Scheduling in Green Vehicular Infrastructure with Multiple Roadside Units

SCHEDULING IN GREEN VEHICULAR INFRASTRUCTURE WITH MULTIPLE ROADSIDE UNITS

BY

AMIR M. KHEZRIAN, B.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

MASTER OF APPLIED SCIENCE

Master of Applied Science (2013)	McMaster University
(Electrical & Computer Engineering)	Hamilton, Ontario, Canada

TITLE:	Scheduling in Green Vehicular Infrastructure with Multiple Roadside Units
AUTHOR:	Amir M. Khezrian B.Sc., (Computer Engineering) Iran University of Science and Technology, Tehran, Iran
SUPERVISOR:	Dr. Terence D. Todd
NUMBER OF PAGES:	xii, 81

To my parents who have always supported me in all situations

Abstract

In this thesis we consider low complexity downlink traffic scheduling for green vehicular roadside infrastructure. In multiple roadside unit (RSU) deployments, the energy provisioning of the RSUs may differ, and it is therefore desirable to balance RSU usage from a normalized energy viewpoint. We consider both splittable and unsplittable RSU assignment scheduling (SRA and URA). We first derive an offline integer linear programming bound for the normalized min-max RSU energy usage, which can be solved for a given input sample function. We then show that in the SRA case there is a polynomial complexity 2-approximation bound for the normalized min-max energy schedule. These bounds are used for comparisons with several proposed online scheduling algorithms. The first scheduler is a low complexity Greedy Online Algorithm (GOA) that makes greedy RSU selections followed by minimum energy time slot assignments. A normalized min-max online algorithm is then proposed (TOAA) which is an online version of the 2-approximation bound for SRA scheduling. Then, the Greedy Flow Graph Algorithm (GFGA), which makes greedy RSU selections followed by time slots reassignment whenever a new vehicle is assigned to the same RSU. This is done using a locally optimum integer linear program that can be efficiently solved using a minimum cost flow graph. Two low complexity algorithms are then introduced based on a potential function scheduling approach. The One-Objective algorithm, uses a primary objective based on normalized min-max energy. The second, the Bi-Objective algorithm, uses the same primary objective combined with a total energy secondary objective. These algorithms have provable performance guarantees, in that their worse-case competitive ratio performance is upper bounded. Results from a variety of experiments show that the proposed scheduling algorithms perform well. In particular, we find that in the SRA case, the TOAA and GFGA algorithms perform very close to the lower bound, but at the expense of having to reassign time slots whenever a new vehicle arrives. In the URA case, our low complexity One-Objective algorithm performs better than the others over a wide range of traffic conditions.

Acknowledgements

I would like to thank my supervisor, Dr. Terence D. Todd for giving me the opportunity to pursue my M.A.Sc. studies at McMaster University and for letting me experience an autonomous research career. Thanks also for his invaluable guidance throughout my research. He was exactly the type of the supervisor I needed, by giving enough time for his students to find their way, and by supporting them when it was necessary. I would like to express my sincere gratitude to Dr. Karakostas for supporting me in my research work and also for giving me helpful comments. I am also very thankful to the members of Wireless Networking Lab especially Morteza Azimifar, Shokouh Mostofi and Abdullah Hammad for all their helpful comments and their sincere friendship.

Notation and abbreviations

AP	Access Point
DL	Down-Link
ILP	Integer Linear Program
MCFG	Minimum Cost Flow Graph
MILP	Mixed Integer Linear Program
RSU	Road Side Unit
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VANETs	Vehicular Ad-Hoc Networks
GOA	Greedy Online Algorithm
WAVE	Wireless Access in Vehicular Environment
TOAA	2-Approximation Online Algorithm
SRA	Splittable RSU Assignment
URA	Unsplittable RSU Assignment
ITS	Intelligent Transportation Systems
DSRC	Dedicated Short Range Communications
WLANs	Wireless Local Area Networks

UMTS	Universal Mobile Telecommunication System
WiMAX	Worldwide Interoperability for Microwave Access
RDS	Radio Data System
TMC	Traffic Message Control
MANETs	Mobile Aad Hoc Networks

Contents

Abstract **iv**

Acknowledgements **vi**

Notation and abbreviations **vii**

1 Introduction **1**

1.1 Energy Efficiency 1

1.2 Thesis Motivation and Overview 2

2 A Brief Review of Vehicular Ad Hoc Networks **4**

2.1 Applications and Their Technical Requirements 7

2.2 Communication Technologies 11

2.3 Routing 12

2.3.1 Topology-based Routing Protocols 13

2.3.2 Geographical Routing Protocols 14

2.3.3 Cluster-Based Routing 14

2.3.4 Broadcast 15

2.3.5 Geocast Routing 15

2.4 Data Scheduling in VANETs 16

2.5 Energy Efficiency in Wireless Networks 20

3 Vehicle-to-Infrastructure Scheduling Algorithms 23

3.1 Introduction 23

3.2 Related Work 26

3.3 System Model and Scheduling Definitions 27

3.4 Offline Min-Max Energy Bounds 31

3.4.1 2-Approximation Bound for Splittable RSU Assignment (SRA) 33

3.5 Online Scheduling Algorithms 40

3.5.1 Greedy Online Algorithm (GOA) 40

3.5.2 Greedy Flow Graph (GFGA) Scheduler 41

3.5.3 2-Approximation Online Algorithm (TOAA) for Splittable RSU Assignment 45

3.5.4 One-Objective and Bi-Objective Online Scheduling Algorithms 47

4 Performance Evaluation 57

5 Conclusions 73

List of Figures

3.1 Roadside Unit (RSU) Example. 24

3.2 Multiple Roadside Units (RSUs). 26

3.3 The graph $T(x)$. To simplify the figure, the source node s and the edges connecting it to nodes v_i are not shown. 37

3.4 Minimum Energy Flow Graph Scheduler. 44

4.1 Maximum and Total RSU Energy vs. Vehicle Demand. Splittable RSU Assignment (SRA). 59

4.2 Maximum and Total RSU Energy Use vs. Vehicle Demand. Unspilttable RSU Assignment (URA). 60

4.3 Maximum and Total RSU Energy Use vs. Vehicle Speed. Splittable RSU Assignment (SRA). 63

4.4 Maximum and Total RSU Energy Use vs. Vehicle Speed. Unsplittable RSU Assignment (URA). 64

4.5 Maximum and Total RSU Energy vs. Vehicle Demand. Splittable RSU Assignment (SRA). Shadowing. 67

4.6 Maximum and Total RSU Energy Use vs. Vehicle Demand. Unsplittable RSU Assignment (URA). Shadowing. 68

4.7 Maximum and Total RSU Energy Use vs. Vehicle Speed. Splittable RSU Assignment (SRA). Shadowing. 69

4.8 Maximum and Total RSU Energy Use vs. Vehicle Speed. Unsplittable RSU Assignment (URA). Shadowing. 70

Chapter 1

Introduction

In the future, *Intelligent Transportation Systems* (ITS) will support a large variety of new vehicular applications such as road safety, entertainment, and Internet access services. ITS will be built, in part, on vehicular ad-hoc networking (VANET) technologies which enable both vehicle-to-vehicle and vehicle-to-infrastructure communications. VANETs are currently designed to use dedicated short-range communication in the 5.9 GHz band, which is licensed by the US Federal Communications Commission. This is referred to as Wireless Access in Vehicular Environment and is based on extensions to the IEEE 802.11 wireless LAN standard. Before ITS can realize its full potential however, many technical issues remain to be addressed. This area of research is the focus of this thesis.

1.1 Energy Efficiency

Global energy consumption is rapidly increasing to satisfy growing industrial and personal needs. For this reason, energy use has become a major concern of governments everywhere, and is increasing the push for alternate energy sources. With the growth and pervasiveness of networking technologies, green communications is now receiving a lot of attention from researchers. In addition to decreasing overall energy use, in some cases grid connected power is unavailable, and renewable energy is needed for network operation. In these types of systems, reduced energy consumption may lead to lower capital and operating costs, which motivates the need for energy efficient designs.

Energy efficiency in vehicular ad hoc networks (VANETs) has been recently studied. This has typically not been considered an issue in the vehicles themselves, since the radios are powered by the car engine. And from the roadside infrastructure point of view, most previous work assumes that grid connected power is available at reasonable cost. When this is not the case however, a viable alternative for roadside infrastructure is to use green energy sources. In reference (Peirce and Mauri, 2007), the US Department of Transportations Vehicle Infrastructure Integration (VII) Initiative has studied the deployment of roadside infrastructure for vehicle monitoring and safety applications. It has estimated that a significant fraction of rural roadside infrastructure would have to be solar powered, and clearly the deployment costs can be reduced significantly by energy efficient designs. In vehicular infrastructure, due to the dependence of power consumption on vehicle-to-infrastructure distance, a smart traffic scheduler can significantly conserve communication energy costs.

1.2 Thesis Motivation and Overview

This thesis considers the problem of traffic scheduling in green vehicular roadside infrastructure. In multiple roadside unit (RSU) arrangements, it is often desirable to load balance the energy consumption across the roadside units so that energy provisioning costs can be reduced as much as possible. We first derive an integer linear programming bound for the normalized min-max RSU energy usage. Then, a polynomial complexity 2-approximation bound for the min-max energy schedule is given. Since the ILP bound is NP-hard, the polynomial schedule can be used as an approximation for the formulated problem. These bounds are used for comparisons with several proposed online scheduling algorithms. First-Come-First-Assigned (GOA) is a low complexity algorithm which makes greedy RSU selection decisions followed by a minimum energy time slot assignment. The second, the Greedy Flow Graph Algorithm (GFGA), makes the same RSU selection but reassigns time slots whenever a new vehicle is assigned to the same RSU using a minimum cost flow graph scheduler. A normalized min-max online algorithm (1-obj) is then proposed using a potential function scheduling approach and an online version of our 2-approximation bound (TOAA) is introduced. A bi-objective algorithm motivated by the 1-obj scheduler is also proposed. A variety of results are presented which compare the performance of the proposed schedulers.

Chapter 2

A Brief Review of Vehicular Ad Hoc Networks

Vehicular ad hoc networks (VANETs) are an emerging technology which will eventually provide vehicle safety, information, and entertainment services. In VANETs, two types of data communications are defined, vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I). These modes can be used to transmit data between vehicles and between vehicles and fixed roadside units (RSUs), respectively. Like many wireless ad-hoc networks, VANETs are not necessarily dependent on a fixed infrastructure for communications. From the architectural perspective, VANETs fall within three categories, *Cellular/WLAN*, *pure ad-hoc*, and *hybrid*. In a *Cellular/WLAN* configuration, vehicles use fixed cellular gateways or WLAN access points to connect to the Internet, gather traffic information, or for routing purposes. In the *ad-hoc* configuration, ad-hoc networks are formed where vehicles exchange data using vehicle-to-vehicle communication. In the *hybrid* architecture, a combination of cellular, WLAN, and ad-hoc networks can be used.

Short radio transmission range, self-organization, self-management, and low bandwidth are some of the similarities that VANETs share with other ad-hoc networks (Li and Wang, 2007). However, VANETs have some unique characteristics which distinguish them from other ad-hoc networks, which are described as follows.

- *Highly dynamic topology:* Due to their fast speed, vehicles may remain within wireless transmission range for short time periods, which can cause the VANET topology to change rapidly.

- *Frequent network disconnection:* For the same reasons, the connectivity of vehicles can change frequently, and as a result, there may be a high probability of network disconnection especially when traffic density is low. In some applications such as Internet access, this can lead to problematic service.

- *Sufficient energy and storage:* Since radio equipment is powered by the vehicle's engine, there is typically ample energy and computing power.

- *Geographical type of communication:* In VANETs, end points are often identified with their geographical locations, while in many other networks, end points are defined by a simple identifier.

- *Mobility modeling and predication:* The characteristics of high node mobility and dynamic network topology can magnify the role of mobility prediction in protocol design. However, vehicle locations are constrained to highways, roads, and streets, which makes location predication possible.

- *Communication environments:* VANETs operate in two typical environments. In rural settings, the traffic pattern of the vehicles tend to be straightforward, while in urban settings, the lack of direct line-of-sight communications makes these environments much more complex (Li and Wang, 2007).

- *Timing constraints:* In VANETs, there are some applications which impose hard time constraints such as Cooperative Collision Warning and other types of safety applications.
- *Interaction with on-board sensors:* Vehicles may be equipped with on-board sensors which will provide information such as the vehicle location. This can be used for routing and data scheduling purposes.

Beside the characteristics mentioned above, there are other technical challenges associated with the design and deployment of VANETs. Fading effects, mobility and multipath propagation may have a strong affect on radio operation. Although some applications involve fixed infrastructure, others are expected to work reliably using decentralized communications (Hartenstein and Laberteaux, 2008). In this case, no central coordination can be assumed, and therefore synchronization and management of transmission events may become difficult, and may lead to less efficient use of the channel. In addition, to avoid information abuse, privacy and security requirements must be satisfied. Receivers need to be assured that received information is from a trusted source and message content is not tampered with. Although certificate authorities can be used to secure VANETs, this may compete with privacy issues (Toor *et al.*, 2008). From the application and economic perspective, there are other issues to be addressed. The cost/benefit tradeoffs, their impact on safety and transportation efficiency are some of the concerns which need further study.

2.1 Applications and Their Technical Requirements

VANET applications are continually emerging and evolving, and their associated technical challenges are still being resolved. Typically, applications are categorized as safety, traffic efficiency, and infotainment (Hartenstein and Laberteaux, 2008; Toor *et al.*, 2008; Yousefi *et al.*, 2006). In this section, some of these applications and their requirements are briefly discussed.

Safety applications: Many people are hurt in traffic accidents. Vehicular communications can help to avoid accidents by extending road visibility and by providing real-time safety information. This is one of the main concerns of Intelligent Transportation Systems (ITS). Accident rates can be reduced by providing safety information, which alerts drivers and helps them to react to unforeseen events. Some common safety applications are as follows:

- *Curve speed warning:* This alert gives the driver a warning when approaching a curve too quickly, using GPS and digital maps.

- *Lane change warning:* This system assists drivers who want to perform a lane change, by monitoring the position of the vehicle within the roadway, and by checking blind spots and issuing appropriate warnings.

- *Cooperative collision warning:* This service improves driving safety by giving warnings or by breaking the car automatically when the distance between vehicles becomes less than a set threshold (Khaled *et al.*, 2009). This system can also be used to warn nearby vehicles about an existing accident, making them aware of the situation, and helping them to alter their paths if required.

- *Incident management:* Whenever an accident or a problem occurs on the road, the incident and the traffic flow can be managed through vehicular communication.

Traffic efficiency: In addition to road safety, traffic information and monitoring are another set of capabilities that vehicular ad-hoc networks can provide. These applications can be used to avoid congestion, traffic jams, and to assist personal navigators in route selection. Platooning, vehicle tracking, notification services, traffic jam prevention, and weather reporting are some of the applications that can be defined in this category. We introduce some of them briefly.

- *Platooning:* This service can be used to permit vehicles to travel closely and safely in an efficient way, which can reduce highway congestion (Khaled *et al.*, 2009).
- *Vehicle Tracking:* Such a system allows manufacturers and trusted parties to remotely monitor vehicle statistics and provide supporting services.
- *Notification Services:* Travel information such as weather and traffic flow conditions can be provided via Internet access. This service can also help drivers to find the best routes to their destinations. This can result in road congestion reduction and smoother traffic flow, thus increasing road capacity.

Infotainment applications: The main goal of these types of applications is to provide services such as Internet access, distributed games, instant messaging, tourist and leisure information access, and parking bookings.

- *Parking place management:* This service allows drivers to find and book parking spots. Vehicle parking can also be provided without driver assistance (Khaled *et al.*, 2009).

- *Distributed games:* Entertainment applications can be managed between a limited group of vehicles through V2V communication. Card games and real-time talking are two examples.

- *Peer to peer applications:* This service provides the capability of information sharing via V2V communication without using an application server. Instantaneous messaging, file transfer, and voice over IP are some examples.

- *Internet Connectivity:* The vast usage of the Internet has made Internet access a daily requirement for modern life. Since many user applications require Internet access to provide services, providing this facility to passengers and VANET applications is important (Toor *et al.*, 2008). The viability of Internet access and its technical issues, such as the performance provided by IEEE 802.11 access points, are discussed in many papers (Bychkovsky *et al.*, 2006; Hadaller *et al.*, 2007; Ott and Kutscher, 2004).

In order to ensure good performance for vehicular applications, their requirements must be identified (Khaled *et al.*, 2009). We now briefly introduce some important network related characteristics of VANET applications.

- **Location awareness:** Vehicles exchange information not only with other vehicles, but also with fixed roadside infrastructure. Vehicular information exchange in a geographical area requires reliable and scalable communication capabilities. Safety services, such as alert cooperative collision warning and incident management, needs to know the actual position of the vehicles. This is also true for certain comfort applications such as parking booking. In addition to GPS, many other technologies, such as cellular or WiFi, can be used to obtain vehicle positioning information.

- **Time awareness:** Many applications must satisfy time constraints in order to provide effective service, especially those related to safety. For instance, a driver is able to react quickly if data is delivered without a significant delay following an alert. This requirement may not be easy to achieve, especially when there is high vehicular mobility.

- **Permanent Access:** Permanent access is one the challenging issues for VANETs. In urban areas, the number of stations to which a mobile user can be connected may be high. In contrast, in rural environments, the density of stations may be low, which leads to a high possibility of connection loss. At the application level, some services, such as file transfer or data download, need a relatively permanent connection.

- **Penetration rate dependency:** Penetration rate is defined as the percentage of VANET equipped vehicles. This plays an especially important role in the operation of safety and critical applications. Although low penetration rates can cause problems in safety applications, high penetration rates can lead to transmission problems and poor network performance.

- **Geocast capability:** Geocast is defined as the capability of sending messages to different geographical regions. This is considered efficient if messages are delivered both in sparse and dense geographical areas by an efficient use of bandwidth (Khaled *et al.*, 2009).

- **Mobility:** Permanent access to a network can be affected by mobility. The sender-to-receiver distance has a direct impact on the probability of packet reception. Besides physical distance, existence of buildings or vehicles, and interference with other radio equipment are some of other effects that should be taken into account (Khaled *et al.*, 2009).

2.2 Communication Technologies

There are a large number of wireless technologies which can provide the platform required for V2V and V2I communications. We briefly review the technologies relevant to vehicular networks.

- **Bluetooth**: Bluetooth is a short range communication technology which can be used for vehicular networks. However, its coverage limitation narrows the use of this technology to intra-vehicle applications.

- **WLAN:** Wireless Local Area Networks (WLANs) comprise a set of standards from the IEEE 802.11 group. IEEE 802.11a/b/g technologies typically provide tens of Mbps of throughput over 100 meter distances.

- **DSRC:** Dedicated Short Range Communications (DSRC) is a short to medium range communication service operating at 5.9 GHz. DSRC is designed to support a wide range of V2I and V2V communications. IEEE 802.11p (also referred as WAVE) has been developed to support the application requirements of intelligent transportation systems (ITS). IEEE 802.11p is used as a groundwork in DSRC, which enables data transmission between high speed vehicles and between these vehicles and fixed infrastructure.

