EFFICIENT DATA DISSEMINATION FOR INTELLIGENT CONNECTED VEHICLES

ELMER RAMILO MAGSINO

PhD

The Hong Kong Polytechnic University

2021

The Hong Kong Polytechnic University

Department of Electronic & Information Engineering

Efficient Data Dissemination for Intelligent Connected Vehicles

Elmer Ramilo Magsino

A thesis submitted in partial fulfillment of the requirements

for the degree of Doctor of Philosophy

August 2020

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

______________________________ (Signed)

Elmer Ramilo Magsino (Name of Students)

Abstract

As the number of conventional, intelligent, and connected vehicles plying the roads and streets of major cities around the world increases, the need to provide comfortable, time-efficient, safe, environment- and energy-friendly travel becomes apparent. Daily trips of commuters, private, intelligent, and connected vehicles begin by anticipating traffic jams, accidents, road works, and other event-delaying circumstances to ensure that arriving at their destination is on time and convenient. To address these concerns and ensure the comfort of the daily voyage, intelligent and connected vehicles must obtain and share real-time, relevant, and accurate road information from/to other vehicles or roadside unit (RSU) infrastructure. However, as more vehicles use the road, the quantity of data sources increases, as well as the amount of available road information. In general, there are two categories of vehicular data: (1) control data and (2) environment data. Control data pertain to the safety applications for vehicles to abide by for achieving an organized traffic flow, such as the highway tollgate scenario. In contrast, environment data relate to non-safety and broad reports or applications for further processing, e.g., multi-junction city landscape. This thesis investigates the data dissemination topic for intelligent connected vehicles in vehicular networks to efficiently and effectively exchange control and environment data among vehicles and infrastructure in order to achieve travel objectives. Some of these travel objectives are: minimized queue time and length, optimal road map download based on vehicular demand under

heterogeneous transmissions, maximized information sharing between vehicles and infrastructure, balanced roadside unit loading, and energy efficiency.

We first examine the control data exchange in a tollgate scenario on highways where intelligent connected vehicles enter and exit the freeway for faster travel at the expense of paying toll fees for this convenience. Choosing and queueing at a certain tollgate becomes problematic for these intelligent connected vehicles since their current highway traffic and available tollgate service time knowledge are limited. Single-stage queuing system models the highway tollgate section with exponential service time, c servers, and K capacity, $(M/M/c/K)$ to address the general problem of minimizing average queue time and length for all these vehicles entering/exiting the tollgate section. Also, independent and identical Poisson distributions characterize vehicular arrivals on highway lanes. A centralized fuzzy logic controller (FLC) is developed by considering the system's queue densities (number of vehicles per lane) and tollgate service times. The FLC facilitates the control data exchange where intelligent connected vehicles should line up. Extensive simulation results show approximately 50% improvement in reducing average queue waiting time and length when the central RSU employing an FLC is in-charge of allocating incoming vehicles to its appropriate servicing tollgate. The FLC controller has considered both the homogeneous and non-homogeneous arrivals of intelligent connected vehicles on the highway. As an added feature, the FLC allows early detection of forming high-density queues such that the service time of tollgate servers can be adjusted accordingly.

We then investigate the general environment data dissemination problem considering the city-wide case by employing empirical mobility traces and real road environment LIght Detection And Ranging (LIDAR) data to represent intelligent vehicular mobility and sensed data, respectively. The framework follows the fog computing paradigm, where computing nodes are close to the vehicles, to immediately process received road information, then satisfy all vehicular demands. To initially reduce the file size of the sensed road segment data to be uploaded by the vehicles, octree compression, differential coding, and hashing techniques are implemented at the vehicular level. During the data dissemination stage, we propose an opportunistic index coding-based transmission scheme to optimally reduce the number of transmissions, transmitted data size, and overhead computations according to the vehicular data demand and availability. This setup considers heterogeneous modes of information download from an RSU fog node or base station to the vehicles. The objective of the proposed index coding-based transmission scheme applied at each of the RSU fog nodes is to reduce the reliance of intelligent and connected vehicles from long-range cellular transmissions and better exploit the short-range broadcast capacity at RSU fog nodes. Experiments involving mobile robots as intelligent connected vehicles have been tested to provide feasibility results for the implementation of this technique in the real world. To capture the city-wide data exchange using the proposed index coding transmission scheme, we employed empirical taxi mobility traces and evaluated the scheme's performance at each target junction. Our extensive simulation and mobile robot testbed results show a promising application of efficient data dissemination in an urban scenario utilizing the fog computing paradigm.

Noting that the installation and maintenance of RSUs along public roads and highways can be quite expensive, RSUs must be strategically and economically deployed to support various vehicular fog applications. In the next two studies, we propose two strategies for maximizing information shared in a vehicular network. First, we develop an Information Sharing via Roadside Unit Allocation (ISRA) strategy to maximize information sharing between intelligent connected vehicles and RSU fog nodes. ISRA operates under the constraint that the number of RSU fog nodes is limited and only considers intersections as locations for possible RSU deployment. ISRA targets energy-efficient candidate locations during information exchange while balancing the load among the selected RSU fog nodes for better resource management. Given a set of candidate intersection locations, ISRA discriminates optimal locations from the set of candidate locations by implementing the proposed index coding-based transmission schemes and considering the vicinity's space mean speed and vehicular density. Simulations utilizing empirical mobility traces show that ISRA, on average, shares 20% more information at the energy efficiency of 83%, i.e., fewer packet transmissions, when compared to other deployment schemes.

The second method, Enhanced Information SHAring via RSU Allocation (EISHA-RSU), also attempts to maximize shared information in a vehicular network but considers all locations as possible deployment spots. EISHA-RSU utilizes the Effective Region of Movement (ERM) concept to irregularly partition an urban setup based on the region's vehicular capacity and then discovers effective positions (EPs) within ERMs to deploy RSUs. We compare the performance of EISHA-RSU with three other benchmarks focusing on the amount of shared information, L-M measure, network starvation, vehicle count of 1-hop connectivity, and effectiveness. Extensive simulation employing three empirical urban mobility traces confirmed the efficiency of the proposed information-sharing scheme.

In summary, this thesis investigated the data dissemination in vehicular networks for providing travel comfort and convenience to intelligent connected vehicles. This work has proposed adaptive and efficient data dissemination techniques to address the information exchange challenges in the data- and the source-rich vehicular environment. These techniques present methods of data exchange that are real-time, updated, and easily accessible to both intelligent connected vehicles and roadside infrastructure. It is desired that the thesis' results will provide practical solutions and approaches to the ever-growing and information-rich vehicular environment, particularly in promoting autonomous, connected driving.

List of Publications

Journal papers:

1. I. W. Ho, S. C. Chau, E. R. Magsino and K. Jia, ”Efficient 3D Road Map Data Exchange for Intelligent Vehicles in Vehicular Fog Networks,” in *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3151-3165, March 2020.

2. E. R. Magsino and I. W. Ho, “An Enhanced Information Sharing Roadside Unit Allocation Scheme for Vehicular Networks,” submitted to *IEEE Transactions on Intelligent Transportation Systems.*

Conference paper:

1. E. R. Magsino and I. W.H. Ho, “An intelligent highway tollgate queue selector for improving server utilization and vehicle waiting time,” in *2016 IEEE Region 10 Symposium (TENSYMP).* IEEE, 2016, pp. 271–276.

2. K.F. Chu, E.R. Magsino, I.W.H. Ho, and C.K. Chau, ”Index coding of point cloud-based road map data for autonomous driving,” in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)* IEEE, 2017, pp. 1-7.

3. E. R. Magsino and I. W.H. Ho, ”Roadside Unit Allocation for Fog-based Information Sharing in Vehicular Networks,” in *Proceedings of the 1st ACM International Workshop on Smart Cities and Fog Computing*, ACM, 2018, pp. 7-12.

