding*.

In Clustering, we partition the RSUs with $Y_n > 0$ ($\mathcal{F}$) into *clusters*, each of which will be "centered" around a vehicle that we call the *cluster center*. More specifically, for each vehicle v, we define α_v to be the summation of $\alpha_r \ell_r$ over all its requests. α_v shows the contribution of each vehicle in the total deployment and connection costs, and decreases as the number of vehicles increases, but it increases as the number of requests per vehicle increases. The exact explanation of the use of α_v in the objective function of the dual of (PLPR) can be found in (Levi *et al.*, 2012).

Let $\mathcal{F}_v$ be the set of (partially) opened RSUs that (fractionally) serve vehicle v. Let $\mathcal{S}$ be the set of cluster center candidates (initially the set of all vehicles), and $\mathcal{C}$ be the set of current cluster centers (initially empty). We use $\mathcal{N}_v$ to denote a potential

Algorithm 1 Minimum Cost Route Clustering

```
Let $(X, Y)$ be the solution to (PLPR) and $\alpha$ the dual variables for constraints (3.7)
Let $\eta$ be the clustering threshold
                                                                ▷ Step 1: Clustering
$\mathcal{F} := \{n \in \mathcal{N} : Y_n > 0\}$                                   ▷ (partially) opened RSUs
for all $v \in \mathcal{V}$ do
    $\mathcal{F}_v := \big\{n \in \mathcal{F} : \sum_{t\in\mathcal{T}} \sum_{r\in\mathcal{R}_v} X_{ntr} > 0\big\}$     ▷ RSUs that fractionally serve vehicle v
    $\alpha_v := \sum_{r\in\mathcal{R}_v} \alpha_r \ell_r$.
end for
Initialize:
    $\mathcal{C} := \emptyset$                                                  ▷ cluster centers
    $\mathcal{S} := \mathcal{V}$                                                ▷ cluster center candidates
    $\mathcal{N}_v := \emptyset,\ \forall v \in \mathcal{S}$                            ▷ one potential cluster per vehicle v
while $\mathcal{S} \neq \emptyset$ do
    for all $v \notin \mathcal{C}$ do
        $\mathcal{B}_v := \{n \in \mathcal{F}_v : n \notin \cup_{k\in\mathcal{C}}\mathcal{N}_k, c_{nv} \leq \min_{k\in\mathcal{C}} c_{nk}\}$
    end for
    $\mathcal{S} := \{v \notin \mathcal{C} : \max_{n\in\mathcal{B}_v, t\in\mathcal{T}, r\in\mathcal{R}_v} X_{ntr} > \eta\}$
    Pick $v \in \mathcal{S}$ with the smallest $\alpha_v$ value and if there are more than one, pick the one
    has the largest $\sum_{n\in\mathcal{B}_v} u_n Y_n$.
    Set $\mathcal{N}_v := \mathcal{B}_v$, $\mathcal{C} = \mathcal{C} \cup \{v\}$
end while
$\mathcal{U} := \mathcal{F} - \cup_{k\in\mathcal{C}}\mathcal{N}_k$
for all $n \in \mathcal{U}$ do
    $v := \arg\min_{k\in\mathcal{C}} c_{nk}$
    $\mathcal{N}_v := \mathcal{N}_v \cup \{n\}$
end for
                                                                ▷ Step 2: Rounding
for all Cluster centers $v \in \mathcal{C}$ do
    Open all of the fully opened RSUs in $\mathcal{N}_v$.
    $\mathcal{Q}_v := \{n \in \mathcal{N}_v : Y_n < 1\}$
    $D_{left} := \sum_{n\in\mathcal{Q}_v} u_n Y_n$
    Sort the RSUs in $\mathcal{Q}_v$ in increasing order of $(f_n/u_n + c_{nv})$
    while $D_{left} > 0$ do
        Let $n$ be the next RSU in the sorted list
        Open RSU $n$
        $D_{left} := D_{left} - \min(D_{left}, u_n)$
        $\mathcal{Q}_v := \mathcal{Q}_v \setminus \{n\}$
    end while
end for
```

cluster centered around vehicle v. Initially, $\mathcal{N}_v$ is empty for all vehicles.

At every iteration of lines 13-20, we define a set $\mathcal{B}_v$ for every vehicle v in $\mathcal{S}$, as long as the latter is non-empty. Of all RSUs in $\mathcal{F}_v$, set $\mathcal{B}_v$ contains only those that have not been assigned to any clusters yet and that are "closer" to v than all cluster centers currently in $\mathcal{C}$, according to a closeness function that is based on the average connection cost between a vehicle v and an RSU n, i.e.,

$$c_{nv} = \frac{\sum_{t \in \mathcal{T}_{nv}} P_{n,v}(t)}{|\mathcal{T}_{nv}|} \tag{3.11}$$

where $P_{n,v}(t)$ is the communication cost between vehicle v and RSU n at time slot t, and $\mathcal{T}_{nv}$ is the set of time slots during which vehicle v is inside the coverage area of RSU n. Unlike many clustering techniques that create clusters of clients around facilities, our algorithm is based on forming clusters of facilities (RSUs) around some candidate clients (vehicles) (Levi *et al.*, 2012). Therefore, a dense network with more vehicles does not have much effect on $\mathcal{B}_v$. On the other hand, a dense network with more RSU candidates may increase the size of $\mathcal{B}_v$. This approach prevents the creation of too many clusters, which will eventually reduce the extra costs incurred by opening the fractional RSUs in the clusters. The intuition behind this is as follows: The removal of RSU n from $\mathcal{B}_v$ because it is closer to some other vehicle v', implies that the two vehicles v, v' share part of their routes. Therefore, we can divide the route of v in two parts, the part that is shared with v', and the part that is not; we charge v with only the cost of the latter.

Throughout the Clustering phase, set $\mathcal{S}$ contains all RSUs that are candidates for opening; these are the RSUs that are (partially) open by at least a preset factor η, called the *clustering threshold*. Parameter η can be preset to any value between 0 and

1, but in our case we set $\eta = 0$ to force all partially opened RSUs to be candidates for full opening. In each iteration, we pick the vehicle $v \in \mathcal{S}$ with the smallest α_v value (we break ties by picking the vehicle that has the largest capacity in $\mathcal{B}_v$). We form a cluster centered at v, with $\mathcal{N}_v = \mathcal{B}_v$ and update sets $\mathcal{C}$ and $\mathcal{S}$ accordingly. We continue this procedure while $\mathcal{S}$ is not empty. After that, there can still be RSUs in $\mathcal{F}$ that are not assigned to any cluster. Each of those RSUs is assigned to the cluster whose center is closer to it.

Clustering is followed by Rounding. For each cluster $\mathcal{N}_v$, and after opening all RSUs with $Y_n = 1$, we start opening the rest of the RSUs in this cluster, called $\mathcal{Q}_v$, one-by-one and in increasing order of $(f_n/u_n + c_{nv})$, until all capacity requirements of this cluster are satisfied. $\mathcal{Q}_v$ is the set of cluster members for vehicle v after we remove the fully opened RSUs in the set, i.e., $\mathcal{Q}_v$ is the set of partially opened RSUs in the cluster where v is its cluster center. After enough RSUs have been opened, we schedule the requests to time-slots and RSUs. Since the capacity of the opened RSUs is sufficient to serve all the requests, this offline scheduling problem is feasible, i.e., the additional request drop ratio is zero (recall that we already have that $Z = 0$).

It can easily be shown that if N, V, R, T are the number of candidate locations, vehicles, request units, and time slots, respectively, and under the natural assumption $N \leq V$, the time complexity of the MCRC algorithm is $\mathcal{O}\left(N^2TR + N^3V\right)$.

Multiple-Choice RSU Placement: In the model described above, there is only one choice for the RSU to be opened at a candidate location. We can easily generalize this, by allowing the RSU to be chosen from a set of different types; however, we still require that at most one RSU is opened at any candidate location. The only change needed is the extension of the Y variables, to have one for each choice at a location (instead of one per location), and the extra constraint that the sum of these variables must be at most 1 at every location. Algorithm 1 does not need to change significantly; we just remove the rest of the choices at a location, once we decide to open an RSU of a specific type.

3.6 Performance Results

In this section, the performance of the proposed RSU placement algorithm is considered. In order to evaluate the performance of MCRC, in the first two sets of results, two different on-line scheduling algorithms were used to assess the placements that the algorithm generates. The first is the GMCF scheduler introduced in (Hammad *et al.*, 2013). GMCF schedules requests on a single RSU using a minimum cost flow graph formulation that minimizes total service cost over a finite scheduling window. The second algorithm is the one-objective min-max scheduler presented in (Khezrian *et al.*, 2015). This algorithm schedules requests across multiple RSUs and attempts to minimize the maximum service cost on any of the RSUs. Since our goal is to minimize the total service cost on multiple RSUs, the schedulers are adapted to work in this setting. These two schedulers are referred to as the *Energy Scheduler* and the *Min-Max Scheduler*, respectively, and both are non-preemptive. Since we find that the energy performance of the two schedulers is very close, in most of our results, we use only the Energy Scheduler.

Our proposed algorithm is compared with RSU placements that minimize only CAPEX, referred to as the Minimum Capital Cost Placement (MCCP) algorithm. This is motivated by the work discussed in Section 3.2, which can be adapted to our problem by removing OPEX from the objective function of (ILP). MCCP solves the resulting ILP exactly, to obtain RSU placements that minimize the total CAPEX cost, subject to satisfying the same constraints as MCRC. It is there a lower bound on the cost that can be obtained by any RSU placement algorithm with the objective of minimizing capital cost or minimizing the number of RSUs. The comparisons between MCRC and MCCP therefore show the advantage of taking OPEX costs into account (i.e., MCRC), compared to existing approaches that focus on CAPEX cost reduction alone.

The performance evaluation is done using 10 vehicular traffic trace inputs, where each trace consists of Poisson process vehicular arrivals to the system at the designated mean arrival rate. This is illustrated in Figure 3.1. As shown at the top of the figure, one trace is first used for the offline RSU design and placement, which determines the CAPEX deployment cost. After the design phase, the remaining 9 traces are then used as inputs to the online experiments, which determine the OPEX costs. As discussed previously, it is assumed that the traces represent equilibrium traffic conditions, so that worst-case conditions can be accommodated. The total cost presented in the simulation results is the sum of the two, and the plotted OPEX cost is obtained by averaging the service costs over each simulation run for the 9 online traffic traces. Nine traffic traces were used for the online experiments, since their individual results were found to be very close. The same approach is used in Chapters 4 and 5.

Uniform service request generation is used for all vehicles, i.e., the same arrival rate, size, and time-to-live (TTL). As in (Maia *et al.*, 2013; Dai *et al.*, 2013), we assume that TTL is 40 time slots for each request. The maximum request drop rate is set to 5%. Before doing all of the simulation runs, we experimented with the length of the offline design trace to ensure that it was sufficiently long to obtain algorithm

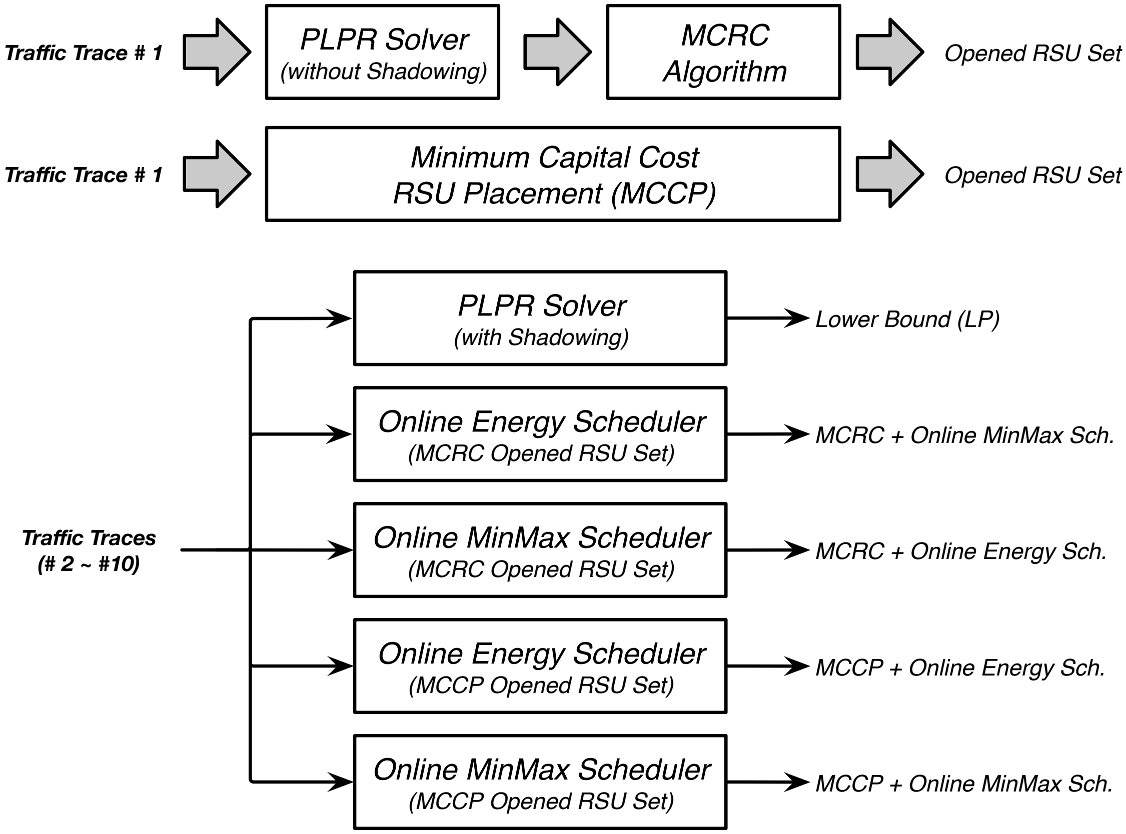


Figure 3.1: Simulation Terminology.

convergence. Our ultimate goal is to make comparisons between MCRC and MCCP. But since MCCP requires the solution of a large ILP, it does restrict the length of the trace that can be used in the RSU placement phase. For the results, MATLAB was used to find the MCRC placements after using CPLEX to solve (PLPR). For the four experiments presented below, the solution times for the MCRC algorithm were quite low, as expected. The per run solution times ranged from, 109 to 283 seconds (150 average), 326 to 740 seconds (421 average), 272 to 740 seconds (421 average), and, 815 to 2411 (1393 average), respectively.

A Manhattan grid road configuration is used, consisting of three horizontal and

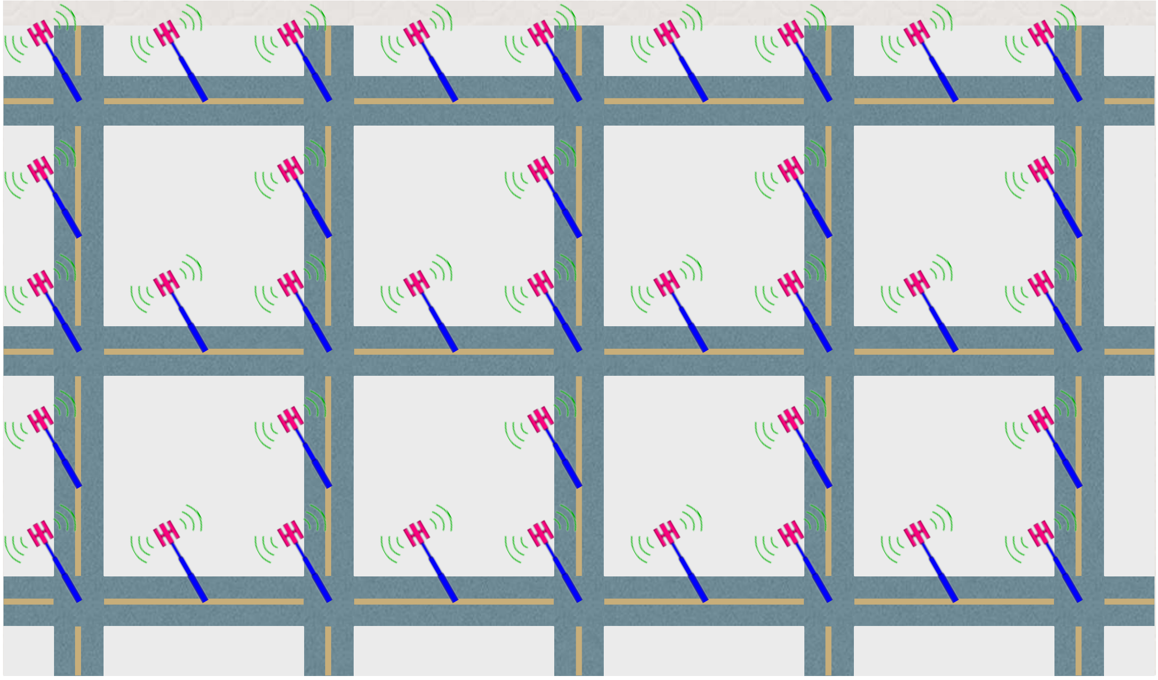


Figure 3.2: City Grid with RSU Candidate Site Locations.

5 vertical streets that are all bidirectional. The Manhattan street network is widely used and is an example of a road network with lots of potential traffic mixing, which tends to make the RSU placement problem difficult. Since it originates from the road configuration in New York City, USA, it is often used to model dense urban traffic scenarios (Maia *et al.*, 2013; Aslam *et al.*, 2012). The smallest block has a 1 km^2 area, which gives a total deployment region of 11.25 km^2. Figure 3.2 shows the city grid with the candidate RSU site locations used as input. To calculate the RSU candidate locations, we divide each street into segments of length equal to twice the RSU coverage range, and the center of each segment is taken as an RSU candidate location. Note that the beginning and the end of each street are under the coverage of the RSUs at the intersections, and therefore, those two sections are subtracted from the length of the street.

It has been shown that microscopic models are the most appropriate for VANET simulations (Martinez *et al.*, 2011; Harri *et al.*, 2009; Spaho *et al.*, 2011). Accordingly, we use the Simulation of Urban MObility (SUMO) tool, which is a microscopic mobility generator along with its other capabilities (Krajzewicz *et al.*, 2012). Vehicles arrive to the city according to a Poisson process. Note that their route selection affects both the RSU placement and the scheduling of requests; for example, if one allows only routes through a single street, the placement of RSUs will be obviously biased towards that street. For our experiments, the source and destination of each vehicle trip are selected uniformly from the set of intersections (Zou *et al.*, 2011; Li *et al.*, 2009), and their route is the shortest path connecting the source to the destination, as calculated by Dijkstra's algorithm, and using the average travel time of each street according to its length, speed limit, and expected traffic density (Patil and Gokhale, 2013; Wang and Chang, 2011; Cascetta, 2009; Song *et al.*, 2014). The vehicle traces are 30 minutes in duration. In Figure 3.2, all streets are two-way. The second street from the top and the third street from the left are 5-lanes with 60 km/h speed limits. The rest are 4-lanes with a 50 km/h speed limit. At intersections, the right-most lane is for right turns and the left-most lane is for left turns. Both also allow straight-through traffic. All intersections are controlled by traffic lights using a standard configuration, as follows. The traffic light logic programs used are similar at all intersections and have 10 phases. In the first (30 second) phase, right turns and straight-through traffic is allowed for vehicles facing a green light. Vehicles are allowed to make a left turn if no vehicle from incoming streams has higher priority. In the second (5 second) phase, green lights turn yellow (amber) and vehicles will decelerate while approaching the intersection for a turn, otherwise they pass through.

The third phase is a left turn light and vehicles have 10 seconds to make a left turn. During this phase, only right turns are allowed if there is no vehicle with higher priority. In the next (5 second) phase, the left turn light turns yellow (amber). During the fifth phase, the lights are red for all directions for 2 seconds. These phases repeat for the opposite direction.

A distance dependent exponential path-loss model with log-normal shadowing (Rappaport, 2001) is used to determine the transmit power needed over a given link. The transmission power between a transmitter and a receiver, $P_{t,r}$, can be expressed by

$$P_{t,r} = P_{t,0} P_{sh} \left(\frac{d_{t,r}}{d_{t,0}} \right)^{\alpha} \tag{3.12}$$

where $d_{t,0}$ is the reference distance, $P_{t,0}$ is the reference power at the reference distance, P_{sh} is a random variable that models the shadowing effect of the channel, α is the path loss exponent, and $d_{t,r}$ is the distance between the transmitter and the receiver. The shadowing effect of the radio channel can be modeled as a random variable with log-normal distribution which has a zero mean (in dB) and a standard deviation of $\sigma_{\text{dB}} = 4$. Table 3.1 summarizes these and the other parameters used in our experiments.

The effect of single RSU capital cost, request size, request arrival rate, request time-to-live (deadline), and vehicle arrival rate is studied. Two experiments were performed that show the trade-off between the two components of the total deployment cost, i.e., the opening and service costs. In the first experiment, which is referred to as "single-choice RSU placement", we give one option for the RSU configuration at each candidate site location. Both MCRC and MCCP algorithms decide the locations where RSUs of that type are placed. In the second, referred to as "multiple-choice

Table 3.1: Simulation Parameters

Parameter Name	Parameter Value(s)
Planning Time Horizon	20 years
Candidate Site Locations	37 sites
RSU Coverage Range	250 m each side
Data Rate	6 Mbps
Vehicle Arrival Rate	0.5 and 1.0 per sec.
Request Arrival Rate	0.0125 per time slot
Request Size	8 time slots
Request Time-to-Live	40 time slots
Street Speed Limit	50 or 60 km/h
Street Number of Lanes	4 or 5
Traffic Light Control	Yes
Path Loss Exponent	$\alpha = 2.7$
Shadowing Standard Deviation	$\sigma_{\mathrm{dB}} = 4$

RSU placement", the output of each algorithm also includes the RSU configuration to be chosen for each selected candidate location.

In some of the experiments, two different vehicular traffic load conditions were considered. The first is when the vehicle arrival rate is one vehicle per time slot, referred to "Low Vehicle Traffic Load". This value is then doubled and the associated experiments are referred to as "High Vehicle Traffic Load".

3.6.1 The Effect of Per RSU Capital Cost

In this first set of results we evaluate the effect of the per RSU capital cost. RSU placements are compared for both single and multiple choice RSU placement. For the first and the third low vehicle load experiments (Figures 3.3 and 3.5), the traffic traces consisted of vehicular arrival numbers ranging from 997 to 1228 and consisting of 2958 to 3733 job requests. In the second and the fourth high traffic load experiments (Figures 3.4 and 3.6), the corresponding numbers ranged from 1961 to 2464 vehicular arrivals and 6545 to 8760 job requests. To properly evaluate the performance of our algorithm, we consider a basic unit cost for each RSU type, and then, we multiply every basic unit cost by the same factor, referred to as the “capital cost factor” during the experiment. The basic unit cost is equal to \$1,000 and \$1,500 for grid-powered and solar-powered RSUs, respectively (J. A. Volpe National Transportation Systems Center, 2008). The single RSU capital cost at each point is equal to its corresponding cost factor multiplied by the basic unit cost. Figures 3.3 and 3.4 show the results of this experiment for single-choice placement, and Figures 3.5 and 3.6 show the multiple-choice case. In the first two sets of results, the OPEX, CAPEX and total cost components are shown in separate subplots. In the remainder of the results, only the total cost is plotted.

In Figures 3.3, 3.4, 3.5 and 3.6, the horizontal axis shows the factor by which the single RSU capital cost is increased. The total cost of RSU deployment are shown in Figures 3.3, 3.4, 3.5 and 3.6. The *Energy* and *Min-Max* schedulers for both MCRC and MCCP RSU placement are shown with different line patterns and markers in the first two figures. Note that because the ILP is too big to solve exactly, we use the relaxed LP as our lower bound. In each subfigure, the LP lower bound service/opening cost component is shown as a black solid line. It is important to note that the LP is only a lower bound on total cost, not on the individual service and opening costs. This can be seen in Figure 3.3, for example, where the service cost falls below the service cost of the LP for capital cost factors greater than about 10.