- **Cellular networks:** UMTS, Universal Mobile Telecommunication System, is a technology which can be used for vehicular network communications. However, coverage, message delay, and usage costs, are a concern.

- **WiMAX:** Worldwide Interoperability for Microwave Access (WiMAX) is a communication technology which fills the gap between 3G and WLANs. IEEE 802.16e is a standard developed for mobile users connected to base stations, making it the most appropriate version of WiMAX for vehicular applications.

- **RDS and TMC:** RDS, or Radio Data System, was developed to carry digital data using the FM radio band. RDS offers a data rate of 1187.5 bps and a transmission range of 80 km. Traffic Message Control (TMC) usually uses RDS to transmit traffic information. TMC systems can be used to broadcast traffic information to users, informing them of road problems.

2.3 Routing

In this section we review routing in VANETs. The main objective in routing is to forward packets with a minimum of delay and use of network resources. Many routing protocols, some of which are derived from mobile ad hoc networks (MANETs), have been developed for VANETs. However, fast vehicle movement, and dynamic information exchange requirements, leads to poor performance in some protocols. For example, the dynamic nature of VANET topologies makes finding and maintaining routes a challenging task. This has been studied recently and many protocols have been proposed, which can be classified into the following categories (Li and Wang, 2007; Kumar and Dave, 2011; Chen *et al.*, 2008): Topology-based, position-based, cluster-based, broadcast, and geocast.

2.3.1 Topology-based Routing Protocols

Topology-based routing protocols use link information to route packets. They can be divided into two sub-categories: Proactive and Reactive.

Proactive routing protocols

Proactive routing protocols continually maintain knowledge of the available paths in the network. For this reason, there is no need to perform route discovery whenever there is a communication request. Typically, each node maintains a table, each entry of which indicates the next hop node towards a particular destination. However, if the network topology changes frequently, maintaining these tables may expend a significant amount of overhead. Some proactive protocols that can be named are DSDV and OLSR (Kumar and Chand, 2010a).

Reactive/ad-hoc routing protocols

In response to the maintenance problem of proactive protocols, reactive protocols were developed. They maintain only the routes which are currently in use, leading to a reduction in network overhead when only a small subset of available paths is needed. However, recent studies have shown that most ad-hoc routing protocols (e.g., AODV and DSR) suffer from the highly dynamic nature of vehicular topologies due to their poor route convergence and low communication throughput. To improve the frequent route breakage problem of AODV, two protocols, PRAODV and PRAODVM, use speed and location information to predict link lifetimes (Li and Wang, 2007).

2.3.2 Geographical Routing Protocols

In location-based routing protocols, the forwarding nodes are selected based on the geographical location of the source, destination, and neighbouring nodes obtained from street maps, traffic models or navigation systems. Greedy Perimeter Stateless Routing (GPSR), one of the best known location-based protocols, forwards packets to a node that is geographically closest to the destination. It uses face routing when greedy routing fails (Li and Wang, 2007). It is argued that geographic routing performs well in evenly distributed node conditions or highway scenarios compared to city settings since there are fewer obstacles. In VANETs, position-based routing faces large challenges in a built-up city environment containing an uneven distribution of vehicles and constrained mobility due to road patterns. A-STAR, GPCR, and GSR are some algorithms developed to deal with these issues using street maps.

2.3.3 Cluster-Based Routing

In cluster-based protocols, nodes are clustered to provide scalability; and within a cluster, a node is selected to play the role of a virtual network infrastructure. The selected node, also known as the cluster head, is responsible for coordinating intra and inter-cluster communications. Intra-cluster communication is performed via direct links while nodes within different clusters communicate through cluster heads. Cluster heads usually have a role in controlling media access and ensuring routing scalability. COIN and LORA_CBF are some examples of cluster-based algorithms. Although scalability is the advantage of cluster-based routing protocols, delay and the overhead of cluster maintenance are main drawbacks of this method.

2.3.4 Broadcast

Broadcast is used in routing methods such as traffic sharing, weather, emergency, road condition and advertisement announcements. It can also be used as a mechanism in unicast routing protocols to find routes. The simplest broadcast service is flooding where each node re-broadcasts any received packet to all nodes except the one it received the packet from. Flooding is easy to implement and it guarantees all nodes will eventually receive the message. The main drawback of flooding however, is that as the number of nodes increases, the algorithm performance becomes poor very quickly since the bandwidth required to broadcast can increase exponentially. BROADCOMM protocol, which is based on a hierarchical structure for a highway network, was developed to improve broadcasting delay and routing overhead. UMB is another protocol designed to overcome interference, packet collisions, and hidden node problems during message dissemination (Li and Wang, 2007).

2.3.5 Geocast Routing

Geocast is a location-based multicast routing. It divides the network into geographical regions where packets from the source node are sent to nodes which are within certain geographical areas. Geocast can be implemented by defining a multicast group to be within a certain geographical area, and can use either flooding or non-flooding approaches to broadcast data. In flooding approaches, to avoid network congestion and message overhead, the flooding is restricted inside a forwarding zone. The non-flooding approaches, which are based on unicast routing, can use regional flooding in the destination node area. Many applications can benefit from geocast routing. For instance, whenever an accident occurs, vehicular sensors can report it to nearby vehicles for proper attention. Vehicles outside the area do not need to be alerted (Li and Wang, 2007).

2.4 Data Scheduling in VANETs

Vehicles typically remain in the coverage area of roadside units (RSU) for relatively short periods of time due to their fast movement, especially in highway settings. In order to serve as many vehicle demands as possible, reduce download delay and energy needed to transmit, smart scheduling is important. In addition, since upload and download requests compete for the same bandwidth, selecting the request to be served at a particular time is critical. Scheduling algorithms are evaluated in terms of the performance metrics which are needed to satisfy application services. Examples of these are fairness, responsiveness, time constraints, data size, service ratio, quality of service, and data quality (Kumar and Chand, 2010a). Some of these are briefly discussed below.

- **Time Constraints:** Due to the speed of vehicles, requests in vehicle-roadside data access may have time constraints and must typically be served before the vehicle leaves the RSU transmission range.

- **Service Ratio:** Service ratio is defined as the ratio of the number of requests served on time, to the total number of requests.

- **Quality of Service:** QoS is associated with time constraints that vehicle requests impose on the scheduler (Xu *et al.*, 2006).

- **Quality of Data:** Data may become stale if the vehicle loses the upload opportunity before it moves out of the RSU range. Data quality is associated with how recent requested data items are updated (Zhang *et al.*, 2010).

Vehicles communicate with roadside infrastructure to upload or download specific data items. Upload requests are mainly used to move data to the Internet when the roadside unit (RSU) acts as an Internet gateway, or to update a buffered data item when the infrastructure is used as a buffering point between vehicles. Value-added advertisement, real-time traffic, and map downloading are some examples which can benefit from data buffering at the RSU.

Unicast and broadcast communications are the methods used to serve single or multiple requests, where appropriate. There are two major broadcast methods: *push-based* and *pull-based*. In *push-based* broadcast, the RSU sends data periodically according to a static schedule, which is based on historic data access statistics or predefined request profiles. Vehicles then listen to the broadcast channel passively. In *pull-based* (on-demand) broadcasting, data items are disseminated according to explicit requests submitted by vehicles. This method is generally scalable to a large and dynamic client population. In addition, no assumption is needed about the data access pattern, which makes it appropriate for dynamic workload environments. Three methods, *opportunistic*, *vehicle-assisted*, and *co-operative*, are approaches used for data delivery in the *pull-based* settings.

- *Opportunistic:* Passing vehicles or the infrastructure provides the required information when a target vehicle needs it (Kumar and Chand, 2010a).

- *Vehicle-Assisted:* Packets are carried with a vehicle and it is delivered to the infrastructure or to other vehicles when they are encountered (Kumar and Chand, 2010a).

- *Co-operative:* Vehicles download part of the desired information and vehicle sharing is used to obtain the complete content.

In reference (Zhang *et al.*, 2010), a scheduling scheme called $D * S$ was proposed which considers both data size and time constraints. In order to take advantage of broadcasting, the scheduler was extended to the $D * S/N$ scheme to serve multiple requests with a single data transmission. To provide trade-offs between service ratio and data quality, and balance between download and upload demands, the Two-Step algorithm was proposed. In the first step of the algorithm, one of the download or upload priority queues is chosen. In the next step, a request from the selected queue is chosen. The $D * S/N$ and $D * S/R$ schemes are used to choose a request from the download and upload queues, respectively. In $D * S/N$, requests are prioritized by the *DSN_Value* measurement, which is defined as the product of urgency and data size, divided by the number of pending requests for the same data item. In $D * S/R$, the *DSR_Value* measurement is defined with the same policy except that data size is divided by the service rate of download requests in the download queue for a data item. A request with the lowest *DSR_Value* (for upload requests) or *DSN_Value* (for download requests) is served first. The paper compared their scheduling algorithms with three naive schemes, FCFS, FDF, and SDF, and showed their performance and limitations. Based on (Zhang *et al.*, 2010), in (Shahverdy *et al.*, 2010), a scheduling algorithm was introduced in which each file is chopped into several segments. If a vehicle is not able to finish downloading all requested segments from a particular RSU, then it has the opportunity to continue downloading the remaining segments from others. The scheduling algorithm works in two steps. In the first step, requests are separated based on whether the requested item is asked for the first time or it is part of a resumed item. In the second step, using the $D * S$ scheme, it selects one of the two queues to serve a request from the chosen queue. In (Kumar and Chand, 2010b), in addition to the deadline and data size, request priority is considered in the scheduling. The proposed algorithm, $D * S/P$, serves requests with higher priority first, otherwise the scheme behaves similar to the method introduced in (Shahverdy *et al.*, 2010). When a request arrives, its importance is evaluated and responded to with respect to its priority, otherwise, it is handled using the $D * S$ scheme. The performance of the proposed algorithm was studied in terms of service ratio. Reference (Xu *et al.*, 2006) introduced another algorithm called *SIN*-α which considers urgency and the number of outstanding requests in its scheduling. Motivated by urgency and improved algorithm efficiency, *SIN*-α defines **sin-**α as the ratio of urgency to the relative weight of the number of outstanding requests, and it serves a request with minimum **sin-**α first. The paper introduced a new metric, *drop rate*, defined as the ratio of the number of requests missing their deadlines, to the total number of requests, and compared the proposed algorithm with three schemes including EDF, MRF, and R*W. In contrast to (Zhang *et al.*, 2010; Kumar and Chand, 2010b; Shahverdy *et al.*, 2010) which assume a single RSU, a group of RSUs is considered in (Gui and Chan, 2011). It proposed MPO, a motion prediction based scheduling scheme for vehicle-to-infrastructure data access. To obtain good performance and a balanced schedule, vehicular requests are transferred among RSUs. They are prioritized based on the $D * S/N$ scheduling scheme proposed in (Zhang *et al.*, 2010). Requests missing their deadlines are discarded in each scheduling cycle, and they are transferred to a nearby RSU using vehicle motion prediction. This is based on the probabilistic mobility nature of the *Manhattan Mobility Model*. If the motion prediction is correct, the transferred request will have a good chance for reasonable service. This paper also introduced MMPO, a multi-item vehicle request scheduling algorithm. The proposed algorithm was studied and compared with other schedulers including First Come First Serve (FCFS), Smallest Data size First (SDF) and Earliest Deadline First (EDF).

2.5 Energy Efficiency in Wireless Networks

Recent studies have proposed many methods to save energy in wireless communication systems (Gong *et al.*, 2010). For example, reference (Mamechaoui *et al.*, 2013) studied green networking in wireless mesh networks. They included a survey of proposed methods to save energy in the three lower layers of the protocol stack. Power consumption is also a critical issue in wireless ad-hoc networks since many nodes are portable devices such as laptops, which are battery powered. The wireless LAN interface is one of power-consuming components of these types of devices. Many papers have proposed various methods to put mobile devices into reduced power sleep modes, and techniques for minimizing transmission and reception activities so that energy consumption can be reduced (Zhou *et al.*, 2005; Li *et al.*, 2003; Ning *et al.*, 2005). Reference (Ning *et al.*, 2005) introduced a packet-driven method called IEEE 802.11NPS which can reduce energy use by 60% compared to the conventional WLAN power save mode. Transmission range control is another approach used for energy savings in wireless ad-hoc networks. Using throughput and throughput per unit of energy as the optimization metrics, (Park and Sivakumar, 2002) analyzed the optimal throughput as a function of transmission power. They demonstrated that the optimum transmission range to maximize system throughput is a function of the network load, network size, and the number of stations. In reference (Chang and Tassiulas,

2000) an energy-efficient routing algorithm was proposed. Given a set of origin and destination nodes and information generation rates at the origin nodes, They proposed flow augmentation algorithms and a flow redirection algorithm to balance the energy consumption rate among nodes so that node lifetime is maximized.

In vehicular ad-hoc networks (VANETs), vehicles communicate by using vehicle-to-vehicle (V2V) communication, or with fixed infrastructure using a vehicle-to-infrastructure (V2I) communication mode. Vehicular radios are less energy constrained than in many other applications since they draw power from the vehicle engine. Nevertheless, vehicular communication involves other wireless devices such as roadside transceivers which may have to be energy-efficient since they may operate from batteries or solar energy. This motivates new designs which optimize the power consumption of the entire vehicular communication system. Reference (Feng *et al.*, 2010) studied green networking (i.e., energy-efficiency) in VANETs and derived a relation between optimal transmission range and network data traffic. Using data traffic input, They enable vehicles to adjust their transmission range as a function of transportation traffic in order to save energy. Reference (Zou *et al.*, 2011) studied energy savings in RSU scheduling, and proposed a two step algorithm. Given a set of RSUs, they find the optimum RSU on/off switching schedule while maintaining vehicle connectivity. Addressing power consumption and quality-of-service, (Toutouh and Alba, 2011) developed the DE_OLSR routing protocol which improves on the standard version of OLSR in terms of energy consumption. Reference (Zhang *et al.*, 2011) developed a novel cross-layer framework for V2I communication networks. Based on this framework, a sub-carrier assignment policy is proposed, which aims to minimize power consumption while guaranteeing delay QoS requirements. Reference

(Hammad *et al.*, 2013) also studied the problem of scheduling for energy-efficient roadside infrastructure. It proposed three online scheduling algorithms all of which attempt to reduce downlink infrastructure-to-vehicle energy communication costs. Using minimum cost flow graph scheduling, it introduced the Greedy Minimum Cost Flow (GMCF) algorithm. Two other algorithms with reduced complexity compared to GMCF, i.e., Nearest Fastest Set (NFS) and Static Scheduler (SS), were proposed. The performance of the online algorithms were compared in terms of energy use.

Chapter 3

Vehicle-to-Infrastructure Scheduling Algorithms

3.1 Introduction

A significant barrier to the widespread deployment of vehicular roadside infrastructure is the cost of providing electrical-grid power connections. In many locations, grid-power is simply unavailable, while in others, the connection costs are too prohibitive to be commercially feasible. An alternative to wired power in these situations is to operate all or some of the RSUs using green energy sources, such as solar or wind power. In (Peirce and Mauri, 2007) a cost-benefit analysis was presented as part of the US Department of Transportation's Vehicle Infrastructure Integration (VII) Initiative. This study includes cost projections for an initial vehicular infrastructure roll-out that focuses on safety and vehicle monitoring applications. In this report it was estimated that 40% of all initial rural free-way roadside infrastructure would have to be solar powered. A breakdown of the deployment costs also showed that

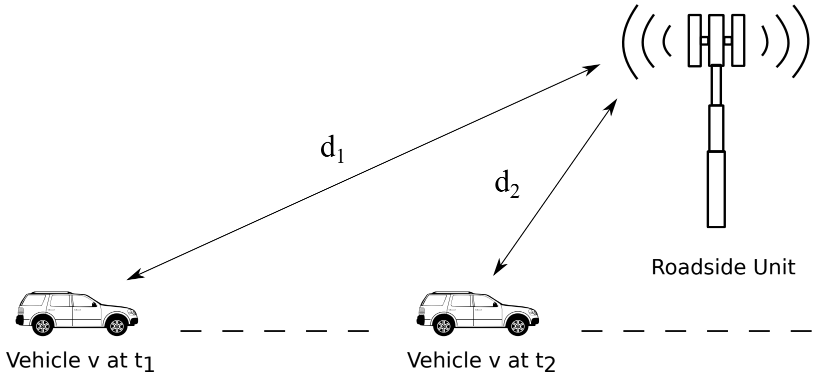


Figure 3.1: Roadside Unit (RSU) Example. Vehicle v is shown at two different times, t_1 and t_2, and distances from the RSU, where $d_1 \gg d_2$. Communication at time t_2 is preferred in terms of downlink RSU energy cost.

over 63% of these roadside unit costs would be consumed by solar energy provisioning, e.g., solar panels, batteries, and their associated electronics. The projected costs of these nodes comes to almost 1B dollars in the proposed initial deployment, and it is clear that these costs can be significantly reduced by energy efficient designs.

In some vehicular settings, vehicle location may be accurately known. This occurs for example, in highway situations where vehicles tend to travel at relatively constant speeds. Reference (Hammad *et al.*, 2010) showed that this knowledge can be used to improve downlink energy consumption by deferring communication until vehicles are in favourable energy positions. An example of this is illustrated in Figure 3.1. In this example, vehicle v is shown at two different times, t_1 and t_2, and at corresponding distances from the RSU given by d_1 and d_2, respectively. Communication at time t_2 may be preferred by the RSU, if the downlink energy costs are lower compared with those at time t_1. In order to exploit this advantage, the downlink communication requirements of the vehicles must be delay tolerant.

In this chapter we consider the delay tolerant RSU scheduling problem when there are more than one roadside units (RSUs) in tandem. In multiple roadside unit (RSU) arrangements, the energy provisioning of the RSUs will often be different, and it is desirable to energy-balance the loading from a normalized energy viewpoint. We first derive an integer linear programming bound for the normalized min-max RSU energy usage, which can be solved for a given input sample function. We then derive a polynomial complexity 2-approximation bound for the min-max energy schedule under splittable RSU assignment scheduling. These bounds are used for comparisons with proposed online scheduling algorithms. The first is a low complexity Greedy Online Algorithm (GOA) scheduler that makes greedy RSU selections followed by a minimum energy time slot assignment. The second, the Greedy Flow Graph Algorithm (GFGA), makes the same RSU selection but reassigns time slots whenever a new vehicle is assigned to the same RSU using a minimum cost flow graph scheduler. A normalized min-max online algorithm, referred to as TOAA, is then proposed which is an online version of our 2-approximation bound. Two algorithms are then proposed based on a potential function scheduling approach. The first, the One-Objective algorithm, uses a primary objective based on normalized min-max energy. Theoretical results are presented which show that this algorithm has provable performance guarantees, based on its worse-case competitive ratio. The second, the Bi-Objective algorithm, uses the same objective, followed by a secondary minimum total energy objective. Results from a variety of experiments show that the proposed scheduling algorithms perform well. The results show that in the SRA scheduling case, the TOAA algorithm performs very close to the lower bound, but it must reassign time slots whenever a new vehicle arrives. In the URA case, our low complexity One-Objective algorithm performs better than the others over a wide range of traffic conditions and does not have this restriction.