Acknowledgments

This Ph.D. journey cannot be completed without the help and guidance of people who were with me along the way, who kept pushing and believing that I can accomplish this feat. These past five years dedicated to Ph.D. have been full of ups and downs.

To the Almighty One, Lord Jesus Christ, thank you for delivering me from this study. Indeed, it was difficult with lots of tears, sweat, and solemn prayers. Mother Mary, thank you for your intercession. May this work serve as a living testament that trust and faith in You provide results according to Your will and time.

I am giving my sincerest and deepest gratitude to my supervisor, Dr. Ivan Wang-Hei Ho, for taking a risk in placing me under his guidance and supervision. His intellect and dedication to his field are of no question. As knowledgeable and capable as he is, Ivan is not only a role model of a supervisor, but more than that became a very good friend. He knows the difficulties I have endured away from home, and his understanding will always be very much appreciated. May you continue to guide young and research-minded students with your passion, drive, and intellect.

I would also like to acknowledge Dr. Sid Chi-Kin Chau for all the help he has extended to me. Thank you for the small chit chats. Thank you to my colleagues and teammates: Kanghao Jia, Kai Fung Chu, Zhenhui Situ, Lingfu Xie, Yu Xie, Yuhao Wang, Tanuja Shanmukhappa, Xinyu Wang, and Josyl

Rocamora. Thank you for all the stories of research and life.

Thank you to the following friends whom I can talk anything but research: Gerald Arada, Catherine Manuela Lee Ramos, Zack Morris, Anjana Balachandran, Alvin Mesa, John Michael L. Macaso, and Val de Grano. Life must have been miserable for me in Hong Kong if you guys were not there, and if not for the spontaneous food trips, hiking, and just hanging around each other.

Thank you to our parents, Millet and Ely and Ayette and Danilo. Nanay and Tatay, I finally did it! Thank you for bringing me into this world and becoming my models for aspiring to be an engineer, while trying to be the best person I can be. Mama, thank you for assisting and checking on me during my first year in Hong Kong. Papa, thank you for looking after my wife and son while I was away. I love you all so much!

Lastly, thank you to my wife, Jessica, and my children, Eugene James (EJ) and Jewel Ellyse (Eisha), for being my inspirations to finish this journey. Jessica and EJ, thank you for allowing me to grow and pursue this endeavor. Three years abroad have been tough for us to be separated from each other, and in the process, I have lost precious moments of being with you during your most important years, particularly with EJ. To Eisha, my apologies for the mishaps when I was in charge of taking care of you. Know that I am always grateful for having you even when you came in the middle of my Ph.D. study. May this accomplishment close the chapter of being literally and figuratively apart from you guys, so that I may now devote my undivided attention to you three. I love you very much, and you are the reasons why I continue to exist. This achievement is dedicated to you, Jessica, EJ, and Eisha. :)

Table of contents

List of figures

List of tables

List of abbreviations

3D-MADS 3D Map dissemination system

ABSM Acknowledged broadcast from static to highly mobile

AP Access point

ARQ Automatic repeat request

BTU Basic traffic unit

C-V2X Cellular V2X

DP Data pouring

DP-IB Data pouring and buffering

DSRC Dedicated short-range communications

DTN Dela tolerant network

EISHA-RSU Enhanced information sharing via roadside unit allocation

EP Effective position

ERM Effective region of movement

ETC Electronic toll collection

FIFO First in first out

FLC Fuzzy logic controller

FLCQ Fuzzy logic-controlled queue

GA genetic algorithm

GPS Global positioning system

IMU Inertial measurement unit

IoT Internet of things

IoV Internet of Vehicles

ISRA Information sharing via roadside unit allocation

LCD Local controllers and databases

LIDAR Light detection and ranging

LTE Long Term Evolution

MDR Map data repository

MF Membership function

PID Proportional-Integral-Derivative

RMS Root mean square

RSU Road side unit

SDN Sotware-defined networking

SQ Shortest queue

SSU Stationary supporting units

V2I Vehicle-to-Infrastructure

V2V Vehicle-to-Vehicle

V2X Vehicle-to-Everything

VANET Vehicular ad hoc network

VFC Vehicular fog computing

XOR Exclusive or

Chapter 1

Introduction

1.1 Background

In 2016, the estimated number of vehicles worldwide was over 1.3 billion, and will double every 20 years. By 2036, there will be an approximate of 2.8 billion vehicles plying the minor and major roads on the planet [1]. While technology progresses, vehicles will become more intelligent and connected because they have more sensing, computational, and communication power, which will provide better travel performance. However, this will also introduce challenges related to environment and energy demands [2]. At the same time, infrastructure nodes such as roadside units (RSUs) have also been deployed. These RSUs are static systems that can also transmit and receive short- and long-range communications to/from vehicles for storage, processing, and information exchange. Collectively, intelligent connected vehicles and the infrastructures are the fundamental building blocks of a vehicular network. Due to these advancements and proliferation of vehicles, ubiquitous monitoring of real-time events and providing up-to-date responses have been possible, even though vehicular networks have a high degree of vehicular mobility and dynamic network availability, connectivity, and topology [3]. In a vehicular network with connectivity and communication among multiple intelligent connected vehicles and infrastructure nodes, various messages can be transmitted. Sharing the local surrounding information enables the delivery of various vehicular applications and services for improving road safety, travel convenience, and coordinated traffic flow [4, 5].

Safety messages are divided into event-driven and routine status messages. Event-driven safety messages can be used for traffic and speed control, accident notification, and traffic updates [6]. These can also be warning messages from emergency vehicles or vehicles having accident information [7]. Other applications include lane change warnings, highway merge assistance, and collision avoidance [8]. On the other hand, non-safety pieces of information are used for convenience and comfort, such as infotainment [5], monitoring locations, planning trips, and discovering available parking space [9]. Non-safety information also includes the static and dynamic road map data like buildings, moving cars, pedestrians, etc.

To reduce traffic congestion and facilitate autonomous driving, appropriate vehicular control instructions, and accurate environment road map data depicting real-time road events are crucial. These information should be exchanged among intelligent connected vehicles and the infrastructure for driving perception, localization, route planning, and control. Given the density or velocity profile of vehicles as the input, the network performance (e.g., delay, throughput, transmission ranges, and contention window sizes) can be calculated as a function of space and time [5]. The vehicles can obtain the optimal network configurations from RSUs and directly set in advance their transmission ranges and contention window sizes to the optimal values before moving into a particular road region [5]. On the other hand, the road map data capture the static (e.g., buildings, road structures) and dynamic (e.g., presence of road accidents, traffic conditions) features of the road setup. A particular type of data that can accurately describe the road environment is the 3D LIDAR point cloud data [10]. The point cloud data, as illustrated in Fig. 1.1 [11], is a set of data points in a 3D coordinate system that represents the surfaces of physical objects in the 3D space.