As seen in Figures 3.3 and 3.4, and, as is the case in the multiple-choice experiments, the MCCP algorithm is insensitive to the service cost as it opens the minimum number of lower opening cost RSUs that are needed to serve requests. For this reason,

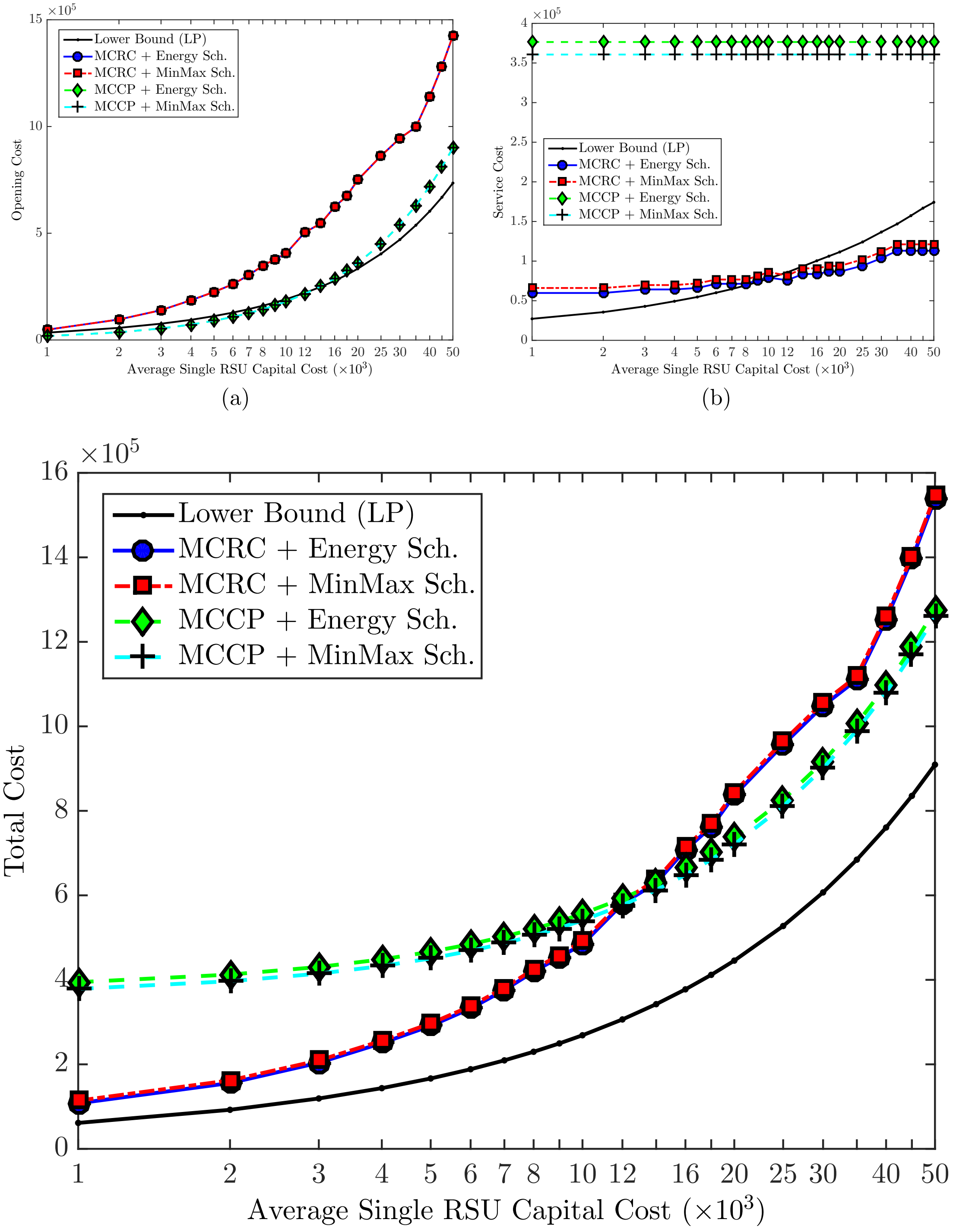
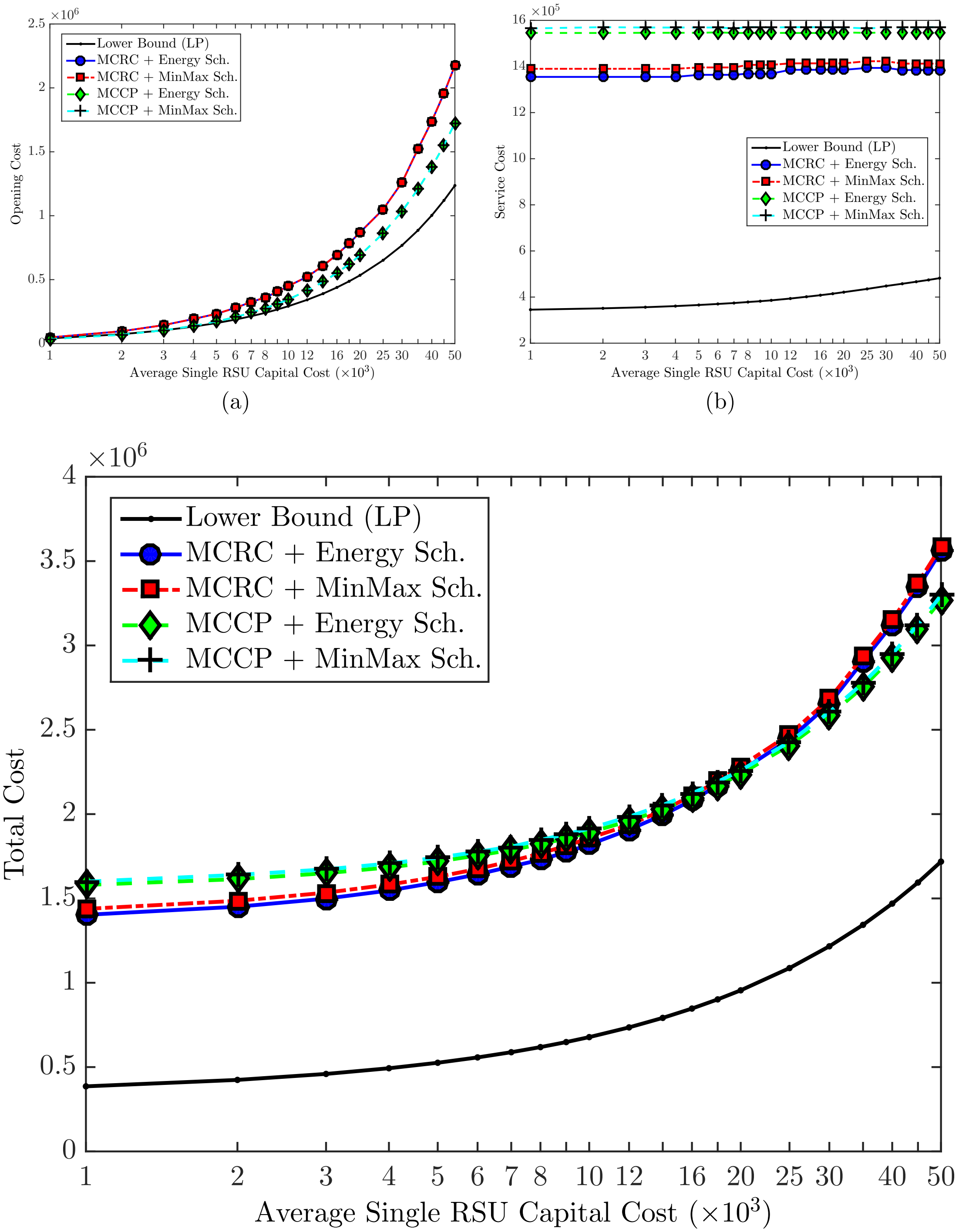


Figure 3.3: The Effect of Per RSU Capital Cost on Single-Choice RSU Placement with Low Vehicle Traffic Load.



(a)

(b)

(c)

Figure 3.4: The Effect of Per RSU Capital Cost on Single-Choice RSU Placement with High Vehicle Traffic Load.

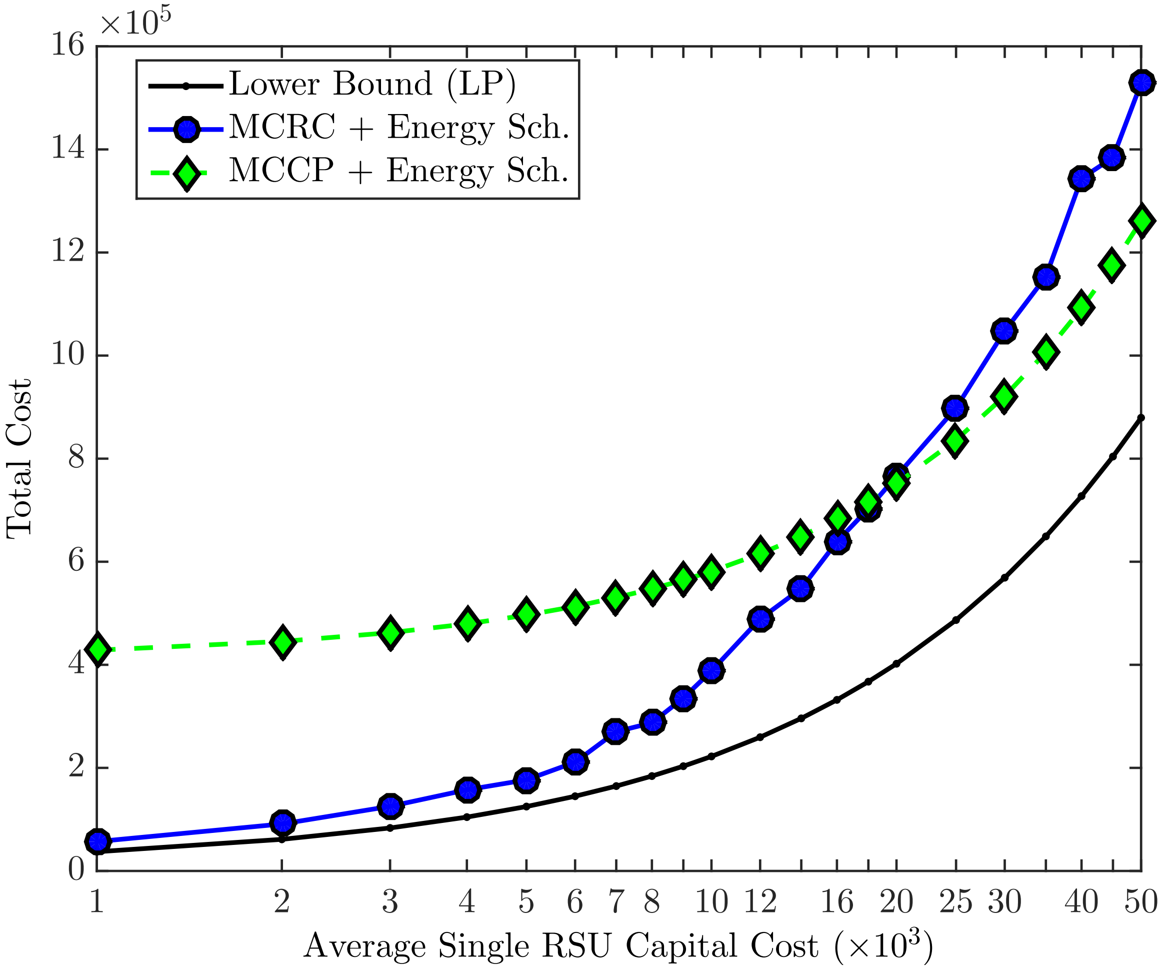


Figure 3.5: The Effect of Per RSU Capital Cost on Multiple-Choice RSU Placement with Low Vehicle Traffic Load.

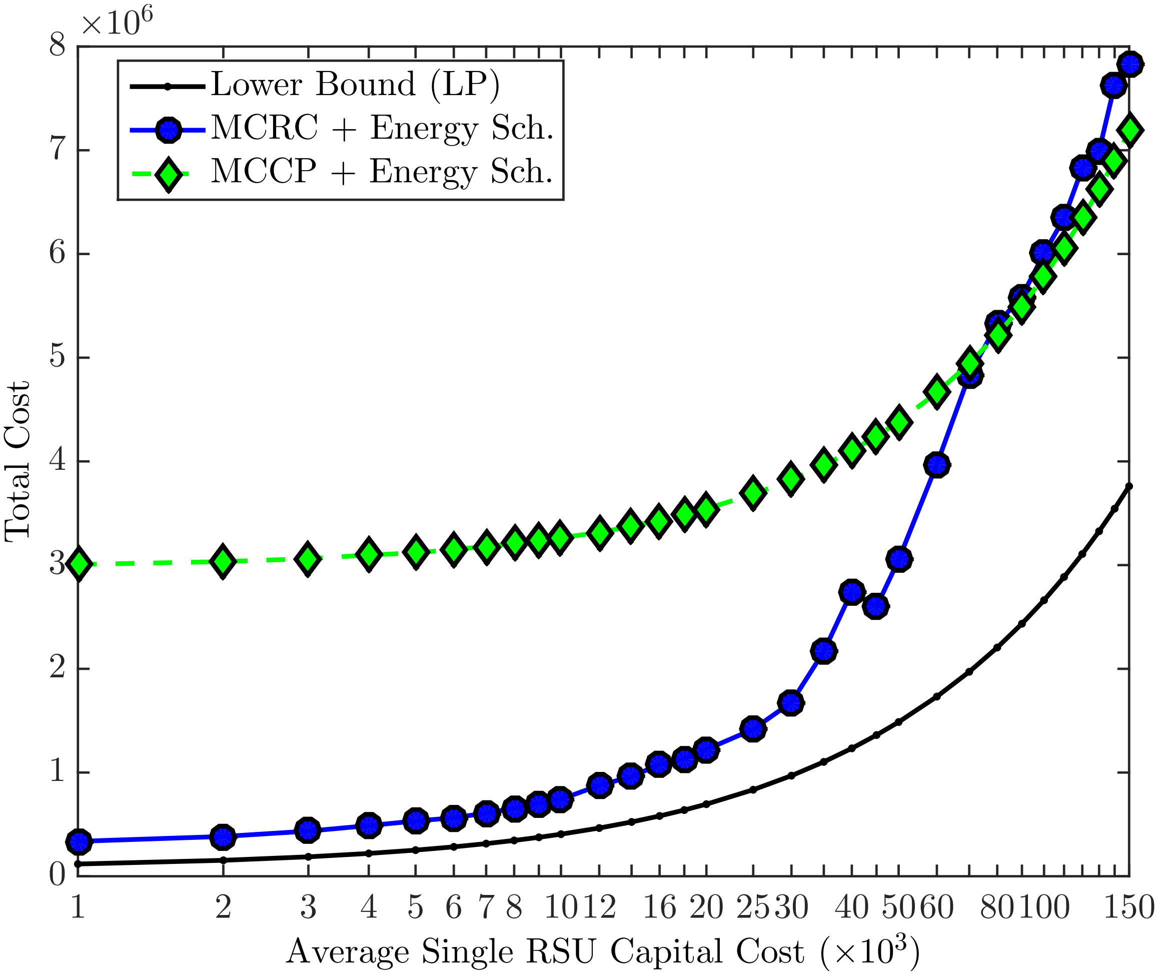


Figure 3.6: The Effect of Per RSU Capital Cost on Multiple-Choice RSU Placement with High Vehicle Traffic Load.

when the per RSU capital cost changes, the MCCP algorithm opens the same set of RSUs. As a result, the opening cost increases linearly with per RSU capital cost and the service cost remains constant for different per RSU capital costs.

On the other hand, the MCRC algorithm tries to trade off the opening and service cost components and tends to outperform MCCP. In Figures 3.3 and 3.4, there are four regions that can be seen in the service cost subfigures. The first region starts with a flat service cost, followed by a smooth increase. In these regions, which correspond to less expensive RSUs, the MCRC algorithm opens more RSUs to reduce the service cost. This approach continues until there is no more decrease in the service cost. This happens either when there are no more RSUs to open, or when opening more RSUs increases the opening cost without improving the service cost.

The third region has a sharper slope compared with the second. As the single RSU capital cost increases, the MCRC algorithm concentrates the requests on a smaller number of RSUs. Although the service cost increases, the overall cost increases at a lower rate. This is because of RSU capacity limitations. After a certain point, there is no way to decrease the number of opened RSUs. This corresponds to the fourth region that has a lower rate by which the service cost increases. There are two reasons for this. Either request deadlines prevent the MCRC algorithm from transferring them from one RSU to another, or, one or more RSUs reach their capacity limit, so that they cannot accept more requests. If both of these happen, the RSU inevitably drops requests.

Note that at the end of the third region and during the fourth, where the MCCP algorithm shows better performance, the opening cost becomes the dominant component of the objective function. The MCRC algorithm opens a smaller fraction of

RSUs to bring down the opening cost.

The four regions discussed above happen at different per RSU capital cost factors and depend on the vehicle traffic load, the data traffic load, and the RSU placement model, i.e., single-choice or multiple-choice. For example, when vehicle traffic load increases, as in Figure 3.4 compared to Figure 3.3, the data traffic load increases, which causes an increase in the service cost. When there is insufficient capacity, opening more RSUs also leads to higher opening costs. When there are no more RSUs to open or when request deadlines do not allow additional loading, increasing the vehicle arrival rate only increases the service cost. Since the MCRC algorithm takes service costs into account during the offline RSU placement, this results in an increased range of capital cost factors over which the total cost is lower than that of the MCCP algorithm. The total cost crossing point of the two algorithms goes from a capital cost factor of 12 in Figure 3.3 to 18 in Figure 3.4. The effect of vehicle arrival rate is discussed in more detail in Section 3.6.5.

A similar behaviour occurs in the multiple-choice experiments, whose total cost is shown in Figures 3.5 and 3.6, but for different reasons. There are two options available at each RSU candidate location, i.e., grid and solar-powered RSUs. The latter are more expensive, but their service cost is lower. This gives the MCRC algorithm more flexibility to trade off these cost components. This can be done using fewer RSUs compared to the similar scenario in single-choice RSU placement. For example, in Figures 3.3 and 3.4, at a capital cost factor of 1, the service cost cannot be reduced, since there are no more RSUs to open. But at the same capital cost factor, the service cost in Figure 3.5 is almost one third of the service cost in Figure 3.3, and the service cost in Figure 3.6 is almost one fourth of the service cost in Figure 3.4. This is true even though there are more RSUs to be opened and happens during the first and second regions by opening more solar-powered RSUs instead of grid-powered RSUs. During the third and the fourth regions, when the single RSU capital cost becomes higher than the service cost, the MCRC algorithm not only concentrates the requests to a smaller number of RSUs, but also prefers grid-powered RSUs. This causes a sharper slope in the service cost.

By comparison, the opening cost remains almost the same for the single and multiple choice cases. Even though the more expensive RSUs are used, a fewer number are opened. As a result, the total cost from the MCRC algorithm in the multiple-choice RSU placement case shows improvement compared to the single-choice placement. The crossing point of the two algorithms moves from a capital cost factor of 12 in Figure 3.3 and from 18 in Figure 3.4, to 20 and 80 in the multiple-choice case, respectively. On the other hand, the MCCP algorithm ignores the service cost, and only opens grid-powered RSUs. As a result, the service cost of the MCCP algorithm in the multiple-choice case is almost double that in Figure 3.4, which degrades its overall performance, as seen in Figure 3.6.

As discussed earlier, if the traffic input surpasses the network capacity, some of the requests will be dropped. In this case, the LP solutions show the regions in which the network is saturated and this can be detected at early stages of the network design. The request drop ratio of the offline LP, lower bound, in Figures 3.4 and 3.6 is equal to 0.1%.

The comparison between the MCRC and MCCP algorithms in terms of the request drop ratio shows that the former has better performance. In the single-choice RSU placement, i.e., Figures 3.3 and 3.4, the request drop ratio of the MCRC algorithm, regardless of the scheduling algorithm is about 0.02%, while the MCCP algorithm has the request drop ratio of 0.6% and 0.5% for the energy scheduler and the min-max scheduler, respectively. As the vehicle arrival rate increases, the competition between vehicles increases. Therefore, more requests are expected to be dropped. In the multiple-choice RSU placement, i.e., Figures 3.5 and 3.6, the request drop ratio of the MCRC algorithm is equal to 2.2% and 1.6% for using the energy scheduler and the min-max scheduler, respectively. The request drop ratio of the MCCP algorithm is equal to 3.0% and 2.4% for the energy scheduler and the min-max scheduler, respectively. Similarly, in multiple-choice RSU placement, shown in Figures 3.5, the request drop ratio of the MCRC algorithm is equal to 0.91% and 0.84% for the energy and the min-max schedulers, respectively. The request drop ratio of the MCCP algorithm is equal to 0.67% and 0.51% for the energy scheduler and the min-max scheduler, respectively. In Figure 3.6, the request drop ratio of the MCRC algorithm is equal to 2.8% and 2.2% for the energy and the min-max schedulers. The request drop ratio of the MCCP algorithm is equal to 3.8% and 3.0% for the energy scheduler and the min-max scheduler. Since the performance of the two schedulers was found to be very close, in the remaining graphs we only consider results for the energy scheduler.

Finally, to further demonstrate the quality of MCRC, Figures 3.7 and 3.8 show the total cost of RSU deployment for Figures 3.3 and 3.4 respectively, when the online experiments use the first vehicular trace that was used to obtain the offline RSU placements. This shows that if the MCRC algorithm is given the correct trace, it still outperforms MCCP. The crossing point of the two algorithms moved from a capital cost factor of about 12 and 18 in Figures 3.3 and 3.4 to 16 and 25 in Figures 3.7 and 3.8 respectively. Note that Figures 3.7 and 3.8 have a slightly higher total cost than

Figures 3.3 and 3.4. This is because of a higher average number of vehicles and job requests than the other traces.

3.6.2 The Effect of Request Size

To evaluate the effect of request size on algorithm performance, we set the capital cost factor to 10. This means that the capital cost of each grid-powered RSU and each solar-powered RSU are equal to \$10,000 and \$15,000, respectively. Because of space limitations, we only present the results for high vehicle traffic load. This means that the vehicle arrival rate is equal to 1 vehicle per second, i.e., 2 vehicles per time slot. Figures 3.9 and 3.10 show these results for the single-choice RSU placement and the multiple-choice RSU placement, respectively. As before, 10 vehicular traces are used, consisting of vehicular arrival numbers ranging from 1961 to 2464 and consisting of 6545 to 8760 job requests. In these results, we increase the size of each individual vehicle request from 1 to 10.

The MCRC algorithm shows a slight advantage over the MCCP algorithm in Figure 3.9. However, in Figure 3.10, the MCRC algorithm significantly outperforms MCCP. As in the previous section, there are four regions. The first two correspond to lower data traffic load. When the request size is small, the service cost is low. Therefore, the MCRC algorithm reduces the number of opened RSUs and transfers requests to the opened RSUs. After this, the only way to bring down the opening cost is to open a smaller number of RSUs. This introduces an extra opening cost in the rounding step of the MCRC algorithm.

In the last two regions, as the request size increases, so does the service cost. However, this increase is not linear. This comes from the fact that vehicles have a

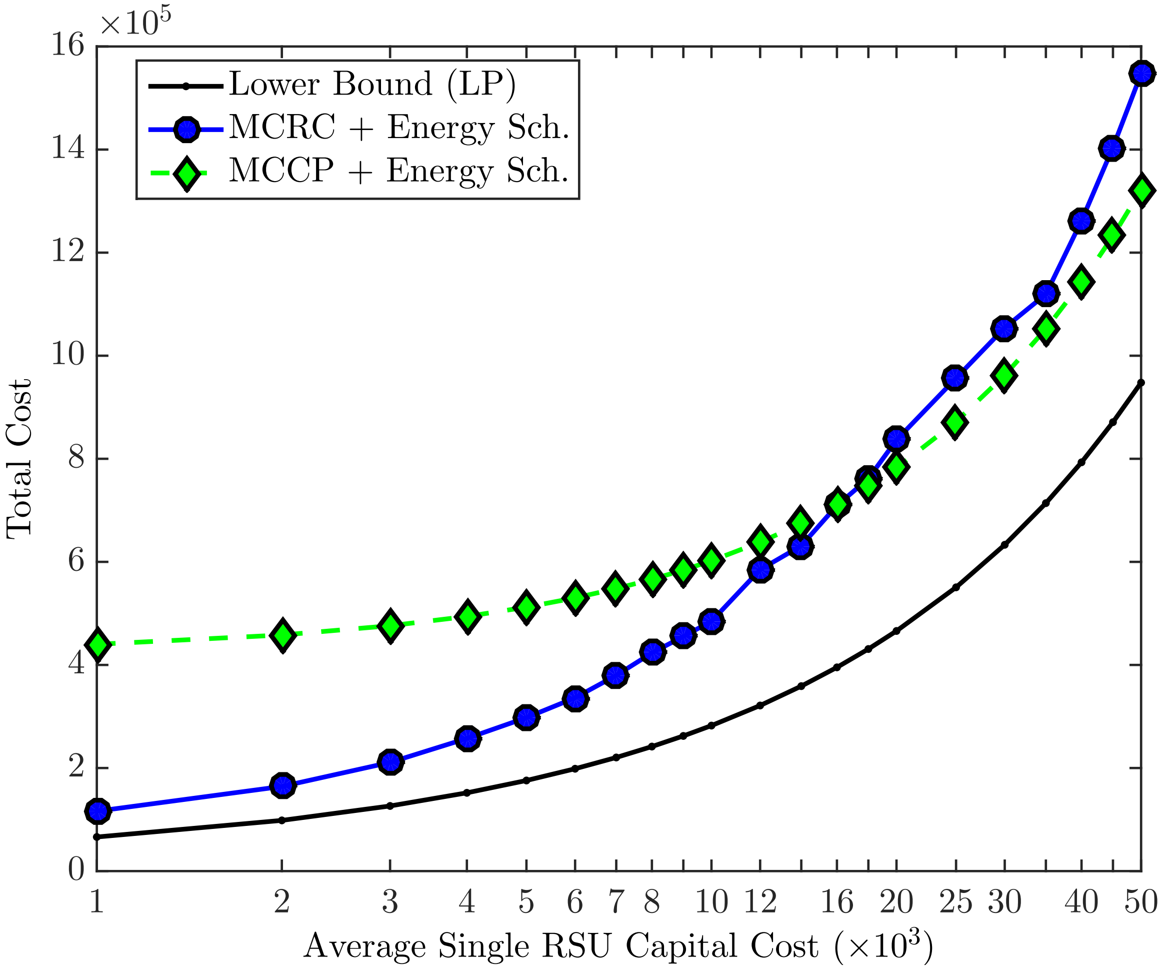


Figure 3.7: The Effect of Per RSU Capital Cost on Single-Choice RSU Placement with Low Vehicle Traffic Load. The Online Experiments Use the Design Trace as Input.