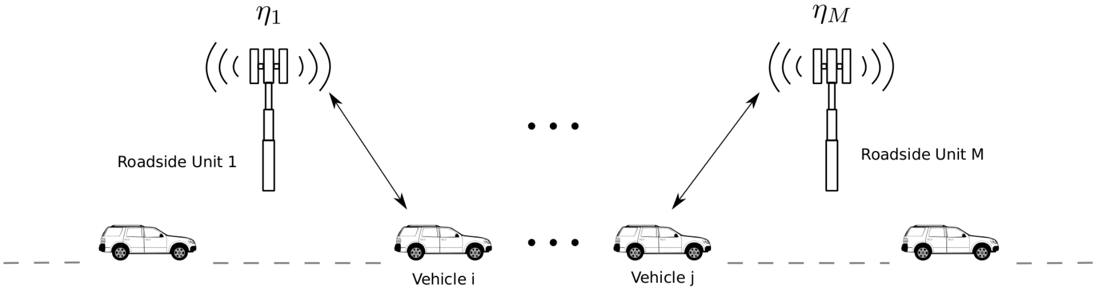


Figure 3.2: Multiple Roadside Units (RSUs). Vehicles pass through the coverage areas of M roadside units, any of which can provide the required vehicular communications. RSU i has a relative energy capacity of η_i.

3.2 Related Work

Recent research in vehicular networks has included topics such as routing algorithms (Li and Wang, 2007), applications (Khaled *et al.*, 2009), security (Zhang *et al.*, 2009), and medium access control performance (Mittag *et al.*, 2008). For example, studies have illustrated the suitability of IEEE 802.11p for highway applications (Bychkovsky *et al.*, 2006; Ott and Kutscher, 2004). In (Jhang and Liao, 2008; Zhao *et al.*, 2008; Nandan *et al.*, 2005), proxy vehicles are used to decrease vehicle contention and improve roadside unit utilization.

Vehicular network connectivity and inter-vehicle interference can be traded-off using transmitter power control. These effects have been extensively studied in (Mittag *et al.*, 2008; Festag *et al.*, 2007; Rawat *et al.*, 2009). However, power control from an energy efficiency viewpoint has normally not been considered, since vehicle-to-vehicle communications is powered from the car engine which has enormous energy reserves. Most work also assumes that grid connected power is readily available at the roadside units, which may not always be the case.

Reference (Zhang *et al.*, 2010) introduced simple schedulers for the roadside unit based on transmission requirement inputs and message deadlines. This work however, did not consider energy efficiency at the RSU. An optimization approach was used in (Alcaraz *et al.*, 2009) to maximize RSU throughput given the locations and velocities of vehicles within radio coverage range. Their scheduler was designed and integrated into the IEEE 802.11e contention free periods. Again however, the focus of this study did not include the energy consumption of the RSU.

RSU scheduling was considered from an energy consumption viewpoint in reference (Hammad *et al.*, 2013). This study focused on minimizing the downlink energy costs in the common case where passing vehicle communication requirements are delay tolerant. It was shown that significant energy savings are possible in this case, using an energy aware scheduler at the RSU. This study is the starting point for our work, however, our focus is on scheduling in the multiple roadside unit case.

3.3 System Model and Scheduling Definitions

A roadside scenario is considered that consists of a tandem set of roadside units (RSUs). This type of arrangement will be common in cases, for example, where high capacity is needed to accommodate evolving peak traffic conditions. An example of this is given in Figure 3.2 where vehicles are shown traveling through a tandem coverage area of M RSUs. For convenience, we describe the system with unidirectional vehicular traffic, but our algorithms and results are applicable to the multi-directional case. In our development we will assume that all vehicles pass by the same set of RSUs, such as in a highway environment, but this is also not a requirement.

The RSUs are assumed to intercommunicate among themselves using back-haul communication links so that scheduling decisions can be coordinated. We also assume that each RSU uses a single radio for downlink communication but the RSUs operate independently from one another without interference. Channel time is assumed to be time-slotted and power control is used on the downlink (i.e., RSU-to-vehicle direction) so that each time slot can carry B bits, regardless of vehicle location within a given RSU coverage area.

It is assumed that when a vehicle j enters the coverage area of RSU 1, its downlink communication requirement is determined, and the system schedules this communication for some future time. In certain situations, such as in a highway scenario, good estimates of RSU downlink energy communication costs can be obtained. In (Hammad *et al.*, 2010) it was shown that when there is a strong deterministic component of path loss versus distance, schedulers can be devised which can greatly reduce downlink RSU energy use. Due to the coverage range associated with the RSUs, the average power consumption of an energy efficient RSU design may be strongly dominated by downlink transmission power. For this reason, the RSUs will generally prefer to communicate with nearby vehicles rather than more distant ones. In the example shown in Figure 3.1, communications at time t_2 is preferable to time t_1 since $d_1 \gg d_2$. To use this option however, requires that there is sufficient packet delay tolerance, and we assume that any vehicle can be served at any time throughout its transition time in the RSU coverage area (Hammad *et al.*, 2013).

The inputs and outputs of our scheduling problem are more formally stated as follows.

INPUT: We are given a set of M tandem RSUs indexed by the set $\mathcal{M} = \{1, 2, \ldots, M\}$ (as shown in Figure 3.2). For each RSU i, there is an energy normalization factor, η_i, which gives its relative energy capacity. We consider a finite sequence of N arriving vehicles indexed by the set $\mathcal{N} = \{1, 2, \ldots, N\}$, where each vehicle j has an RSU-to-vehicle communication requirement of H_j, in units of time slots. It is assumed that H_j must be fulfilled by a given subset, $\mathcal{M}_j \subset \mathcal{M}$, of the RSUs. The entire time considered consists of T time slots given by the set $\mathcal{T} = \{1, 2, \ldots, T\}$ over which the scheduling is to occur. We are also given estimates of the per time slot energy cost for downlink communications from RSU i to vehicle j during time slot t, defined by $\epsilon_{i,j,t}$. Below, we will also use the notion of *normalized energy cost* of vehicle j on RSU i during time-slot t, which is defined as $\tilde{\epsilon}_{i,j,t} := \epsilon_{i,j,t}/\eta_i$, i.e., the normalized energy cost for downlink communications from RSU i to vehicle j during time slot t. This setting is quite general, since it also allows for a vehicle j to choose never to be processed by a particular RSU i, by setting $\epsilon_{i,j,t} = \infty$ for all time-slots $t \in \mathcal{T}$. For example, it may be the case that an RSU i hosts the content of a specific type that vehicle j doesn't have any demand for; therefore j will never schedule any demand on i.

OUTPUT: A scheduler output gives an RSU-to-vehicle transmission schedule. We define

$$x_{i,j,t} = \begin{cases} 1 & \text{if RSU } i \text{ sends to vehicle } j \text{ in time slot } t, \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

Given the inputs defined above, the objective of the scheduler is to select values for $x_{i,j,t}$ such that all vehicle communication requirements are satisfied, and the maximum normalized downlink RSU energy cost across all RSUs is minimized, i.e.,

$$\min_{x_{i,j,t}} \max_{i \in \mathcal{M}} \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{N}} x_{i,j,t} \, \epsilon_{i,j,t} \, / \, \eta_i \tag{3.2}$$

Two RSU assignment options are considered, as follows.

1. Splittable RSU Assignment (SRA): The H_j communication requirement for vehicle j may be satisfied by any combination of the RSUs in the set $\mathcal{M}_j$.

2. Unsplittable RSU Assignment (URA): The H_j communication requirement for vehicle j must be satisfied by a single RSU selected from the set $\mathcal{M}_j$.

The splittable RSU assignment case (RSA) can be reduced to the case of all vehicles having a unit communication requirement, by splitting vehicle j with communication requirement H_j into H_j unit-requirement vehicles. Therefore, from now on we will assume that the communication requirements in the splittable case are $H_j = 1$ for all vehicles j. Obviously, in the unsplittable RSU assignment (URA) case, H_j can be any non-negative integer.

The generated schedules can be either offline or online. In *offline scheduling*, the complete set of inputs is provided to the scheduler all at once, i.e., the former has knowledge of all past and future vehicular inputs. Offline scheduling is used in Section 3.4 to derive lower bounds on energy performance which are then used for comparisons in Chapter 4. In *online scheduling*, the inputs are provided to the scheduler in real time and it must assign the values of $x_{i,j,t}$ in a causal fashion based solely on past and current input values.

3.4 Offline Min-Max Energy Bounds

In this section we derive offline bounds for normalized min-max RSU energy use. The formulation given considers the unsplittable RSU assignment (URA) case where a vehicle's requirements must be assigned to a specific RSU. SRA scheduling is a special case of this, as discussed below. Accordingly, we define the following set of binary scheduling variables.

$$a_{i,j} = \begin{cases} 1 & \text{if RSU } i \text{ is assigned to vehicle } j, \\ 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

The RSU assigned to j must belong to the set $\mathcal{M}_j$. Using the $x_{i,j,t}$ and $\epsilon_{i,j,t}$ variables defined in Section 3.3, a lower bound on min-max energy use can then be computed using the following integer linear program (ILP) referred to as ILP-Bound.

$$\begin{aligned}
\underset{x_{i,j,t}}{\text{minimize}} \quad & \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{N}}\sum_{i\in\mathcal{M}} \epsilon_{i,j,t}\; x_{i,j,t} && && \text{(ILP-Bound)} \\
\text{subject to} \quad & \sum_{t\in\mathcal{T}} x_{i,j,t} = a_{i,j}\; H_j, && \forall j \in \mathcal{N}, i \in \mathcal{M}_j && (3.4) \\
& \sum_{i\in\mathcal{M}_j} a_{i,j} = 1, && \forall j \in \mathcal{N} && (3.5) \\
& \sum_{j\in\mathcal{N}}\sum_{t\in\mathcal{T}} x_{i,j,t}\; \epsilon_{i,j,t}/\eta_i \le \mathcal{E}, && \forall i \in \mathcal{M} && (3.6) \\
& \sum_{j\in\mathcal{N}} x_{i,j,t} \le 1, && \forall i \in \mathcal{M}, \forall t \in \mathcal{T} && (3.7) \\
& x_{i,j,t} \in \{0,1\}, && \forall \{i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T}\} && (3.8) \\
& a_{i,j} \in \{0,1\}, && \forall \{i \in \mathcal{M}, j \in \mathcal{N}\} && (3.9)
\end{aligned}$$

In ILP-Bound, the objective function is simply the total downlink energy used by the RSUs. Constraint (3.4) ensures that vehicle communication requirements are fulfilled by summing the appropriate values of $x_{i,j,t}$ over all RSUs. The $a_{i,j}$ term ensures that the entire vehicle communication requirement is served by the assigned RSU. Constraint (3.5) requires that vehicle j is assigned to one RSU. Constraint (3.6) places a common upper bound, $\mathcal{E}$, on the total normalized energy used by each RSU where η_i is the normalization factor for RSU i. Since $\mathcal{E}$ is not known a priori, we can do a binary search on its value, solving ILP-Bound each time, to get the minimum value of $\mathcal{E}$ that achieves an optimal normalized min-max energy bound, i.e., a bound on the best load balancing possible. We will refer to $\mathcal{E}$ as the *minmax bound*, and we will denote by $\mathcal{E}^{opt}$ the best possible minmax bound. Constraint (3.7) ensures that a given time slot can only contain a single transmission, but allows for simultaneous operation of the M RSUs. Note that only minor simplifications to ILP-Bound are needed to obtain the splittable RSU assignment (SRA) case. Constraints (3.5) and (3.9) are removed and constraint (3.4) is changed to $\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{M}_j} x_{i,j,t} = H_j, \forall j \in \mathcal{N}$

ILP-Bound can be solved directly using branch and bound techniques and CPLEX 8.1.0 has been used with data generated from MATLAB. These results are used for comparisons with online algorithms to be introduced in Section 3.5. However, the complexity of computing ILP-Bound is very high, and in the next section we show that there is a polynomial complexity 2-approximation bound for the splittable RSU assignment case.

3.4.1 2-Approximation Bound for Splittable RSU Assignment (SRA)

In this section we consider the splittable RSU assignment (SRA) case where each vehicle has a unit time slot communication requirement. We notice that our problem is a version of the offline load-balancing (i.e., minimum makespan) machine scheduling problem, or, even more specifically, an extension of the Generalized Assignment Problem. Based on the classic algorithm by Shmoys and Tardos (Shmoys and Tardos, 1993), we develop a bi-objective algorithm that attempts to achieve objective (3.2), and minimize the total energy used to achieve that objective. The process is as follows.

First, we show how to compute a schedule of minmax bound at most $2\mathcal{E}$ of total energy cost C, if a schedule of cost C and bound $\mathcal{E}$ exists. Note that C is the minimum total energy needed to achieve a minmax bound of $\mathcal{E}$. Then we apply this algorithm repeatedly over different choices of $\mathcal{E}$ until we get a schedule close to $C^{opt}, 2\mathcal{E}^{opt}$, where $\mathcal{E}^{opt}$ is the optimal minmax bound that can be achieved and C^{opt} is the minimum total energy needed to achieve this bound.

For any $\mathcal{E}_0 \geq \mathcal{E}$, a solution to the following integer program, ILP($\mathcal{E}_0$), will give a schedule of all vehicles that can be scheduled on some RSU with normalized energy at most $\mathcal{E}_0$ and a minmax bound at most $\mathcal{E}$: if $x_{i,j,t} = 1$ then vehicle j is scheduled on RSU i at time-slot t.

$$\text{minimize} \quad \sum_{j\in\mathcal{N}}\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{M}} \epsilon_{i,j,t}\, x_{i,j,t} \tag{ILP($\mathcal{E}_0$)}$$

$$\text{subject to} \quad \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{M}_j} x_{i,j,t} = 1 \qquad \forall j \in \mathcal{N} \tag{3.10}$$

$$\sum_{j \in \mathcal{N}} x_{i,j,t} \leq 1 \qquad \forall t \in \mathcal{T}, i \in \mathcal{M} \tag{3.11}$$

$$\sum_{j \in \mathcal{N}} \sum_{t \in \mathcal{T}} \tilde{\epsilon}_{i,j,t} \; x_{i,j,t} \leq \mathcal{E} \quad \forall i \in \mathcal{M} \tag{3.12}$$

$$x_{i,j,t} \in \{0,1\} \qquad \forall i \in \mathcal{M},\; j \in \mathcal{N},\; t \in \mathcal{T} \tag{3.13}$$

$$x_{i,j,t} = 0 \qquad \text{if } \tilde{\epsilon}_{i,j,t} > \mathcal{E}_0 \tag{3.14}$$

Constraint (3.12) enforces the minmax bound. Constraint (3.10) ensures that a vehicle is fully processed, and constraint (3.11) ensures that only one vehicle is processed in each time slot on each RSU. The last constraint makes sure that vehicles with energy requirement on an RSU greater than $\mathcal{E}_0 \geq \mathcal{E}$ are not scheduled at that RSU. Note that, due to constraints (3.10) and (3.11), the $x_{i,j,t}$ variables can only take values in $\{0,1\}$.

Unfortunately, the integer program ILP($\mathcal{E}_0$) is NP-hard to solve since it solves the Generalized Assignment problem, which is NP-complete. Therefore, we have relaxed it by allowing x to be fractional. In what follows, LP($\mathcal{E}_0$) refers to the fractional relaxation of the integer program ILP($\mathcal{E}_0$), and it can be solved using standard LP algorithms in polynomial time. Algorithm 1 performs a binary search on the minmax bound $\mathcal{E}$, in order to compute a fractional solution of minimum total cost. This happens in the while statements 4 to 12 using the binary search accuracy parameter ξ. Then it rounds the solution to an integral one, by applying a modification of the well-known Generalized Assignment Algorithm by Shmoys and Tardos (Shmoys and Tardos, 1993).

Algorithm 1 2-Approximation Bound

```
$\xi \rightarrow$ Binary search accuracy
$\mathcal{E} := \sum_{j \in \mathcal{N}} \sum_{t \in \mathcal{T}} \tilde{\epsilon}_{i,j,t}$
$P := \mathcal{E}$
while $P/\mathcal{E} < \xi$ do
    $(x, C) :=$ (Solution, Objective) of LP($\mathcal{E}_0$) (if feasible)
    $P := P/2$
    if LP($\mathcal{E}_0$) is feasible then
        $\mathcal{E} := \mathcal{E} - P$
    else
        $\mathcal{E} := \mathcal{E} + P$
    end if
end while
Round fractional matching $x$ of vehicles to time-slots and RSUs, to an integer matching $x_{round}$.
For each $x_{round}(i, j, t) = 1$, schedule job $j$ in time slot $t$ on RSU $i$.
```

To simplify the exposition, we assume that, for each RSU i, the normalized energies $\tilde{\epsilon}_{i,j,t}$ are ordered as follows:

$$\tilde{\epsilon}_{i,1,1} \geq \tilde{\epsilon}_{i,1,2} \geq \ldots \geq \tilde{\epsilon}_{i,1,l} \geq \tilde{\epsilon}_{i,2,1} \geq \tilde{\epsilon}_{i,2,2} \geq \ldots \geq \tilde{\epsilon}_{i,2,l} \geq \ldots \geq \tilde{\epsilon}_{i,n,l}$$

Theorem 1. *If LP($\mathcal{E}_0$) has a feasible solution with a total energy cost of C, then there exists a schedule of makespan at most $\mathcal{E} + \mathcal{E}_0$ and a total energy cost of at most C.*

The theorem is proven by an algorithm that rounds a feasible (fractional) solution of LP($\mathcal{E}_0$) to a schedule. First, we solve LP($\mathcal{E}_0$) to get a fractional solution x. Then we construct a tripartite graph $T(x) = (V \cup Q \cup W, E)$, with node set $V \cup Q \cup W$, defined as follows: V is the set of *vehicle nodes*:

$$V = \{v_j : j = 1, ..., n\}.$$

Q is the set of (time-slot, RSU) nodes, called *time nodes*:

$$Q = \{q_{(t,i)} : t = 1, ..., l,\ i = 1, ..., m\}.$$

W is the set of *RSU nodes*:

$$W = \{w_{is} : i = 1, ..., m, s = 1, ..., h_i\},$$

where $h_i = \lceil \sum_j \sum_t x_{i,j,t} \rceil$. Note that every RSU i generates h_i nodes in W. The edge set of $T(x)$ is $E = \{(v_j, q_{(t,i)}) \in V \times Q : \tilde{\epsilon}_{i,j,t} < \infty\} \cup \{(q, w) \in Q \times W\}$. The edge costs are 0 for all edges in $Q \times W$, and $c(v_j, q_{(t,i)}) := \epsilon_{i,j,t}$. The graph $T(x)$ is shown in Figure 3.3. For every path $v_j, q_{(t,i)}, w_{is}$ we define a value $x'(v_j, q_{(t,i)}, w_{is})$ as follows:

- If $\sum_{j=1}^{n} \sum_{t=1}^{l} x_{i,j,t} \leq 1$ for a RSU i, there is only one node, w_{i1}, corresponding to RSU i, in set W. In this case, we set $x'(v_j, q_{(t,i)}, w_{i1}) := x_{i,j,t}$.
- Otherwise, for RSU i, there are minimum j_1 and t_1 which $\sum_{j=1}^{j_1} \sum_{t=1}^{t_1} x_{i,j,t} \geq 1$. It should be noted that $\sum_{j=1}^{j_1} \sum_{t=1}^{t_1-1} x_{i,j,t} < 1$. We set $x'(v_j, q_{(t,i)}, w_{i1}) := x_{i,j,t}$, for $1 \leq j \leq j_1,\ 1 \leq t \leq t_1 - 1$. We also set $x'(v_{j_1}, q_{(t_1,i)}, w_{i1}) := 1 - \sum_{j=1}^{j_1} \sum_{t=1}^{t_1-1} x_{i,j,t}$. Up to here, we are sure that the sum of the components of x' for the paths incident to w_{i1} is exactly 1. If $\sum_{j=1}^{j_1} \sum_{t=1}^{t_1} x_{i,j,t} > 1$, a node w_{i2} is created to set the rest of the unassigned value of x_{i,j_1,t_1} to it. Therefore

  $$x'(v_{j_1}, q_{(t_1,i)}, w_{i2}) := x_{i,j_1,t_1} - x'(v_{j_1}, q_{(t_1,i)}, w_{i1}) = \sum_{j=1}^{j_1} \sum_{t=1}^{t_1} x_{i,j,t} - 1.$$

  Next, we apply the same strategy to the rest of vehicles $j \geq j_1$ and times $t > t_1$. In general, for each $s = 2, ..., h_i - 1$, we find the minimum indices j_s

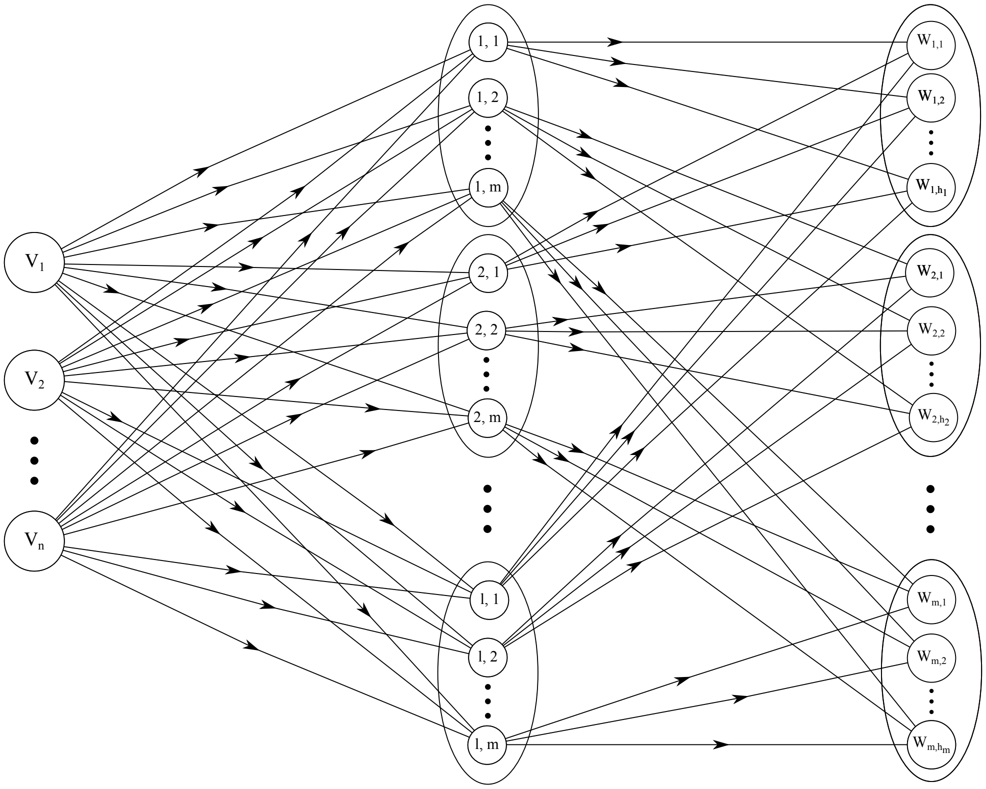


Figure 3.3: The graph $T(x)$. To simplify the figure, the source node s and the edges connecting it to nodes v_i are not shown.

and t_s, such that $\sum_{j=1}^{j_s}\sum_{t=1}^{t_s} x_{i,j,t} \geq s$. We set $x'(v_j, q_{(t,i)}, w_{is}) := x_{i,j,t}$, and $x'(v_{j_s}, q_{(t_s,i)}, w_{is}) := 1 - \sum_{j=j_{s-1}+1}^{j_s}\sum_{t=t_{s-1}+1}^{t_s-1} x_{i,j,t}$. If $\sum_{j=1}^{j_s}\sum_{t=1}^{t_s} x_{i,j,t} > 1$, a node $w_{i(s+1)}$ is created to set the rest of the unassigned value of x_{i,j_s,t_s} to it. Therefore

$$x'(v_{j_s}, q_{(t_s,i)}, w_{i(s+1)}) := x_{i,j_s,t_s} - x'(v_{j_s}, q_{(t_s,i)}, w_{is}) = \sum_{j=1}^{j_s}\sum_{t=1}^{t_s} x_{i,j,t} - s.$$

If (j_{h_i-1}, t_{h_i-1}) is the last (vehicle,time) pair for which this general method applies, then for $t > t_{h_i-1}$ and $j \geq j_{h_i-1}$ we create a node w_{ih_i} and set $x'(v_j, q_{(t,i)}, w_{ih_i}) := x_{i,j,t}$.

Note that the value of every variable $x_{i,j,t}$ is assigned to either a single $x'(v_{j_s}, q_{(t_s,i)}, w_{is})$ or is split between two $x'(v_{j_s}, q_{(t_s,i)}, w_{is}), x'(v_{j_s}, q_{(t_s,i)}, w_{i(s+1)})$, therefore

$$x_{i,j,t} = \sum_{s=1}^{h_i} x'(v_j, q_{(t,i)}, w_{is}), \;\forall j, t, i. \tag{3.15}$$

Let $\tilde{\epsilon}_{is}^{max}, \tilde{\epsilon}_{is}^{min}$ denote the maximum and minimum, respectively, of the normalized energy needed for communication $\tilde{\epsilon}_{i,j,t}$ corresponding to used $(x'(v_j, q_{(t,i)}, w_{is}) > 0)$ paths $(v_j, q_{(t,i)}, w_{is})$ ending at node w_{is}. Note that since vehicle j is (fractionally) assigned by LP($\mathcal{E}_0$), $\tilde{\epsilon}_{is}^{max} \leq \mathcal{E}_0, \;\forall i, s$. A non-negative vector z of edges weights on the edges of a bipartite graph is a *fractional matching* if, for each node u, the sum of the weights on the edges incident to u is at most 1. The fractional matching *exactly matches a node u* if this sum is exactly 1. A fractional matching z is a *matching* if each component of z is 0 or 1. The cost of a fractional matching is the summation of the cost of each edge e weighted by its edge weight z_e. The graph $T(x)$ is tripartite, but we can extend the definition of fractional matching to this case by thinking of paths $(v_j, q_{(t,i)}, w_{is})$ as “edges” connecting V to W (through Q). Then the following lemma is easy to prove by the construction of $T(x)$, the definition of x', and the fact that x is feasible for LP($\mathcal{E}_0$):

Lemma 1. *The vector x' is a fractional matching in $T(x)$ of total energy at most C. Also, we have $\tilde{\epsilon}_{is}^{min} \geq \tilde{\epsilon}_{i(s+1)}^{max}$ for each $i = 1, ..., m, \; s = 1, ..., h_i - 1$.*

Proof. **(of Theorem 1)** This is an extension of the proof of Shmoys and Tardos (1993). By Lemma 1, x' is a fractional matching in graph $T(x)$ of cost at most C, which matches all the vehicles exactly to the RSU’s through time slots. It can be extended to a flow in a standard way as follows: We give capacity 1 to all edges of

$T(x)$. We add a source node s and edges (s, v_j), $\forall j$ with capacity 1 (the demand of vehicle j) and cost 0 each. We also set $x'(s, v_j) := 1$, $\forall j$. It is easy to see that the extended x' is a flow that saturates all edges from s and has total energy cost C, and since all capacities are integral, we can compute an integral minimum cost flow x'' of cost at most C, that also satisfies the same total demand. This new flow gives us an assignment of all vehicles to RSU's through a single time slot each, i.e., a schedule, of cost at most C. An obvious upper bound for the energy load on each RSU i is

$$\begin{aligned}
\sum_{s=1}^{h_i} \tilde{\epsilon}_{is}^{max} &= \tilde{\epsilon}_{i1}^{max} + \sum_{s=2}^{h_i} \tilde{\epsilon}_{is}^{max} && (3.16)\\
&\leq \tilde{\epsilon}_{i1}^{max} + \sum_{s=1}^{h_i-1} \tilde{\epsilon}_{is}^{min} && (3.17)\\
&\leq \tilde{\epsilon}_{i1}^{max} + \sum_{s=1}^{h_i-1} \sum_{j=1}^{n} \sum_{t=1}^{l} \tilde{\epsilon}_{i,j,t} x'(v_j, q_{(t,i)}, w_{is}) && (3.18)\\
&\leq \tilde{\epsilon}_{i1}^{max} + \sum_{s=1}^{h_i} \sum_{j=1}^{n} \sum_{t=1}^{l} \tilde{\epsilon}_{i,j,t}\; x'(v_j, q_{(t,i)}, w_{is}) && (3.19)\\
&\overset{(3.15)}{=} \tilde{\epsilon}_{i1}^{max} + \sum_{t=1}^{l} \sum_{j=1}^{n} \tilde{\epsilon}_{i,j,t}\; x_{i,j,t} && (3.20)\\
&\leq \mathcal{E}_0 + \mathcal{E} && (3.21)
\end{aligned}$$

where the last inequality is due to the fact $\tilde{\epsilon}_{is}^{max} \leq \mathcal{E}_0$, $\forall i, s$. □

3.5 Online Scheduling Algorithms

The results in Section 3.3 give lower bounds on the downlink min-max RSU energy needed to fulfill vehicular packet requirements. In this section we consider online algorithms for performing the required scheduling in real time. As previously discussed in Section 3.3, in the online case the scheduling decisions can only be made using data consisting of past and current inputs. In the following sections we present low complexity algorithms that operate to load balance the normalized energy use across the RSUs, with varying levels of complexity and performance.

3.5.1 Greedy Online Algorithm (GOA)

The GOA algorithm uses both a greedy RSU selection and a greedy assignment of time slots. The details are shown in Algorithm 2. When a vehicle enters the network of RSUs, taking into account the energy that RSUs require to serve it, the RSU with the least accumulated normalized energy usage is selected. This is shown in Step 8 and this vehicle is added to the set of vehicles assigned to that RSU, i.e., $\mathcal{S}_r$. Time slots (H_v for vehicle v) are then assigned from those available (i.e., from the set $U_{r,t}$), which minimize the cost of serving that vehicle, which is shown in Step 6. Once these time slots have been allocated, in Step 9 they are removed from $U_{r,t}$ and are unavailable to any subsequently arriving vehicles. The value of $\mathcal{C}_{i,t}$ is also updated. As previously discussed, in the SRA case, rather than assigning the vehicle to a particular RSU, we view the H_v time slot requirement for vehicle v to be H_v separate single time slot vehicles, allowing them to be assigned across multiple RSUs. A major advantage of GOA is that it is very simple to implement.

Algorithm 2 Greedy Online Algorithm (GOA)

```
1: $U_{r,t} \rightarrow$ Set of unassigned RSU $r$ time slots at time $t$.
2: $\mathcal{C}_{r,t} \rightarrow$ Accumulated normalized energy usage of RSU $r$ at time $t$.
3: $P_r \rightarrow$ Normalized energy usage of RSU $r$ for a new vehicle.
4: for all $t \in \{0, 1, \ldots\}$ do
5:     for each vehicle $v$ that arrives to the system (at time $t$) do
6:         Find time slots from set $U_{r,t}$ that minimizes the energy cost of communication
           with vehicle $v$ for all RSUs in the set $\mathcal{M}_v$.
7:         Update $P_r$, for each RSU, if the time slots obtained in step 6 are used to
           serve vehicle $v$.
8:         Assign $v$ to RSU $r = \arg\min_{i \in \mathcal{M}_v} \mathcal{C}_{i,t} + P_i$.
           (i.e., $\mathcal{S}_r \leftarrow \mathcal{S}_r \cup v$.)
9:         Update $U_{r,t}$ by removing those time slots from $r$ assigned in Step 6 and
           update the value of $\mathcal{C}_{i,t}$ to account for the new assignments.
10:        Update the schedule for RSU $r$ using the solution generated in Step 6.
11:    end for
12:    Using the current schedule for each RSU, continue RSU-to-vehicle transmission
       (at time slot $t$).
13: end for
```

3.5.2 Greedy Flow Graph (GFGA) Scheduler

As in the GOA algorithm, in the Greedy Flow Graph (GFGA) algorithm, RSU selection is made for a newly arrived vehicle based on minimum RSU energy use plus the energy that RSUs require to serve it. Once the target RSU is chosen, time slots are allocated to the vehicles which are assigned to that RSU based on a minimum cost energy schedule. This is computed using all currently available vehicular information and remaining backlog, and is recomputed whenever a new vehicle arrival occurs. The algorithm is shown in Algorithm 3 and is described in detail as follows.

When a new vehicle arrives to the system, it is assigned to the RSU which currently has the minimum accumulated normalized energy usage. $C_{i,t}$ is defined to be the normalized energy usage for RSU i at time t as shown in Step 10. $\mathcal{S}_r \subseteq \mathcal{N}$ is defined to be the set of active vehicles currently assigned to RSU r. In Step 7 we update the set of unassigned time slots for which all vehicles assigned to RSU r will be active. Then in Step 8 we use these updated inputs to find the minimum energy time slot assignment by solving the following ILP.

$$\begin{aligned}
\underset{x_{r,j,t}}{\text{minimize}} \quad & \sum_{t\in\mathcal{T}_r}\sum_{j\in\mathcal{S}_r} \epsilon_{r,j,t}\; x_{r,j,t} && && (3.22)\\
\text{subject to} \quad & \sum_{t\in\mathcal{T}_r} x_{r,j,t} = \widetilde{H}_j, && \forall j \in \mathcal{S}_r && (3.23)\\
& \sum_{j\in\mathcal{S}_r} x_{r,j,t} \leq 1, && \forall t \in \mathcal{T}_r && (3.24)\\
& x_{r,j,t} \in \{0,1\}, && \forall \{j \in \mathcal{S}_r, t \in \mathcal{T}_r\} && (3.25)
\end{aligned}$$

ILP 3.22 to 3.25 is similar to ILP-Bound except that it solves for the minimum energy schedule for a single RSU r, using the currently available inputs $\mathcal{S}_r$ and $\mathcal{T}_r$. The objective function therefore only considers the energy cost for RSU r. Constraint (3.23) satisfies the *residual* (i.e., remaining unserved) transmission requirement for vehicle j, denoted by $\widetilde{H}_j$. The other constraints follow similarly from ILP (3.4) to (3.9). In Step 11, once the new assignments are made, the value of $\mathcal{C}_{i,t}$ is updated. As previously discussed, in the SRA case, rather than assigning the vehicle to a particular RSU, we view the H_j time slot requirement of vehicle j to be H_j separate one time slot vehicles, allowing them to be assigned across multiples RSUs.

The form of ILP (3.22) to (3.25) is such that it can be solved in time complexity which is polynomial in the number of time slots, using a minimum cost flow graph formulation (Ahuja *et al.*, 1994). This is shown for RSU r in Figure 3.4, where $G = (V, E)$ is defined by a set V of vertices (nodes) and a set E of edges (arcs) connecting the nodes. For each edge $(i, j) \in E$ there is a capacity $u_{i,j}$ that gives the

Algorithm 3 Greedy Flow Graph (GFGA) Scheduler

```
1: $\mathcal{C}_{r,t} \rightarrow$ Accumulated normalized energy usage of RSU $r$ at time $t$.
2: $\mathcal{S}_r \rightarrow$ Set of vehicles currently assigned to RSU $r$.
3: $\mathcal{T}_r \rightarrow$ Set of unassigned time slots for which vehicles in $\mathcal{S}_r$ are within RSU $r$
   coverage.
4: $P_r \rightarrow$ Normalized energy usage of RSU $r$ for a new vehicle.
5: for all $t \in \{0, 1, \ldots\}$ do
6:    for each vehicle $v$ that arrives to the system (at time $t$) do
7:       For each RSU, update $\mathcal{T}_r$ to the union of unassigned time slots for which all
           vehicles in $S_r$ are within the RSU $r$ coverage.
8:       Solve ILP 3.22 to 3.25 for each RSU $r$ using the vehicles in $\mathcal{S}_r$ for the time
           slots in $\mathcal{T}_r$. This can be done in time which is polynomial in $|\mathcal{T}_r|$ using a
           minimum cost flow graph.
9:       Update $P_r$, for each RSU, if the time slots obtained in step 8 are used to
           serve vehicle $v$.
10:      Assign $v$ to RSU $r = \arg\min_{i \in \mathcal{M}_i} \mathcal{C}_{i,t} + P_i$.
           (i.e., $\mathcal{S}_r \leftarrow \mathcal{S}_r \cup v$.)
11:      Update the schedule for RSU $r$ using the solution generated in Step 8. Update
           the value of $\mathcal{C}_{i,t}$ to account for the new assignments.
12:   end for
13:   Using the current schedule for each RSU, continue RSU-to-vehicle transmission
        (at time slot $t$).
14: end for
```

maximum flow on the edge, and an associated cost, $c_{i,j}$, that denotes the cost per unit of flow on that edge. These are written as ordered pairs, $(u_{i,j}, c_{i,j})$, on each graph edge in Figure 3.4.