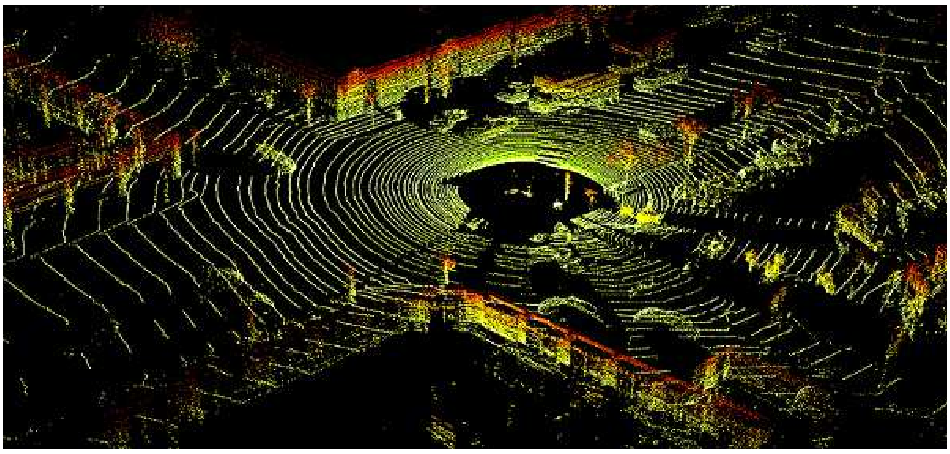

Figure 1.1: A single junction with four road segments represented by 3D LIDAR data.

For control data dissemination to assist road drivers of incoming road danger and emergencies, the broadcast transmission is best used to quickly provide drivers and passengers safety-related information by covering multiple vehicles and infrastructure within the transmission range [12]. On the other hand, the unicast transmission is suitable when the transmitted messages require an automatic repeat request (ARQ) because unicast involves a message retransmission mechanism and a recipient confirmation technique from the origin after a successful reception [13]. Unicast transmission can also avoid the broadcasting storm problem [14].

Unlike control data, environment data, such as 3D LIDAR point cloud data, are usually huge. Commercial LIDAR with 64 laser sensors can generate up to 2.2M points per second for the 3D representation of its surrounding environment [15]. Given this, the exchange of 3D map data from one vehicular node to another is a challenging task. Overcoming the bandwidth limitation in highly-dynamic vehicular networks for exchanging 3D point cloud data can enable collaborative perception among intelligent connected vehicular nodes for extending their sights to reach hidden and distant on-road objects or pedestrians.

Consider the scenario in Fig. 1.2 where the intelligent connected vehicles will cross the intersection with the guidance of a roadside infrastructure because they have a blind perception of the current situation. Vehicle-to-Everything (V2X) communications support the information exchanges (e.g., Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I)), which can be realized by either a short-range local broadcast or long-range unicast via the cellular network. For smooth traffic flow, the RSU, upon studying the current situation, can directly instruct the intelligent connected vehicle on each road section to proceed to its desired course. For highways and multiple intersections with high vehicular densities, a more sophisticated control strategy can be applied to provide a smoother approach to traffic control.

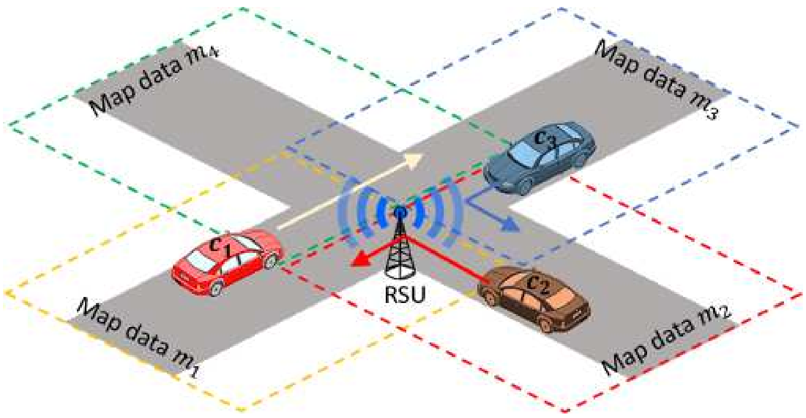

Figure 1.2: A four-way junction with three vehicles and indicated direction of their travel. The RSU coordinates traffic flow such that vehicles do not need to slow down as it approaches the intersection. It can also disseminate road map data upon request.

Highways are a vital transportation infrastructure that usually have more vehicle lanes than the conventional roads to accommodate a higher vehicle volume. Expressways have no traffic light signals, and there are entry/exit ramps, and tollgates. On a freeway, there are three volume-density-based control regulations to prevent or eliminate congestion, namely: (1) on-ramp (most efficient), (2) speed control, and (3) merging control [16]. If the approaches above fail to maintain stable traffic flow, a large volume of vehicles bunches up at these exit tollgates. Queues are another source of inconvenience to vehicle owners and travelers [17].

In a more general intersection and multiple junction scenarios, some intelligent connected vehicles may prefer to update their stored knowledge and rely on up-to-date environment data before making any turning/travel decisions. Given this, we can extend the capability of RSUs to become central databases for all map data, both local and global, i.e., become a Map Data Repository (MDR) infrastructure. MDR is a type of cloud server that stores the global view of map data over time, merges multiple map data sources, and extracts useful information to assist decision-making at individual vehicles.

The traditional broadcasting scheme broadcasts each road map data as a message, one by one, regardless of whether the incoming intelligent connected vehicle already has that piece of road map data or not. However, if certain intelligent connected vehicles already have partial knowledge regarding the environment of surrounding, then transmitting the encoded pieces of road map data will reduce the overall number of transmissions on average [18,19]. Generally, in an n-way junction, there are a various number of intelligent connected vehicles, each possibly carrying map data regarding some road segments. Thus, from the RSU, there are many possible encoded message combinations for transmission that can provide the least number of transmissions for the limited bandwidth condition. Given this, an efficient data dissemination scheme can determine which encoded message should be transmitted based on the overall data demand and availability in the network [4].

1.2 Research Motivation and Contribution

There is an abundant amount of information available in vehicular networks, e.g., traffic density, flow, and state, and static (buildings) and dynamic (presence/absence of accidents) road map data. Hence, the uploading and downloading of data among intelligent connected vehicles and the road infrastructure are time-consuming and take up a considerable amount of network resources. Besides, many V2X applications are time-critical, and failure in transmission may lead to accidents, casualties, or traffic congestions, and inconveniences [20]. Therefore, a pivotal challenge in a heterogeneous V2X network is how to manage the information exchange among vehicles and the infrastructure effectively. An efficient information exchange allows network nodes to acquire the appropriate control and environment data on time for making real-time on-road decisions.

In the first part of the thesis, the control data dissemination problem is studied by considering a highway tollgate scenario for implementation. On-ramp control is the most efficient method to eliminate congestion, but this passively controls the entry of vehicles into the highway, resulting in an uneven distribution of vehicular density at various speeds. Most of the time, vehicles closely following one another tend to develop ghost traffic jams [21] because of their various speeds. To address this congestion, for a given number of highway lanes and tollgates and vehicular capacities, we designed a centralized fuzzy logic controller (FLC) to appropriately assign an incoming intelligent connected vehicle to a server queue in real-time. This automatic queue assignment reduces the overall average waiting time for each intelligent connected vehicle at the tollgate while ensuring fair server utilizations.

Extensive simulation of the real-time decision-making controller based on a homogeneous assumption of service times, homogeneous and non-homogeneous vehicle arrival rates, and fixed queue capacities has been conducted to verify the network performance. The proposed integration of an FLC in a highway tollgate setup addresses the growing vehicle volume while keeping the highway infrastructures constant.

The main contributions of the first part of the study are summarized as follows:

1. We develop a discrete-event model of a highway tollgate scenario that captures the entry/exit behavior of vehicles following the Shortest Queue (SQ) and the proposed Fuzzy Logic-controlled Queue (FLCQ) policies. The SQ policy allows approaching intelligent connected vehicles to select servers currently having the shortest queue. On the other hand, the FLCQ policy assigns the approaching vehicle to a server upon considering the server's lane density and service time.