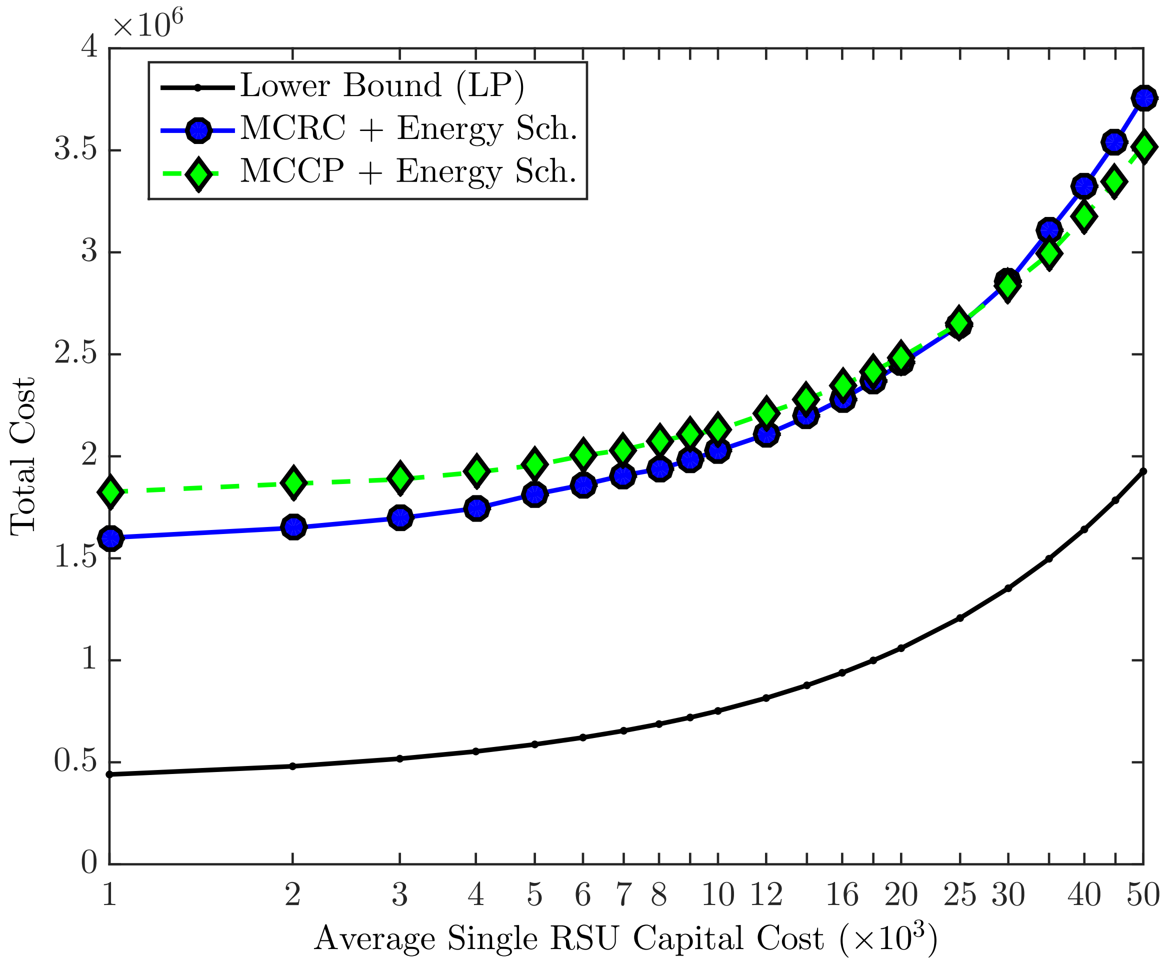


Figure 3.8: The Effect of Per RSU Capital Cost on Single-Choice RSU Placement with High Vehicle Traffic Load. The Online Experiments Use the Design Trace as Input.

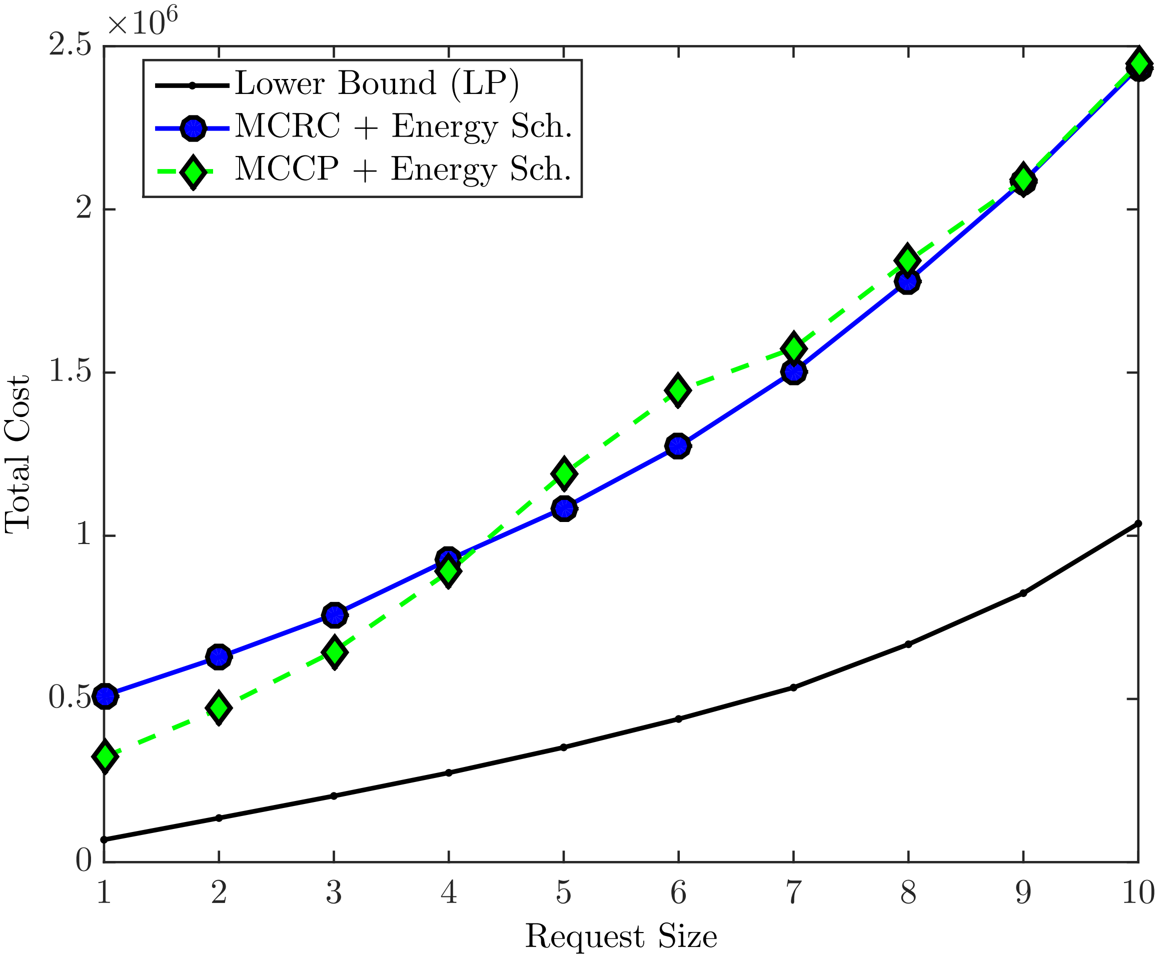


Figure 3.9: The Effect of Request Size on Single-Choice RSU Placement with High Vehicle Traffic Load.

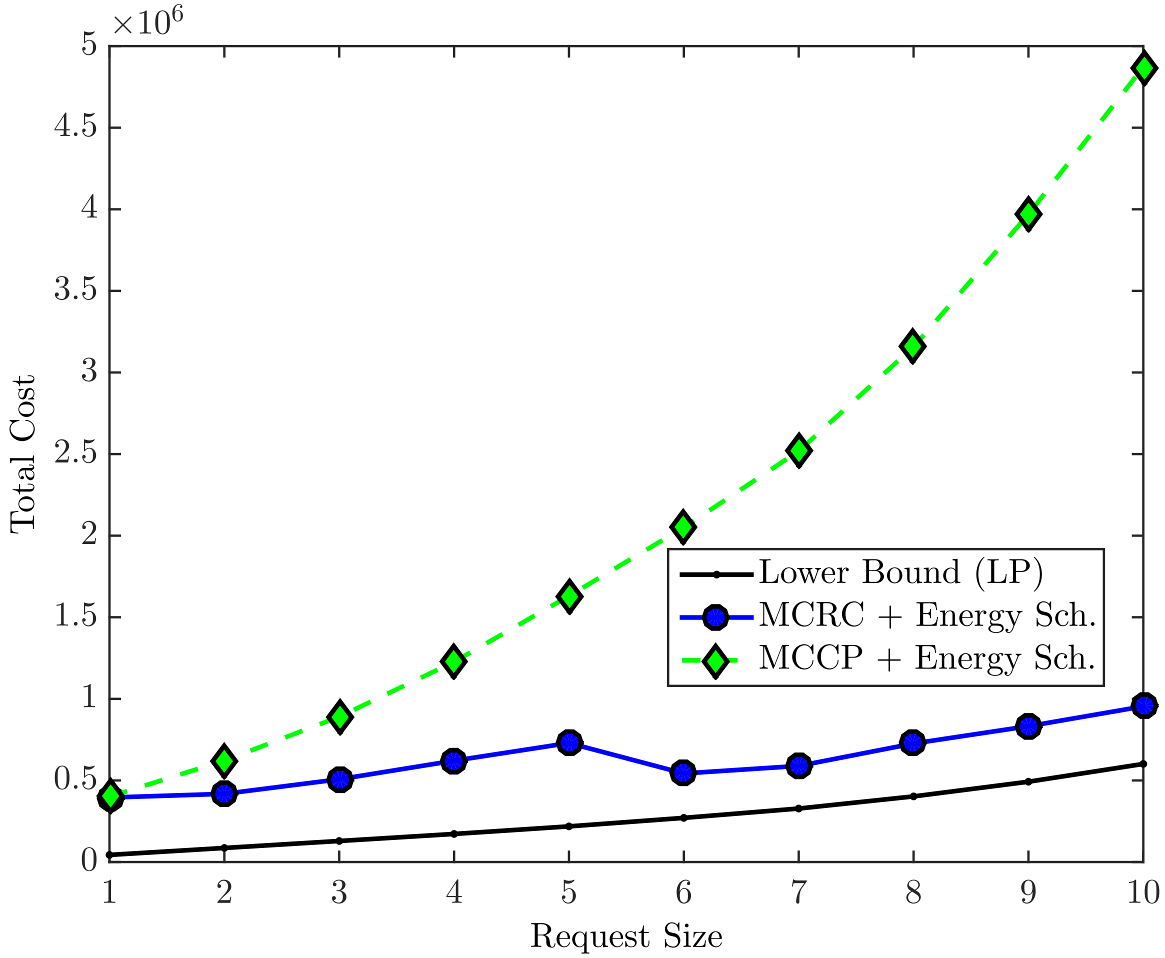


Figure 3.10: The Effect of Request Size on Multiple-Choice RSU Placement with High Vehicle Traffic Load.

capacity limitation since they are equipped with single-radio transceivers. If some parts of the request cannot be transferred to the next RSU, it is more likely that they will be served when the vehicle is farther from the RSU. Also, the increase in the request size requires more network capacity and as a result, both algorithms open more RSUs. The MCRC algorithm opens more RSUs to moderate the service cost increase. In Figure 3.10, the MCRC algorithm also switches to solar-powered RSUs to take advantage of their low service cost. In terms of request drop ratio, the MCRC algorithm shows better performance as before. In both algorithms, the request drop ratio increases rapidly with the request size. Above request sizes of 8, even the LP drops requests.

3.6.3 The Effect of the Request Arrival Rate

For the per RSU capital cost and the vehicle arrival rate, the same values are selected as in Section 3.6.2. In these experiments, the 10 vehicular traces consisted of vehicular arrival numbers ranging from 1961 to 2464. As the request arrival rates increase, the average job requests vary between 2237 and 8149. Figures 3.11 and 3.12 show the results of the single-choice and multiple-choice RSU placements, respectively.

To evaluate the MCRC algorithm, we increase the rate by which vehicles generate their requests from 0.0025 requests per time slot to 0.015. All vehicles have the same request arrival rate and each request has a size of 8. The results are similar to that of Section 3.6.2 and the same arguments apply here. There is only one significant difference, i.e., the rate that the service cost increases. The service cost increases almost linearly with the request arrival rate, since there is more flexibility for load balancing. In Section 3.6.2, the request arrival rate was fixed and we increased the

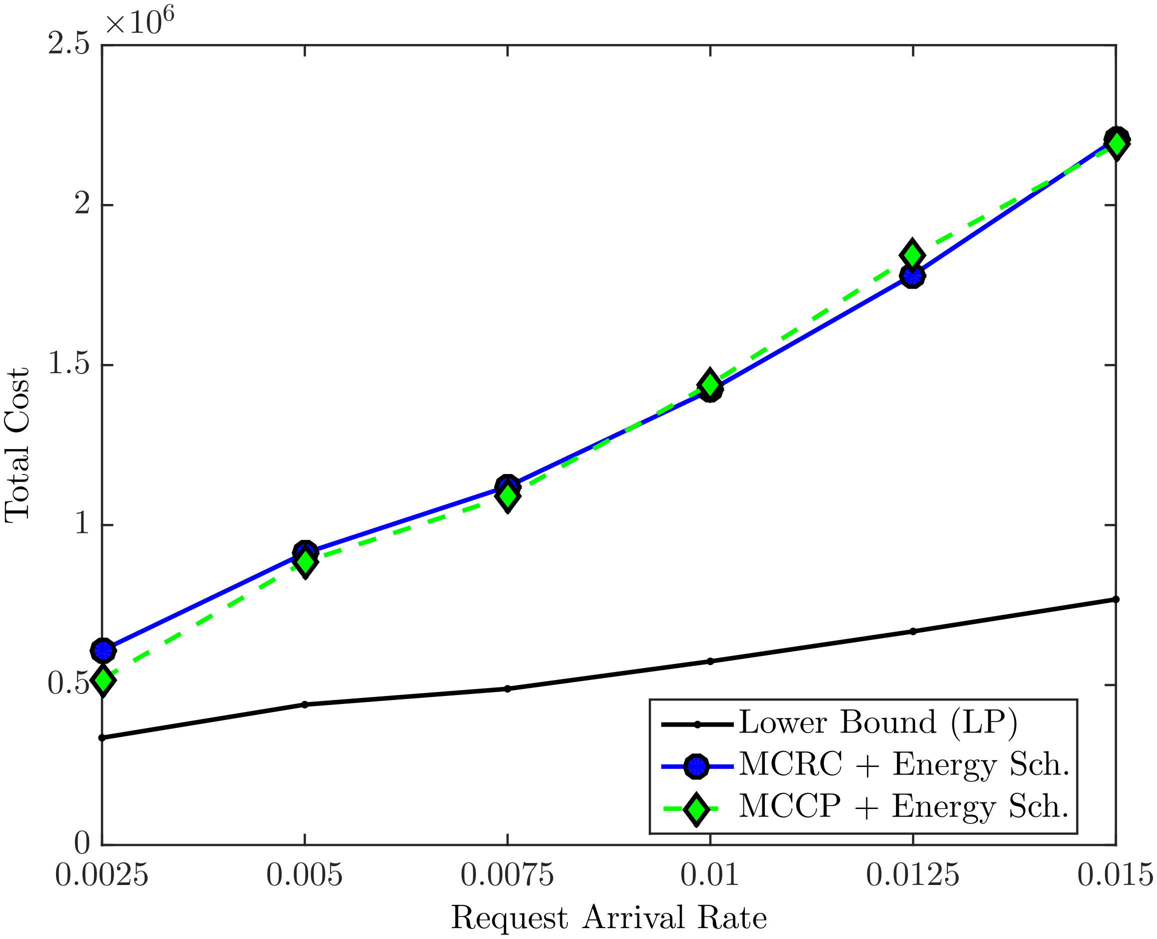


Figure 3.11: The Effect of Request Arrival Rate on Single-Choice RSU Placement with High Vehicle Traffic Load.

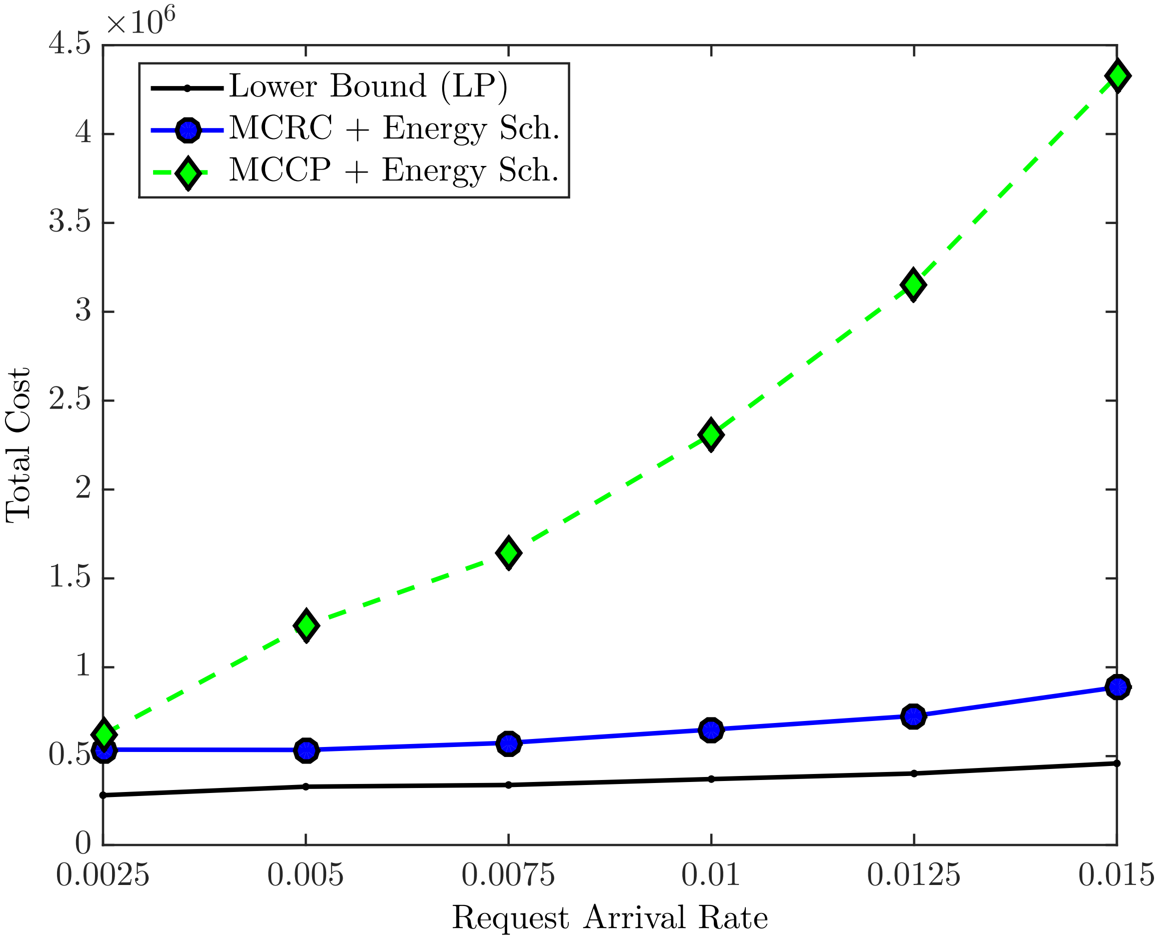


Figure 3.12: The Effect of Request Arrival Rate on Multiple-Choice RSU Placement with High Vehicle Traffic Load.

size of the requests. However, in this case we fix the request size and increase the request arrival rate.

3.6.4 The Effect of the Request Time-to-Live (Deadline)

These results are presented in Figures 3.13 and 3.14. The single RSU capital cost factor is 10, the vehicle arrival rate is 2 per time slot, the request arrival rate is 0.0125 requests per time slot, and the request size is 8. As in Section 3.6.2, 10 vehicular traces are used, consisting of vehicular arrival numbers ranging from 1961 to 2464 and 6545 to 8760 job requests.

In this section, we change the request time-to-live (TTL) from 20 to 160 time slots. It can be seen from Figures 3.13 and 3.14 that as TTL increases, the RSU deployment cost, mainly because of the service cost, decreases. At the lowest TTL value, both algorithms in both the single-choice and multiple-choice RSU placement case, open RSUs at all candidate site locations. This is because the short TTL does not allow any request transfer between RSUs for the purpose of load concentration. In this case, the request TTL is shorter than the travelling time of a vehicle inside the coverage area of an RSU. However, as requests become more delay tolerant, the MCRC algorithm transfers more requests between RSUs, so that it can both reduce the number of opened RSUs and also serve more requests at energy favourable positions. As a result, the opening cost and the service cost, and consequently, the total cost of RSU deployment decreases. Similar to the previous sections, the MCRC algorithm shows its advantage in the multiple-choice RSU placement by selecting more solar-powered RSUs over grid-powered RSUs.

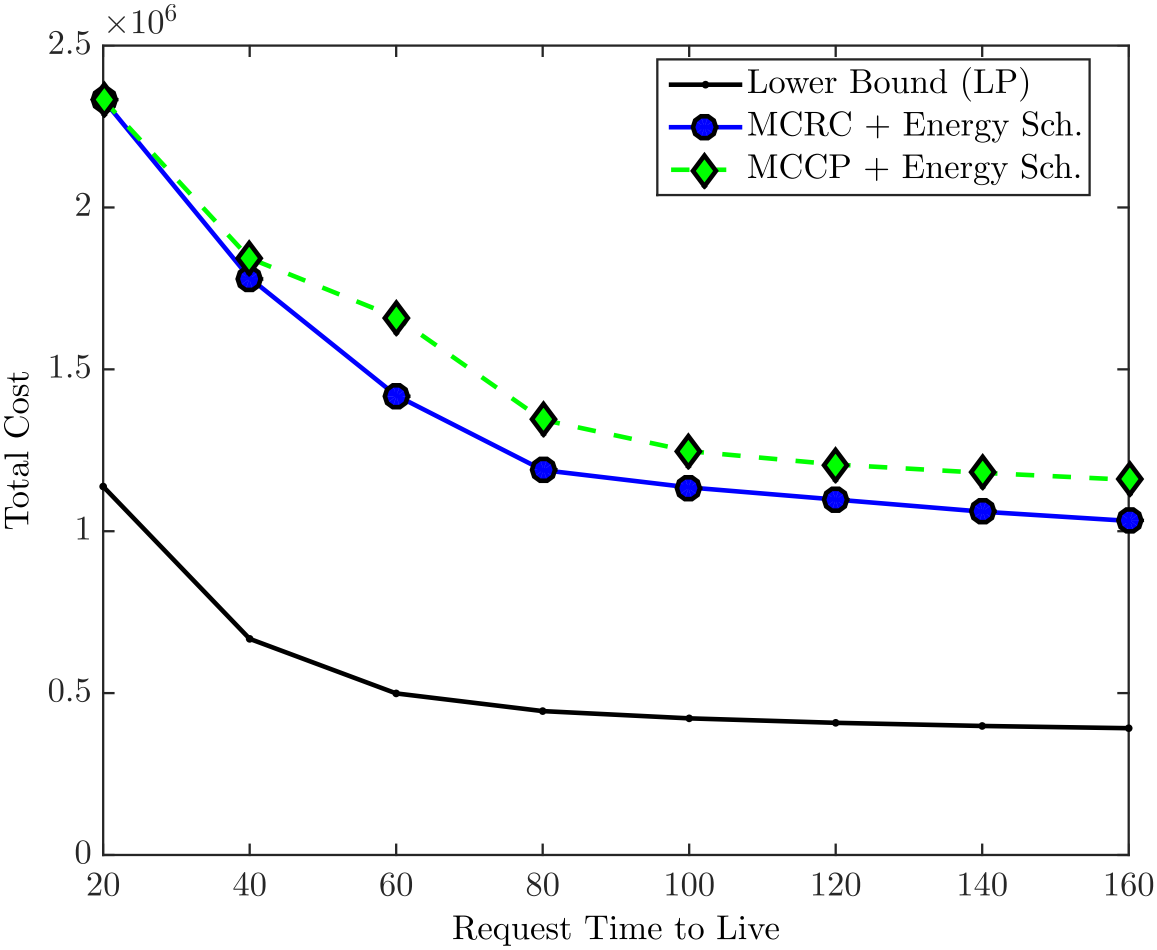


Figure 3.13: The Effect of Request Time-to-Live on Single-Choice RSU Placement with High Vehicle Traffic Load.