The flow enters and exits the graph at dummy nodes S and D, respectively. The first column of nodes represents all vehicles in $\mathcal{S}_r$, where $N_r = |\mathcal{S}_r|$. The second column represents all time slots in $\mathcal{T}_r$, where $T_r = |\mathcal{T}_r|$. Each vehicle node has edges connected to the time slot nodes during which the vehicle is inside the RSU r coverage area. The capacity for an edge from the source S to a vehicle node is the residual communication requirement for vehicle j in time slots. The capacity for an edge from

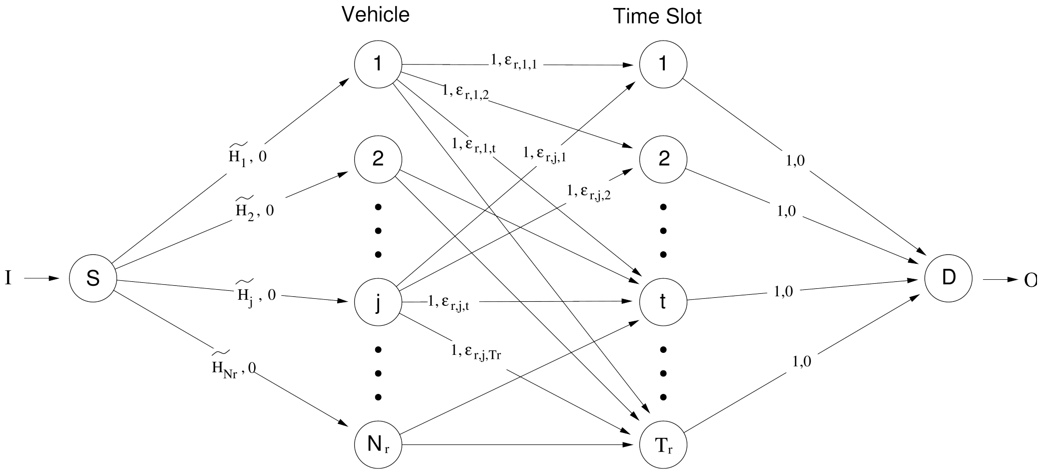


Figure 3.4: Minimum Energy Flow Graph Scheduler. Each edge is labeled with an ordered pair, $(u_{r,j,t}, \epsilon_{r,j,t})$, where $u_{r,j,t}$ and $\epsilon_{r,j,t}$ are the capacity and cost of using edge (i, j). The input and output links, I and O, carry a flow of $\sum_{i=1}^{N} \widetilde{H_i}$ with a 0 edge cost.

any time slot node to the destination D is 1 which prevents time slots from being used more than once. The edges between vehicle and time slot nodes also have a capacity of 1 which ensures that only one unit of transmission requirement can be assigned to a given time slot. The cost for using the edges originating from Node S or terminating at Node D is zero. Finally, the cost of the edges between the vehicle and time slot nodes is given by $\epsilon_{r,j,t}$ which is the energy cost of communication from RSU r to vehicle j at that time. Finding the minimum cost flow for graph G provides the minimum energy the RSU must consume to schedule vehicle transmission requirements for the given set of inputs. The Integrality Property Theorem (Ahuja *et al.*, 1994) ensures that provided input flows and capacities are integer, the resulting minimum cost flow will also be integer. Since our vehicle to time slot edge capacities are 1, the resulting path flows are binary and give the optimum values for $x_{r,j,t}$.

Once the schedule for RSU r has been updated via Step 11 in Algorithm 3, this becomes the *active* schedule for that RSU. Finally, in Step 13 all RSUs will transmit

Algorithm 4 2-Approximation Online Algorithm (TOAA)

```
1: $\mathcal{C}_r$ → Accumulated normalized energy usage of RSU $r$.
2: $\mathcal{S}$ → Set of vehicles currently in the coverage area of RSUs.
3: $\mathcal{W}$ → Set of unassigned time slots for which vehicles in $\mathcal{S}$ are within RSUs coverage.
4: for all $t_c \in \{0, 1, \ldots\}$ do
5:     for each vehicle $v$ that arrives to the system (at time $t_c$) do
6:         ($\mathcal{S} \leftarrow \mathcal{S} \cup v$.)
7:         Update $\mathcal{W}$ to the union of unassigned time slots for which there is at least a vehicle within the coverage of an RSU.
8:         Solve (ILP($\mathcal{E}_0$)) for time slots $\mathcal{W}$ using vehicles in $\mathcal{S}$. Algorithm 1 can be used to solve the ILP.
9:         Update the schedule for all RSUs using the solution generated in Step 8.
10:    end for
11:    Using the current schedule for each RSU, continue RSU-to-vehicle transmission (at time slot $t_c$). Update the value of $\mathcal{C}_i$ to account for the new transmission.
12:    Remove vehicles which are served in step 11, from $\mathcal{S}$.
13: end for
```

if time slot t has been assigned.

3.5.3 2-Approximation Online Algorithm (TOAA) for Splittable RSU Assignment

In this section we present an online version of Algorithm 1, referred to as the 2-Approximation Online Algorithm (TOAA). RSU and time slot selection are made for a newly arrived vehicle based on minimizing both normalized makespan (i.e. maximum normalized RSU energy), and total time (i.e., total energy). This is computed using currently available vehicular information and remaining backlog, and is computed whenever a new vehicle arrives. Recall that this algorithm is for the splittable RSU assignment case (SRA) and $H_j = 1$ for all vehicles j. This algorithm is shown in Algorithm 4 and described in detail as follows.

$\mathcal{C}_i$ is defined as accumulated normalized energy usage of RSU i. $\mathcal{S}$ is defined to be the set of active vehicles currently within the coverage area of RSUs. In step 7, $\mathcal{W}$ is updated to the set of unassigned time slots for which there is at least a vehicle in the coverage area of an RSU. In step 8 we use these updated inputs to find the best minimum and a balanced energy time slot assignment by solving the following ILP.

$$
\begin{aligned}
\text{minimize} \quad & \sum_{j\in\mathcal{S}}\sum_{t\in\mathcal{W}}\sum_{i\in\mathcal{M}} \epsilon_{i,j,t}\ x_{i,j,t} && && (\mathrm{ILP}(\mathcal{E}_0))\\
\text{subject to} \quad & \sum_{t\in\mathcal{W}}\sum_{i\in\mathcal{M}_j} x_{i,j,t} = 1 && \forall j\in\mathcal{S} && (3.26)\\
& \sum_{j\in\mathcal{S}} x_{i,j,t} \le 1 && \forall t\in\mathcal{W},\ i\in\mathcal{M} && (3.27)\\
& \sum_{j\in\mathcal{S}}\sum_{t\in\mathcal{W}} \tilde{\epsilon}_{i,j,t}\ x_{i,j,t} + \mathcal{C}_i \le \mathcal{E} && \forall i\in\mathcal{M} && (3.28)\\
& x_{i,j,t} \ge 0 && \forall i\in\mathcal{M},\ j\in\mathcal{S},\ t\in\mathcal{W} && (3.29)\\
& x_{i,j,t} = 0 && \text{if } p_{i,j,t} > t_0 && (3.30)
\end{aligned}
$$

Note that $(\mathrm{ILP}(\mathcal{E}_0))$ is similar to the 2-Approximation Bound that solves the bi-objective ILP to minimize the makespan and total energy, using currently available inputs $\mathcal{S}$ and $\mathcal{W}$. The objective function is minimizing total energy needed to schedule vehicles in set $\mathcal{S}$. Constraint (3.26) ensures that vehicle demands are satisfied. Each time slot for a given RSU can be assigned to a vehicle by constraint (3.27). Taking into account the accumulated normalized energy usage of the RSUs, constraint (3.28) places an upper bound on the normalized energy needed to serve vehicles in set $\mathcal{S}$ for each RSU. If C^{opt} and $\mathcal{E}^{opt}$ are the optimal bound for the ILP, Algorithm 1 can be used to find a schedule with C^{opt} and $2\mathcal{E}^{opt}$.

3.5.4 One-Objective and Bi-Objective Online Scheduling Algorithms

As in Section 3.4.1, we note that our problem can be viewed as an online load-balancing machine scheduling problem where time is replaced by energy. Based on this observation, we develop online algorithms for both the splittable (SRA) and unsplittable (URA) RSU assignment cases. Recall that in the splittable case, $H_j = 1$ for all vehicles j, and $\epsilon_{i,j,t}$ is the energy needed for RSU i communicating with j during time-slot t. In the unsplittable case, we will abuse notation, and we will also use $\epsilon_{i,j,t}$ to denote the energy needed by RSU i to communicate with j during *a set* of H_j time-slots t which have been chosen *greedily*, i.e., t is the set of H_j most energy-efficient time-slots available for RSU i. The normalized energies $\tilde{\epsilon}_{i,j,t}$ are defined analogously.

Suppose that the vehicles are ordered according to their arrival order, and that newly arrived vehicle j is being scheduled on RSU i (while vehicles $1, 2, \ldots, j-1$ have already been scheduled on RSU's). Then the *load* $l_r(j)$ on RSU $r = 1, 2, \ldots, m$ after j vehicles have been scheduled is recursively defined as

$$l_r(j) = \begin{cases} l_r(j-1) + \epsilon_{r,j,t}, & r = i \\ l_r(j-1), & \text{otherwise} \end{cases}$$

(obviously we set $l_r(0) = 0$ for all r). The *normalized* load of RSU i after the scheduling of j vehicles is defined as $\tilde{l}_i(j) = l_i(j)/\eta_i$. The two objective functions that are the objectives of our scheduling problem, given that k vehicles arrive, become:

Min-max: Compute a schedule that minimizes $L(k) := \max_i \tilde{l}_i(k)$.

Total energy: Compute a schedule that minimizes $E(k) := \sum_{r=1}^{m} l_r(k)$.

Note that *min-max* is a fairness objective doing load-balancing across all RSU's, so it uses the normalized loads, while *total energy* is a quantity that doesn't depend on the relative capacities of the RSU's, and, therefore, uses the regular loads.

Since the vehicles arrive in real time, we design *online* algorithms that compute a solution to our scheduling problem while their input arrives in a piece-meal fashion; this is in contrast to our *offline* algorithms that assume the whole input is available from the start, and which we use in order to calculate lower bounds for the online ones. In this section, we present two online algorithms that are variations of the well-known algorithm of Aspnes et al. (J. Aspnes and *et al.*, 1997). The One-Objective algorithm attempts to optimize the *min-max* objective, and the Bi-Ojective algorithm attempts to do so with the minimum *total energy.* Intuitively, every time a new vehicle arrives and the One-Objective algorithm is run, it attempts to estimate the optimal min-max value and then make a placement decision for the vehicle requirement that will keep the schedule close to its estimate. This is done by using an exponential potential function of the RSU loads. The algorithm returns the updated load and its estimate of the optimal. The Bi-Objective algorithm still solves the *min-max* problem like the One-Objective algorithm does, but, in this case, it also tries to estimate the minimum total energy needed to achieve the min-max, by doing a binary search over the total energy "budget" needed to achieve this maximum load.

We present a theoretical analysis of the performance of our One-Objective algorithm, by extending the proofs of (J. Aspnes and *et al.*, 1997) for bounding the *competitive ratio.* The competitive ratio is a standard measure of performance for online algorithms and is defined as the *worst-case* ratio of the objective value achieved by the online algorithm over the objective value achieved by the optimal offline algorithm, i.e., the competitive ratio measures the worst (over all possible input sequences) gap between the solution quality of the online algorithm and that of the optimal all-knowing offline algorithm. Obviously, the competitive ratio is a pessimistic measure, and one would expect an online algorithm to do better in practice.

Let $\mathcal{A}^*$ be the offline algorithm that, given a sequence of k vehicles, calculates a schedule with RSU loads $l_i^*(k)$, that minimizes the normalized maximum load $L^*(k) = max_i\ \tilde{l}_i^*(k)$, using the minimum possible total energy $E^*(k) = \sum_{i=1}^n l_i^*(k)$. If $\mathcal{A}$ is one of our online algorithms, and $L(k)$ is the worst maximum RSU load it achieves over all possible input sequences of length k, then its competitive ratio is $L(k)/L^*(k)$. We will show that for the One-Objective algorithm, this ratio is bounded by $O(\log m)$. In what follows, the parameters Λ and Δ, such that $\Lambda \geq L^*$ and $\Delta \geq E^*$, are the algorithms' estimates of min-max and the corresponding optimal total energy, respectively.

One-Objective Algorithm

The One-Objective algorithm is run every time there is a new arrival j; it takes as input the current estimate Λ of the min-max L^* and the current RSU load vector $\vec{l}$ (together with the set of time-slots that are already occupied for every RSU). Its output is a new estimate as well as a new load vector, after the new arrival has been scheduled. Note that none of the previous arrivals is rescheduled. Its analysis will use parameters a, β, γ. Once a has been set, β and γ are also determined, as described below. The following lemma shows that the competitive ratio for min-max is upper-bounded by $4\beta = O(\log m)$, where the factor 4 is due to the fact that the algorithm is

Algorithm 5 One-Objective(Λ, l)

```
$\Lambda$ $\rightarrow$ Estimate of $L^*$ initialized with $\min_{(i,t)}\epsilon_{i,1,t}$
$\beta$ $\rightarrow$ Designed performance guarantee of the algorithm
$j$ $\rightarrow$ Vehicle currently being scheduled
for all $i, t$ do
    $\bar{l}_i := l_i/(\Lambda * \eta_i)$
    $\bar{\epsilon}_{i,t} := \epsilon_{i,j,t}/(\Lambda * \eta_i)$
    $\Gamma_{i,t} := (a^{\bar{l}_i+\bar{\epsilon}_{i,t}} - a^{\bar{l}_i})$
end for
$(i', t') = \arg\min_{(i,t)}\Gamma_{i,t}$
while $(l_{i'} + \epsilon_{i',j,t'})/\eta_{i'} > \beta\Lambda$ do
    $\Lambda = 2 * \Lambda$
    for all $i$ do
        $l_i := 0$
    end for
    for all $i, t$ do
        $\bar{l}_i := l_i/(\Lambda * \eta_i)$
        $\bar{\epsilon}_{i,t} := \epsilon_{i,j,t}/(\Lambda * \eta_i)$
        $\Gamma_{i,t} := (a^{\bar{l}_i+\tilde{\epsilon}_{i,t}} - a^{\bar{l}_i})$
    end for
    $(i', t') = \arg\min_{(i,t)}\Gamma_{i,t}$
end while
Schedule vehicle $j$ to RSU $i'$ at time-slot(s) $t'$
$l_{i'} := l_{i'} + \epsilon_{i',j,t'}$
return $(\Lambda, l)$
```

using the estimate Λ instead of $L^*(k)$ in its calculations. Initially, and assuming that j is the vehicle that we are scheduling now, $\Lambda = min_{i,t}\ \epsilon_{i,j,t}$. We give the description of the algorithm for the splittable (SRA) case, but the same description, as well as Theorem 2 that follows, also work for the URA case, if t in the indexing of $\epsilon_{i,j,t}$ is the greedily-chosen set of time-slots reserved on RSU i by the (possibly non-unit) requirement H_j of vehicle j.

Theorem 2. *If $L^* \leq \Lambda$, then there exists $\beta = O(\log m)$ such that the One-Objective algorithm produces a valid schedule. Therefore, the load on an RSU never exceeds*

$\beta\Lambda$.

Proof. This is an extension of the proof in (J. Aspnes and *et al.*, 1997). We name the vehicles using their order in their arrival ordering, i.e., vehicle j is also the j-th vehicle arrival. In what follows, we define $\bar{l}_i(j) := l_i(j)/(\Lambda * \eta_i)$ and $\bar{\epsilon}_{i,t}(j) := \epsilon_{i,j,t}/(\Lambda * \eta_i)$, for all vehicles j and all RSU i and (sets of) time-slots t. We define the potential function

$$\Phi(j) = \sum_{r=1}^{m} a^{\bar{l}_r(j)}(\gamma - \bar{l}_r^*(j)), \quad \Phi(0) = m\gamma, \tag{3.31}$$

where m is the number of RSUs, and a, $\gamma > 1$ are constants to be defined later. If we assume that vehicle $j+1$ is assigned to RSU i' by the online algorithm and to RSU i by the offline algorithm, then

$$\begin{aligned}\Phi(j+1) - \Phi(j) &= (\gamma - \bar{l}_{i'}^*(j))(a^{\bar{l}_{i'}(j+1)} - a^{\bar{l}_{i'}(j)}) - \bar{\epsilon}_{i,j+1,t}a^{\bar{l}_i(j+1)} \\ &\leq \gamma(a^{\bar{l}_{i'}(j)+\bar{\epsilon}_{i',j+1,t'}} - a^{\bar{l}_{i'}(j)}) - \bar{\epsilon}_{i,j+1,t}a^{\bar{l}_i(j)} \\ &\leq \gamma(a^{\bar{l}_i(j)+\tilde{\epsilon}_{i,j+1,t}} - a^{\bar{l}_i(j)}) - \bar{\epsilon}_{i,j+1,t}a^{\bar{l}_i(j)} \\ &= a^{\bar{l}_i(j)}(\gamma(a^{\bar{\epsilon}_{i,j+1,t}} - 1) - \bar{\epsilon}_{i,j+1,t}). \end{aligned} \tag{3.32}$$

The first inequality comes from the fact that $\bar{l}_r^*(j)$ is non-negative. The last inequality is true since $(i', t') = \arg\min_{(v,s)}(a^{\bar{l}_v(j)+\tilde{\epsilon}_{v,j+1,s}} - a^{\bar{l}_v(j)})$.

Inequality $0 \leq \bar{\epsilon}_{i,j+1,t} \leq L^*(j+1)/\Lambda \leq 1$ is true, since the optimal algorithm assigns vehicle $j+1$ to RSU i at time-slot(s) t. Hence the potential function does not increase if, for $x \in [0,1]$, we have $\gamma(a^x - 1) \leq x$; this is true if we set γ to satisfy $a = 1 + 1/\gamma$. By adding-up (3.32) from 0 to $j-1$, we get

$$\sum_{r=1}^{m} a^{\bar{l}_r(j)}(\gamma - 1) \leq \gamma m,$$

Then

$$a^{\bar{l}_r(j)} \leq \sum_{r=1}^{m} a^{\bar{l}_r(j)} \leq \frac{\gamma m}{\gamma - 1} \Rightarrow \bar{l}_r(j) \leq \log \frac{\gamma m}{\gamma - 1}, \ \forall r$$

which implies that

$$L(j) = \max_i \frac{l_i(j)}{\eta_i} \leq \Lambda \log \frac{\gamma m}{\gamma - 1} = \Lambda O(\log m). \tag{3.33}$$

□

If we pick a, then γ is defined by $a = 1 + 1/\gamma$ and β by $\beta = \log_a \frac{\gamma m}{\gamma - 1}$, so that Theorem 2 holds.

Theorem 3. *The One-Objective algorithm produces a schedule where the competitive ratio of the min-max energy load on the RSUs is upper bounded by* 4β.

Proof. Theorem 2 gives a bound on the min-max performance of the algorithm for a series of vehicles scheduled by One-Objective when the upper bound on min-max energy, Λ, is known and fixed. This information however, will not be given a priori, and therefore we must use estimates of Λ. The algorithm works by starting at a small estimate of Λ, and each time we schedule a downlink transmission, we test that the bound is still valid by ensuring that the updated energy load on the RSU in question does not exceed $\beta\Lambda$ (from Equation (3.33)). Assuming that j is the initial vehicle to be scheduled at time t, the starting value for Λ is given by $min_{i,t}\ \epsilon_{i,j,t}$. We refer to a period of time where Λ remains fixed as a “phase”, and each time the test on $\beta\Lambda$ fails, we double Λ. This is shown in Algorithm 5 in the while loop. When we enter this loop, a new phase begins and we reset our energy load variables and begin processing vehicles with the new (doubled) value of Λ.