2. Extensive simulation verified the FLCQ's performance against the SQ policy. With varying tollgate service times and regardless of traffic conditions (light or heavy and homogeneous or non-homogeneous vehicular arrival), a knee separates slow from fast service times. When compared to the standard SQ policy, the FLCQ method presented a maximum 50% improvement in the reduction of expected vehicular queuing time, while having a 20% decrease in average utilization among all servers. Also, to provide a more real-time reaction to highway traffic conditions, the FLCQ policy detects early build-ups for each tollgate. For a given queue density threshold, tollgate servers provide faster service accordingly.

To further exploit the advantages of employing a centralized fuzzy logic controller in the tollway gate servers, the other sections must also be managed accordingly. Highway on-ramp entries/exits are junctions allowing continuous and free flow of traffic. The first part of the study is then extended in Chapter

5, wherein, once RSUs are deployed on the selected optimal intersections, the tollway gate servers' Access Point/RSU also sends out to these RSUs its early warning signal. The received warning signal lets the highway section RSU to control the vehicular density and space mean speed appropriately. This control mechanism will allow density and speed balance such that there is little to no traffic build-up in the tollgate section, effectively, eliminating ghost traffic jams, and further reducing the average queue waiting time.

The second part of the thesis considers a vehicular network employing the fog computing paradigm that implements a scheduled information dissemination mechanism utilizing index coding, data hashing, and heterogeneous transmission options. We employed the fog computing paradigm [22] to bring the cloud computing servers closer to the vehicles to reduce latency and to effectively manage the information exchange among vehicles and the infrastructure in a heterogeneous V2X network. Fog computing has an additional computing layer covering a local area within the vehicular network that processes data coming from its local vehicular and infrastructure nodes and immediately transmits its results to the appropriate vehicular/infrastructure node. Aside from reducing latency, fog computing offers 1) effective operation even at the absence of reliable data connection and bandwidth, 2) enhanced privacy and security because data are not uploaded to remote servers, and 3) reduced operational expenses such as transmission costs over cloud and edge computing platforms [23].

Unlike fog computing, the cloud-edge pairing deploys computation- and storage-enabling technologies closer to end-devices resulting to a localized application [24]. Service-sharing happens only in the cloud, therefore, multiple Internet of Things (IoT) applications are not supported, leading to a high resource-contention rate at the cloud level [25]. Edge computing focuses on the edge- or device-level, while fog computing presents an infrastructure level solution [26]. In the data dissemination scenario in vehicular networks (which is illustrated in Fig. 4.1), vehicles individually perceive their surroundings for objects and obstacles. However, to keep their plan trips updated with important road information, intelligent connected vehicles need to share their environment data and access other location's data in real-time, especially for time-critical cases such as road accidents and vehicular breakdown. This can be accomplished when vehicles interact with an RSU fog node and local controllers and databases for data exchange, where local and global road data are both available in real-time and minimal latency.

Intelligent connected vehicles may serve as mobile fog nodes for implementing localized computational tasks and can directly communicate with nearby vehicles via Dedicated short-range communications (DSRC) or Cellular V2X (C-V2X) [27], especially when vehicles are beyond the infrastructure's coverage. On the other hand, infrastructure nodes, such as roadside units (RSUs), traffic lights, base stations, etc., can act as fog nodes for efficiently communicating with intelligent connected vehicles within its transmission range. These infrastructure nodes can also store massive amount of data and perform computationally-intensive processing and calculations instead of allowing the cloud to do it, thereby, providing real-time and reliable vehicular applications, e.g., autonomous driving in a dynamic environment.

Meanwhile, a local controller facilitates the data exchange among closely related infrastructure fog nodes in a local region. Local controllers determine the transmission mode, i.e., either long-range unicast (Long Term Evolution or LTE) or short-range broadcast via DSRC or C-V2X [27]. They also decide whether specific road map data are to be forwarded to the map data repository (cloud) [28] or stored in local databases. The access of road information from other local regions is administrated by the super software-defined networking (SDN) controller, which is the network component with global intelligence [29] that orchestrates data traffic and manages resources among local controllers and databases [30]. The SDN controllers also perform scheduling of tasks among fog nodes. Finally, the map data repository is a cloud node with a global knowledge of an urban area for monitoring and control at a city-wide level [31].

Applying index coding in the data dissemination scenario has the following advantages. First, the RSU fog nodes, while employing index coding-based transmission schemes, can disseminate information in a more compact packet size given the side information at vehicles, thus, increasing throughput. Second, when binary-coded packets are used, a relatively low-complexity mechanism dissemination is implemented. Thus, computational complexity is minimal, despite the added process for encoding road map data. Finally, when the RSU fog node broadcasts an encoded message, a vehicle cannot decode it unless it has side information. Therefore, the coded packets are secured during the broadcast, which is useful in certain scenarios. One mild drawback of index coding is the generation of overheads in the network. However, since we only employ binary-coded packets in our scheme, our proposed data dissemination scheme introduces minimal processing overhead.

The major contributions of the second part of the thesis are summarized as follows.

1. Under the vehicular fog computing framework, we integrate the index coding algorithm to optimally disseminate high-definition 3D road map data among intelligent vehicles and the roadside infrastructure to reduce the number of required transmissions and data load while satisfying the vehicular demands.

2. We propose fog-based opportunistic scheduling algorithms based on vehicular trip plans for map data downloading in citywide vehicular net-

works. These dynamic schedulers determine the mode of transmission (short-range broadcast or long-range unicast) based on the available resources at fog devices to reduce the overall operating cost of the network. Also, differential coding and hashing techniques for 3D point cloud data uploading at the vehicular level is proposed to avoid data redundancy, and hence reduce the processing and computation load of roadside fog nodes.

3. Utilizing empirical mobility traces and 3D LIDAR data of city streets, we rigorously evaluate the performance of the proposed algorithms and system. We have also implemented our system in a multi-robotic vehicle testbed for practical evaluation.

Developed dissemination strategies are then deployed in roadside infrastructures. The study of deploying RSUs was first proposed in [32] and was aimed to aid vehicles in information dissemination in a vehicular network. In Fig. 1.2, the RSU is positioned at the center of the junction. The deployment is the same on highway tollgates. However, when a citywide scenario of multiple junctions is investigated, RSUs cannot be allocated at all intersections due to its cost and maintenance. Because of this, optimal RSU locations should be determined to optimize information dissemination and minimize deployment costs.

In the third part of the thesis, we explore where to optimally allocate RSU fog nodes for maximized information sharing, given these constraints:

(a) multiple candidate junctions and

(b) any urban spatial locations in a citywide setup by employing empirical mobility traces.

For the multiple candidate junction constraint in (a), we propose an Information Sharing via Roadside Unit Allocation (ISRA) strategy for deploying hotspots at road intersections to support various fog computing applications. ISRA exploits the traffic and communications statistics derived from these mobility traces to select the appropriate spot where the intelligent connected vehicles and the RSU fog nodes can exchange maximum information. In summary, ISRA has the following significant contributions below.

1. Based on a seven-day taxi Global Positioning System (GPS) dataset plying on the first and the second rings of Beijing City, the junction's daily average V2I download and upload contact densities and space mean speeds are extracted.

2. In order to determine energy-efficient and information-rich candidate hotspots, we applied an index-coding based transmission scheme to identify the candidate locations' minimum total number of packet transmissions and transmitted data size to satisfy the information demands of nearby vehicles.

3. Given the empirical findings, ISRA is proposed to identify the optimal positions for the RSU fog centers such that the information shared among vehicles and RSU fog nodes are maximized.

Finally, given a general set of urban spatial locations to deploy RSUs, we develop an Enhanced Information SHAring via RSU Allocation (EISHA-RSU) algorithm to address the information exchanges, connectivity, and coverage issues in the network. While most works have already identified intersections as candidate positions for assigning RSUs (like ISRA), EISHA-RSU extends the search for candidate locations to the whole considered area. This constraint (b) may include landmarks, a particular location along a road segment, and other spots where RSU deployment is possible. EISHA-RSU irregularly partitions an urban area based on its network characteristic to determine regional priorities. Then, in each region, the most suitable positions to deploy the RSUs are determined based on the optimization objectives and constraints. The major contributions of the EISHA-RSU strategy are summarized below.