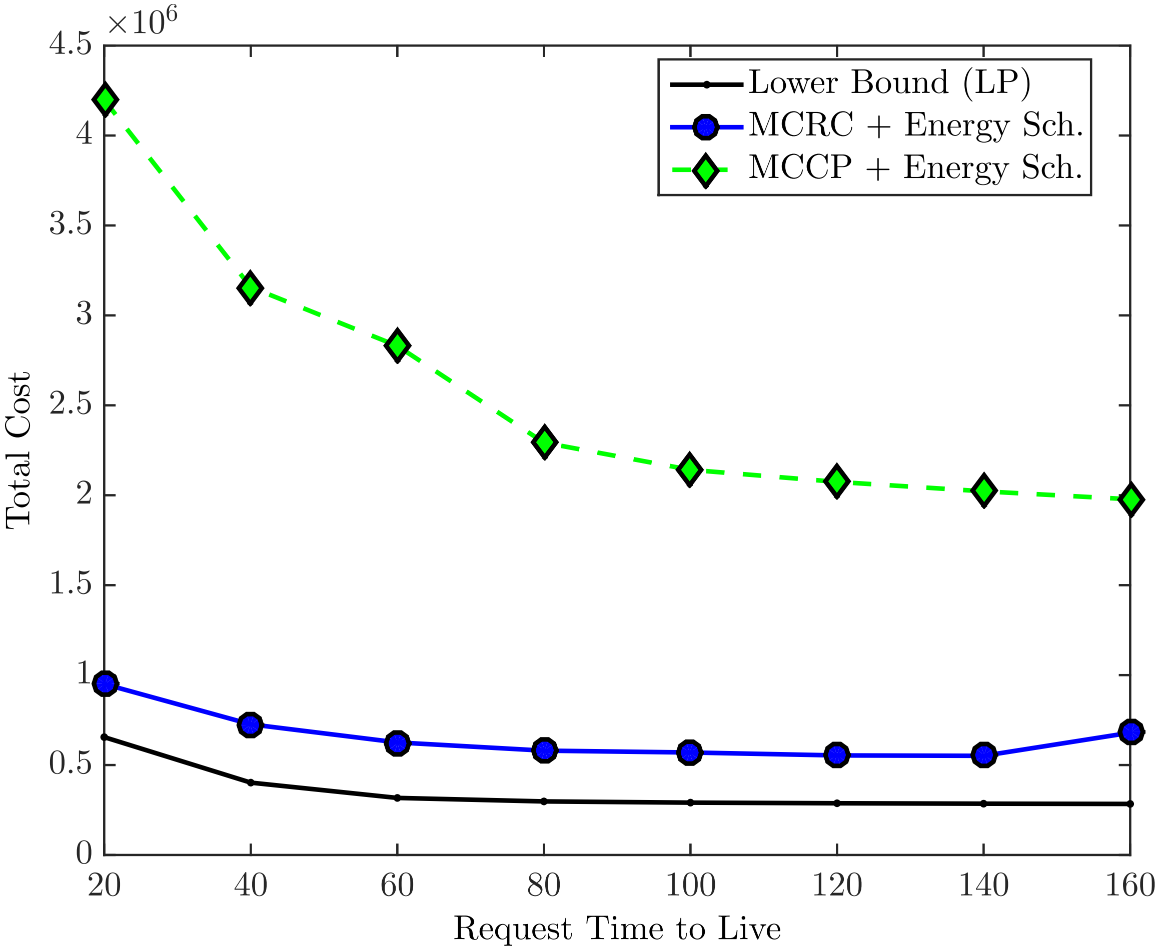


Figure 3.14: The Effect of Request Time-to-Live on Multiple-Choice RSU Placement with High Vehicle Traffic Load.

3.6.5 The Effect of the Vehicle Arrival Rate

The effect of the vehicle arrival rate on the RSU placement algorithms was briefly discussed in Section 3.6.1. In this section, we further evaluate this effect. The single RSU capital cost factor is 10, the vehicle arrival rate is 2 per time slot, the request arrival rate is 0.0125 requests per time slot, the request size is 8, and the request TTL is 40 time slots. Figures 3.15 and 3.16 show the results for the single-choice and multiple-choice RSU placements, respectively. The vehicle arrival rate is changed from 0.5 vehicles per time slot to 2.5 vehicles per time slot. As before, 10 vehicular traces are used. Since our goal in these experiments is to investigate the effect of vehicle arrival rate on our algorithm performance, the average vehicular arrival numbers in these traces vary between 515 to 2553 consisting of job requests between 1504 and 9599.

It can be seen in Figures 3.15 and 3.16 that the MCRC algorithm is as good as MCCP in single-choice RSU placement, but not in the multiple-choice case. Similar to the previous results, when the data traffic load is low, the difference between the fractional solution and the rounded solution pushes the opening cost of the MCRC algorithm to higher values. However, as the vehicle arrival rate increases, it balances the load between more RSUs. Specifically, in Figure 3.16, as the service cost increases, the MCRC algorithm switches to more solar-powered RSUs, which brings down the service cost and consequently, improves the total cost of the deployment. The results also show that the request drop ratio increases rapidly after the vehicle arrival rate of 1.5 vehicles per time slot. Also, after 2 vehicles per time slot, the offline LP starts to drop more requests. Consistently with the previous cases, the MCRC algorithm has again a smaller request drop ratio than the MCCP algorithm.

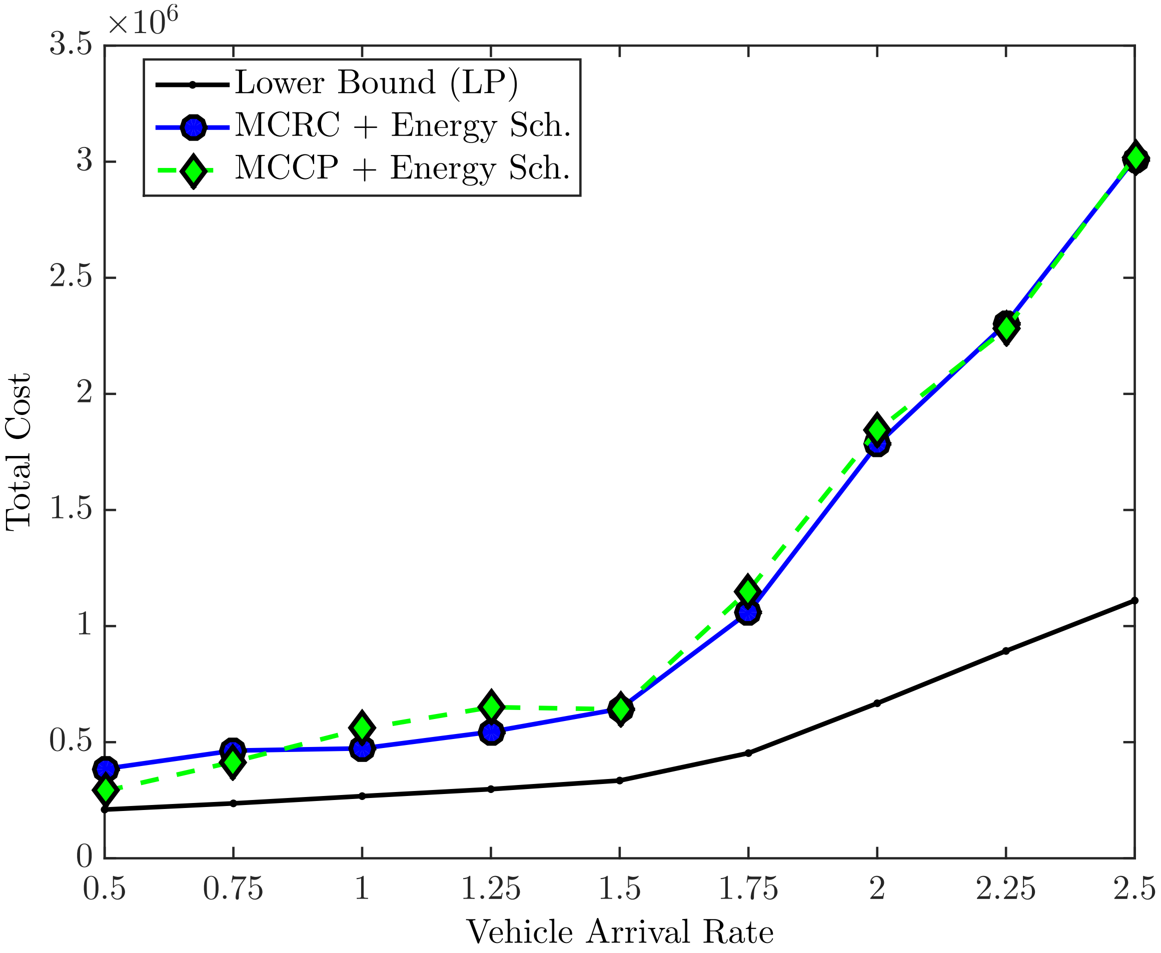


Figure 3.15: The Effect of Vehicle Arrival Rate on Single-Choice RSU Placement.

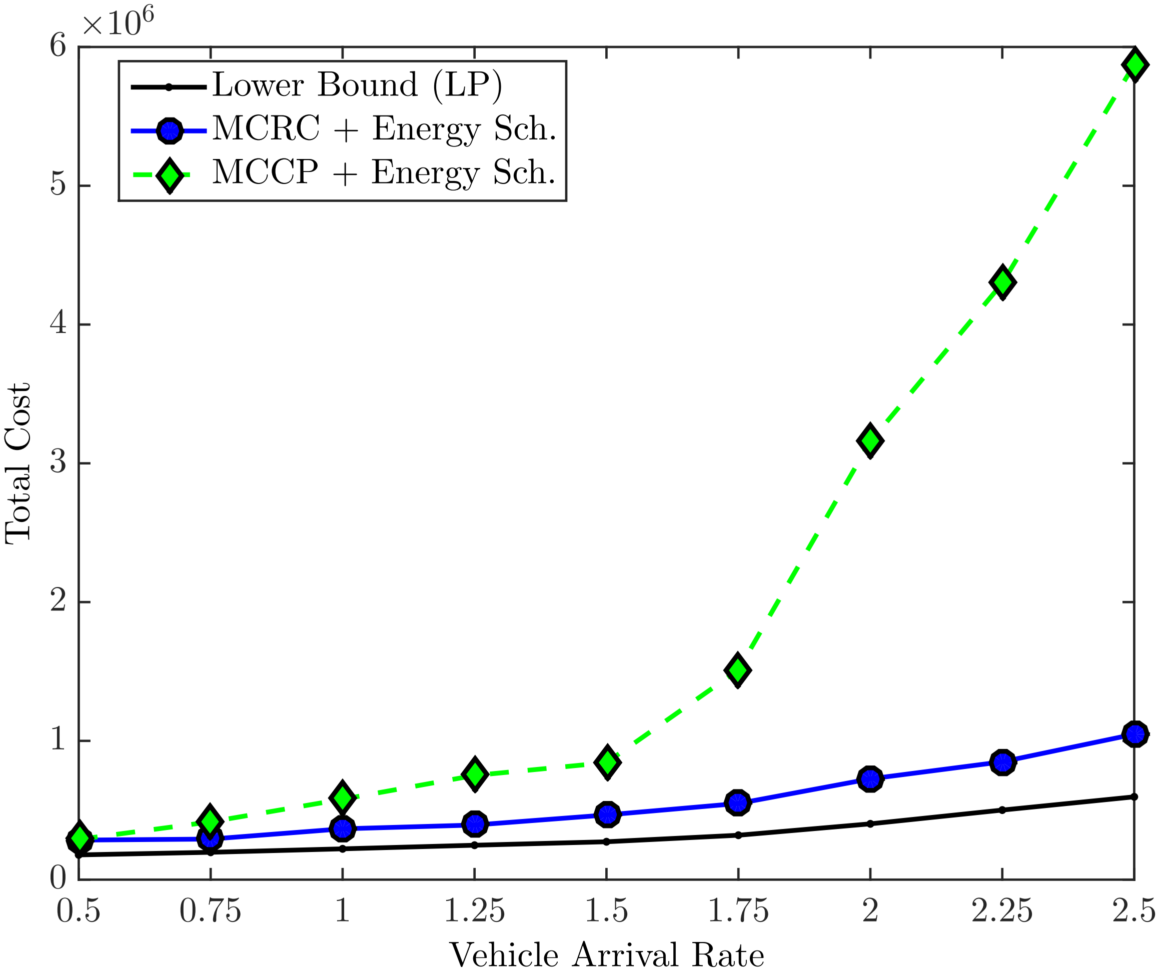


Figure 3.16: The Effect of Vehicle Arrival Rate on Multiple-Choice RSU Placement.

3.7 Conclusions

In this chapter, we have considered the problem of road-side unit (RSU) placement so that the sum of installation and operating costs is minimized. In this type of system, the total cost is a function of both capital expenditure (CAPEX) installation/opening costs, and long-term energy operating (OPEX) costs. An integer linear program (ILP) was first formulated that gives the minimum cost placement using a given set of inputs. This was used as a lower bound on total cost and is attainable for small problem sizes where the solution complexity is reasonable. To address larger problems, an algorithm was then proposed that solves a relaxed version of the ILP and uses a novel rounding procedure to obtain RSU placements, referred to as Minimum Cost Route Clustering (MCRC). The placement decisions take into account the service costs associated with the energy used to operate the RSUs, which is done using a minimum energy online scheduler.

The performance of the MCRC algorithm was investigated in different scenarios, where per RSU capital cost, request parameters (such as arrival rate, size, and time-to-live), and vehicle arrival rate were changed. MCRC was compared with the Minimum Capital Cost Placement (MCCP) algorithm that generates RSU placements that only minimize capital costs. As was discussed in Section 3.6, MCRC outperforms MCCP in RSU deployment cost, and it has less request drop ratio. In contrast to the MCCP algorithm, which is insensitive to the service cost, the MCRC algorithm creates a balance between the opening and service cost components. As a result, for different per RSU capital costs, MCRC outperforms MCCP through load concentration and load balancing, whichever is more appropriate. As we increase the size of the vehicle requests, the MCRC algorithm shows a slight advantage over the MCCP algorithm. However, in multiple-choice RSU placement, the MCRC algorithm significantly outperforms MCCP. Similar results were found as the request arrival rate and vehicle arrival rate were increased. By increasing the request time-to-live (i.e., extending the request deadline), the RSU deployment cost decreases through better load concentration. The MCRC algorithm also shows its advantage in the multiple-choice RSU placement case by selecting more solar-powered RSUs over grid-powered RSUs, where appropriate.

Chapter 4

Capacity Augmentation in Energy Efficient Vehicular Roadside Infrastructure

4.1 Introduction

Many future in-vehicle applications will be enabled using vehicular ad-hoc communications and networking. Roadside infrastructure is a key component of these systems and will eventually provide a platform for new local vehicular services. In these types of systems, road-side unit (RSU) costs include that of both RSU installation, i.e., CAPEX (capital expenditure) costs, and long-term operating, i.e., OPEX (operating expenditure) costs. The latter costs include those associated with wired energy usage over long time periods (Farbod and Todd, 2007; Badawy *et al.*, 2010). An RSU deployment that minimizes the sum of these cost components must jointly consider both the initial RSU placement and their associated long-term service costs (Nikookaran *et al.*, 2017).

A characteristic of many network design methods is that they do not take into account the *causality* present in the traffic design traces that are used for the offline design, i.e., they consider the set of design trace requests as known at the beginning of the design process, and then proceed into designing a network that can accommodate them. This is typically done by solving an Integer Program optimization. As a result, and because of the causal nature of the stream of incoming requests, the causal online scheduling during the operational life of the network may be suboptimal. In this chapter, we consider *RSU capacity augmentation* as a method of adjusting the initial network design, and to counterbalance its failure of taking causality into account. By capacity augmentation we mean the upgrade of radio capacity for RSUs that have already been placed.

Capacity augmentation in general, is not a novel idea. Once RSUs have been deployed for example, capacity augmentation is needed to update the system, thus accommodating evolving traffic conditions. Similarly, the output of an RSU design placement algorithm may not meet the packet loss targets specified in the original design specifications, which is the case considered in this chapter. RSU capacity augmentation can be used in this case to provision the RSUs, thus meeting their original performance objectives. This work introduces a procedure referred to as the Capacity Augmentation (CA) Algorithm that can be used to perform this function.

We are given, as input, the RSU locations and their initial radio capacities, as well as any set of historical vehicular traffic flow traces used by the design algorithm, i.e., the (*design traces*). These traces are representative of the expected traffic flow to be accommodated by the augmented design. The objective of the design is to obtain a minimum total cost RSU radio capacity assignment that meets a given packet loss rate target, and subject to packet deadline constraints. The CA Algorithm iterates over the RSUs, selecting candidates for capacity augmentation based on a combination of the RSU loss rate sensitivities and their capacity augmentation costs. The selection of the RSU whose capacity is to be augmented is done in every iteration by running the request scheduling algorithm on the design traces, treating them *as an online (causal) input*, i.e., under actual operational conditions. The CA Algorithm terminates when the request drop ratio improvement is below a preset threshold.

The intuition behind the CA Algorithm is to trade off CAPEX (paying for the extra capacity) for decreasing the drop ratio during operation. A variety of results is presented that characterize and compare the performance of the CA Algorithm using a simple greedy online packet scheduler. The comparisons also use energy-optimal offline scheduling obtained by solving an integer linear program formulation. It is shown that the CA Algorithm achieves a very significant decrease of the drop ratio with only a very moderate (if any) increase of the network total cost.

The remainder of the chapter is organized as follows. Section 4.2 briefly overviews the related work. In Section 4.3, a detailed description of our system model is presented. Then, in Section 4.4, a heuristic algorithm referred to as capacity augmentation (CA) is introduced. Performance results are presented and discussed in Section 4.5. Finally, this chapter is concluded in Section 4.6.

4.2 Related Work

Capacity augmentation has been previously studied for a variety of different networking scenarios. Reference (Ashraf, 2015) for example, discusses capacity augmentation in wireless mesh networks in order to maximize the aggregate throughput for all network flows, and in Reference (Ahdi and Subramaniam, 2016), augmentation is proposed using free-space optical (FSO) links to enhance the capacity of RF wireless mesh networks. Two combinatorial optimizations are used in (Lin, 1994) for link set capacity augmentation in networks supporting switched multi-megabit data service (SMDS). The goal is to determine the amount of additional capacity required and its location. The objective of the first formulation is to minimize the total routing cost subject to a budget constraint, while in the second, the total capacity augmentation cost is minimized subject to a set of shortest-path-routing constraints.

To the best of our knowledge, our work is the first that proposes a method for road-side unit capacity augmentation in vehicular networks. Our approach is unique in that the objective is to minimize the sum of capital expenditure and long-term operating costs, such that a packet loss target is achieved subject to delay deadline constraints. This is done by incorporating energy aware scheduling into the design process.

4.3 System Model

Let $\mathcal{S}$ be the set of candidate RSU locations, and $\mathcal{N}_s = \{1, \ldots, N_s\}$ be the set of RSU configurations. Different site locations are allowed to have a different set of configurations, e.g., different capacities, but at most one of these configurations can eventually be installed at each site location, and let $\mathcal{N} = \cup_{s \in \mathcal{S}} \mathcal{N}_s$. Let $\mathcal{V}$ be the set of vehicles serviced by the installed RSUs, each with a set of requests $\mathcal{R}_v$ for a total of $\mathcal{R} = \cup_{v \in \mathcal{V}} \mathcal{R}_v$ requests. Request r has a release date of a_r and a deadline (due date) of d_r. In this work, we assume that any job request of size ℓ_r time slots is splittable into ℓ_r unit-size (in time slots) requests with the same deadline, that can be serviced by different RSUs.

The problem that is addressed in this chapter is *capacity augmentation* of an existing RSU network. Once an RSU configuration (placement and capacity provisioning) is given, the scheduling of requests are done so that at most one request of each vehicle is being serviced by any RSU, each RSU serves at most one request during each time slot, and requests are serviced before their deadline in order not to be dropped.

The energy cost for servicing a request is defined by a distance-dependent exponential path-loss model with log-normal shadowing (Rappaport, 2001). The transmission power between a transmitter and a receiver, $P_{t,r}$, can be expressed by $P_{t,r} = P_{t,0}P_{sh}\left(\frac{d_{t,r}}{d_{t,0}}\right)^{\alpha}$, where $d_{t,0}$ is the reference distance, $P_{t,0}$ is the reference power at the reference distance, P_{sh} is a random variable that models the shadowing effect of the channel, α is the path loss exponent, and $d_{t,r}$ is the distance between the transmitter and the receiver. The shadowing effect of the radio channel is modeled as a random variable with log-normal distribution which has a zero mean (in dB) and a standard deviation of $\sigma_{\mathrm{dB}} = 4$.

After getting an initial placement and provisioning of an RSU network (by solving the ILP formulation in Chapter 3, for example), our capacity augmentation algorithm CA (described in the following section) is run.

4.4 Capacity Augmentation Algorithm

The starting point of our proposed algorithm is a placement of RSU's and their provisioning with capacities calculated by a placement and provisioning algorithm. Although any starting placement and provisioning can be used, in this work we will use the initial placement $\mathcal{N}_{\mathcal{O}}$ of RSU's, and capacities $U = u_n, \forall n \in \mathcal{N}_{\mathcal{O}}$ calculated by the algorithm in (Nikookaran *et al.*, 2017). The algorithm used for the on-line scheduling of vehicle demands will be referred to as the *Scheduling Algorithm*, while we will refer to our algorithm as the *Capacity Augmentation* algorithm. Throughout its running, the set of RSU locations $\mathcal{N}_{\mathcal{O}}$ will never change. In each iteration, our algorithm will increase the capacity of one RSU, and will test the new capacities, U, by running the Scheduling Algorithm: if there is no "substantial" improvement in the loss rate for the traffic case we are using, then the algorithm terminates.

More specifically, before every iteration (lines 6-24 in Algorithm 2), the Scheduling Algorithm is run with the current capacities, and its loss rate is calculated (lines 3-5 and 21-23). In case this loss rate is smaller than the target loss rate, defined by the input parameter ξ (line 6), or the loss rate improvement from the previous iteration is not more than input parameter ζ (line 8), the algorithm terminates. The loss rate improvement is defined as the decrease of the Scheduling Algorithm loss rate within a "window" of M iterations (where M is another input parameter).