After all vehicles have been processed, assume that the system finishes while in phase ϕ, counting from an initial phase 0. From Theorem 2 we know that in phase p, the bound on the min-max performance of the algorithm is given by $\beta\, 2^p\, \Lambda_0$ where Λ_0 is the initial value chosen for Λ during phase 0. This is due to the doubling of Λ at the start of each phase. Therefore, the total bound on competitive ratio is given by $\sum_{p=0}^{\phi} \beta\, 2^p\, \Lambda_0 = \beta\, (2^{\phi+1} - 1)\, \Lambda_0$. From Theorem 2 it can be shown that when entering phase ϕ, the min-max performance of the optimal offline algorithm must exceed $2^{\phi-1}\Lambda_0$, otherwise a new phase would not have started. Combining these last two results, we see that the competitive ratio of Algorithm 5 is bounded by $4\,\beta$. □

Bi-Objective Algorithm

Motivated by the exponential potential function approach used in Section 3.5.4, we modify One-Objective to obtain the Bi-Objective algorithm. Although the Bi-Objective algorithm uses an exponential decision function as in the One-Objective algorithm, it also tries to accomplish this with minimum total energy. Initially the algorithm works exactly as in One-Objective as it sets the current energy budget Δ to, essentially infinity, and finds the best min-max estimate Λ. Once such a Λ has been identified, it's value is held constant, and the algorithm does a binary search on the total energy budget Δ. The idea is to find the smallest Δ that maintains Λ as a good estimate for min-max performance. Initially, and assuming that the vehicle we are scheduling now is vehicle j, $\Lambda = min_{i,t}\, \epsilon_{i,j,t}$ and $\Delta = \sum_{i=1}^{m} \eta_i \beta \Lambda$.

As before, we give the algorithm description for the SRA case, but the same description works for the URA case. Note that since the algorithm performs a binary search on the estimate Δ, the final Δ is doubled to account for the case the last step of the binary search brings Δ below E^*. β is defined by $\beta = \log_a \frac{\gamma(m+1)}{\gamma-1}$, where γ and a are selected as they were picked for One-Objective.

Algorithm 6 Bi-Objective(Λ, l)

```
$\Lambda$ $\rightarrow$ Current estimate of $L^*$ initialized with $\min_{(i,t)}\epsilon_{i,1,t}$
$\Delta$ $\rightarrow$ Estimate of $E^*$ initialized with $\sum_{i=1}^{m}\eta_i\beta\Lambda$
$\beta$ $\rightarrow$ Designed performance factor
$j$ $\rightarrow$ Vehicle currently being scheduled
**for all** $i, t$ **do**
    $\bar{l}_i := l_i/(\Lambda * \eta_i)$
    $\bar{\epsilon}_{i,t} := \epsilon_{i,j,t}/(\Lambda * \eta_i)$
    $\hat{l}_i := l_i/\Delta$
    $\hat{p}_{i,t} := \epsilon_{i,j,t}/\Delta$
    $\Gamma_{i,t} = (a^{\bar{l}_i+\bar{\epsilon}_{i,t}} - a^{\bar{l}_i}) + (a^{\sum_{r=1}^{m}\hat{l}_r+\hat{p}_{i,t}} - a^{\sum_{r=1}^{m}\hat{l}_r})$
**end for**
$(i', t') = \arg\min_{(i,t)}\Gamma_{i,t}$
**while** $(l_{i'} + \epsilon_{i',j,t'})/\eta_{i'} > \beta\Lambda$ **do**
    $\Lambda = 2 * \Lambda$
    $\Delta = \sum_{i=1}^{m}\eta_i\beta\Lambda$
    **for all** $i$ **do**
        $l_i := 0$
    **end for**
    **for all** $i, t$ **do**
        $\bar{l}_i := l_i/(\Lambda * \eta_i)$
        $\bar{\epsilon}_{i,t} := \epsilon_{i,j,t}/(\Lambda * \eta_i)$
        $\hat{l}_i := l_i/\Delta$
        $\hat{p}_{i,t} := \epsilon_{i,j,t}/\Delta$
        $\Gamma_{i,t} = (a^{\bar{l}_i+\bar{\epsilon}_{i,t}} - a^{\bar{l}_i}) + (a^{\sum_{r=1}^{m}\hat{l}_r+\hat{p}_{i,t}} - a^{\sum_{r=1}^{m}\hat{l}_r})$
    **end for**
    $(i', t') = \arg\min_{(i,t)}\Gamma_{i,t}$
**end while**
**while** $\sum_{r=1}^{m} l_r + \epsilon_{i',j,t'} \leq \beta\Delta$ **do**
    $\Delta = \Delta/2$
    **for all** $i, t$ **do**
        $\hat{l}_i := l_i/\Delta$
        $\hat{p}_{i,t} := \epsilon_{i,j,t}/\Delta$
        $\Gamma_{i,t} = (a^{\bar{l}_i+\bar{\epsilon}_{i,t}} - a^{\bar{l}_i}) + (a^{\sum_{r=1}^{m}\hat{l}_r+\hat{p}_{i,t}} - a^{\sum_{r=1}^{m}\hat{l}_r})$
    **end for**
    $(i', t') = \arg\min_{(i,t)}\Gamma_{i,t}$
    **if** $(l_{i'} + \epsilon_{i',j,t'})/\eta_{i'} > \beta\Lambda$ **then**
        break
    **end if**
**end while**
```

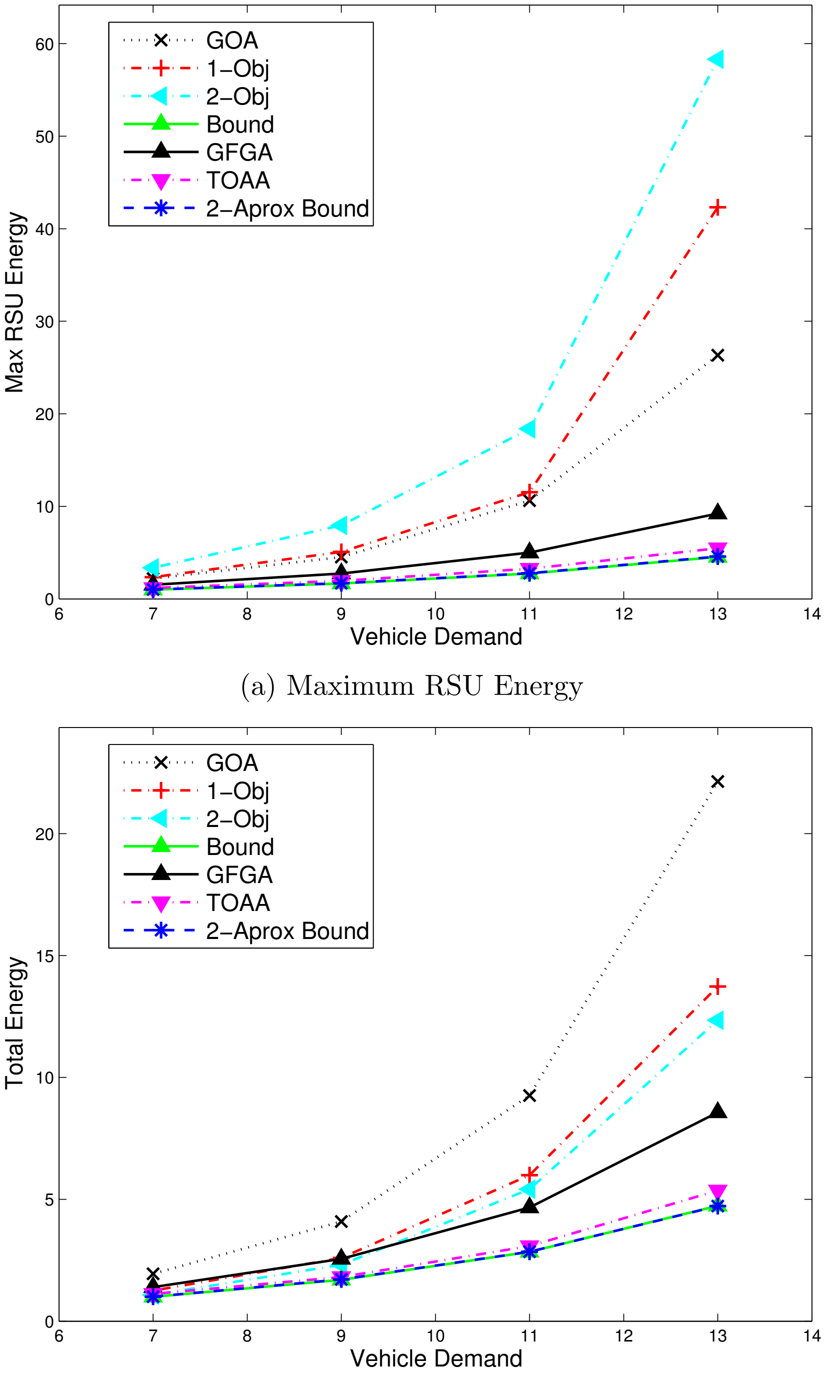
```
$\Delta = \Delta * 2$
**for all** $i, t$ **do**
    $\hat{l}_i := l_i/\Delta$
    $\hat{p}_{i,t} := \epsilon_{i,j,t}/\Delta$
    $\Gamma_{i,t} = (a^{\bar{l}_i+\bar{\epsilon}_{i,t}} - a^{\bar{l}_i}) + (a^{\sum_{r=1}^{m} \hat{l}_r+\hat{p}_{i,t}} - a^{\sum_{r=1}^{m} \hat{l}_r})$
**end for**
$(i', t') = \arg\min_{(i,t)} \Gamma_{i,t}$
Schedule vehicle $j$ to RSU $i'$ at time-slot(s) $t'$
$l_{i'} = l_{i'} + \epsilon_{i',j,t'}$
return $(\Lambda, l)$
```

Chapter 4

Performance Evaluation

In this chapter we discuss the performance of the proposed algorithms. A wide variety of experiments have been performed and we will present samples which are representative of the results that we have obtained. The theoretical bound for min-max RSU energy consumption as derived in Section 3.3 is referred to as *Bound* in the graphs and is compared to the online algorithms discussed in Section 3.5. The input data to the schedulers is taken from a highway environment where vehicles are assumed to maintain constant speed throughout the RSU coverage areas (Khabazian and Ali, 2008) (Hammad *et al.*, 2013). It has been shown that in this type of scenario, good estimates of energy costs can be readily made (Wang, 2005)(C. Sommer and Dressler, 2011). The associated energy cost inputs are based on vehicle position and associated estimates of downlink transmission energy costs. We assume that the energy costs come from a distance dependent exponential path loss model with a path loss exponent of $\alpha = 3$. However, in many practical systems there will be dominant deterministic propagation with random components due to effects such as shadowing. Therefore, we also include results which incorporate errors due to strong shadowing components. The models used assume Poisson process vehicle arrivals as in (Wang, 2005), (Khabazian and Ali, 2008) and (Bilstrup *et al.*, 2008). For all the experiments we assume that $M = 4$ and that the RSU coverage radius is 1 Km. The relative energy capacities of the four RSUs are set to 1, .67, .33, and .167, respectively. Three combinations of RSUs are assumed from the set of all RSUs. When a vehicle is assigned to one of these subsets, the vehicle demands have to be satisfied by the RSUs specified within the subset. RSUs (4,3,1), (4,2), and (3,2) form the three RSUs subsets. Vehicles are assigned to a subset with a uniform distribution. The values of the points in the graphs are normalized to the first point of the Bound graph in each figure. Each figure includes a pair of graphs, consisting of plots for maximum normalized RSU energy, and total RSU energy.

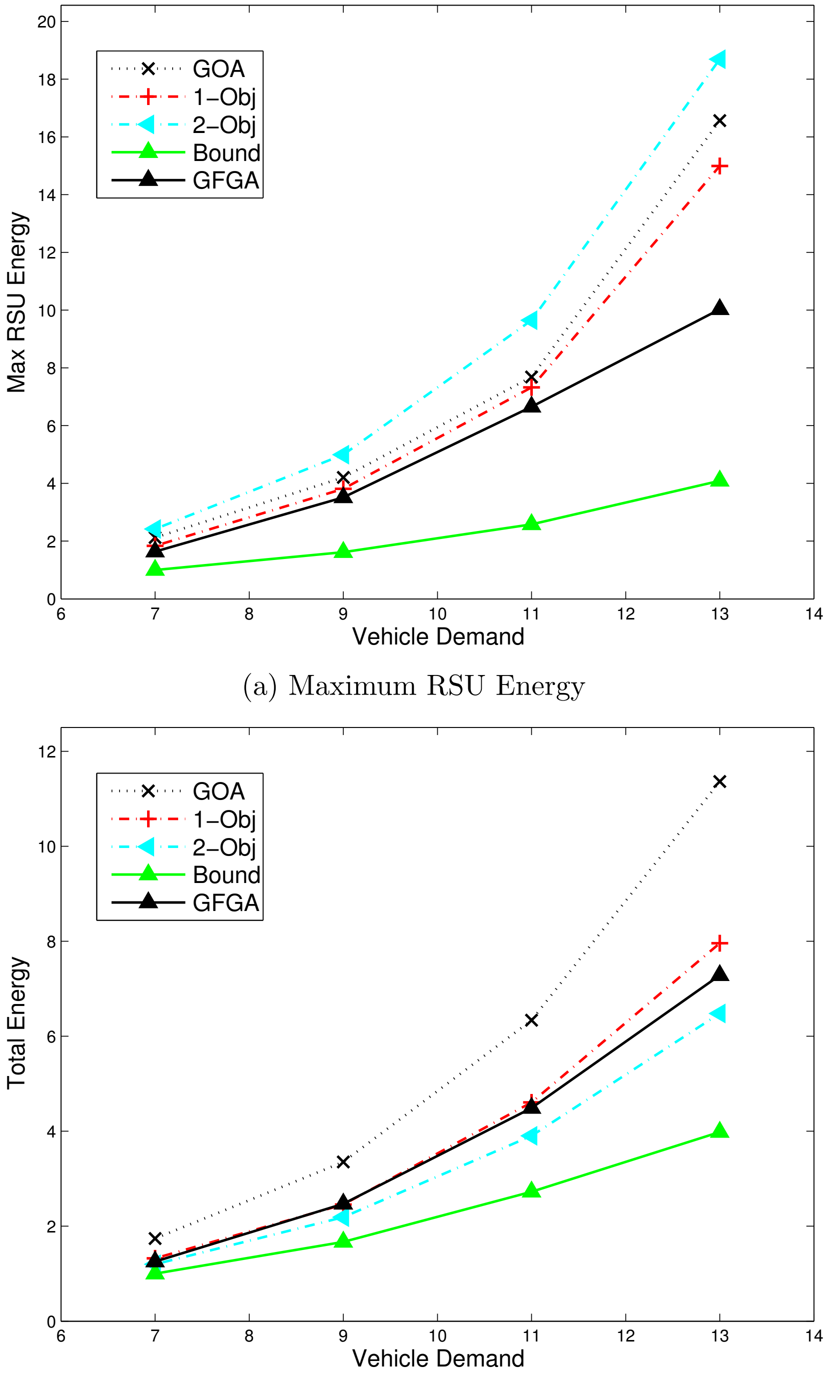
In the first set of results we consider three classes of vehicles with the same arrival rate but different speeds of 20, 25 and 30 m/s, i.e., vehicles within each class have the same speed. In these results, 15% of the demands are “bursty” meaning that those vehicle’s average demands are 20 unit demands higher than the others. To obtain a mix of traffic conditions, the total simulation time is divided into 5 cycles. In the first period, vehicle arrival rate is 8/100 vehicles/sec and the following time cycles arrival rate are 1.4, 0.4, 1.5, and 1.0 times the arrival rate of the first cycle respectively.



(a) Maximum RSU Energy

(b) Total RSU Energy

Figure 4.1: Maximum and Total RSU Energy vs. Vehicle Demand. Splittable RSU Assignment (SRA).



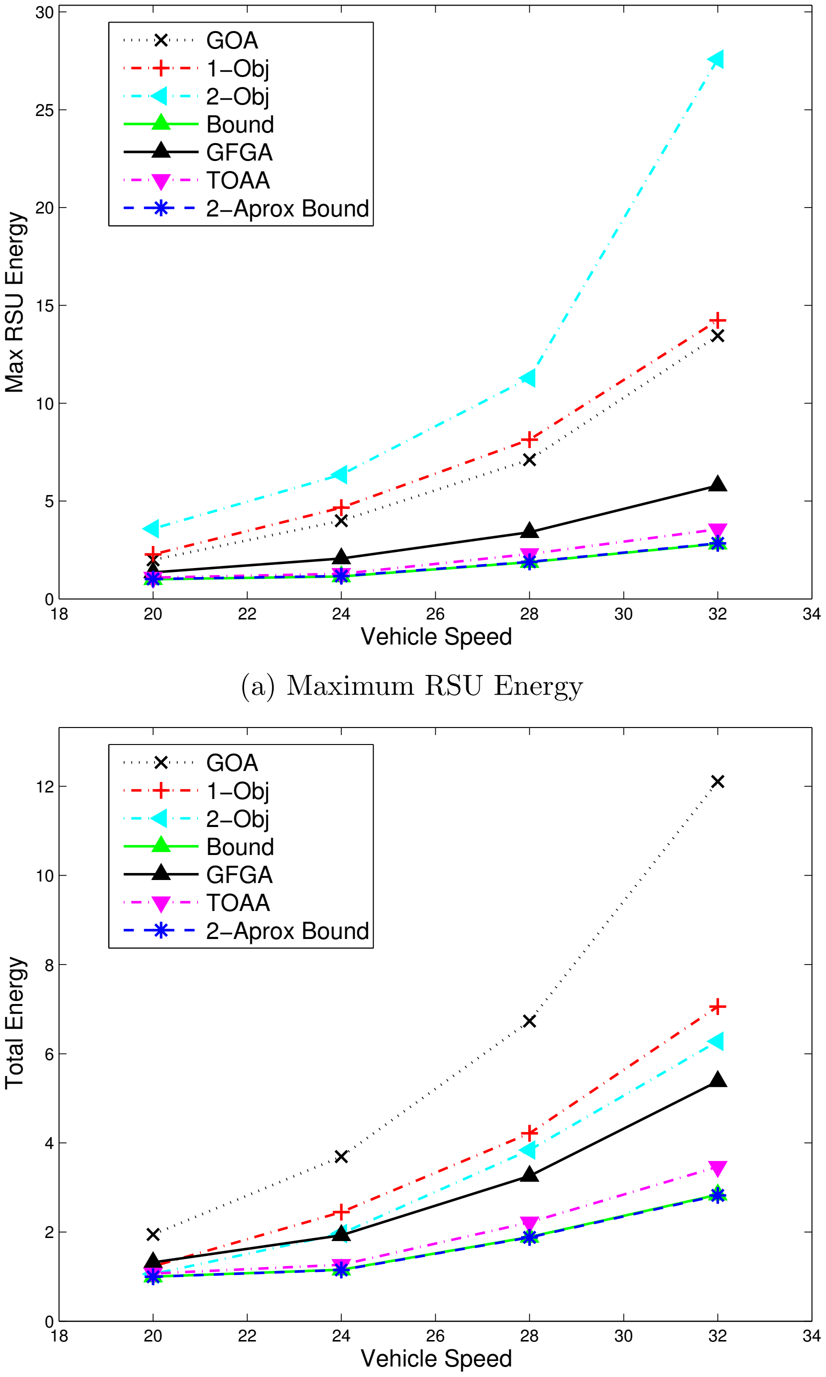
(a) Maximum RSU Energy

(b) Total RSU Energy

Figure 4.2: Maximum and Total RSU Energy Use vs. Vehicle Demand. Unspilttable RSU Assignment (URA).