1. We propose the novel EISHA-RSU deployment scheme as a vehicular-mobility-aware scheme that irregularly partitions an urban city according to its effective regions of movement (ERMs) characterized by vehicular capacity homogeneity. For each formed ERM, effective positions (EPs) are located to ensure urban-wide connectivity, wider coverage, and maximum shared information. Since ERMs are two-dimensional areas, all physical locations are considered as possible EPs for RSU allocation.

2. Extensive simulation utilizing three urban empirical mobility traces and locations is carried to evaluate EISHARSU's efficiency performance. EISHA-RSU fairly allocates RSUs at EPs to achieve the problem objectives, particularly identifying locations with maximized information-rich sources and data carriers.

3. By comparing EISHA-RSU with three other benchmarks (Uniform, City-wide, and MaxInfo Deployment) and employing three urban empirical mobility datasets, our proposed allocation scheme requires on the average 16%, 21%, and 113% less number of RSUs, respectively, while satisfying the problem objectives. Also, based on the Effectiveness metric, EISHA-RSU performs the best when concerning coverage area and the amount of information shared.

1.3 Organization of the Thesis

This thesis is organized as follows.

Chapter 2 reviews the general control and environment data dissemination published works and differentiates the contributions of this thesis.

Chapter 3 presents control data dissemination by highway tollgate servers for optimizing server utilization and minimizing vehicle waiting time. A fuzzy logic controller implemented in an infrastructure node is proposed to attain such objectives.

Chapter 4 investigates the availability and demand of environment data in a citywide scenario. It also proposes an efficient opportunistic data dissemination scheme based on index coding that can optimally reduce the number of transmissions (whether cellular or roadside unit downloads), transmitted data size, and processing overheads. Extensive simulations using empirical taxi mobility traces are employed to verify the efficiency of the proposed algorithm. A robotic experiment setup is also able to confirm its feasibility.

Chapters 5 and 6 discuss optimal allocation techniques for maximizing information sharing among intelligent connected vehicles and between roadside units. Chapter 5 covers only intersections and proposes ISRA to determine possible RSU candidate locations, while Chapter 6 presents EISHA-RSU to consider all spatial locations. Using the optimal index coding technique of Chapter 4, information-rich, and energy-efficient intersections are identified by ISRA and have an RSU allocated at that spot. ISRA also manages the load of each deployed RSU equally. On the other hand, EISHA-RSU employs the Effective Regions of Movement and Effective Positions concepts to determine RSU allocation places.

Chapter 7 summarizes the thesis' contributions and tackles the future direction of the research study.

Chapter 2

Literature Review

In this chapter, related works about the data dissemination in vehicular networks will be discussed. The literature review is divided into two sections discussing accomplishments, issues, and challenges on (1) control and environment data dissemination and exchange and (2) deployment of roadside infrastructures to aid distribution of data in vehicular networks.

2.1 General Data Dissemination Problem

In this work, the data dissemination problem is categorized into two, namely, (1) control data and (2) environment data. Control data are safety-related applications and commands for intelligent connected vehicles to follow for orderly traffic flow. In contrast, environment data are non-safety reports/applications for further processing by the intelligent connected vehicle.

2.1.1 Highway Control Data Dissemination

Earlier works focused on the optimal allocation and management of highway tollgates. [33] derived the optimal number of operating tollgates based on the economic cost per unit time equal to the sum of the toll booth's operating cost and the customer's waiting cost. The tollgate system was based on an $M/D/1$ model, whether in the inbound or outbound direction. Another model used was the Basic Traffic Unit (BTU) that represented tollgates as nodes where roads enter and exit [34]. Having a combined limit of 12 operating tollgates for both inbound and outbound, four and eight tollgates were needed for inbound and outbound directions, respectively. This allocation assumed that the inbound and outbound arrival rates were 8.33 and 18.33 cars/min, respectively. Also, the toll booths were assumed to have the same service times in any direction. However, if we considered peak-hour traffic of 30 cars/min (1800 cars/hr) per road lane as in [35], then the total necessary number of operating tollgate will exceed the set limit of 12 to accommodate a single (inbound/outbound) lane.

In 2011, a 100-km traffic jam, which took 12 days to clear, appeared on an expressway in Inner Mongolia. This standstill was caused by the reconstruction and expansion of the Beijing-Tibet Highway [36]. On the other hand, take the case of the "Carmageddon" that happened in 2015 in China. Thousands of vehicles were stuck on a 50-lane highway and experienced gridlock on the G4 Beijing-Hong Kong-Macau Expressway on their way home after the holidays. According to the article, the queuing of motorists at the toll booths triggered the gridlock [37].

The traffic delays in the tollgates at Port Authority were studied to address the staffing requirement and its equivalent cost, the traffic condition, and the grade of service simultaneously. Efficient scheduling of its workforce to reduce traffic delays and minimize its operational costs was proposed to effectively handle traffic with the least number of tollgate collectors with consistently excellent service while not exhausting the tollgate collectors. The study was done in 14 months and was able to achieve scheduling efficiencies of 95% or better [38]. In [39], toll booth delays and its elimination were also discussed. Adding tollgates was the first and straightforward improvement that just required additional right-of-way, but has diminishing effects because of weaving maneuvers and driver confusion.

Other research studies analyzing tollgate scheduling scenarios were seen in [40,41]. The work in [40] presented a MODSIM III-based simulation tool, called Tollsim, for evaluating the tollgate scenario in Venezuela and help in the traffic management decision-making. Electronic Toll Collection (ETC) services were introduced and evaluated in [41] to hasten tollgate service times. ETC tollgates were dedicated lanes for automatic vehicle identification that do not provide toll fee change and have a uniform and constant service times. In this study, the operational planning and viability for deploying ETC lanes were presented. ETCs do not allow full vehicle stop when compared to an operational tollgate [39]. This discussion about optimizing tollgate management has also been extended into parking systems by including Pay-on-Foot alternative instead of Pay-At-Exit only [42] and predicting waiting delays at intersections [43].

Due to the ever-increasing number of vehicles, the addition of tollgates become less economical in terms of cost and benefits, while making optimal scheduling of tollgate servers not applicable anymore. For instance, in the Philippines, the ratio of the number of vehicles and tollgates is very high, e.g., the average daily vehicle-to-tollgate ratio is 2,814 for the North Luzon Expressway (with 26 southbound lanes), 5,912 for the South Luzon Expressway (with 18 southbound lanes), and 11,405 for the Skyway (with 6 southbound lanes) [44]. In addressing such a dilemma, real-time and robust systems can be used, such as fuzzy systems. Fuzzy logic control (FLC) systems have been considered as one of the most critical technologies that can play a significant role in intelligent transportation systems because they can provide real-time decision-making systems and are quite easy to design [45–47]. When compared to other control mechanisms, fuzzy logic has the following advantages [48–50]. Since traffic is normally characterized by subjective descriptions such as light, moderate, or heavy, assigning control values and set points become flexible and intuitive and can easily be adjusted based on the current traffic conditions. Therefore, simultaneously addressing ambiguity derived from uncertain information. FLC, nowadays, has user-friendly interfaces that allow non-technical decision makers to derive its control strategy that can be based on simple or complicated membership functions. Finally, fuzzy logic controllers can be incorporated with other conventional and modern controllers, e.g., Proportional-Integral-Derivative (PID), Neural Networks, Sliding Mode Control, and State Feedback.