For every RSU n, we calculate the *distributed loss rate* z_n as follows (line 14):

$$z_n = \sum_{r \in \mathcal{R}} \left(Z_r \cdot \frac{u_n \left|\mathcal{T}_{nr}\right|}{\sum_{k \in \mathcal{N}_{\mathcal{O}}} u_k \left|\mathcal{T}_{kr}\right|} \right), \tag{4.1}$$

where $\mathcal{T}_{nr}$ is the set of time slots during which request r can be served by RSU n. Z_r is defined to be 1 if request r is lost and zero otherwise. The idea behind this calculation is to distribute the loss of request r over all RSU's that could serve r. Each RSU n receives a fraction of r equal to the fraction of total potential (capacity unit, time slot) pairs that can be used to service r that belongs to n. Therefore, the larger the ability (more capacity and/or more time a dropped request spends within range)

Algorithm 2 Capacity Augmentation (CA) Algorithm

```
Input:
    • Placement of opened RSUs $\mathcal{N}_\mathcal{O}$, capacities $U = \{u_n, \forall n \in \mathcal{N}_\mathcal{O}\}$
    • Traffic trace with requests $\mathcal{R}$, time slots $\mathcal{T}$, and communication cost matrix $C = [c_{ntr}]_{\mathcal{N}_\mathcal{O} \times \mathcal{T} \times \mathcal{R}}$
    • Parameters $\xi, \zeta, M$
Output: Adjusted capacities $U = \{u_n, \forall n \in \mathcal{N}_\mathcal{O}\}$
$\mathcal{T}_{nr} :=$ {time slots during which request $r$ can be served by RSU $n$} for all $n, r$
$i := 0$                                                    ▷ Iteration Counter
Run Scheduling Algorithm ($\mathcal{N}_\mathcal{O}, U, R, T, C$)
$Z_r := 1$ if request $r$ is dropped, 0 otherwise, $\forall r \in \mathcal{R}$
$loss[0] := \frac{\sum_{r \in \mathcal{R}} Z_r}{|\mathcal{R}|}$          ▷ loss rate of Scheduling Alg.
while $loss[i] > \xi$ do
    if $i \geq M - 1$ then
        if $\frac{loss[i-M+1] - loss[i]}{loss[i-M+1]} < \zeta$ then
            break
        end if
    end if
    $i := i + 1$
    for all $n \in \mathcal{N}_\mathcal{O}$ do
        $z_n := \sum_{r \in \mathcal{R}} (Z_r \cdot \frac{u_n |\mathcal{T}_{nr}|}{\sum_{k \in \mathcal{N}_\mathcal{O}} u_k |\mathcal{T}_{kr}|})$
        $\delta_n :=$ (capital cost after increasing RSU $n$ capacity) - (capital cost with current RSU $n$ capacity)
        $ratio_n := z_n / \delta_n$                 ▷ loss rate per unit of cost increase
    end for
    $\mathcal{H} := \{n \in \mathcal{N}_\mathcal{O} : u_n < u_n^{\max}\}$
    $n_0 := \arg\max_{n \in \mathcal{H}} \{ratio_n\}$
    Increase $u_{n_0}$ to the next available capacity for RSU $n_0$
    Run Scheduling Algorithm ($\mathcal{N}_\mathcal{O}, U, R, T, C$)
    $Z_r := 1$ if request $r$ is dropped, 0 otherwise, $\forall r \in \mathcal{R}$
    $loss[i] := \frac{\sum_{r \in \mathcal{R}} Z_r}{|\mathcal{R}|}$
end while
```

of an RSU to service r, the higher is its responsibility for dropping r. Nevertheless, the total responsibility of RSU n for dropped requests must take into account the capital cost of increasing its capacity (to the next higher available capacity). This cost δ_n is calculated in line 15 (in case the capacity of n cannot increase further, we set $\delta_n := \infty$). The more expensive it is to increase the capacity of n, the less responsible it should be for the loss rate. Therefore, we assign to each RSU n a score $ratio_n = z_n/\delta_n$ (line 16). The RSU with the highest score is chosen (line 19), and its capacity is increased to the next available capacity level (line 20).

4.5 Performance Results

In this section, we evaluate the performance of the proposed capacity augmentation algorithm. The set of opened RSUs and their capacities $\mathcal{N}_{\mathcal{O}}, U$ are the ones resulting from solving the (offline) ILP formulation of the problem, using a given *design traffic trace* as described in Chapter 3. Given the RSU placement and provisioning, a greedy non-preemptive online scheduler is used. The scheduler tries to minimize the total service cost of scheduled job requests, by assigning each job request to the energy-wise cheapest time-slot amongst all RSU's with available capacity, as long as the deadline job constraints are met. In addition to the greedy online scheduler, in our results we will use the optimal *offline* scheduling algorithm, which produces the minimum cost schedule that satisfies the deadline constraints when the job requests are known ahead of time. The optimal offline schedule is not implementable in our setting, since the requests are not known ahead of time, but its performance is a lower bound for *any* online scheduling algorithm (not just the greedy online scheduler used here). We will see that the simple greedy scheduler we use is not far from the optimal offline scheduler in its performance, and, therefore, we will be able to assign any performance improvements to the RSU capacity adjustments performed by algorithm CA (Algorithm 2) rather than using a particular scheduler.

The performance evaluation is done using 10 vehicular traffic traces as input. The vehicular arrivals in each trace are generated by a Poisson process with a predetermined mean arrival rate (here 1.25 vehicles per second, i.e., 2.5 vehicles per time slot). Vehicles also generate job requests according to a Poisson process, with mean arrival rates uniformly selected between 0.01 to 0.02 per time slot. The sizes of vehicular requests are generated from an exponential distribution, with mean value selected uniformly between 4 and 8 time slots. Note that each request of size larger than one time slot is divided into multiple requests of size one with the same release and due dates, since we have assumed that job requests are splittable. In order to define the deadline for each request, the number of time slots from its release date to its due date (i.e., the request time-to-live) is chosen uniformly at random between 80 to 160 time slots. The traces used in our simulations are 30 minutes in duration. The number of vehicle arrivals ranges between 1705 and 2286 (with an average of 2174), and the number of total requests ranges between 28335 and 35693 (with an average of 33939). The first trace is used by both the initial placement algorithm of Chapter 3, and by algorithm CA in order to do its capacity augmentations. The reason for this choice is the fact that both algorithms are offline, run on known past traces before an RSU placement is implemented. Then, the other nine traces are used to evaluate the effects of algorithm CA on both the cost and the drop rate.

The vehicle routes were generated by using SUMO (Song *et al.*, 2014). The source and destination of vehicle trips are selected uniformly from the set of intersections, and each vehicle follows the shortest path from its source to its destination. The average travel time of each street is calculated according to its length, speed limit, and expected traffic density. Vehicles travel on a Manhattan grid with three horizontal and five vertical streets, which are all bidirectional. All intersections are controlled by traffic lights. The smallest block has a 1 km square area, which gives a total deployment region of 11.25 km^2. All RSUs are placed either at intersections or at the middle point between two intersections. There are three available RSU types, with capacity of 2, 4, or 6 respectively. All three types have a coverage range of 250 m on each side. Following (Mahdian *et al.*, 2006; Holmberg, 1994), the model we use for the CAPEX f_s of an RSU s is an affine function of capacity $f_s = f_{0s} + f_{1s} \times u_s$ for all $s \in \mathcal{N}_s$, where f_{0s} is the fixed cost for opening an RSU, and $f_{1s} \times u_s$ is the part of CAPEX which depends on RSU capacity u_s. In our simulations, we use the same CAPEX coefficients for all RSUs; these coefficients are $f_{0s} = f_0 = 4000, f_{1s} = f_1 = 2000$, in accordance with (J. A. Volpe National Transportation Systems Center, 2008; Kumrai *et al.*, 2014). This means that the CAPEX for any RSU s is $f_s = 4000 + 2000 \times u_s$.

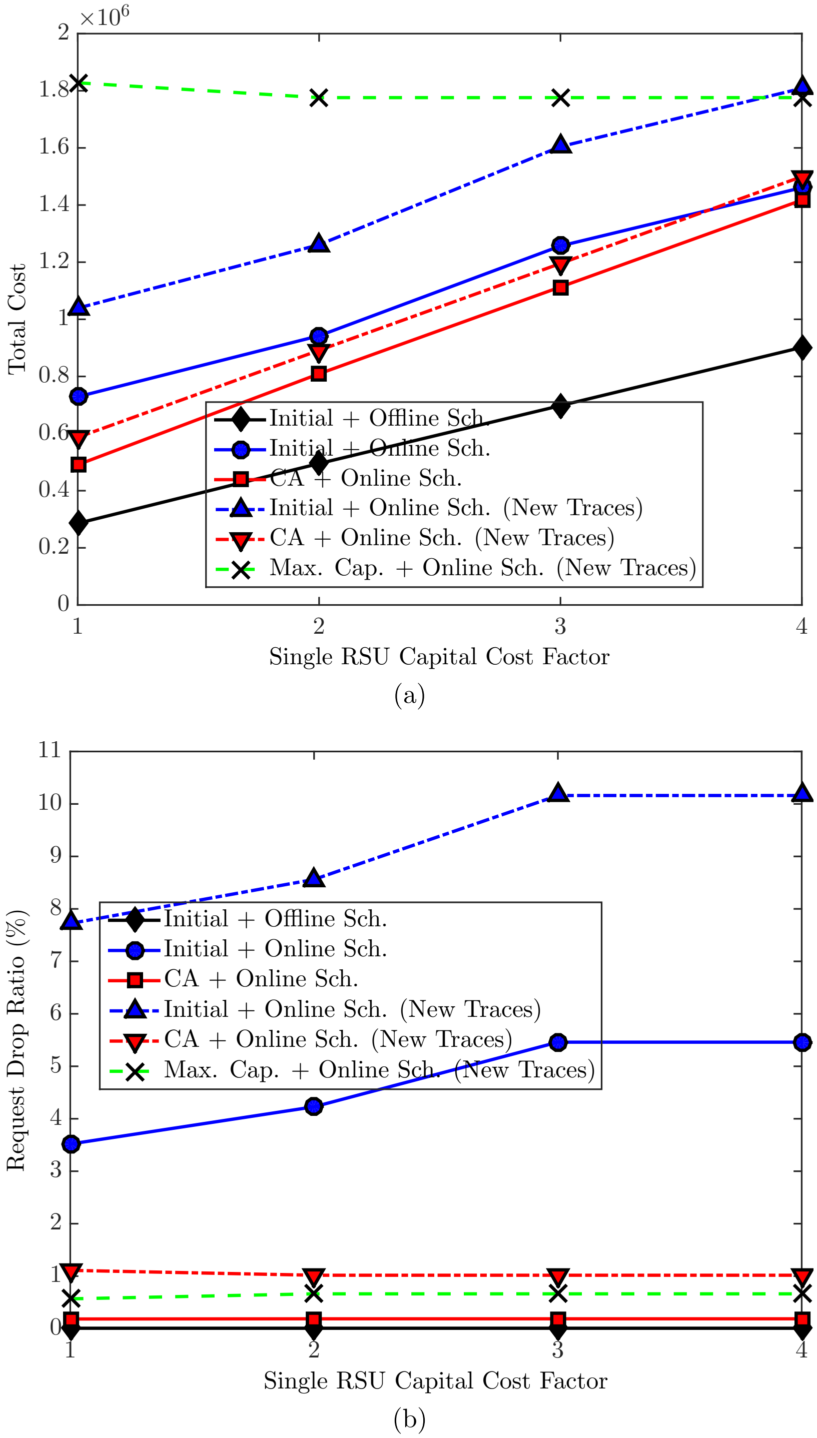
In assessing the total cost (CAPEX plus OPEX) in our simulations, we would like to study different weightings of CAPEX in relation to OPEX. In order to do that, we will multiply the CAPEX of an RSU with a factor we call the *single RSU capital cost factor*. We will use four different multiplicative single RSU capital cost factors $(1, 2, 3,$ and 4) in our simulations; obviously, the effect of CAPEX on the total cost increases as the single RSU capital cost factor increases. The goal of these experiments will be to assess the effectiveness of algorithm CA when the influence of CAPEX to the total cost ranges from lighter to heavier.

To summarize, these are the generic steps we follow in each experiment:

1. Using the first traffic trace generated as explained above, solve the ILP in Chapter 3 to calculate the RSUs placement and initial capacities.
2. Run the CA algorithm to calculate the adjusted capacities of the RSUs.
3. Run the optimal offline and the greedy online schedulers using the initial RSU configuration on the design traffic trace used in step 1.
4. Run the greedy online scheduler using the initial RSU configuration on the remaining 9 traffic traces (and average the results).
5. Run the greedy online scheduler using the CA placement on the traffic traces used in step 4 (and average the results).

Each experiment was run for each of the 4 possible single RSU capital cost factors. The rationale behind step 3 (i.e., running both schedulers on the trace used to calculate the initial RSU configuration), is the following: Using the initial placement with the optimal offline scheduler and with the design trace gives a lower bound on the performance of the initial configuration. Using the initial placement with the greedy online scheduler and with the design trace gives us an idea of how detrimental to the initial configuration is the online nature of job scheduling (although we use the design trace itself).

Figure 4.1 shows the performance of using algorithm CA to adjust the initial capacities. The total cost CAPEX+OPEX is shown in Figure 4.1a, and the drop ratio achieved is shown in Figure 4.1b. Both the total cost and the drop rate are shown for each of the 4 single RSU capital cost factors ($1, 2, 3$, and 4 on the x-axis). The number of opened RSUs for each one of these factors are 25, 24, 24, and 24, respectively.



(a)

(b)

Figure 4.1: The Effect of Capacity Augmentation Algorithm on RSU Placement Scheme.

Running the initial RSU configuration with the optimal offline scheduler and the design trace (i.e., the lower bound shown as “Initial + Offline Sch.”) is shown with a solid black line and diamond markers. Obviously, it achieves the minimum possible total cost, and the minimum drop rate. Note that the minimum drop rate is not always 0%; this is due to the fact that the requests and their deadlines generated by our random processes cannot always be serviced (i.e., there may be requests that cannot be accommodated in any schedule). The results of using the online scheduler and the design trace on the initial configuration output are denoted by “Initial + Online Sch.”. Switching to the online scheduler that has to schedule the design trace requests as they come, significantly increases the OPEX (and, hence, the total cost) and the drop ratio, as can be seen in the figure.

We compare these results with the performance of applying algorithm CA on the initial configuration, using the greedy online scheduler on the design trace, i.e., “CA + Online Sch.”. Observe that, while the total cost is similar to the cost incurred by the initial configuration, the drop ratio achieved is very close to the lower bound. This means that by investing more in CAPEX by buying more RSU capacity, algorithm CA compensates for this cost increase by reducing OPEX by a similar amount, while almost completely achieving its main goal, i.e., the reduction of drop ratio as much as possible.

While the previous results are encouraging for the use of the CA algorithm, the more important test is clearly running CA on the 9 traces that were not used in the design phase. These 9 traces were generated with the same statistics as the design trace. The averaged results of running the online scheduler with the initial configuration, and with the CA configuration are shown in Figure 4.1 denoted by “Initial

+ Online Sch. (New Traces)" and "CA + Online Sch. (New Traces)", respectively. For comparison purposes, we have added results when we maximize the capacities of all opened RSUs, i.e., the "Max. Cap. + Online Sch. (New Traces)" in Figure 4.1). These results correspond to the case of using as much capacity as possible in order to achieve the best possible drop ratio, while being oblivious to any cost increases.

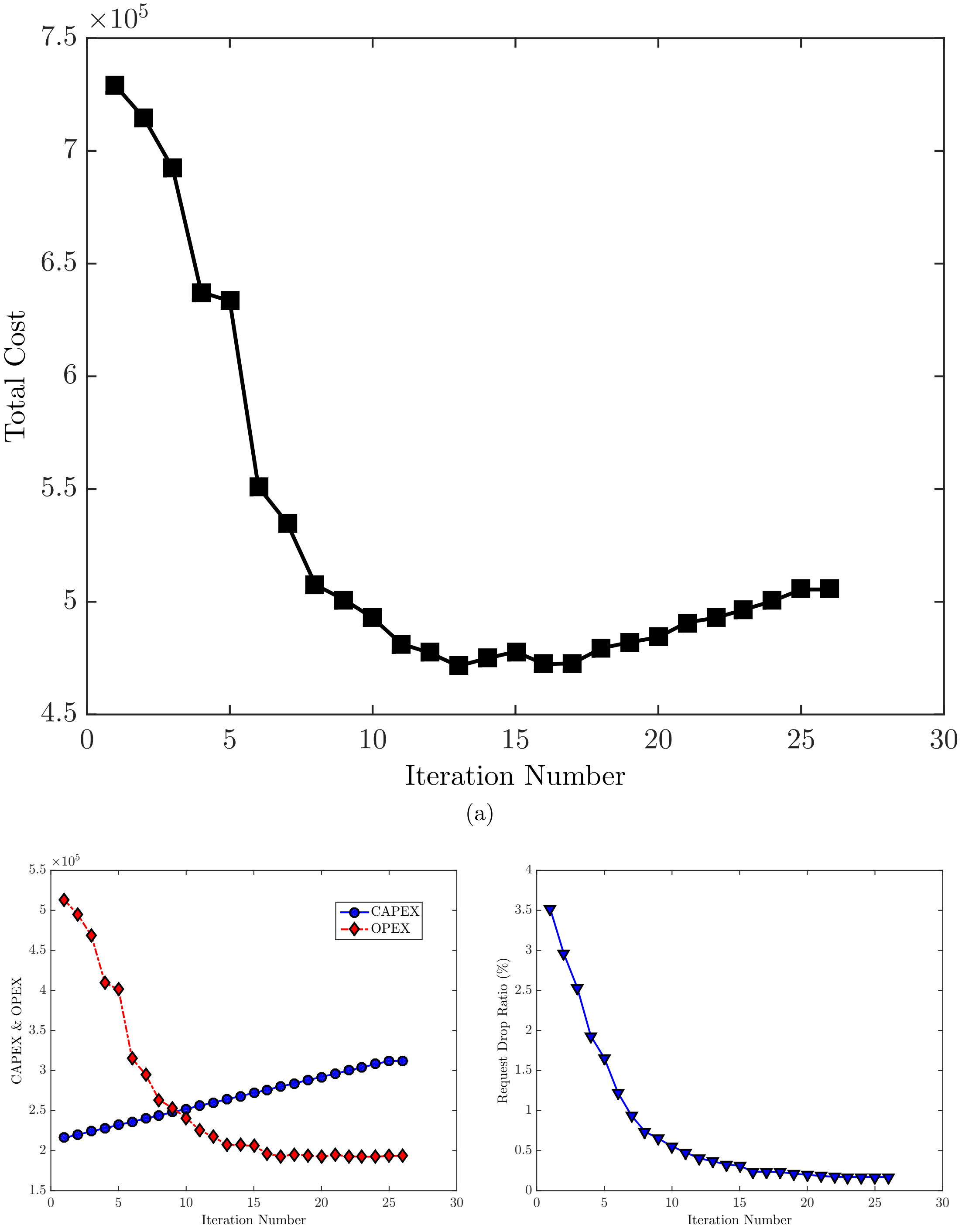
As expected, running the three different configurations (Initial, CA, and Max. Cap.) on the 9 new traces incurs larger total costs than running Initial and CA on the design trace (Figure 4.1a). Obviously, the increase for the Initial and CA configurations is due to increased OPEX costs, while the high total cost of max capacity reflects its high CAPEX cost. Note though, that the discrepancy between the latter and the costs of the Initial and CA configurations decreases as the CAPEX cost becomes more dominant in the total cost (i.e., the single RSU capital cost factor increases). This is due to the fact that as the CAPEX contribution to the total cost increases, the initial configuration opens fewer RSUs and equips them with more (up to max) capacity. Nevertheless, the cost incurred by maximizing all capacities is always significantly greater than that using the Initial or CA configurations. On the other hand, its drop ratio is very close to the lower bound (Figure 4.1b). Therefore, if one is oblivious to costs, maximizing all capacities will give the best drop ratio. If the total cost is a consideration, then Figure 4.1b shows that the CA configuration has a much smaller drop ratio than the Initial configuration, while incurring almost the same (actually smaller) cost as the Initial configuration, as seen in Figure 4.1a.

To better understand the performance of the CA algorithm, we show the performance of the configuration resulting after each one of its iterations in Figure 4.2, for the case of the single RSU capital cost factor of one (the leftmost points in Figure

4.1). The total cost and its components, i.e., OPEX and CAPEX, are shown in Figures 4.2a and 4.2b, respectively. Figure 4.2c shows the request drop ratio for each iteration. The results for the other single RSU capital cost factors is similar, and, therefore, we concentrate on the case of single RSU capital cost factor of one.

As can be seen in Figure 4.2, there are two phases in the graphs. The first phase (up to iteration 9) corresponds to a sharp decrease in the drop ratio. During this phase, by adding capacity to those RSUs with the highest impact on the drop ratio, algorithm CA creates opportunities to serve more vehicles at their favourable positions relative to RSUs (from an energy point of view), while, at the same time, it decreases the contention between requests for service. This causes both the OPEX and the drop ratio to decrease. On the other hand, increasing RSU capacities increases CAPEX. Hence, after a certain point (iteration 9), increasing capacities does not affect OPEX by much, while CAPEX continues to increase, and as a result, the total cost is increasing. During this second phase, the improvement of the drop ratio is slow, especially when compared to its rapid drop during the first phase. This is to be expected, since after a certain point the capacities of the RSUs are no longer an issue, and increasing them does not improve significantly the OPEX or the drop ratio.

In all previous experiments, the RSUs were chosen from three types, with a maximum capacity of 6. Given that vehicular networks are built with an operational horizon measured in decades, in the future it may be possible to increase RSU capacities much beyond the upper bound of 6. In order to assess the performance of algorithm CA in this case, we run it with the same CAPEX model as before, but without an upper bound on the available capacities. The results are shown in Figure 4.3. The curve "CA (Unlimited Cap.) + Online Sch. (New Traces)" shows the results



(a)

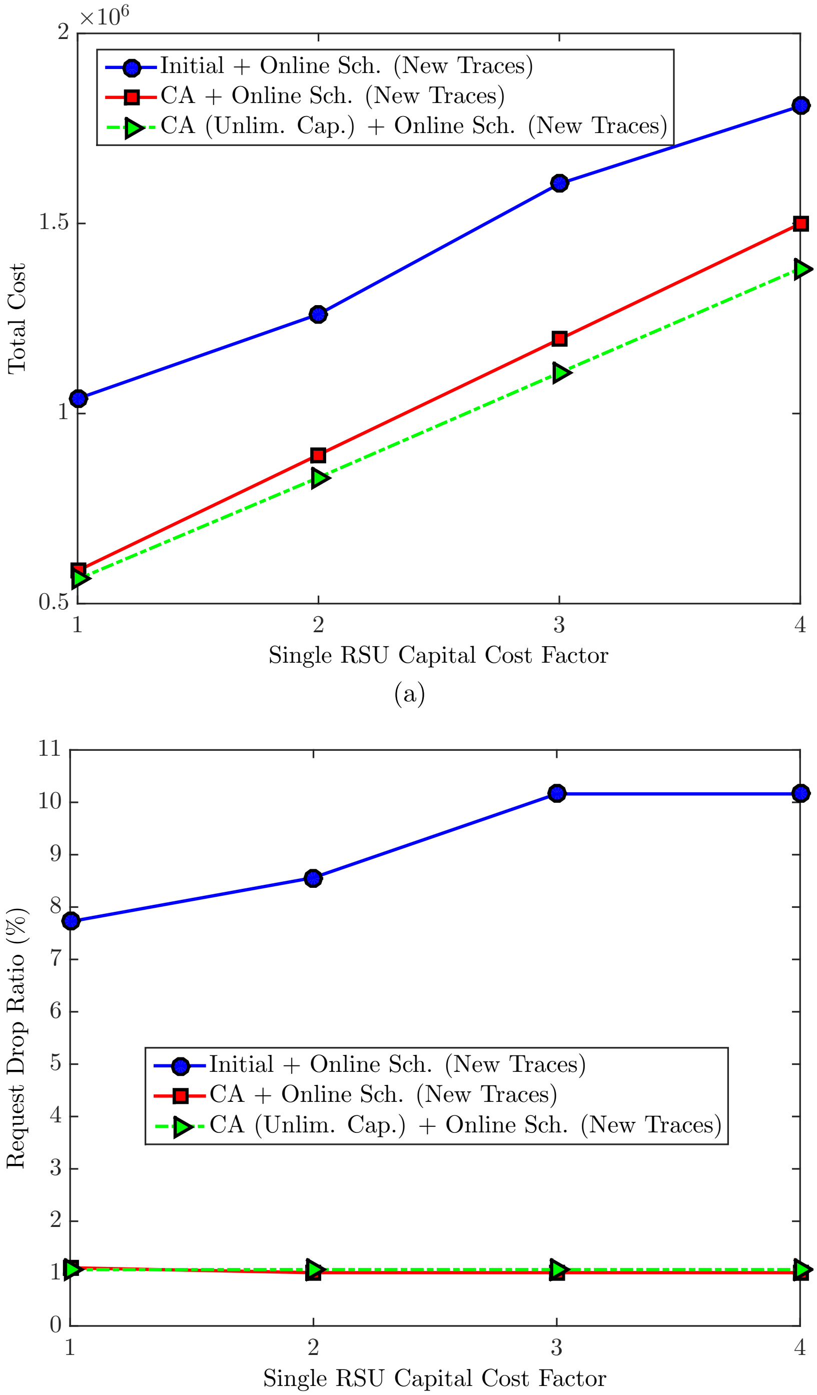
(b)

(c)

Figure 4.2: The Capacity Augmentation Algorithm Progress in each of Iteration.

of running algorithm CA without RSU capacity upper bounds. The other curves are the result of the Initial and CA configurations from Figure 4.1, where RSUs have a maximum capacity of 6. In these experiments, the highest capacities reached by algorithm CA are 8, 9, 9, and 9 for single RSU capital cost factors $1, 2, 3, 4$ respectively. Note that these maximum capacities are not much higher from the upper bound of 6 used before.