Figures 4.1 and 4.2 show min-max normalized energy use and total energy used across the RSUs as a function of varying vehicular demands with the uniform distribution of $\mathcal{U}(1, 13)$ for the SRA and URA cases, respectively. From these figures, it can be seen that as vehicular demands increase, min-max normalized energy and total energy increase, as would be expected. This is due to having to serve vehicles at increasingly less favourable locations as downlink time slot contention increases. From Figure 4.1, it can be seen that the energy cost of the 2-approximation offline algorithm is close to the energy cost of the bound obtained from CPLEX for this input data. It can also be seen that its online version, TOAA, performs close to this optimum performance, which it is, in part, due to the fact that it reschedules time slot assignments whenever there is a new vehicle arrival. This increased complexity is clearly helping the min-max performance of the algorithm. In addition, GFGA performs well as it reschedules time slot assignments, but, since it makes greedy RSU selections for load-balancing, its performance is worse than TOAA. In the SRA case the GOA, 1-, and 2-Objective algorithms perform significantly worse that the bound especially when vehicle demand is high. The GOA algorithm performs the best of these three, but it operates with significantly higher total energy use than the other two. It can be concluded that when vehicle demands can be serviced across multiple RSUs (i.e., SRA case), the most important factor is to ensure that scheduling is efficient from a normalized energy viewpoint. In the SRA (versus URA) cases that we have considered, it is generally easier for a sophisticated algorithm to obtain results which are close to the theoretical minimum min-max energy. However, these algorithms require time slot re-scheduling, which is undesirable from an implementation viewpoint.

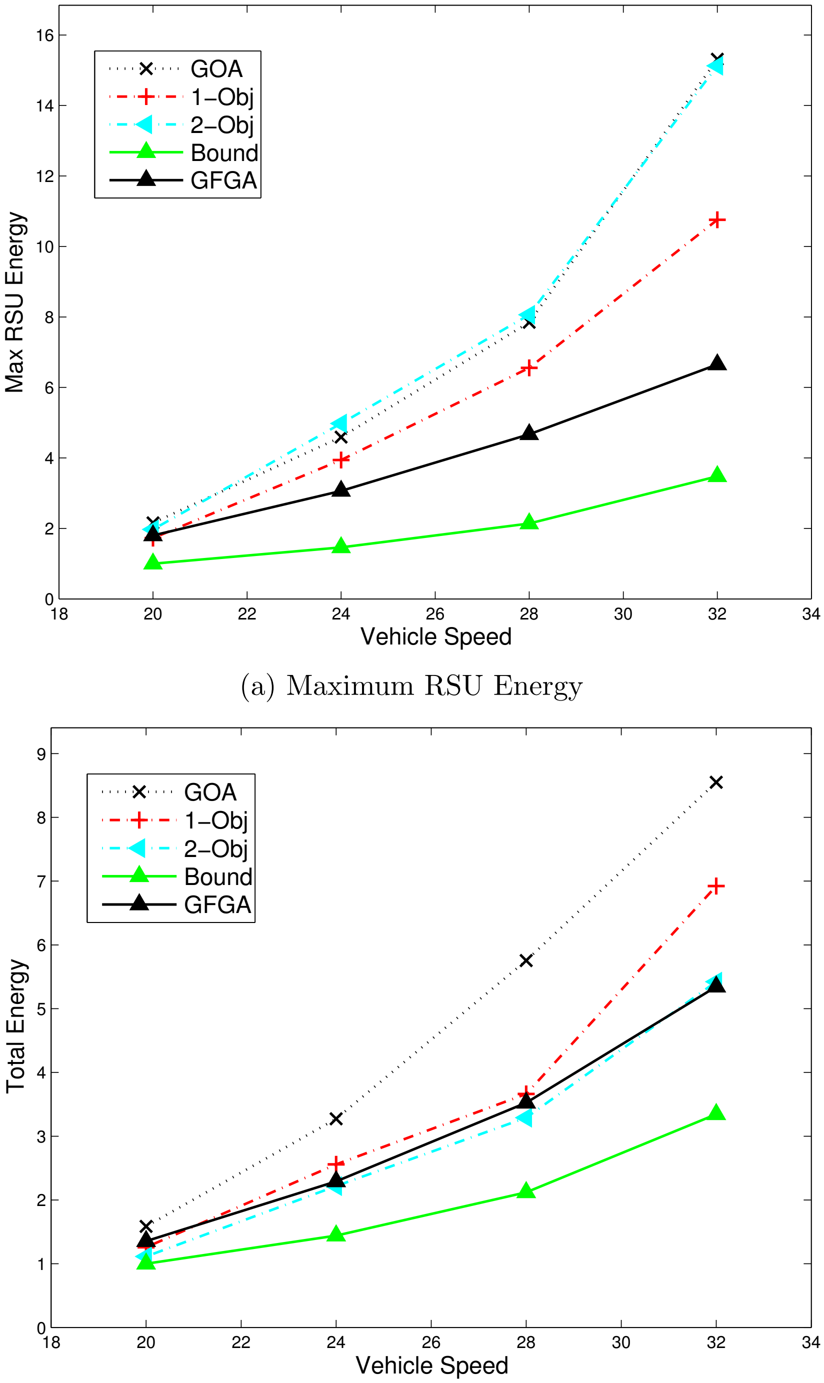
The situation is quite different when vehicular requests must be assigned to a specific RSU, i.e., the unsplittable RSU assignment (URA) case shown in Figure 4.2. This scenario is clearly more difficult from an algorithm complexity perspective, but it is probably a more desirable feature in practical implementations. When this is the case, the 2-approximation algorithm and it's online version (TOAA) are not available. In the GFGA algorithm, due to the constraint that the URA case imposes and the greedy RSU selections of the scheduler, the gap between the GFGA algorithm and the bound is much higher than in the SRA case from both normalized min-max and total energy viewpoints. It can also be seen that GOA performs significantly worse than the other algorithms in terms of total energy, and slightly worse than the 1-Objective algorithm from a min-max normalized energy viewpoint. With respect to total energy, the performance of the Bi-Objective algorithm is good in comparison with other online algorithms; however, its performance in terms of min-max normalized energy is not as good as the scheduling produced by the One-Objective algorithm. This result is significant in that the best online min-max performance that we obtained (i.e., the 1-Objective algorithm) also performs better than the GOA algorithm from a total energy viewpoint. It can be seen in Figure 4.2 that the 2-Objective algorithm trades off normalized min-max energy use for total RSU energy as one would expect.



(a) Maximum RSU Energy

(b) Total RSU Energy

Figure 4.3: Maximum and Total RSU Energy Use vs. Vehicle Speed. Splittable RSU Assignment (SRA).



(a) Maximum RSU Energy

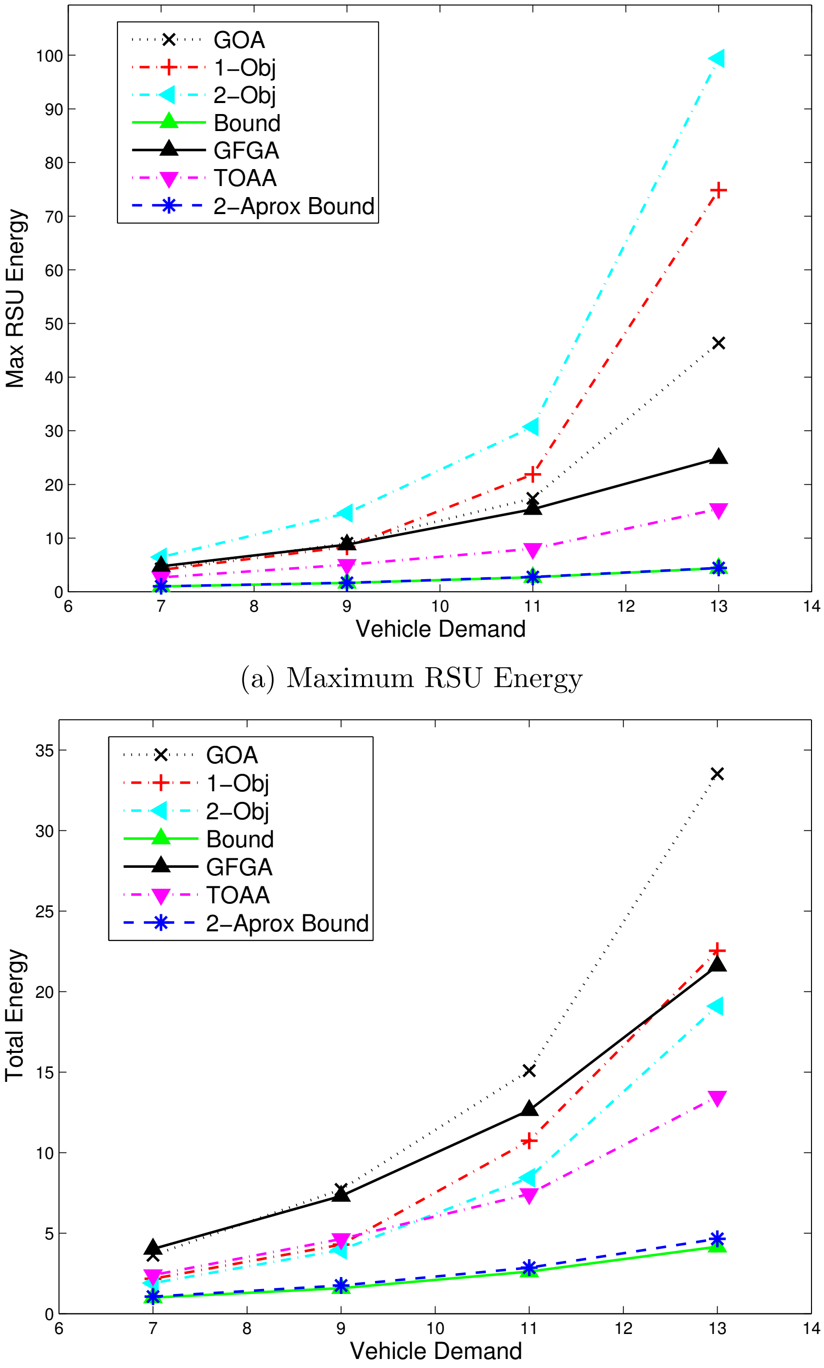
(b) Total RSU Energy

Figure 4.4: Maximum and Total RSU Energy Use vs. Vehicle Speed. Unsplittable RSU Assignment (URA).

In Figures 4.3 and 4.4 we now show results for the SRA and URA cases, respectively, where there are two classes of vehicles. The first class operates at a speed of 20 m/sec and the speed of the second class is varied over the values 20, 24, 28, and 32 m/sec, which are plotted on the x-axis. The vehicle arrival rates for the two classes in the SRA and URA cases are 11/100 and 1/10 vehicles/sec, respectively. All other parameters remain as they were in the first set of experiments. In these figures it can be seen that both the normalized min-max and total energy increases with vehicular speed, as one would expect. This is caused by the fact that as vehicle speed increases, vehicles on average, spend less time in energy favourable locations.

It can be seen from Figure 4.3 that in the SRA case, the TOAA algorithm performs the best from a min-max normalized energy viewpoint as in the previous set of results. As before, this algorithm achieves results which are very close to the ILP and 2-approximation bounds. Taking advantage of rescheduling, GFGA still performs significantly better than the other online algorithms from both min-max normalized and total energy viewpoints. In this particular case however, the GOA algorithm performs slightly better than 1-Objective. When splittable RSU assignment is permitted, it is generally difficult to predict which of these algorithms will perform better in terms of min-max normalized energy. Interestingly, although the two algorithms are very similar min-max normalized energy-wise, the GOA algorithm operates with significantly higher total energy consumption, so from that point of view the 1-Objective algorithm is making better overall decisions. It is able to achieve similar min-max energy performance but with lower overall total energy expenditure.

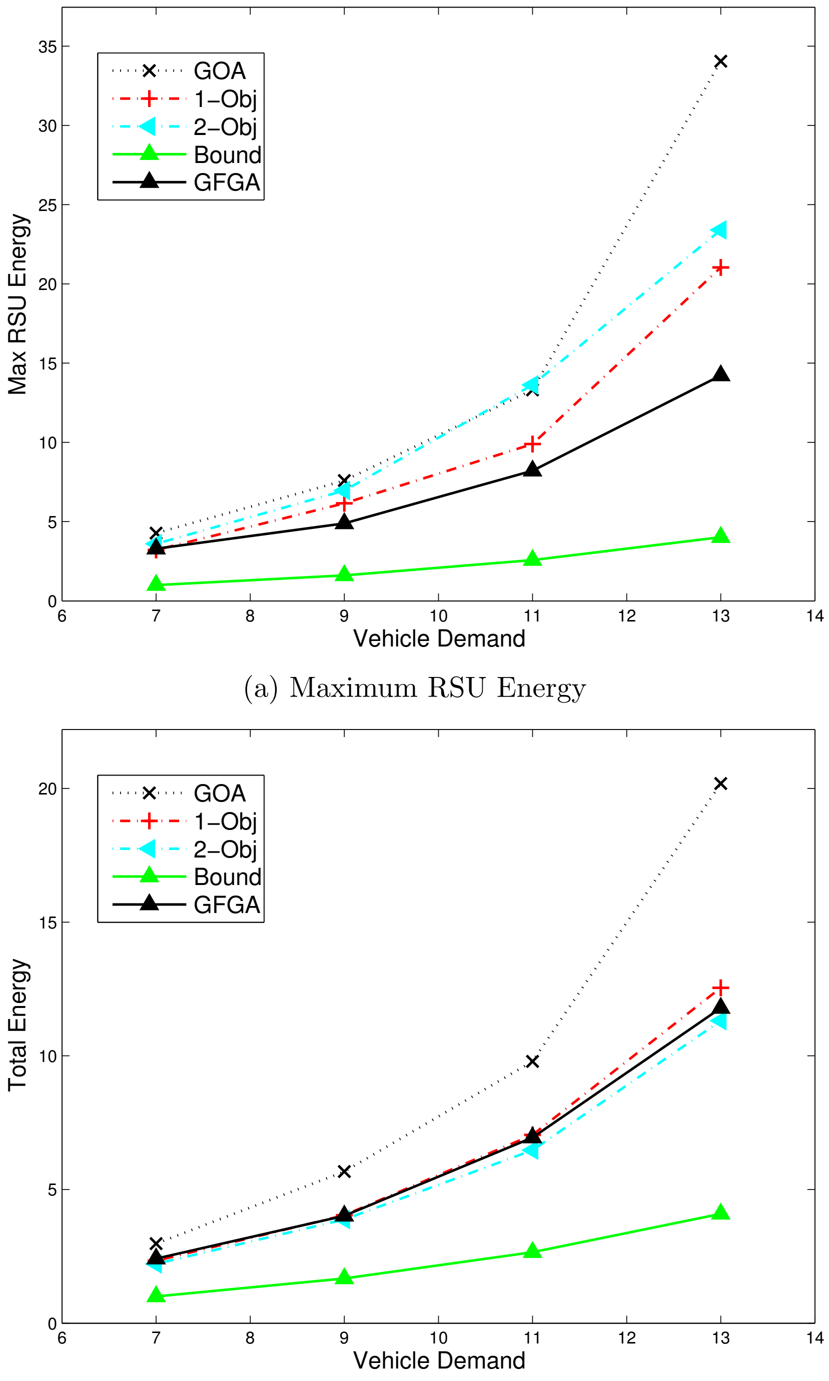
In the URA case, Figure 4.4, the results are similar to that obtained in the first set of results. The 1-Objective algorithm achieves the best min-max normalized energy and it does not do worse than the GOA algorithm in terms of total energy. In general we find that, as one would expect, an inefficient schedule in the URA case incurs larger energy penalties than in the SRA case. We also find that that the gap between the best online algorithms and the offline bound is much larger for the URA results. Although we have no way of knowing how tight the ILP bounds are, this large gap gives us some hint of the complexity of the scheduling problem under the URA assumption.



(a) Maximum RSU Energy

(b) Total RSU Energy

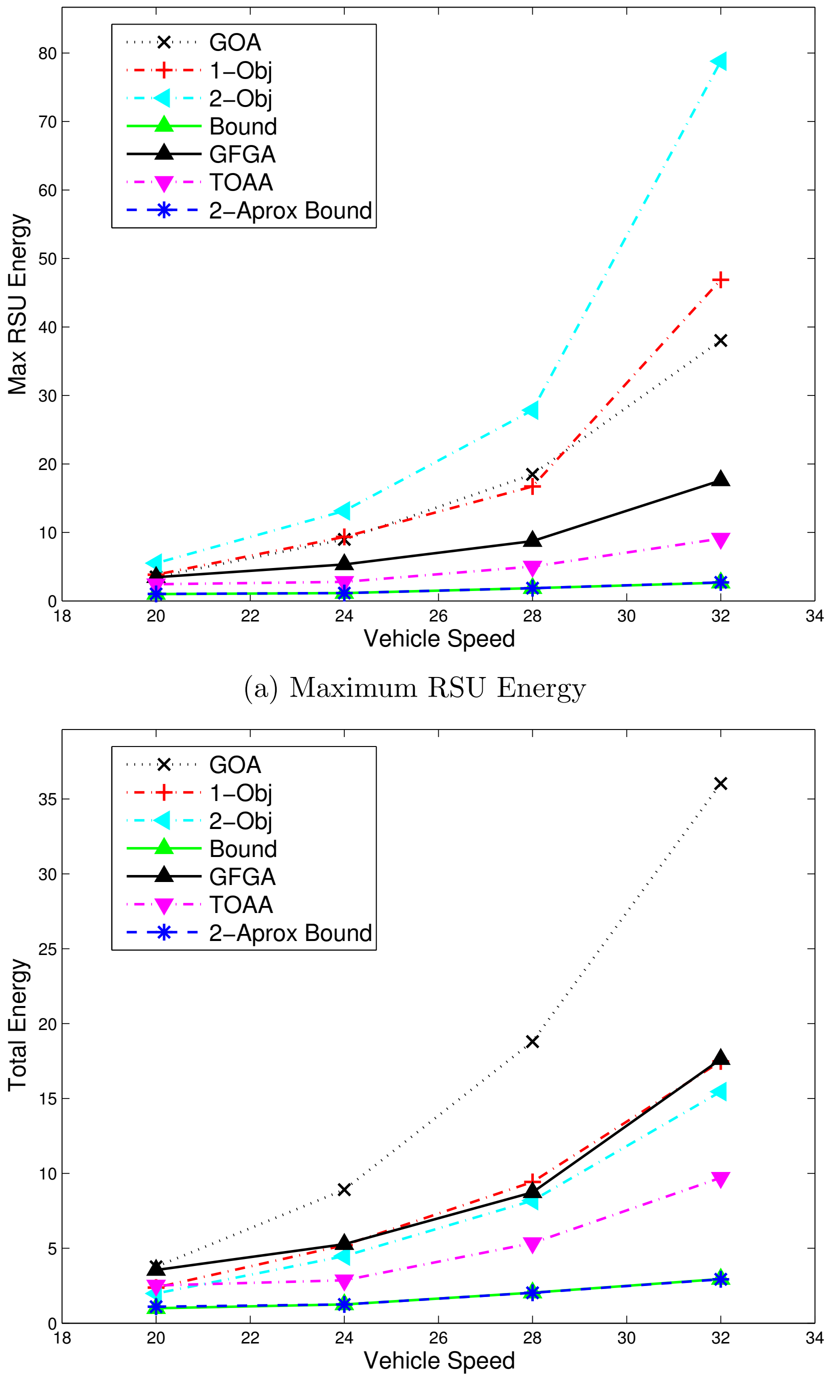
Figure 4.5: Maximum and Total RSU Energy vs. Vehicle Demand. Splittable RSU Assignment (SRA). Shadowing.