Also, deploying roadside infrastructure in critical highway sections will allow intelligent connected vehicles to obtain early warning signals. Therefore, these vehicles will adjust their speeds accordingly.

2.1.2 Environment Data Dissemination

There have been plenty of flourishing developments for intelligent connected vehicles in the past decade. Intelligent connected vehicles, on their own, can perceive and model their environment, build a map and localize themselves in it, independently plan their paths, and make appropriate decisions and control their motion [51]. They have a communication system capable of short-range broadcast and cellular communications for information sharing among intelligent connected vehicles and infrastructure nodes, such as roadside units (RSUs), base stations, local controllers, databases, and cloud servers.

Most intelligent connected vehicles use 3D LIDAR sensors to perceive the environment in a 3D view. However, 3D-view data require large memory for storage and transmission. Sharing such a considerable amount of data with other intelligent connected vehicles and infrastructure requires large communication bandwidth. Therefore, data dissemination in vehicular networks becomes a challenging task because of its dynamic network topology and frequently disconnected network [52, 53]. To overcome the bandwidth limitation of map data dissemination, communications, and cooperation among vehicles and roadside nodes can extend the sights of vehicles in V2X networks.

The work in [54] analytically studied the horizontal data dissemination in 1D vehicular networks via V2V. A reference car downloaded materials from the Internet and became the source in distributing the contents to other interested parties in only one direction. This model was similar to the SPAWN-CarTorrent protocol [55, 56] that relied on a gossip mechanism for data propagation over multiple hops. To cover more vehicles, [57] proposed an opportunistic publish/subscribe dissemination mechanism to deliver information in a geographical region. Another work that does not rely on infrastructure for data dissemination was seen in [58] by developing Acknowledged Broadcast from Static to highly Mobile (ABSM) protocol [59]. In the ABSM protocol, the vehicle forwards a received broadcast message if it will not create a broadcast redundancy. Periodic beacon transmissions achieved this. TrafficView [60] also disseminated environment data via V2V mode for driving conveniences.

Subsequent works incorporated the use of roadside infrastructures in data dissemination. The authors in [61] proposed a crowdsensing framework that collects available RSU resources to guarantee the level of quality of disseminated data. The authors in [62] investigated the scheduling of significant file distribution problems. It assumed that each intersection had an RSU deployed, and a high-speed backbone interconnected all the RSUs. Scheduled downloads were done in two ways, either downloading from scratch or resuming download. Another cooperative V2I data dissemination was presented in [63]. It discussed a V2I communication model that tackled real-time delay and tolerance for efficiently communicating between RSUs and vehicles. Longer transmission time was expected for substantial data files, but this was less than the time spent by the vehicle in an RSUs range. The same approach of RSU load balancing was previously investigated in [64], which considered request delay tolerance, transferee RSUs current load, and heading of moving vehicle.

The VVID architecture [65] combined the V2V, V2I, and delay-tolerant networks (DTNs) in addressing the timely data dissemination by considering geographical scalability and sparse vehicular density. V2V allowed the propagation of valuable information through the network, while V2I permitted information movement in a vast region. When there is node sparsity, the network used DTNs. [66] proposed the Data Pouring (DP) and Buffering (DP-IB) methods. DP periodically transmitted data on selected roads, while DP-IB rebroadcasted these data on the intersections. DP-IB increased the data delivery ratio and maximized dissemination capacity. The work in [67] relied on cluster formation of V2V and V2I groups for continuous emergency signal data transmission in a one-dimensional road.

Environment data in vehicular networks come in various items and can be requested by different vehicular nodes [68]. The content distribution in vehicular ad hoc networks (VANETs) [69] can be enhanced by network coding, e.g., index coding [70]. For example, [71] discussed how network coding configurations, such as resource constraints affect content distribution performance. In [72], network coding was used for bandwidth efficiency improvement and data service enhancement. Network coding techniques cancelled interference caused by infrastructure transmitting information to designated vehicles [73].

These studies did not optimize the uploading and downloading of the abundant 3D map data for dissemination to intelligent connected vehicles. Also, data processing leads to massive network resources consumption, thereby introducing unwanted delays, and eventually affecting time-critical vehicular applications. Therefore, there is a need to exchange information effectively among intelligent connected vehicles and roadside infrastructures in a vehicular network that supports different transmissions. A more recent approach is by incorporating the fog computing paradigm into vehicular networks [74]. Fog computing brings the processing and communication capabilities among nodes and servers closer to each other.

Fog computing, first coined and introduced by Cisco Systems in 2012, is a recent paradigm bringing cloud computing closer to the network edges to reduce the latency in various real-time services [22, 74]. The incorporation of fog computing into vehicular networks establishes the Internet of Vehicles (IoV) concept or the vehicular fog computing (VFC) paradigm. Extensive fog computing surveys [31, 75, 76] have outlined the possible application of such computing paradigm in vehicular networks. In vehicular fog networks, intelligent connected vehicles act as sensing devices that gather and preprocess surrounding data before uploading. Some data coding and hashing techniques can also be done at the network edges to alleviate the traffic load as well. Intelligent connected vehicles may serve as mobile fog nodes for implementing localized computational tasks and can directly communicate with nearby vehicles via DSRC/C-V2X, especially when intelligent connected vehicles are beyond the infrastructure's coverage.

On the other hand, infrastructure nodes, such as roadside units (RSUs), traffic lights, and base stations, can act as fog nodes for efficiently communicating with intelligent connected vehicles within its transmission range [77]. These infrastructure nodes can also store huge amounts of data and perform computationally-intensive processing and calculations instead of allowing the cloud to do it, thereby, providing real-time and reliable vehicular applications, e.g., autonomous driving in a dynamic environment. There have already been several application scenarios employing RSUs as fog nodes [78–80]. The most common use-case scenario is a traffic management system for improving traffic flow and collection of environmental data.

2.2 Deployment of Roadside Infrastructures

The data dissemination problem in VANETs offers limited wireless bandwidth and intermittent connectivity. One of the strategies to address these issues was to place roadside units (RSUs) strategically, e.g., broadcasting safety application services on highways [81] and information sharing among roadside units in Intelligent Transportation Systems [82]. As more intelligent connected vehicles roam the roads, RSUs will serve as the vehicles' support layer for real-time or on-demand environment sensing, storage, processing, and dissemination. There is a need to efficiently situate RSUs in a city-wide scenario such as Beijing City due to deployment cost and maintenance. Placing RSUs on intersections is the simplest but not the most practical and economical solution [83–85].

The work in [32] considered using a minimal number of specialized and networked infrastructures, called Stationary Supporting Units (SSUs), to provide a dramatic improvement in data dissemination. The locations on which to place these SSUs were based on a heuristic approach. Follow-up work in [86] employed the use of a genetic algorithm (GA) based on travel time savings to determine the proper positions of SSUs. GA was also used in [87] to minimize the deployed number of RSUs while ensuring that a level quality of service was met. In [88], RSUs were deployed by using Affinity Propagation to determine the candidate position from formed clusters based on the traffic statistics of synthetic mobility traces. The performance was evaluated by measuring the delivery ratio, average delay, and hop count in the network.

The published works in [89–94] addressed the RSU deployment problem by maximizing RSU coverage for maintaining high connectivity among vehicles and infrastructures, maximizing throughput, and reducing incident reporting time while using the least number of deployed RSUs in a given urban setup.

In [95], the RSU allocation problem was addressed via the maximum coverage problem by maximizing the number of unique V2I contacts. The maximum distance between RSUs was studied in [96] by incorporating effective bandwidth theory and capacity content. These research works, [96–98], also tackled message delay dissemination in disconnected RSU deployment.