We note that algorithm CA has a higher drop ratio when run with unlimited capacities compared to the limited capacity case (although the total costs have the reverse relationship). This can be explained as follows: First, reaching the capacity upper bound on an RSU forces the algorithm to distribute its capacity increases to other RSUs in order to decrease the overall drop ratio, and as a result, this distribution of extra capacity eventually helps to service more requests within their deadlines. Second, there is the pathological situation of a vehicle generating more requests than can be serviced before the vehicle leaves the servicing RSU coverage area; therefore, an RSU can have the highest impact on the request drop ratio and, at the same time, increasing its capacity does not reduce the number of dropped requests. This causes algorithm CA to focus on the wrong RSU and to continuously increase its capacity, until it detects that the drop ratio has not improved (the length of window M in Algorithm 2) and terminates.



(a)

(b)

Figure 4.3: The Effect of Unlimited Capacity on the Capacity Augmentation Algorithm Performance.

4.6 Conclusions

This chapter has addressed the issue of capacity augmentation in energy efficient road-side unit (RSU) deployments. The objective is to find RSU radio capacity augmentation assignments that minimize the total capital expenditure and long-term operating expenditure costs. This is subject to meeting packet deadline constraints with a given packet loss rate target. An algorithm, referred to as the Capacity Augmentation (CA) Algorithm, was proposed that iterates over the RSUs, selecting candidates for capacity augmentation based on their packet loss rate sensitivities. Results were presented that characterize and compare the performance of the CA Algorithm using a greedy online packet scheduler. It was shown that the CA Algorithm is an efficient way to assign RSU radio capacity that can achieve the desired packet loss rate target while reducing the sum of operating and capital expenditure costs.

Chapter 5

Energy Efficient Roadside Unit Transmission Scheduling with Unknown Vehicle Routes

5.1 Introduction

In many applications, competing vehicular job requests may span across multiple RSU coverage areas as the vehicle travels through the coverage region (Khezrian *et al.*, 2015). When the vehicle routes are known by the scheduler (Zou *et al.*, 2011; Ali *et al.*, 2014a), then this information can be incorporated into the scheduling in a straightforward manner. This may eventually be possible once self-driving vehicle technology becomes widespread, since route information may be communicated directly to the roadside infrastructure (Paden *et al.*, 2016). However, when vehicle routes are unknown, which is currently the case, the scheduling problem becomes significantly more complicated.

In this chapter we consider the problem of roadside unit job scheduling when vehicle routes are unknown. The scheduler is given the topology of an urban road network and historical traffic traces that are used to extract vehicular motion statistics. Given this information, the scheduler must process online vehicular job requests that are subject to hard deadline constraints and incurring a small packet loss. The objective is to perform this scheduling in an energy efficient fashion, such that the long-term energy service costs, i.e., operating costs (OPEX) of the RSUs are minimized. A scheduler referred to as the Route Coverage Expectation Scheduler (RCES) is proposed. RCES uses the historical input traffic traces to estimate vehicular motion and the associated energy communication costs. RCES uses this information to schedule job requests across multiple RSUs whenever possible. This is done by scheduling part of a request on the current RSU and deferring the remainder to RSUs that the vehicle may encounter in the future. The decision on request deferment is based on the expected free capacity the vehicle will encounter in the near future. This expectation is computed over all possible routes (up to a certain length) using the provided route statistics as an input.

A wide variety of results is presented that show the performance of the proposed scheduler. In particular, we compare RCES to optimal offline scheduling, where routes are assumed to be known in advance. We also compare it with a simple greedy online scheduler, which also knows vehicle routes at the time of scheduling. This algorithm greedily assigns each job request to the energy-minimum time-slots among the RSU's with available capacity that the vehicle will encounter. RCES is also compared to the scheduler in (Ali *et al.*, 2014b), which attempts to assign all requests to the RSUs on the current street using an earlier-deadline-first (EDF) scheduling policy. The results show that deploying RCES when vehicle routes are not known by the network achieves a drop ratio similar to the drop ratio achieved when these routes are known, with only a modest increase in energy cost.

We emphasize that these results were achieved under quite restrictive constraints, since, apart from the vehicle routes that are not known, the online scheduler has to comply with the request deadlines and is not allowed to preempt already scheduled jobs. Hence, our results hold for a network satisfying few assumptions, and as a result, are fit for more practical settings. Nevertheless, it may be the case that historical data are not available, or it is not feasible to gather and store a large volume of such data. Therefore, we explore the possibility of using classic Machine Learning (ML) algorithms in order to estimate the statistical information used in scheduling. In Section 5.5 we use a Bayesian estimator (Alpaydin, 2014) to deduce vehicle turning probabilities at intersections, and use them to run RCES. It is shown that, while using estimates produces worse performance, as expected, the performance degradation is rather small. We leave the further exploration of ML techniques for future work.

The remainder of the chapter is organized as follows. Section 5.2 briefly overviews the related work. In Section 5.3, a detailed description of our system model is presented. Then, in Section 5.4, a heuristic algorithm referred to as route coverage expectation scheduler (RCES) is introduced. Performance results are presented and discussed in Section 5.5. Finally, this chapter is concluded in Section 5.6.

5.2 Related Work

Reference (Gui and Chan, 2012) introduces a motion prediction-based scheduling scheme for the case user routes are not known, in which RSUs cooperatively balance their loads by transferring part of their requests to nearby non-overloaded RSUs. Each RSU schedules requests according to their priorities (i.e., remaining valid time and unserved size) and if a request in the current queue cannot be served before its deadline, it will be transferred to the next RSU with the highest probability to encounter. If transfer is not possible, the RSU with the second highest probability will be considered. The request will be dropped if none of the RSUs can serve this request. Each request can only be transferred once.

Reference (Ali *et al.*, 2014b) enhances the model introduced in (Gui and Chan, 2012), referred to as the cooperative load balancing (CLB) scheduler. It considers the current street RSUs before considering those in the next encountered intersection. It calculates the load situation of the transferee RSU at the arrival time of vehicle whose request is about to transfer. If the request cannot be transferred to any RSU, it will be dropped. Note that, unlike our setting, preemption of a job already assigned to an RSU is allowed.

Reference (Ali *et al.*, 2014a) modifies the CLB scheduler in (Ali *et al.*, 2014b) by changing the assumption to the case that the vehicles routes are known at the time of scheduling. Therefore, it guarantees more load balancing than the CLB scheduler. However, at every request transfer, the scheduler also transfers the request to the RSU at the next intersection, in case the vehicle deviates from its route.

To the best of our knowledge, our work is the first that proposes a roadside unit scheduler that operates with unknown routes and with the objective of minimizing long-term energy operating costs under job deadlines and with small packet loss.

5.3 System Model

In this work, vehicles moving within an urban network of RSUs are generating transmission requests that have to be serviced by the RSUs. The basic premise of our work is that the route of each vehicle within the city is *not known*. In order to schedule the request transmissions, the scheduler is given the city topology (represented as a directed graph), and historical data (e.g., past traffic traces) out of which it can deduce: (i) A probability matrix $P = [p_{s,s'}]$, where $p_{s,s'}$ is the probability of a vehicle turning from street s to street s' at an intersection. (ii) The average travel times for all streets.

Let $\mathcal{N}$ be the set of given deployed RSUs with capacities $\mathcal{U} = \{u_n, \forall n \in \mathcal{N}\}$ and $\mathcal{V}$ be the set of vehicles serviced by the deployed RSUs, each with a set of requests $\mathcal{R}_v$ for a total of $\mathcal{R} = \cup_{v \in \mathcal{V}} \mathcal{R}_v$ requests. Request r has a release date of a_r and a deadline (due date) of d_r. In this work, we assume that any job request of size ℓ_r time slots is splittable into ℓ_r unit-size (in time slots) requests with the same deadline, that can be serviced by different RSUs.

Let $\mathcal{T}$ be the set of time slots; within a time slot, RSU n has the capacity to transmit to at most u_n vehicles, and a vehicle can communicate with at most one RSU during a time slot. Scheduling a unit-size request r means the assignment of r to an RSU n for service during a time-slot t. In our setting, scheduling is *non-preemptive*, i.e., interrupting the service of a request, or changing the RSU and time-slot assignment of a request is not allowed. When vehicle v is within the coverage area of RSU n during time-slot t, the energy cost for servicing request r is c_{ntr}, defined by a distance-dependent exponential path-loss model with log-normal shadowing (Rappaport, 2001), otherwise $c_{ntr} = \infty$. The transmission power between a transmitter and a receiver, $P_{t,r}$, can be expressed by $P_{t,r} = P_{t,0} P_{sh} \left(\frac{d_{t,r}}{d_{t,0}}\right)^{\alpha}$, where $d_{t,0}$ is the reference distance, $P_{t,0}$ is the reference power at the reference distance, P_{sh} is a random variable that models the shadowing effect of the channel, α is the path loss exponent, and $d_{t,r}$ is the distance between the transmitter and the receiver. The shadowing effect of the radio channel is modeled as a random variable with log-normal distribution which has a zero mean (in dB) and a standard deviation of $\sigma_{\text{dB}} = 4$.

5.4 The Route Coverage Expectation Scheduler (RCES)

The input to our proposed algorithm RCES, shown as Algorithm 3, is a road network with RSUs installed, vehicular traffic statistics (mean traveling time of streets and turning probabilities at intersections) based on historical traffic data, and a communication cost matrix. New vehicular job requests that need to be scheduled arrive in an on-line fashion. Algorithm RCES greedily finds the time-slots of minimum communication cost to serve requests, trying to minimize the total energy cost of scheduling. Note that the algorithm does not make any assumptions about the vehicle routes. RCES schedules part of a request on the RSU which currently covers the vehicle's current location (if such an RSU exists), and leaves the rest to be serviced by RSUs that the vehicle may encounter in the future. Any portion of the first part that could not be scheduled before the vehicle leaves an RSU coverage area, will be postponed along with the second part. The decision on how requests should be divided is based on the expected value of free capacity the vehicle will encounter in the next $\xi_{\max}$ streets or before its request expires (whichever happens first), where $\xi_{\max}$ is a preset algorithm parameter upper-bounding its lookahead. The expectation is calculated over all possible routes within these limits, using the probabilities given as an input. In our experiments we set $\xi_{\max} = 3$, as a trade-off between accuracy and efficient computability.

Algorithm 3 Route Coverage Expectation Scheduler (RCES)

```
Input:
    • Deployed RSUs, initial capacities
    • Incoming requests $\mathcal{R}$, time slots $\mathcal{T}$, and communication costs for all requests,
      RSUs and time-slots
    • City graph $G$, street traveling times, intersection turning probability matrix $P$
Output: Online schedule of all non-dropped requests
 1: for all $t \in \mathcal{T}$ do
 2:     $\mathcal{R}^t$ = set of postponed or released at time $t$ requests
 3:     Drop from $\mathcal{R}^t$ requests whose deadlines cannot be met
 4:     while $\mathcal{R}^t \neq \emptyset$ do
 5:         $\mathcal{R}_d^t := \{$same vehicle, RSU requests in $\mathcal{R}^t$ with smallest deadline$\}$
 6:         $\gamma_n :=$ overall available capacity in RSU $n$
 7:         $\Gamma_i :=$ expected overall available capacity in route $i$, $\forall$ routes $i$
 8:         $ratio_n := \frac{\gamma_n}{\gamma_n + \sum_{\text{route } i} \Gamma_i}$
 9:         $\mathcal{P}_d :=$ first $(ratio_n \cdot |\mathcal{R}_d^t|)$ requests in $\mathcal{R}_d^t$
10:         Schedule requests in $\mathcal{P}_d$ in current RSU
11:         Postpone requests in $\mathcal{R}_d^t \backslash \mathcal{P}_d$
12:     end while
13: end for
```

More specifically, set $\mathcal{R}^t$ contains the requests released during time-slot t, together with all other requests that were previously postponed (line 2). If any request expires before it can be served, it will be dropped, otherwise the request will be added to $\mathcal{R}^t$ when the vehicle announces its arrival to an RSU coverage area. In lines 4-12, RCES goes through all requests in $\mathcal{R}^t$ and either schedules them or postpones them. We use an earlier-deadline-first policy to pick all requests in $\mathcal{R}^t$ with the same earliest deadline, that were submitted to the same RSU by the same vehicle (line 5). In line 6, γ_n is the overall available capacity of the RSU currently covering the vehicle, while in line 7, Γ_i is the expected overall available capacity in route i following the current RSU. By 'overall available capacity' we mean the total number of time slots of capacity available for the request we are currently serving (for example, if the vehicle will be in an RSU's coverage area during time slots 3, 4, 5, and 6, and during these time slots the RSU has available capacity 2, then the overall available capacity in this RSU is 8). In line 8, we calculate the ratio $ratio_n$ by which we partition the set $\mathcal{R}_d^t$ into two parts, $\mathcal{P}_d$ and $\mathcal{R}_d^t \backslash \mathcal{P}_d$. We postpone the requests in $\mathcal{R}_d^t \backslash \mathcal{P}_d$, and schedule the ones in $\mathcal{P}_d$, if possible (lines 10-11).

Ratio $ratio_n$ (line 8) is the ratio of the overall capacity of the current RSU γ_n over the expected overall capacity in all possible routes a vehicle may follow, including the current RSU. To calculate the denominator, we enumerate all possible routes of length at most $\xi_{\max}$ or the request deadline (whichever comes first), and use the given traffic statistics. The intuition behind this ratio, is the obvious motivation of assigning to an RSU a portion of total remaining requests that is proportional to the available capacity of this RSU over the expected available capacity the vehicle is (or is going to be) encountering.

5.5 Performance Results

In this section, we evaluate the performance of algorithm RCES. We compare it with the optimal *offline* scheduling algorithm, which produces the minimum cost schedule that satisfies the deadline constraints when the job requests are known ahead of time and the vehicle routes are *known*. Obviously this optimal schedule is not implementable in our setting, since the requests and the vehicle routes are not known ahead of time, but its performance is a lower bound for *any* online scheduling algorithm (not just the online scheduler used here). We also compare it with a simple greedy online scheduler that knows the routes of vehicles at the time it schedules their requests. It greedily assigns each job request to the energy-wise cheapest time-slot among RSU's with available capacity that this vehicle encounters before the request deadline. We use this simple algorithm as a benchmark for the advantages gained when the vehicle routes are known, even when the simplest possible scheduler is used.

We compare the results of our proposed algorithm with an alternative scheduler based on the algorithm of (Ali *et al.*, 2014b) that attempts to assign all requests to the RSUs located in the current street where a vehicle is traveling in a balanced manner, and following the earlier-deadline-first (EDF) scheduling policy. We refer to this algorithm as the *Cooperative Load Balancing (CLB)* algorithm. To have a proper comparison, we have adapted algorithm CLB to our setting. Hence, the implemented CLB algorithm is non-preemptive, since preemption (i.e., interrupting the service of a request, or changing the RSU and time-slot assignment of a request) is not allowed. Also, and unlike (Ali *et al.*, 2014b), requests for CLB can be generated even when vehicles are outside the RSUs coverage area, but vehicles communicate their job requests to the first RSU they encounter. Requests can have different sizes (multiple time-slots), but we assume they are splittable into multiple unit-size (in time slots) requests with the same release and due dates, that can be served by different RSUs. We use a time window of 5 time slots and $\beta = 0.25$ for the exponential weighted moving average used in algorithm CLB.

The performance evaluation is done using 10 vehicular traffic traces as input. The vehicular arrivals in each trace are generated by a Poisson process with a predetermined mean arrival rate (1.25 vehicles per second, i.e., 2.5 vehicles per time slot). Vehicles also generate job requests according to a Poisson process, with mean arrival rates uniformly selected between 0.01 and 0.02 per time slot. The sizes of vehicular requests are generated by an exponential distribution, with mean value selected uniformly between 4 and 8 time slots. In order to define the deadline for each request, the number of time slots from its release date to its due date (i.e., the request time-to-live) is chosen uniformly at random between 80 to 160 time slots. The traces used in our simulations are 30 minutes in duration. The first trace is used to calculate the statistics used in our algorithm, i.e., the turning probabilities at intersections and the mean traveling time of each street in the given city topology. We also use the first trace to calculate the initial RSU placement. Then, the other nine traces are used to evaluate the effects of the proposed scheduling algorithm on both the cost and the drop rate. The average results from these nine traces are presented here.

When they are assumed to be known, vehicle routes are generated by using SUMO (Song *et al.*, 2014). The origin and the destination of vehicle trips are selected uniformly from the set of intersections, and each vehicle follows the shortest path from its origin to its destination. The average travel time of each street is calculated according to its length, speed limit, and expected traffic density. Vehicles travel on a Manhattan grid with three horizontal and five vertical streets, which are all bidirectional. All intersections are controlled by traffic lights. The smallest block has a 1 km square area, which gives a total deployment region of 11.25 km^2. All RSUs are placed either at intersections or at the middle point between two intersections. Three RSU types with capacity of 2, 4, or 6 are used. All deployed RSUs have coverage range of 250 m on each side.

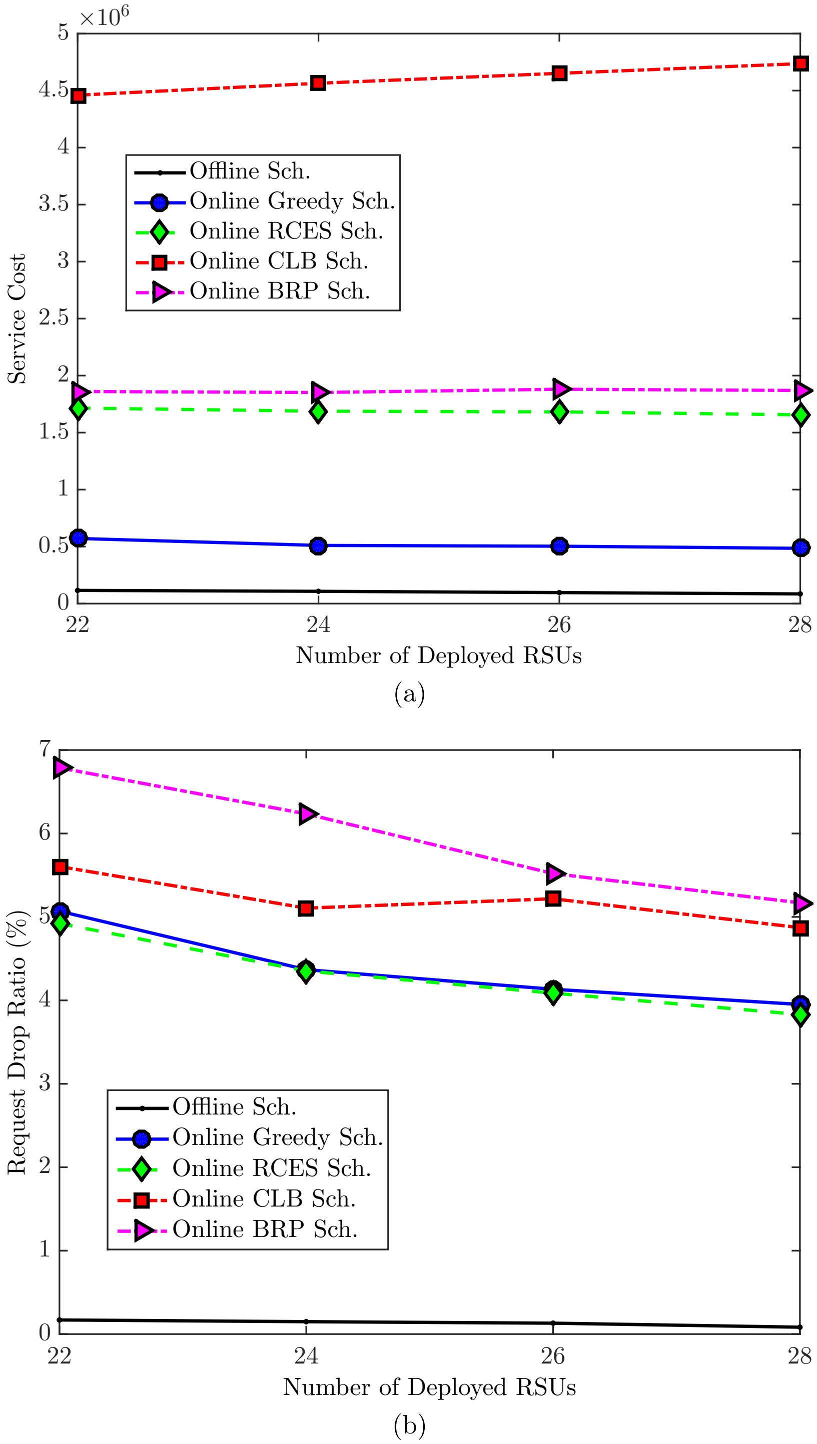
In all experiments, the results shown are the ones of the optimal offline scheduler, the greedy online scheduler with known routes, as well as algorithms RCES and CLB.

5.5.1 The Effect of the Number of Deployed RSUs

In the first set of experiments (Figures 5.1-5.2), we investigate the effect of the number of deployed RSUs on the algorithms.

The horizontal axes show the number of deployed RSUs and the vertical axes show either the total cost of scheduling or the percentage of dropped requests. We consider two different RSU placement policies. First, we use the RSU placement algorithm of Chapter 3 (MCRC algorithm). The number of RSUs is 22, 24, 26, and 28 with a total capacity of 60, 66, 68, and 70, respectively. These results are shown in Figure 5.1. Second, for the results shown in Figure 5.2, RSUs are placed at all intersections (i.e., 15 RSUs) and 4 additional streets are randomly chosen to randomly place additional RSUs. Thus we get cities with 19, 23, and 27 RSUs, respectively. All these RSUs have a capacity of 2.

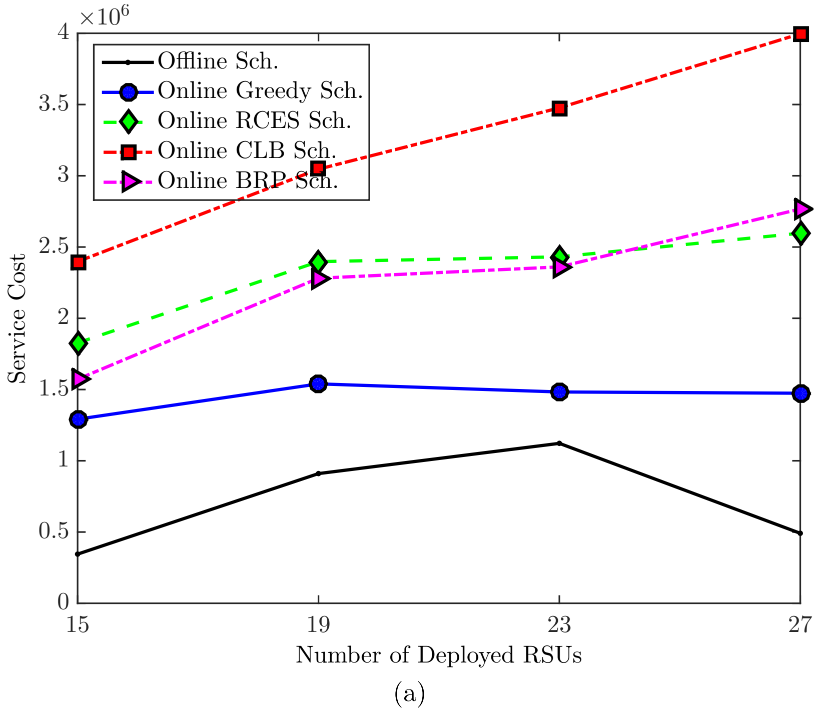
As shown in Figures 5.1b, 5.2b, the drop ratio for all algorithms are close to each other (and far from the lower bound). While this is not surprising for algorithms RCES and CLB, and can be attributed to the fact that these algorithms do not know the vehicle routes thus missing future opportunities for assigning requests to available capacity, it is rather surprising that the greedy scheduler that knows the routes performs similarly. This is due to the objective of the latter algorithm, which is the minimization of service cost without much regard to how many requests are dropped. Note that our algorithm RCES has a lower drop ratio than CLB, since



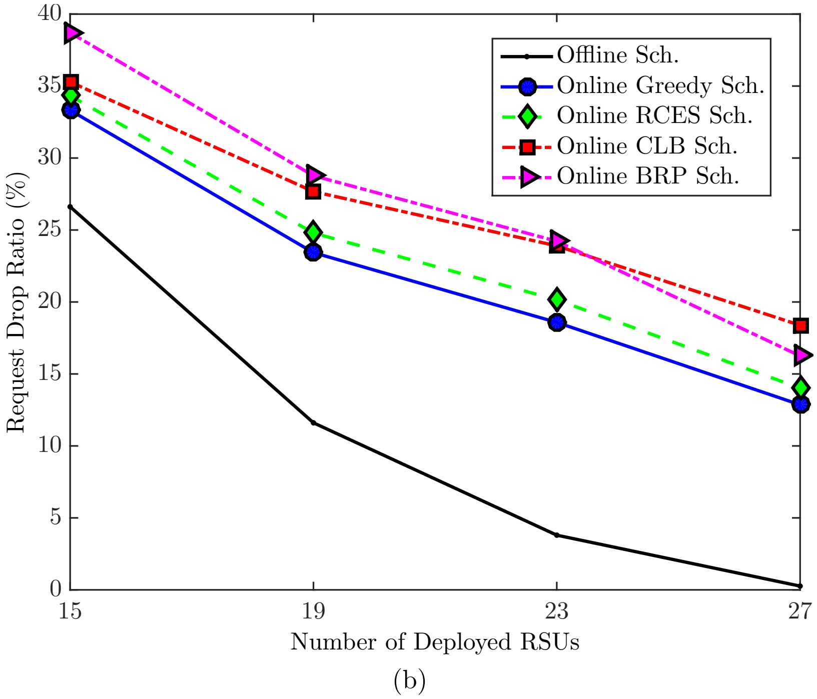
(a)

(b)

Figure 5.1: The Effect of Number of Deployed RSUs (the First Policy).