(a) Maximum RSU Energy

(b) Total RSU Energy

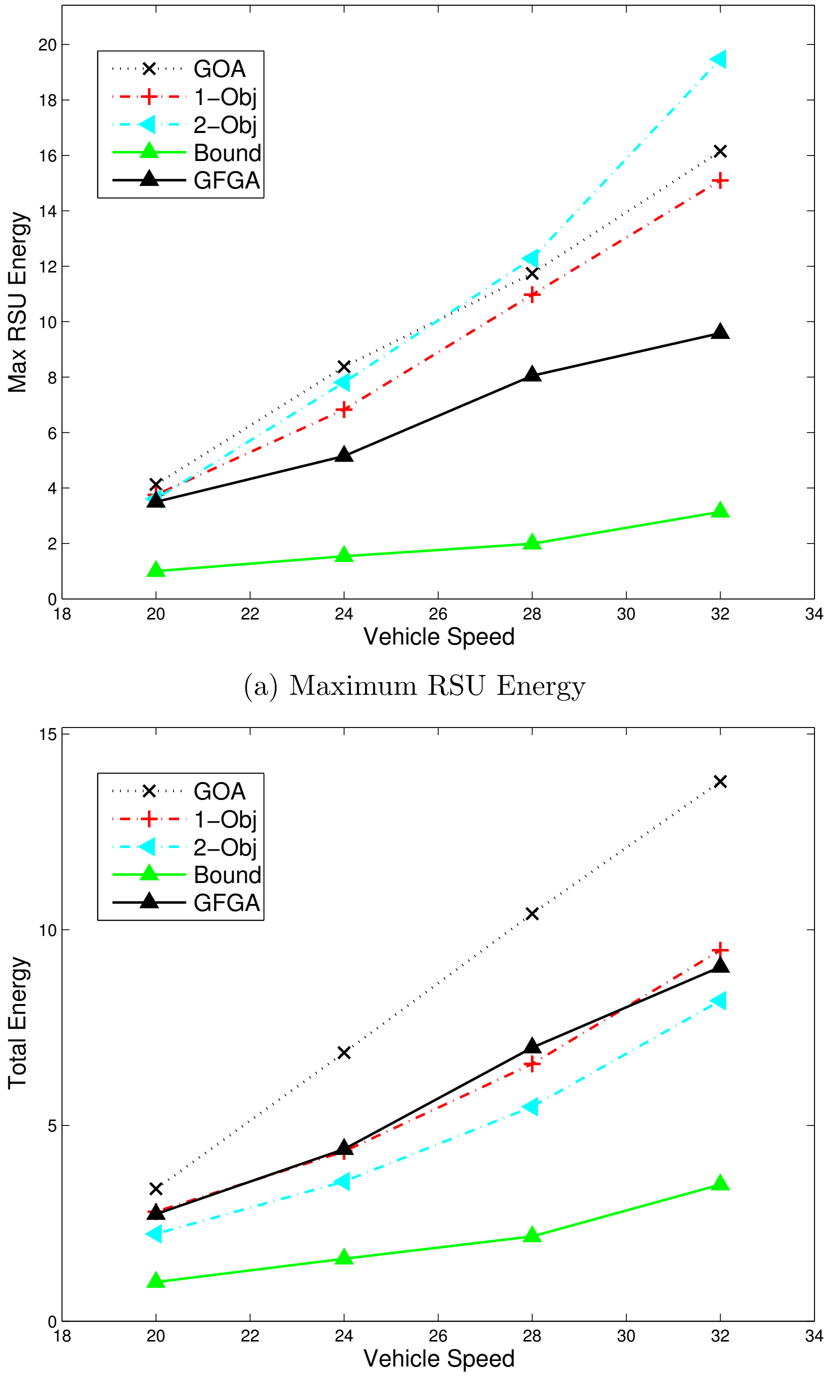
Figure 4.6: Maximum and Total RSU Energy Use vs. Vehicle Demand. Unsplittable RSU Assignment (URA). Shadowing.



(a) Maximum RSU Energy

(b) Total RSU Energy

Figure 4.7: Maximum and Total RSU Energy Use vs. Vehicle Speed. Splittable RSU Assignment (SRA). Shadowing.



(a) Maximum RSU Energy

(b) Total RSU Energy

Figure 4.8: Maximum and Total RSU Energy Use vs. Vehicle Speed. Unsplittable RSU Assignment (URA). Shadowing.

In the next set of graphs, the same assumptions hold as in the results presented thus far, except that the energy cost data that is fed to the schedulers includes random components. The reason that this is done is to ensure that when scheduler input data is not ideal, that the algorithms do not produce results which would be biased in some way, due to this randomness. In the results presented, this was done by adding propagation shadowing effects to the extracted data, which result in unpredictable randomness in these estimates. The scheduling is therefore based on this input, but the actual costs incurred include the energy perturbations due to the random components. A conventional log-normal shadowing model is used with zero mean and standard deviation of $\sigma = 4$ dB. When uncertainty increases, as one would expect, the energy cost of the algorithms increase and their schedules are worse than the case we have an exact prediction of the energy cost. Figures 4.5, 4.6, 4.7 and 4.8 are the experiments corresponding to Figures 4.1, 4.2, 4.3 and 4.4 from before.

Comparing these sets of graphs with the previous ones clearly show that GOA, GFGA, and TOAA total RSU energy usage can often be about twice that for the previous cases, and over 20% higher for the Bi-Objective and One-Objective algorithms (Figures 4.5, 4.6). The comparisons are similar for the max RSU energy results. As one would expect, the introduction of random components into the input data results in poorer scheduling decisions, as was noted in (Hammad *et al.*, 2013). However, other than creating a greater split between the performance of the algorithms, it can be seen that the relative algorithm performance is the same as it was previously.

Figure 4.6 shows that the same argument is true for the URA case. The gap between the GOA algorithm and both the One-Objective and Bi-Objective algorithms increases as uncertainty is added to the system, which shows that these two algorithms are low-penalty schedulers.

Figures 4.5, 4.6, 4.7 and 4.8 also show that the performance of the online algorithms compared with the Bound are much less favourable when there are random components in the energy cost data inputs to the schedulers. This is to be expected since the scheduling decisions are based on less accurate information. In Figure 4.7 for example, TOAA is about 400% higher than the bound at the highest speed value, compared with almost exactly matching the bound from Figure 4.3. Overall, in the URA case, the better behavior of both the One-Objective and Bi-Objective algorithms over GOA can clearly be seen. It can also be concluded that when there is uncertainty in the energy cost predictions, the GOA algorithm energy cost growth is more than other algorithms.

Chapter 5

Conclusions

In this thesis we have considered the issue of low complexity energy efficient scheduling in vehicular networks when there are multiple roadside units (RSUs) in tandem. The objective is to minimize energy use as much as possible while load balancing the normalized energy costs across the roadside units. The thesis considered both the splittable (SRA) and unsplittable (URA) RSU assignment cases. In SRA, downlink communication requests can be split across multiple RSUs, whereas, in URA, a single RSU must satisfy the request. An offline integer linear program was derived which provides a lower bound for the normalized min-max RSU energy usage. For the splittable RSU assignment case, a polynomial complexity 2-approximation bound for the normalized min-max energy schedule was introduced. Low complexity online scheduling algorithms were then introduced. The Greedy Online Algorithm makes greedy RSU selections followed by minimum energy time slot assignments. The second, the Greedy Flow Graph Algorithm (GFGA), makes the same RSU selection but reassigns time slots whenever a new vehicle is assigned to the same RSU, using a minimum cost flow graph scheduler. An algorithm was then proposed (TOAA) which is an online version of the 2-approximation bound for the SRA scheduling case. Two low complexity algorithms were then introduced using a potential function scheduling approach. The One-Objective algorithm, uses an objective based on normalized min-max energy. This algorithm was shown to have provable performance guarantees, i.e., its worse-case competitive ratio compared with the lower bound is upper bounded. The second, the Bi-Objective algorithm, uses the same objective, but combined with a total energy secondary objective.

Results from a variety of experiments show that the proposed scheduling algorithms perform well. In particular, we find that in the SRA case, the TOAA algorithm performs very close to the lower bound, but at the expense of having to reassign time slots whenever a new vehicle arrives, which is undesirable in many practical scenarios. In the URA case, our low complexity One-Objective algorithm performs better than the others over a wide range of traffic conditions. Since the Bi-Objective algorithm also tries to minimize the total energy usage, its performance in min-max normalized energy is not as good, but it is still better than GOA in the URA case.

In this thesis, energy load-balancing in a tandem of RSUs has been considered from a normalized energy point of view. However, it was assumed that the vehicle requirements are delay tolerant which may not always be the case. In some cases, such as in safety applications, time constraints play a significant role, and it is necessary to consider real-time deadlines. In some applications a level of QoS is required and if it is not guaranteed, the provided service may not be feasible. While power consumption was the focus of this thesis, other performance metrics, such as QoS, service ratio and throughput, can be used to evaluate the scheduling algorithms. These could be considered in future work.

Bibliography

Ahuja, R., Magnanti, T., and Orlin, J. (1994). Network Flows: Theory, Algorithms, and Applications. *Journal of the Operational Research Society*, **45**(11), 1340–1340.

Alcaraz, J., Vales-Alonso, J., García-Haro, J., *et al.* (2009). Control-Based Scheduling with QoS Support for Vehicle to Infrastructure Communications. *IEEE Communications.*

Bilstrup, K., Uhlemann, E., Strom, E., and Bilstrup, U. (2008). Evaluation of the IEEE 802.11p MAC Method for Vehicle-to-Vehicle Communication. In *IEEE 68th Vehicular Technology Conference, 2008. VTC 2008-Fall.*

Bychkovsky, V., Hull, B., Miu, A., Balakrishnan, H., and Madden, S. (2006). A Measurement Study of Vehicular Internet Access Using In Situ Wi-Fi Networks. In *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*. ACM.

C. Sommer, D. Eckhoff, R. G. and Dressler, F. (2011). A Computationally Inexpensive Empirical Model of IEEE 802.11p Radio Shadowing in Urban Environments. In *Eighth International Conference on Wireless On-Demand Network Systems and Services*, pages 84–90.

Chang, J.-H. and Tassiulas, L. (2000). Energy Conserving Routing in Wireless Ad-Hoc Networks. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 22–31 vol.1.

Chen, W., Guha, R., Kwon, T., Lee, J., and Hsu, I. (2008). A Survey and Challenges in Routing and Data Dissemination in Vehicular Ad-Hoc Networks. In *Vehicular Electronics and Safety, 2008. ICVES 2008. IEEE International Conference on*, pages 328–333.

Feng, W., Alshaer, H., and Elmirghani, J. M. H. (2010). Green Information and Communication Technology: Energy Efficiency in a Motorway Model. *Communications, IET*, **4**(7), 850–860.

Festag, A., Baldessari, R., and Wang, H. (2007). On Power-Aware Greedy Forwarding in Highway Scenarios. In *5th International Workshop on Intelligent Transportation (WIT), Hamburg, Germany*.

Gong, J., Zhong, X., and Xu, C.-Z. (2010). Maximizing Rewards in Wireless Networks with Energy and Timing Constraints for Periodic Data Streams. *Mobile Computing, IEEE Transactions on*, **9**(8), 1187–1200.

Gui, Y. and Chan, E. (2011). Data Scheduling for Vehicular Ad-hoc Networks Augmented with Road Side Units. *IJIEE*, **2**(1), 88–95.

Hadaller, D., Keshav, S., Brecht, T., and Agarwal, S. (2007). Vehicular Opportunistic Communication under the Microscope. In *Proceedings of the 5th International*

Conference on Mobile Systems, Applications and Services, MobiSys '07, pages 206–219, New York, NY, USA. ACM.

Hammad, A., Todd, T. D., Karakostas, G., and Zhao, D. (2013). Downlink Traffic Scheduling in Green Vehicular Roadside Infrastructure. *IEEE Transactions on Vehicular Technology*, **62**, 1289–1302.

Hammad, A. A., Badawy, G. H., Todd, T. D., Sayegh, A. A., and Zhao, D. (2010). Traffic Scheduling For Energy Sustainable Vehicular Infrastructure. In *IEEE Global Communications Conference (IEEE GLOBECOM'2010), Miami, Fla.*

Hartenstein, H. and Laberteaux, K. (2008). A Tutorial Survey on Vehicular Ad Hoc Networks. *Communications Magazine, IEEE*, **46**(6), 164–171.

J. Aspnes and, Y. A., Fiat, A., Plotkin, S., and Waarts, O. (1997). On-line Routing of Virtual Circuits with Applications to Load Balancing and Machine Scheduling. *Journal of the ACM*, **44(3)**, 486–504.

Jhang, M. F. and Liao, W. (2008). On Cooperative and Opportunistic Channel Access for Vehicle to Roadside (V2R) Communications. *GLOBECOM.*

Khabazian, M. and Ali, M. (2008). A Performance Modeling of Connectivity in Vehicular Ad Hoc Networks. *Vehicular Technology, IEEE Transactions on*, **57**(4), 2440–2450.

Khaled, Y., Tsukada, M., Santa Lozano, J., and Ernst, T. (2009). On The Design of Efficient Vehicular Applications. *Proc. of IEEE VTC09, Barcelona, Spain, April.*

Kumar, R. and Dave, M. (2011). A Comparative Study of Various Routing Protocols in VANET. *Computing Research Repository*, **abs/1108.2094**.

Kumar, V. and Chand, N. (2010a). Data Scheduling in VANETs : A Review. *IJCSC*, **1**(2), 399–403.

Kumar, V. and Chand, N. (2010b). Efficient Data Scheduling in VANETs. *Journal of Computing*, **2**, 33–37.

Li, F. and Wang, Y. (2007). Routing in Vehicular Ad Hoc Networks: A Survey. *IEEE Vehicular Technology Magazine*, **2**(2), 12–22.

Li, N., Wang, H., and ren Zheng, S. (2003). An Energy-Saving Scheme For Wireless LANs. In *Communication Technology Proceedings, 2003. ICCT 2003. International Conference on*, volume 2, pages 1242–1245 vol.2.

Mamechaoui, S., Didi, F., and Pujolle, G. (2013). A Survey on Energy Efficiency for Wireless Mesh Network. *Computing Research Repository*, **abs/1304.3904**.

Mittag, J., Schmidt-Eisenlohr, F., Killat, M., Harri, J., and Hartenstein, H. (2008). Analysis and Design of Effective and Low-Overhead Transmission Power Control for VANETs. In *Proceedings of the Fifth ACM International Workshop on Vehicular Inter-Networking*. ACM.

Nandan, A., Das, S., Pau, G., Gerla, M., and Sanadidi, M. (2005). Co-operative Downloading in Vehicular Ad-Hoc Wireless Networks. *WONS*.

Ning, L., Yuan, X., and Sheng-li, X. (2005). A Power-saving Protocol for Ad Hoc Networks. In *Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on*, volume 2, pages 808–811.

Ott, J. and Kutscher, D. (2004). Drive-thru Internet: IEEE 802.11b for Automobile Users. *INFOCOM 2004*, **1**.

Park, S.-J. and Sivakumar, R. (2002). Quantitative Analysis of Transmission Power Control in Wireless Ad-Hoc Networks. In *Parallel Processing Workshops, 2002. Proceedings. International Conference on*, pages 56–63.

Peirce, S. and Mauri, R. (2007). Vehicle-Infrastructure Integration (VII) Initiative: Benefit-Cost Analysis: Pre-Testing Estimates. *Intelligent Transportation Systems Joint Program Office, United States Department of Transportation, Washington, DC.*

Rawat, D., Yan, G., Popescu, D., Weigle, M., and Olariu, S. (2009). Dynamic Adaptation of Joint Transmission Power and Contention Window in VANET. *VTC*.

Shahverdy, M., Fathy, M., and Yousefi, S. (2010). Scheduling Algorithm for Vehicle to Road-Side Data Distribution. In Y. Wu and Q. Luo, editors, *High Performance Networking, Computing, Communication Systems, and Mathematical Foundations*, volume 66 of *Communications in Computer and Information Science*, pages 22–30. Springer Berlin Heidelberg.

Shmoys, D. and Tardos, E. (1993). An Approximation Algorithm for the Generalized Assignment Problem. *Mathematical Programming: Series A*, **62(3)**, 461–474.

Toor, Y., Muhlethaler, P., and Laouiti, A. (2008). Vehicle Ad Hoc Networks: Applications and Related Technical Issues. *Communications Surveys Tutorials, IEEE*, **10**(3), 74–88.

Toutouh, J. and Alba, E. (2011). An Efficient Routing Protocol for Green Communications in Vehicular Ad-Hoc Networks. In *Proceedings of the 13th Annual*

Conference Companion on Genetic and Evolutionary Computation, GECCO '11, pages 719–726, New York, NY, USA. ACM.

Wang, S. (2005). The Effects of Wireless Transmission Range on Path Lifetime in Vehicle-Formed Mobile Ad Hoc Networks on Highways. In *IEEE International Conference on Communications, 2005. ICC'2005.*, volume 5, pages 3177–3181. IEEE.

Xu, J., Tang, X., and chien Lee, W. (2006). Time-Critical On-Demand Data Broadcast: Algorithms, Analysis, and Performance Evaluation. *IEEE Transactions on Parallel and Distributed Systems*, **17**, 2006.

Yousefi, S., Mousavi, M., and Fathy, M. (2006). Vehicular Ad Hoc Networks (VANETs): Challenges and Perspectives. In *ITS Telecommunications Proceedings, 2006 6th International Conference on*, pages 761–766.

Zhang, H., Ma, Y., Yuan, D., and Chen, H.-H. (2011). Quality-of-Service Driven Power and Sub-Carrier Allocation Policy for Vehicular Communication Networks. *Selected Areas in Communications, IEEE Journal on*, **29**(1), 197–206.

Zhang, L., Wu, Q., Solanas, A., and Domingo-Ferrer, J. (2009). A Scalable Robust Authentication Protocol for Secure Vehicular Communications. *IEEE Transactions on Vehicular Technology*.

Zhang, Y., Zhao, J., and Cao, G. (2010). Service Scheduling of Vehicle-Roadside Data Access. *Mobile Networks and Applications*, **15**(1), 83–96.

Zhao, J., Arnold, T., Zhang, Y., and Cao, G. (2008). Extending Drive-Thru Data Access by Vehicle-to-Vehicle Relay. *VANET '08: Proceedings of the Fifth ACM International Workshop on Vehicular Inter-Networking.*

Zhou, Y., Tan, Y.-Y., Laurenson, D., and McLaughlin, S. (2005). Impact of Power Saving MAC Scheme on Ad Hoc Network Routing Protocol. In *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, volume 4, pages 2463–2467 Vol. 4.

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.