Information download, security, and privacy can also be used as constraints to deploy RSUs efficiently. Driving time, extra overhead time, and file downloading quality were constraints to the optimization problem [99–101]. However, simulations were done on a small scale city map. Also, in [101], RSU allocation was based on the most popular used routes, the most dominant intersection pairs, and the most critical intersection to allow vehicles to update their certificates before it expires.

In order to reduce RSU deployment cost, other works suggested using parked cars as temporary RSUs [102, 103]. Such configuration not only offers a short-term solution in providing aid to an intelligent connected vehicle's navigation and environment data request but also opens discussions about privacy issues like access and energy consumption.

Chapter 3

An Intelligent Highway Tollgate Queue Selector for Improving Server Utilization and Vehicle Waiting Time

This chapter studies the centralized dissemination of control data for a highway tollgate system. On a highway setup, an intelligent connected vehicle will most probably use a tollgate server that has the shortest queue, thinking that it is the fastest exit. However, without extensive knowledge of the other service times and queue lengths, a vehicular build-up is formed. To resolve this, we develop an intelligent highway tollgate queue selector using a fuzzy logic controller. It aims to automatically select the most appropriate tollgate server for an intelligent connected vehicle to ensure the shortest waiting time while trying to balance the server's utilization.

The organization of the chapter is as follows: The first section discusses the development of the highway model for simulation and compares the Shortest Queue (SQ) and Fuzzy Logic-Controlled Queue (FLCQ) policies. The second section provides the simulation results and discussion of the two policies based on a set of performance metrics. Finally, a summary wraps the work done in the chapter.

3.1 Development of an Intelligent Highway Tollgate Queue Selector

We discuss the model development of the highway system in Matlab/Simulink and the shortest queue and fuzzy logic controlled policies.

3.1.1 Matlab/Simulink Highway System Development

Fig. 3.1 shows a section of a highway having N vehicle lanes and i exit tollgates with corresponding l_i queue capacities that is considered. Fig. 3.1 also represents a symmetric queue capacity configuration. An access point (AP) is placed on the highway system where the vehicle communicates to obtain which server to queue according to the proposed intelligent highway queue selector. This is modeled in Matlab/Simulink incorporating modeling blocks from the SimEvents Library as shown in Fig. 3.2.

The following assumptions are used in the highway model.

1. Each lane $L_{z=1,\ldots,N}$ is characterized by an independent, identical Poisson distribution with its own (non)homogeneous arrival rate of vehicles.

2. All vehicles have the same length. Practical deployment of tollgates allows only vehicles of the same class to line up on a tollgate to further hasten entry/exit.

3. Tollgate queue capacities $(l_{x=1,\ldots,i})$ are fixed and based on the number of vehicles it can accommodate.

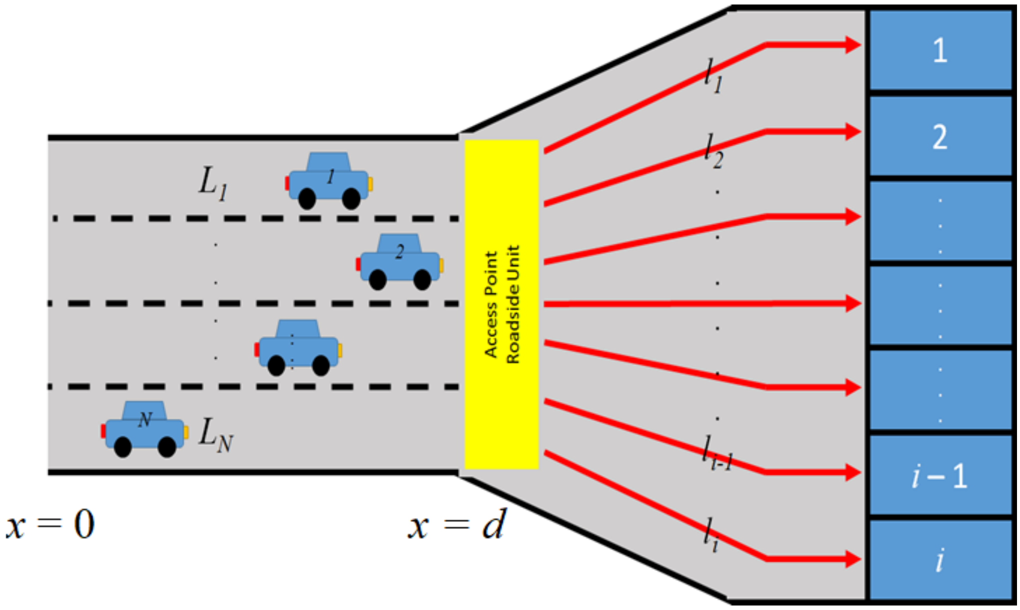

Figure 3.1: Highway Tollgate System.

4. Tollgate service times ($T_{Sy=1,\ldots,i}$) are characterized by a deterministic (constant) or exponential distribution.

Fig. 3.2 depicts the Matlab/Simulink model incorporating modeling blocks from SimEvents Library of a Fuzzy-controlled Tollgate system for the highway segment shown in Fig. 3.1. The three blocks highlighted by the dashed box represent the highway tollgate system following the SQ policy.

The three main blocks are: (1) VehicleGeneration, (2) TollgateQueues and (3) TollGateServers. The VehicleGeneration block generates vehicles following a (non)homogeneous Poisson distribution. The stochastic traffic model used came from [35] and assume that there is no car joining or leaving the highway segment before the tollgate. The highway scenario in this study is modeled as a single state queueing system with a Poisson arrival distribution, exponential service time, c servers and K capacity, $(M/M/c/K)$. Though vehicles originate in more than one highway lane before $x = d$ (see Fig. 3.1), service will be dependent on a first come, first served basis (FIFO).

The TollgateQueues block determines which tollgate a vehicle will use to

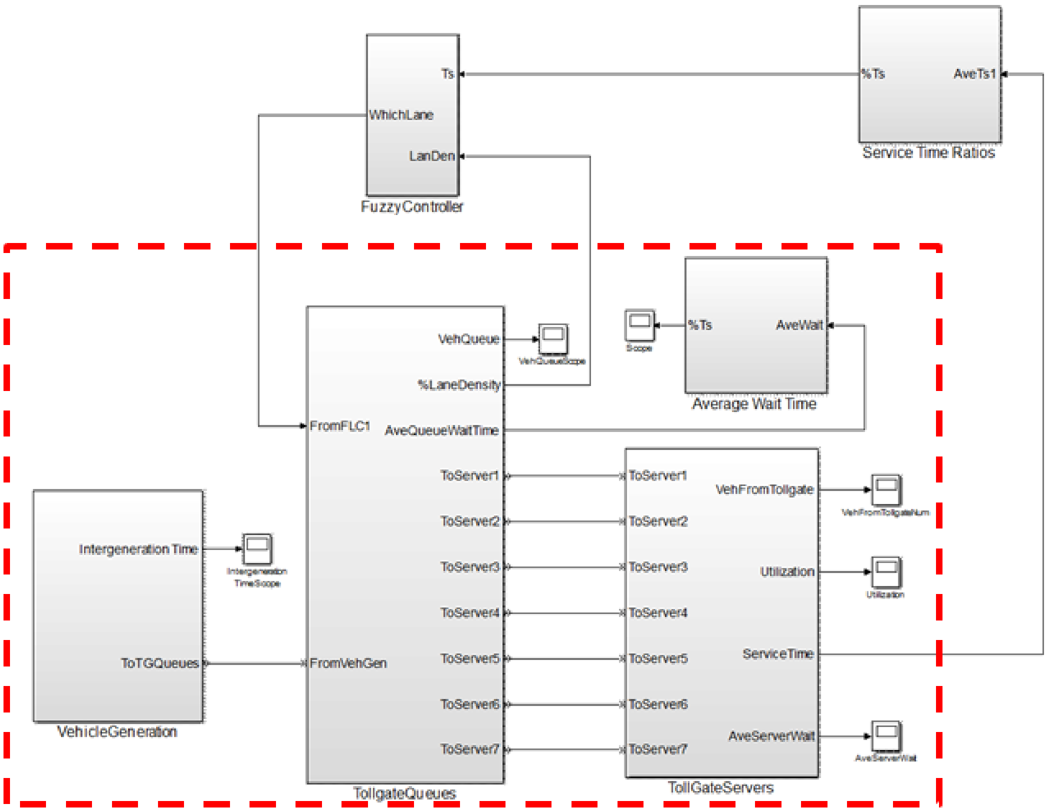

Figure 3.2: Matlab/Simulink System Model. Dashed box represents the highway tollgate system.

exit the highway. The queue capacities are defined before any simulation run depending on the tollgate configuration. The TollGateServers block contains all the available servers. Service time (T_S) can be set to be a constant or exponential distribution.