(a)



(b)

Figure 5.2: The Effect of Number of Deployed RSUs (the Second Policy).

the former is basing its estimates on the given traffic statistics, while the latter does not make use of them; it just assigns as many requests as it can in the immediate neighbourhood that is certain to be on a vehicle's route (current street). Obviously, the greater the number of deployed RSUs, the lower the drop ratio for all algorithms.

In terms of service cost (Figures 5.1a, 5.2a), there is a rather large difference between the greedy scheduler that knows the routes and the algorithms that do not (RCES and CLB), as expected. Nevertheless, the more sophisticated use of traffic statistics by RCES as described above leads to a much lower service cost in comparison to CLB. We can deduce that while not knowing the routes significantly increases the service cost, use of known statistics by our algorithm greatly limits the difference. Note that the service costs *increase* with the increase in the number of RSUs. This is due to the previously dropped requests that can now be served, thus increasing the service cost, but also reducing the drop ratio. Since the drop ratio is similar for all algorithms, i.e., they drop roughly the same number of requests, the comparison of their service costs is valid, i.e., no algorithm achieves a better service cost by dropping many more requests than the others.

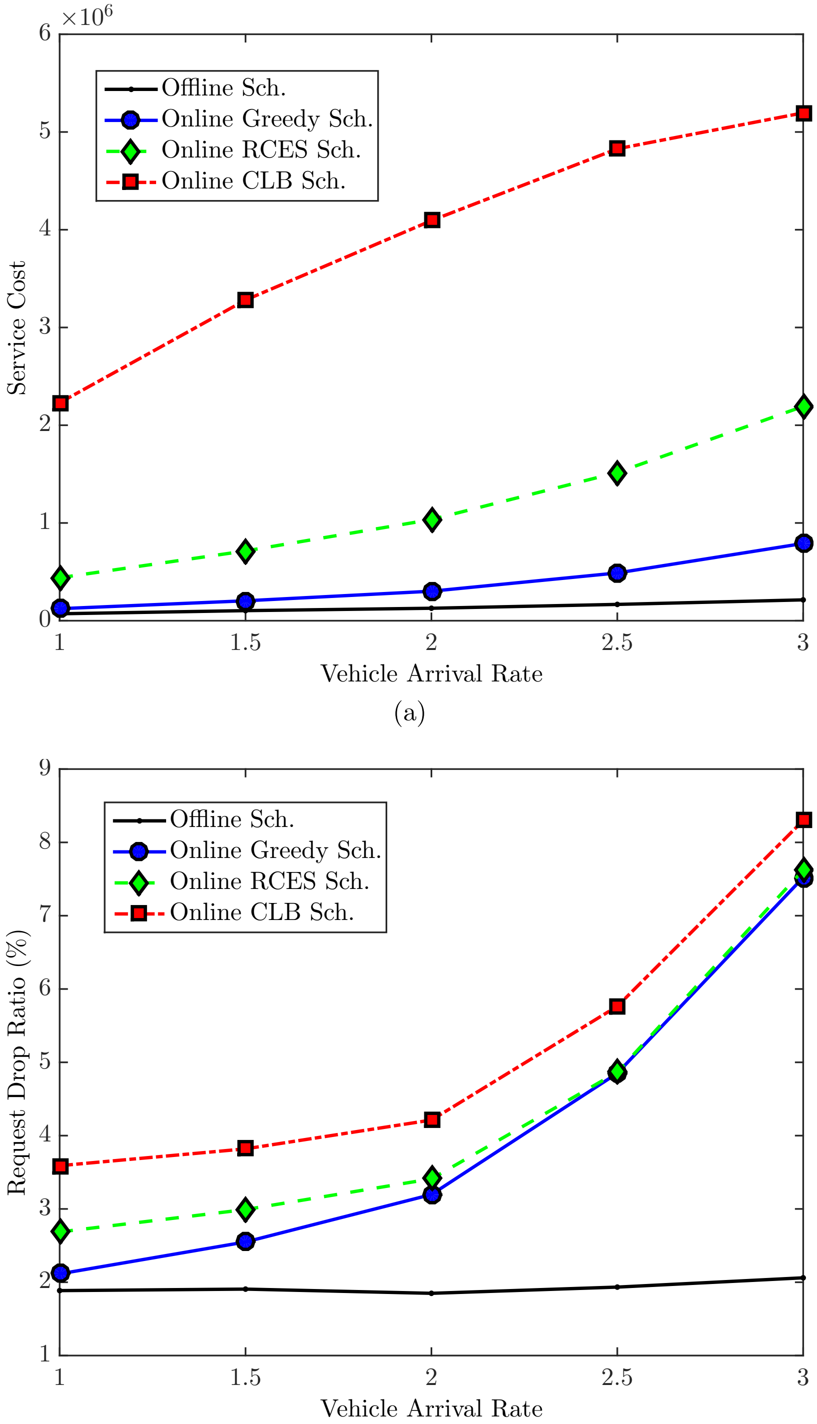
Finally, we include our results of running the RCES scheduler on vehicle motion probability estimates produced by a classic Bayesian estimator (Alpaydin, 2014) (denoted as the online BRP scheduler in Figures 5.1-5.2), which is trained using the first of our traces. Note that the drop ratio is slightly larger than that of RCES when run on exact probabilities calculated from past data, but its energy cost is not significantly larger (due, in part, to the larger number of dropped requests). We leave further exploration of Machine Learning techniques for future work.

5.5.2 The Effect of Vehicle Arrival Rate and Request Arrival Rate

In this set of experiments, we assess our algorithm performance as the vehicle arrival rate or the request arrival rate increase. These results are shown in Figure 5.3-5.4. Figure 5.3 shows the effect of the vehicle arrival rate and Figure 5.4 shows the effect of the request arrival rate.

We use a fixed set of RSUs as we increase the vehicle or request arrival rates. There are 22 deployed RSUs with a total capacity of 70. In subfigures 5.3a and 5.3b, the horizontal axes show the rate by which vehicles arrive per time slot. In subfigures 5.4a and 5.4b, the horizontal axes show the middle point of the interval from which the rate by which a vehicle generates its requests is randomly chosen. As mentioned before, the per time-slot request arrival rate is chosen uniformly at random between 0.01 and 0.02. As before, the results shown are produced by averaging over all (except the first) traffic traces used in our experiments.

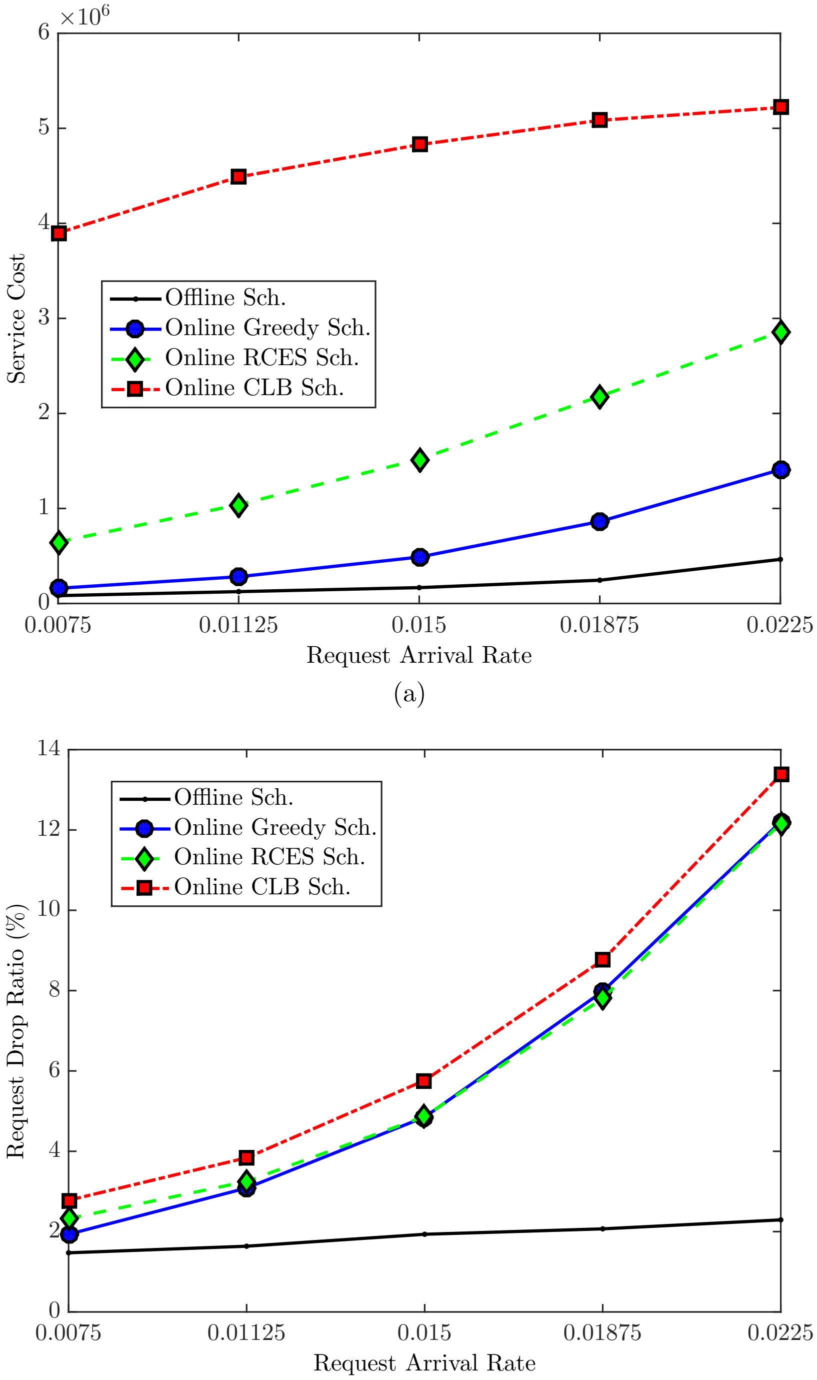
As expected, increasing the number of vehicles or requests will increase both the service cost and the request drop ratio. However, note that the request arrival rate increase has a worse effect than the vehicle arrival rate increase, on both the service cost and the drop ratio. This is due to the limitation of at most one request per vehicle serviced during each time slot: While two requests from two different vehicles can be served by a single RSU (using two units of capacity), two requests from the same vehicle cannot do the same. Again, RCES outperforms CLB in terms of service cost, for the same reasons as above.



(a)

(b)

Figure 5.3: The Effect of Vehicle Arrival Rate.



(a)

(b)

Figure 5.4: The Effect of Request Arrival Rate.

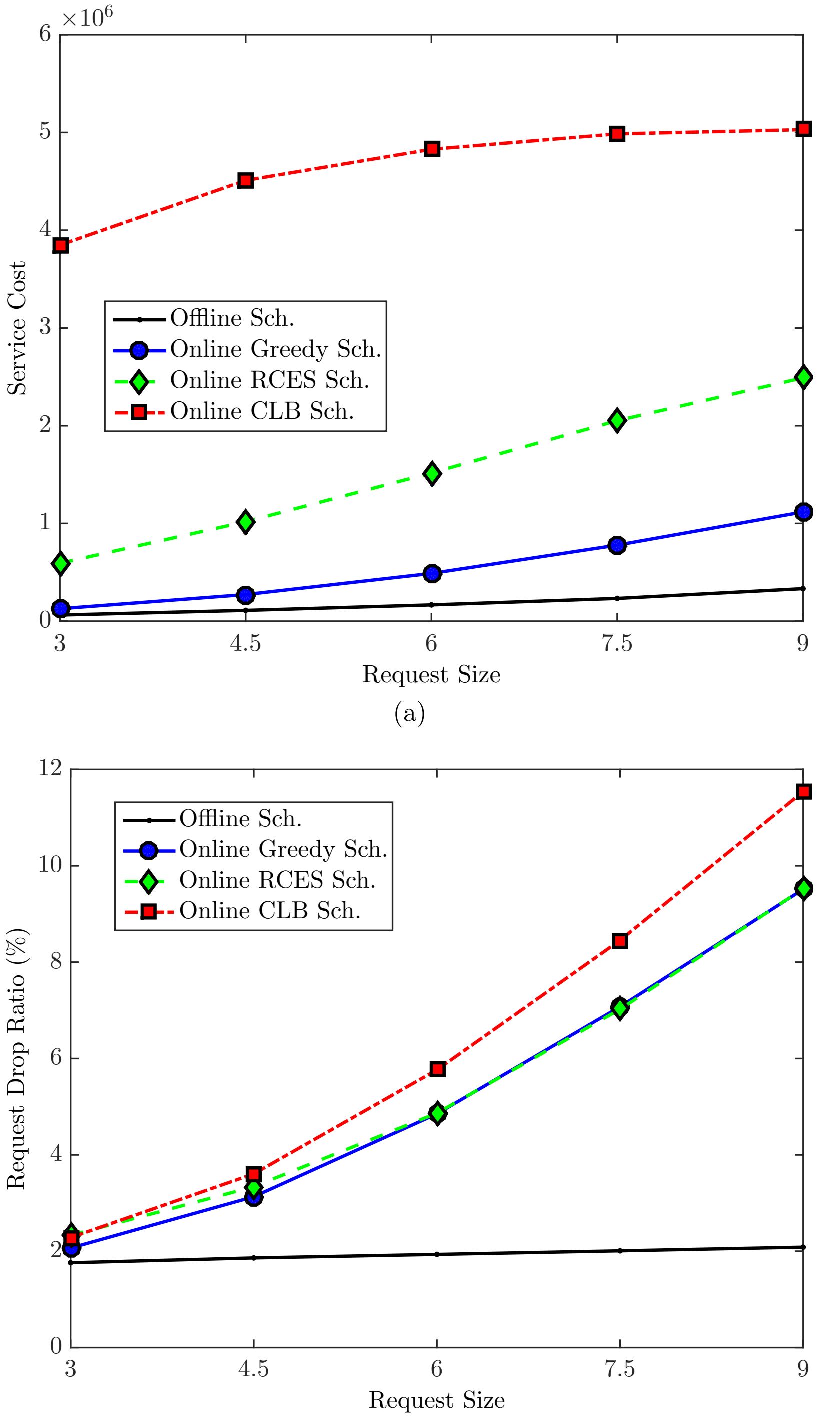
5.5.3 The Effect of Request Size and Request Time-to-Live

Finally, we evaluate the effects of the request size and the request time-to-live on the algorithms (Figures 5.5-5.6). Figure 5.5 shows the effect of the request size, while Figure 5.6 shows the effect of the request time-to-live.

We use the same fixed set of deployed RSUs as in the previous case. To increase the request sizes, we increase the lower and upper limits from which the mean request sizes for each vehicle are randomly chosen; the middle point of these intervals are shown on the horizontal axes in subfigures 5.5a and 5.5b. Similarly, we increase the lower and upper limits from which the mean request time-to-live values are randomly chosen; the middle point of the intervals are shown on the horizontal axes of subfigures 5.6a and 5.6b.

The results show that increasing the request size has the same effects that increasing the request arrivals has on both the service cost and the drop ratio. But, as can be seen, this case has slightly lower service costs and drop ratios. This is due to the fact that two requests of size one introduce more uncertainties in the input to the scheduler than a single request of size two, since the first request will be scheduled without knowing the effect of the later second request. On the other hand, a single request of size two can be broken into two unit-size requests that can be better scheduled. Again, RCES performs better than CLB, for the same reasons as the ones discussed above.

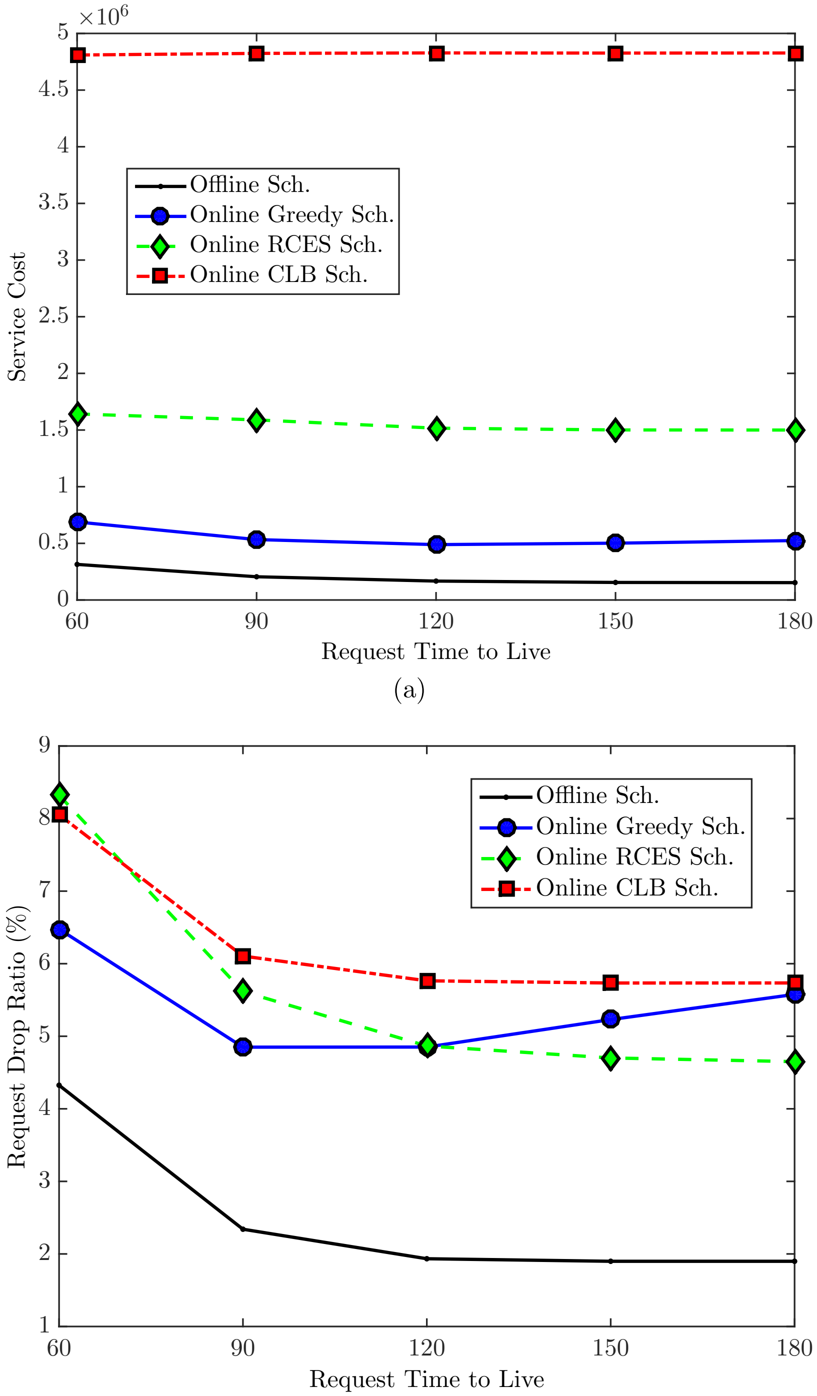
Subfigures 5.6a and 5.6b show that increasing the request time-to-live is mostly beneficial for the drop ratio (as expected), since it allows the scheduler to postpone requests that cannot be serviced at the current RSU to the future. On the other hand, there isn't much of an effect on the service costs. Again, RCES performs better



(a)

(b)

Figure 5.5: The Effect of Request Size.



(a)

(b)

Figure 5.6: The Effect of Request Time-to-Live (TTL).

than CLB. Note that the greedy scheduler with route knowledge does worse for higher time-to-live values, because it greedily looks for time slots with smaller communication cost. Oftentimes, this scheduler prefers to serve requests during much later time slots, because in this way it saves some energy cost, but it suffers higher drop ratio as a result.

5.6 Conclusions

This chapter considered roadside unit job scheduling when vehicle routes are unknown. A scheduler referred to as the Route Coverage Expectation Scheduler (RCES) was proposed. RCES uses historical input traffic traces to estimate vehicular motion and the energy communication costs associated with RSU-to-vehicle transmission. The scheduler processes online vehicular job requests that are subject to hard deadline constraints and a small packet loss. The objective is to schedule vehicular jobs in an energy efficient fashion, so that the long-term energy service costs of the RSUs are minimized. RCES does this by scheduling job requests across multiple RSUs, scheduling the initial part of a request on the current RSU and deferring the remainder to RSUs that the vehicle may encounter in the future. A variety of results were presented that show the performance of the proposed scheduler. Comparisons were made to optimal offline scheduling, where routes are assumed to be known in advance, a simple greedy online scheduler, which also knows vehicle routes, and a known scheduler that attempts to assign all requests to the RSUs on the current street using an earlier-deadline-first (EDF) policy. The results showed that deploying RCES when vehicle routes are not known by the network achieves a drop ratio similar to the drop ratio achieved when these routes are known, with only a modest increase in energy cost.

Chapter 6

Conclusions and Future Work

Vehicular Ad-hoc Networks (VANETs) will play an essential role in the future of intelligent transportation systems (ITS) that will eventually enable a wide range of services, from safety applications to infotainment services. Roadside infrastructure is a key component of these systems, either through data collection and dissemination or as a gateway to the Internet. The cost of deploying roadside units (RSUs) consists of installation cost, i.e., CAPEX (capital expenditure), and long-term operating cost, i.e., OPEX (operating expenditure). An RSU deployment that minimizes the sum of these cost components must jointly consider both the initial RSU placement and their associated long-term service costs.

This thesis investigated an RSU deployment strategy in which the location and the configuration of RSUs are decided such that the sum of CAPEX and OPEX costs is minimized and all vehicular traffic requirements are met. We used historical vehicular traffic traces, a set of RSU candidate locations, and a set of RSU configurations for each candidate location as inputs. The traces include vehicular communication requests with associated time deadlines. The RSU placement problem consisted of two phases. The first phase is the design, where historical traces are used to identify the RSU locations and their configurations. In the second phase, the deployed RSU network is exposed to a new online vehicular traffic flow. In this phase, the vehicular traffic demands are processed in a causal fashion.

In the first part of the thesis, we studied the RSU placement and configuration problem in VANETs. First, the offline RSU placement and configuration problem was formulated as an integer linear program (ILP), which was used as a lower bound on total cost. Solving the ILP becomes impractical as the problem size grows. A heuristic algorithm called Minimum Cost Route Clustering (MCRC) was then proposed to address large-scale problems. The MCRC algorithm is an LP-based relaxation of the ILP optimization problem, which used a novel clustering technique and rounding procedure to obtain RSU placements and configurations. A variety of simulation results was presented that suggests that unlike RSU placement algorithms that focus only on the number of deployed RSUs or the capital cost of deploying RSUs, the joint optimization of capital and operational expenditure costs has a lower total cost as well as lower request drop ratio. The significant advantage of the MCRC Algorithm, especially in multiple-choice RSU placement, was also pointed out since MCRC takes into account the service costs associated with the energy used to operate the RSUs during the placement and configuration decisions.

In the second part of the thesis, the problem of capacity augmentation of the RSU network was studied as a solution to vehicular demand growth over time and the necessity for higher RSU capacities. Many network design methods are based on offline traffic traces. As a result, and because of the causal nature of the stream of incoming requests, the causal online scheduling during the operational life of the network may be suboptimal. Therefore, the objective of the design is to obtain a minimum total cost RSU radio capacity assignment that meets a given packet loss rate target, and subject to packet deadline constraints. A heuristic algorithm, referred to as the Capacity Augmentation (CA) Algorithm, was then introduced that iterates over the RSUs, selecting candidates for capacity augmentation based on their packet loss rate sensitivities. A variety of results was presented to show how the CA Algorithm counterbalances the lack of causality in the RSU network design, and achieves the desired packet loss rate target, while reducing the sum of operating and capital expenditure costs.

In the third part of the thesis, we investigated another challenge that faces the offline RSU network design, i.e., the effect of the a priori scheduler knowledge of vehicular routes. In many applications, it may not be possible to fully serve a vehicle request by a single RSU, and the request should be served by multiple RSUs as the vehicle travels through their coverage areas. In such circumstances, the knowledge of vehicle routes can be valuable. However, this information may not be available to the network. The thesis considered the RSU placement that uses such past knowledge during the offline design, but it does not have access to current route information during the job scheduling process. A scheduler referred to as the Route Coverage Expectation Scheduler (RCES) was proposed. RCES uses historical input traffic traces to estimate vehicular motion and the energy communication costs associated with RSU-to-vehicle transmission. The objective is to schedule vehicular jobs across multiple RSUs in an energy efficient fashion so that the long-term energy service costs of the RSUs are minimized. A variety of experiments was presented that shows that employing the RCES Algorithm when vehicle routes are not known by the network can achieve a request drop ratio similar to the request drop ratio achieved when these routes are known, with only a modest increase in energy cost.

The work in this thesis can be extended in the future by considering the expansion and reconfiguration of RSU networks. In this thesis, we assumed that the statistics of vehicular demands are stable for the period of planning and operating the RSU network. Also, we assumed that at the beginning of the planning phase, there is no existing deployed RSU. The model that was introduced in Chapter 3 can be extended in the future to accommodate not only already existing deployed RSUs but also the possibility of installing new RSUs, closing, and reconfiguring existing RSUs. Another extension is to consider a hybrid scenario in which unicast and multicast requests can be processed by RSUs.

Future extensions to the algorithms proposed in Chapter 4 can take into account the possibility of installing a new RSU in addition to the capacity augmentation of the current RSUs. Another extension to this work would be the reduction of RSU capacities wherever it is appropriate. This can be considered when the traffic pattern changes and operating an RSU with higher capacity may be costly to the network. Employing more sophisticated machine learning techniques would be another extension of the proposed algorithm in Chapter 5, and can further improve the system performance.

Bibliography

Abdrabou, A. and Zhuang, W. (2011). 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.

Abdrabou, A., Liang, B., and Zhuang, W. (2013). Delay Analysis for Sparse Vehicular Sensor Networks with Reliability Considerations. *Wireless Communications, IEEE Transactions on*, **12**(9), 4402–4413.

Ahdi, F. and Subramaniam, S. (2016). Capacity Enhancement of RF Wireless Mesh Networks Through FSO Links. *IEEE/OSA Journal of Optical Communications and Networking*, **8**(7), 495–506.

Al-Sultan, S., Al-Doori, M. M., Al-Bayatti, A. H., and Zedan, H. (2014). A Comprehensive Survey on Vehicular Ad Hoc Network. *Journal of Network and Computer Applications*, **37**, 380–392.

Ali, G. G. M. N., Mollah, M. A. S., Samantha, S. K., and Mahmud, S. (2014a). An Efficient Cooperative Load Balancing Approach in RSU-based Vehicular Ad Hoc Networks (VANETs). In *Control System, Computing and Engineering (ICCSCE), 2014 IEEE International Conference on*, pages 52–57.

Ali, G. G. M. N., Chan, E., and Li, W. (2014b). On Scheduling Data Access with Cooperative Load Balancing in Vehicular Ad Hoc Networks (VANETs). *The Journal of Supercomputing*, **67**(2), 438–468.

Alpaydin, E. (2014). *Introduction to Machine Learning*. MIT Press.

Ashraf, U. (2015). Capacity Augmentation in Wireless Mesh Networks. *IEEE Transactions on Mobile Computing*, **14**(7), 1344–1354.

Aslam, B. and Zou, C. (2011). Optimal Roadside Units Placement along Highways. In *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*, pages 814–815.

Aslam, B., Amjad, F., and Zou, C. (2012). Optimal Roadside Units Placement in Urban Areas for Vehicular Networks. In *Computers and Communications (ISCC), 2012 IEEE Symposium on*, pages 423–429.

Aslam, B., Wang, P., and Zou, C. C. (2016). *An Economical, Deployable and Secure Architecture for the Initial Deployment Stage of Vehicular Ad-Hoc Network*, pages 487–519. Springer International Publishing, Cham.

Badawy, G., Sayegh, A., and Todd, T. (2010). Energy Provisioning in Solar-Powered Wireless Mesh Networks. *Vehicular Technology, IEEE Transactions on*, **59**(8), 3859–3871.

Barrachina, J., Garrido, P., Fogue, M., Martinez, F. J., Cano, J. C., Calafate, C. T., and Manzoni, P. (2013). Road side unit deployment: A density-based approach. *IEEE Intelligent Transportation Systems Magazine*, **5**(3), 30–39.

Bazzi, A., Masini, B. M., Zanella, A., and Pasolini, G. (2015). IEEE 802.11p for Cellular Offloading in Vehicular Sensor Networks. *Computer Communications*, **60**, 97 – 108.

Brahim, M. B., Drira, W., and Filali, F. (2014). Roadside Units Placement within City-Scaled Area in Vehicular Ad-hoc Networks. In *2014 International Conference on Connected Vehicles and Expo (ICCVE)*, pages 1010–1016.

Campolo, C. and Molinaro, A. (2013). Multichannel Communications in Vehicular Ad Hoc Networks: A Survey. *Communications Magazine, IEEE*, **51**(5), 158–169.

Campolo, C., Cozzetti, H. A., Molinaro, A., and Scopigno, R. (2011). Vehicular Connectivity in Urban Scenarios: Effectiveness and Potential of Roadside, Moving WAVE Providers and Hybrid Solutions. *EURASIP Journal on Wireless Communications and Networking*, **2011**(1), 1–10.

Cascetta, E. (2009). *Transportation Systems Analysis: Models and Applications*, volume 29 of *Springer Optimization and Its Applications*. Springer US, 2 edition.

Cavalcante, E. S., Aquino, A. L., Pappa, G. L., and Loureiro, A. A. (2012). Roadside Unit Deployment for Information Dissemination in a VANET: An Evolutionary Approach. In *Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '12, pages 27–34, New York, NY, USA. ACM.

Cheng, H., Fei, X., Boukerche, A., and Almulla, M. (2015). GeoCover: An Efficient Sparse Coverage Protocol for RSU Deployment over Urban VANETs. *Ad Hoc*

Networks, **24, Part B**, 85 – 102. Modeling and Performance Evaluation of Wireless Ad-Hoc Networks.

Chi, J., Jo, Y., Park, H., Hwang, T., and Park, S. (2013a). An Effective RSU Allocation Strategy for Maximizing Vehicular Network Connectivity. *International Journal of Control and Automation*, **6**(4), 259–270.

Chi, J., Jo, Y., Park, H., and Park, S. (2013b). Intersection-Priority based Optimal RSU Allocation for VANET. In *Ubiquitous and Future Networks (ICUFN), 2013 Fifth International Conference on*, pages 350–355.

Cunha, F., Villas, L., Boukerche, A., Maia, G., Viana, A., Mini, R. A., and Loureiro, A. A. (2016). Data Communication in VANETs: Protocols, Applications and Challenges. *Ad Hoc Networks*, **44**, 90 – 103.

Dai, C., Chow, C.-Y., and Zhang, J. (2013). Utilizing Road-side Infrastructure for Location-based Services in Vehicular Ad-hoc Networks. In *Communications and Networking in China (CHINACOM), 2013 8th International ICST Conference on*, pages 546–551.

Eftekhari, H. R., Bashirzadeh, A. J., and Ghatee, M. (2015). Binary Programming Model to Optimize RSU Placement for Information Dissemination. In *2015 International Conference on Connected Vehicles and Expo (ICCVE)*, pages 112–115.

Farbod, A. and Todd, T. (2007). Resource Allocation and Outage Control for Solar-Powered WLAN Mesh Networks. *Mobile Computing, IEEE Transactions on*, **6**(8), 960–970.

Force, U. T. T. (2012). The High Cost of Congestion in Canadian Cities. Technical report, Council of Ministers Responsible for Transportation and Highway Safety.

Frey, B. J. and Dueck, D. (2007). Clustering by Passing Messages Between Data Points. *Science*, **315**(5814), 972–976.

Gui, Y. and Chan, E. (2012). Data Scheduling for Multi-item Requests in Vehicle-Roadside Data Access with Motion Prediction Based Workload Transfer. In *Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference on*, pages 569–574.

Hammad, A., Todd, T., Karakostas, G., and Zhao, D. (2013). Downlink traffic scheduling in green vehicular roadside infrastructure. *Vehicular Technology, IEEE Transactions on*, **62**(3), 1289–1302.

Hammad, A., Todd, T., and Karakostas, G. (2015). Variable Bit Rate Transmission Schedule Generation in Green Vehicular Roadside Units. *Vehicular Technology, IEEE Transactions on*, **PP**(99), 1–1.

Harri, J., Filali, F., and Bonnet, C. (2009). Mobility Models for Vehicular Ad Hoc Networks: A Survey and Taxonomy. *Communications Surveys Tutorials, IEEE*, **11**(4), 19–41.

Holmberg, K. (1994). Solving the Staircase Cost Facility Location Problem with Decomposition and Piecewise Linearization. *European Journal of Operational Research*, **75**(1), 41 – 61.

Hu, Y., Xiao, M., Liu, A., Cheng, R., and Mao, H. (2016). Nearly Optimal Probabilistic Coverage for Roadside Data Dissemination in Urban VANETs. In H. Gao,

J. Kim, and Y. Sakurai, editors, *Database Systems for Advanced Applications: DASFAA 2016 International Workshops: BDMS, BDQM, MoI, and SeCoP, Dallas, TX, USA, April 16-19, 2016, Proceedings*, pages 238–253, Cham. Springer International Publishing.

J. A. Volpe National Transportation Systems Center (2008). Vehicle-Infrastructure Integration (VII) Initiative Benefit-Cost Analysis, Version 2.3. Technical report, Intelligent Transportation Systems Joint Program Office, United States Department of Transportation, Washington, DC.

Jiang, J.-H., Shie, S.-C., and Tsai, J.-Y. (2014). Roadside Unit Deployment Based on Traffic Information in VANETs. In J.-S. Pan, V. Snasel, S. E. Corchado, A. Abraham, and S.-L. Wang, editors, *Intelligent Data analysis and its Applications, Volume I: Proceeding of the First Euro-China Conference on Intelligent Data Analysis and Applications, June 13-15, 2014, Shenzhen, China*, pages 355–365, Cham. Springer International Publishing.

Kchiche, A. and Kamoun, F. (2014). Traffic-Aware Access-Points Deployment Strategies for VANETS. In N. Mitton, A. Gallais, E. M. Kantarci, and S. Papavassiliou, editors, *Ad Hoc Networks: 6th International ICST Conference, ADHOCNETS 2014, Rhodes, Greece, August 18-19, 2014, Revised Selected Papers*, pages 15–26, Cham. Springer International Publishing.

Khezrian, A., Todd, T., Karakostas, G., and Azimifar, M. (2015). Energy-Efficient Scheduling in Green Vehicular Infrastructure With Multiple Roadside Units. *Vehicular Technology, IEEE Transactions on*, **64**(5), 1942–1957.

Kim, D., Velasco, Y., Yang, Z., Wang, W., Hussain, R., and Uma, R. N. (2016).

Cost Effective Mobile and Static Road Side Unit Deployment for Vehicular Adhoc Networks. In *2016 International Conference on Computing, Networking and Communications (ICNC)*, pages 1–5.

Kolliopoulos, S. G. and Moysoglou, Y. (2015). Integrality Gaps for Strengthened Linear Relaxations of Capacitated Facility Location. *Mathematical Programming*, pages 1–43.

Krajzewicz, D., Erdmann, J., Behrisch, M., and Bieker, L. (2012). Recent Development and Applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, **5**(3&4), 128–138.

Kumrai, T., Ota, K., Dong, M., and Champrasert, P. (2014). RSU Placement Optimization in Vehicular Participatory Sensing Networks. In *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*, pages 207–208.

Lee, J. and Kim, C. M. (2010). 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. Springer-Verlag.

Levi, R., Shmoys, D. B., and Swamy, C. (2012). LP-based Approximation Algorithms for Capacitated Facility Location. *Mathematical Programming*, **131**(1-2), 365–379.

Li, P., Liu, Q., Huang, C., Wang, J., and Jia, X. (2015). Delay-bounded Minimal Cost Placement of Roadside Units in Vehicular Ad Hoc Networks. In *2015 IEEE International Conference on Communications (ICC)*, pages 6589–6594.

Li, Z., Zhu, Y., and Li, M. (2009). Practical Location-based Routing in Vehicular Ad Hoc Networks. In *Mobile Adhoc and Sensor Systems, 2009. MASS '09. IEEE 6th International Conference on*, pages 900–905.

Li, Z., Chigan, C., and Gao, C. (2013). STCP2: Short-Time Certificate-based Privacy Protection for Vehicular Ad Hoc Networks. In *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1762–1767.

Liang, Y., Liu, H., and Rajan, D. (2012). Optimal Placement and Configuration of Roadside Units in Vehicular Networks. In *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th*, pages 1–6.

Lin, F. Y. S. (1994). Link Set Capacity Augmentation Algorithms for Networks Supporting SMDS. In *Communications, 1994. ICC '94, SUPERCOMM/ICC '94, Conference Record, 'Serving Humanity Through Communications.' IEEE International Conference on*, pages 624–629 vol.1.

Lin, P.-C. (2012). Optimal Roadside Unit Deployment in Vehicle-to-Infrastructure Communications. In *ITS Telecommunications (ITST), 2012 12th International Conference on*, pages 796–800.

Liu, C., Huang, H., Du, H., and Jia, X. (2014). Optimal RSUs Deployment in Vehicular Networks. In Y. Chen, W.-T. Balke, J. Xu, W. Xu, P. Jin, X. Lin, T. Tang, and E. Hwang, editors, *Web-Age Information Management: WAIM 2014 International Workshops: BigEM, HardBD, DaNoS, HRSUNE, BIDASYS, Macau, China, June 16-18, 2014, Revised Selected Papers*, pages 236–246, Cham. Springer International Publishing.

Liu, C., Huang, H., and Du, H. (2016). Optimal RSUs Deployment with Delay Bound Along Highways in VANET. *Journal of Combinatorial Optimization*, pages 1–15.

Liu, Y., Niu, J., Ma, J., and Wang, W. (2013). File Downloading Oriented Roadside Units Deployment for Vehicular Networks. *Journal of Systems Architecture*, **59**(10, Part B), 938 – 946. Advanced Smart Vehicular Communication System and Applications.

Liya, X., Chuanhe, H., Peng, L., and Junyu, Z. (2013). A Randomized Algorithm for Roadside Units Placement in Vehicular Ad Hoc Network. In *Mobile Ad-hoc and Sensor Networks (MSN), 2013 IEEE Ninth International Conference on*, pages 193–197.

Lu, N., Cheng, N., Zhang, N., Shen, X., and Mark, J. (2014). Connected Vehicles: Solutions and Challenges. *Internet of Things Journal, IEEE*, **1**(4), 289–299.

Mahdian, M., Ye, Y., and Zhang, J. (2006). Approximation algorithms for metric facility location problems. *SIAM Journal on Computing*, **36**(2), 411–432.

Maia, G., Villas, L., Boukerche, A., Viana, A., Aquino, A., and Loureiro, A. (2013). Data Dissemination in Urban Vehicular Ad hoc Networks with Diverse Traffic Conditions. In *Computers and Communications (ISCC), 2013 IEEE Symposium on*, pages 000459–000464.

Makkawi, A., Daher, R., and Rizk, R. (2015). RSUs Placement Using Cumulative Weight Based Method for Urban and Rural Roads. In *Reliable Networks Design and Modeling (RNDM), 2015 7th International Workshop on*, pages 307–313.

Malandrino, F., Casetti, C., Chiasserini, C. F., Sommer, C., and Dressler, F. (2012). Content Downloading in Vehicular Networks: Bringing Parked Cars Into the Picture. In *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC)*, pages 1534–1539.

Malandrino, F., Casetti, C., Chiasserini, C.-F., and Fiore, M. (2014). Content Download in Vehicular Networks in Presence of Noisy Mobility Prediction. *Mobile Computing, IEEE Transactions on*, **13**(5), 1007–1021.

Martinez, F. J., Toh, C. K., Cano, J.-C., Calafate, C. T., and Manzoni, P. (2011). A Survey and Comparative Study of Simulators for Vehicular Ad Hoc Networks (VANETs). *Wirel. Commun. Mob. Comput.*, **11**(7), 813–828.

Mehar, S., Senouci, S. M., Kies, A., and Zoulikha, M. M. (2015). An Optimized Roadside Units (RSU) Placement for Delay-Sensitive Applications in Vehicular Networks. In *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, pages 121–127.

Meireles, T., Fonseca, J., and Ferreira, J. (2016). *Deterministic Vehicular Communications Supported by the Roadside Infrastructure: A Case Study*, pages 49–80. Springer International Publishing, Cham.

Nikookaran, N., Karakostas, G., and Todd, T. (2017). Combining Capital and Operating Expenditure Costs in Vehicular Roadside Unit Placement. *IEEE Transactions on Vehicular Technology*, **PP**(99), 1–1.

Paden, B., Čáp, M., Yong, S. Z., Yershov, D., and Frazzoli, E. (2016). A Survey of

Motion Planning and Control Techniques for Self-Driving Urban Vehicles. *IEEE Transactions on Intelligent Vehicles*, **1**(1), 33–55.

Patil, P. and Gokhale, A. (2013). Voronoi-based Placement of Road-side Units to Improve Dynamic Resource Management in Vehicular Ad Hoc Networks. In *Collaboration Technologies and Systems (CTS), 2013 International Conference on*, pages 389–396.

Patra, M., Mishra, S., and Murthy, C. S. R. (2014). An Analytic Hierarchy Process Based Approach for Optimal Road Side Unit Placement in Vehicular Ad Hoc Networks. In *Vehicular Technology Conference (VTC Spring), 2014 IEEE 79th*, pages 1–5.

Rappaport, T. S. (2001). *Wireless Communications: Principles and Practice*. Prentice Hall, 2 edition.

Rashidi, M., Batros, I., Madsen, T., Riaz, M., and Paulin, T. (2012). Placement of Road Side Units for Floating Car Data Collection in Highway Scenario. In *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2012 4th International Congress on*, pages 114–118.

Rizk, R., Daher, R., and Makkawi, A. (2014). RSUs Placement Using Overlap Based Greedy Method for Urban and Rural Roads. In *Communication Technologies for Vehicles (Nets4Cars-Fall), 2014 7th International Workshop on*, pages 12–18.

Silva, C. M., Aquino, A. L., and Jr., W. M. (2015). Deployment of Roadside Units Based on Partial Mobility Information. *Computer Communications*, **60**, 28 – 39.

Silva, C. M., Meira, W., and Sarubbi, J. F. M. (2016). Non-Intrusive Planning the Roadside Infrastructure for Vehicular Networks. *IEEE Transactions on Intelligent Transportation Systems*, **17**(4), 938–947.

Song, J., Wu, Y., Xu, Z., and Lin, X. (2014). Research on Car-Following Model Based on SUMO. In *Advanced Infocomm Technology (ICAIT), 2014 IEEE 7th International Conference on*, pages 47–55.

Song, Y., Lin, B., Tian, Y., Ding, N., He, R., and Zhao, T. (2015). Heterogeneous Roadside Unit Placement in Eco-Sustainable Communication Networks for Intelligent Transportation. In *2015 Ninth International Conference on Frontier of Computer Science and Technology*, pages 214–219.

Spaho, E., Barolli, L., Mino, G., Xhafa, F., and Kolici, V. (2011). VANET Simulators: A Survey on Mobility and Routing Protocols. In *Broadband and Wireless Computing, Communication and Applications (BWCCA), 2011 International Conference on*, pages 1–10.

Sun, Y., Lin, X., Lu, R., Shen, X., and Su, J. (2010). Roadside Units Deployment for Efficient Short-Time Certificate Updating in VANETs. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1–5.

Tao, J., Zhu, L., Wang, X., He, J., and Liu, Y. (2014). RSU Deployment Scheme with Power Control for Highway Message Propagation in VANETs. In *2014 IEEE Global Communications Conference*, pages 169–174.

Tonguz, O. K. and Viriyasitavat, W. (2013). Cars as Roadside Units: A Self-Organizing Network Solution. *IEEE Communications Magazine*, **51**(12), 112–120.

Trullols, O., Fiore, M., Casetti, C., Chiasserini, C., and Ordinas, J. B. (2010). Planning Roadside Infrastructure for Information Dissemination in Intelligent Transportation Systems. *Computer Communications*, **33**(4), 432 – 442.

U.S. Federal Communications Commission (2004). Report and Order FCC 03-324.

Vageesh, D., Patra, M., and Murthy, C. S. R. (2014). Joint Placement and Sleep Scheduling of Grid-Connected Solar Powered Road Side Units in Vehicular Networks. In *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2014 12th International Symposium on*, pages 534–540.

Vazirani, V. V. (2003). *Approximation Algorithms*. Springer Berlin Heidelberg.

Wang, C., Li, X., Li, F., and Lu, H. (2014). A Mobility Clustering-based Roadside Units Deployment for VANET. In *Network Operations and Management Symposium (APNOMS), 2014 16th Asia-Pacific*, pages 1–6.

Wang, S.-W. and Chang, M.-Y. (2011). Roadside Units Allocation Algorithms for Certificate Update in VANET Environments. In *Communications (APCC), 2011 17th Asia-Pacific Conference on*, pages 472–477.

Wang, Y., Zheng, J., and Mitton, N. (2016). Delivery Delay Analysis for Roadside Unit Deployment in Vehicular Ad Hoc Networks With Intermittent Connectivity. *IEEE Transactions on Vehicular Technology*, **65**(10), 8591–8602.

Wu, T.-J., Liao, W., and Chang, C.-J. (2012). A Cost-Effective Strategy for Road-Side Unit Placement in Vehicular Networks. *Communications, IEEE Transactions on*, **60**(8), 2295–2303.

Wu, X., Subramanian, S., Guha, R., White, R., Li, J., Lu, K., Bucceri, A., and Zhang, T. (2013). Vehicular Communications Using DSRC: Challenges, Enhancements, and Evolution. *Selected Areas in Communications, IEEE Journal on*, **31**(9), 399–408.

Xie, B., Xia, G., Chen, Y., and Xu, M. (2013). Roadside Infrastructure Placement for Information Dissemination in Urban ITS Based on a Probabilistic Model. In C.-H. Hsu, X. Li, X. Shi, and R. Zheng, editors, *Proceedings of the 10th IFIP International Conference on Network and Parallel Computing, NPC 2013, Guiyang, China, September 19-21, 2013.*, pages 322–331, Berlin, Heidelberg. Springer Berlin Heidelberg.

Xiong, Y., Ma, J., Wang, W., and Tu, D. (2013). RoadGate: Mobility-Centric Roadside Units Deployment for Vehicular Networks. *International Journal of Distributed Sensor Networks*, **9**(3), 690974.

Yan, T., Zhang, W., Wang, G., and Zhang, Y. (2014). Access Points Planning in Urban Area for Data Dissemination to Drivers. *IEEE Transactions on Vehicular Technology*, **63**(1), 390–402.

Zhang, B., Zhu, G., Xu, S., and Zhang, N. (2015). Energy-Efficient Roadside Units Deployment in Vehicular Ad hoc Networks. In *6th International Conference on Wireless, Mobile and Multi-Media (ICWMMN 2015)*, pages 6–10.

Zheng, K., Zheng, Q., Chatzimisios, P., Xiang, W., and Zhou, Y. (2015). Heterogeneous Vehicular Networking: A Survey on Architecture, Challenges and Solutions. *Communications Surveys Tutorials, IEEE*, **PP**(99), 1–1.

Zou, F., Zhong, J., Wu, W., Du, D.-Z., and Lee, J. (2011). Energy-Efficient Roadside Unit Scheduling for Maintaining Connectivity in Vehicle Ad-hoc Network. In *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication*, ICUIMC '11, pages 64:1–64:8, New York, NY, USA. ACM.