3.1.2 Shortest Queue and Fuzzy Logic-Controlled Policies

There are two policies governing how a vehicle queues in a server that are presented: (1) Shortest Queue (SQ) [40] and (2) the proposed Fuzzy logic-controlled Queue (FLCQ). The SQ Policy is used at a certain instance by any approaching vehicle to select which server currently has the shortest queue. This selection does not consider which server has the quickest service time or the highest utilization. On the other hand, the FLCQ Policy determines which lane the approaching vehicle should queue. The FLCQ takes into consideration the server's current lane density and its service time to make the decision.

The SQ Policy pseudocode is shown in Algorithm 1 below. A vehicle approaching a tollgate server makes its decision based on the current server i lane density (QD_i). Vehicles select tollgates with possible minimum QD based on its perception. These selected tollgates are defined as possible queues PQ where vehicles can line for entry/exit.

Pseudo-codes 1 Shortest Queue Algorithm

Get Lane Densities, QD_i
Determine ALL Possible Queues (PQ) with minimum QD
if $PQ > 1$ **then**
    Select a queue based on a uniform random number
end if
Output Queue number

Queue Density (QD_i) is the ratio between the number of vehicles in a queue and the queue capacity. Mathematically, it is expressed as:

$$QD_i = \frac{n_i}{l_i} \tag{3.1}$$

where: n_i = number of cars in queue i and l_i = queue i car capacity

Normally for a symmetrical tollgate configuration, if there are servers having the same number of vehicles in queue, a vehicle will likely choose any of the innermost servers because of the traveled distance involved and the concern of changing of lanes. Therefore, during a light traffic flow, tollgate servers having queues with the least queue capacities will be highly utilized.

For the FLCQ policy, the crisp inputs chosen to be fuzzified are the lane density and service time ratio (TS_{Ratio}). TS_{Ratio} is defined as:

$$TS_{Ratio} = \frac{T_{Si}}{T_{Smax}} \tag{3.2}$$

where: T_{Si} = service time of server i and T_{Smax} = maximum service time from all servers

The service time ratio allows the fuzzy logic controller to have an idea on the relative service time of a certain tollgate with respect to the other tollgates. The defuzzified output is the probability of the tollgate being chosen. Fuzzy control is then developed using the rule base that is generally implemented by:

$$QP = fuzzy(TS_{Ratio}, QD) \tag{3.3}$$

where the server time ratio and queue density are the preconditions while the consequent is the queue probability (QP). QP is dynamically updated once a new vehicle joins. Creating the fuzzy rules are based on the idea that when the server time ratio is small (thus, being the fastest), and the lane density is low (thus, being the shortest), the probability of choosing the server queue is high. The fuzzy rule base is shown in Table 3.1.

Table 3.1: FLCQ Policy Fuzzy Rule Base.

QD \ TS_{Ratio}	F	OK	S
SQ	HP	HP	HP
OK	HP	OK	OK
LQ	HP	OK	LP

The LD inputs are categorized as: SQ (short queue), OK (nearing half-filled queue) and LQ (long queue). On the other hand, the TS_{Ratio} inputs are characterized by: F (fast), OK (just right) and S (slow). There are two types of membership functions (MFs) used to represent the inputs, namely, (1) triangular MFs and (2) trapezoidal MFs. These are shown in Fig. 3.3.

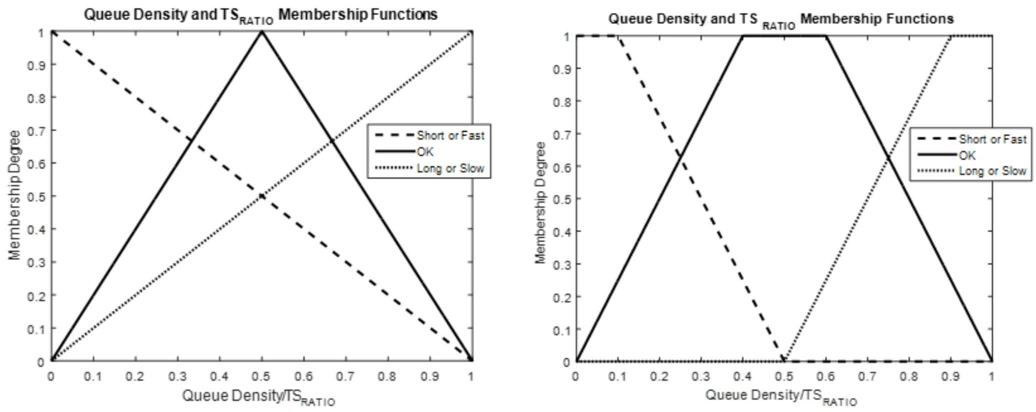

Figure 3.3: Triangular (left) and Trapezoidal (Right) Membership Functions.

The membership degree of an MF describes the relationships between input events. [104]

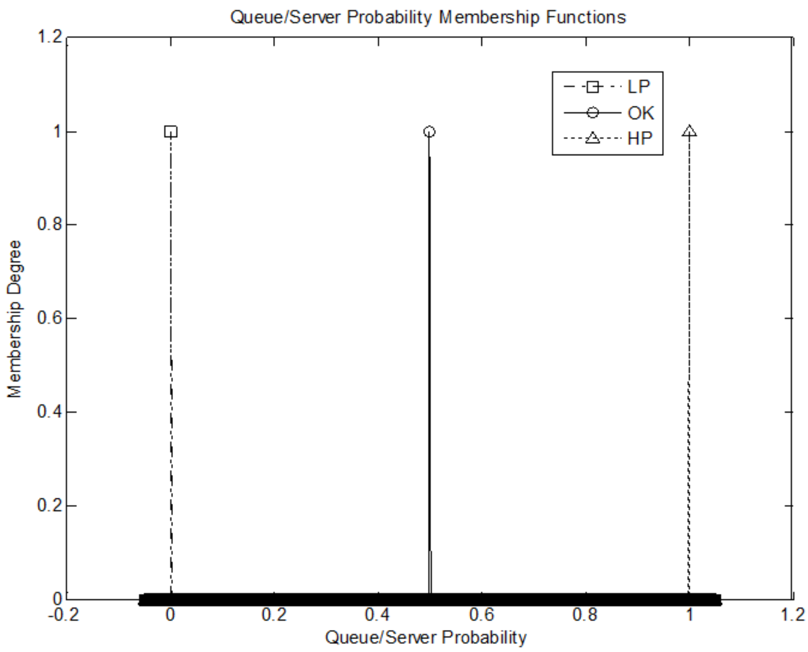

Figure 3.4: Output Membership Function .

The possible outputs for these two approaches in input membership functions are: HP (high probability), OK (more or less 50% of being selected) and LP (low probability). These are depicted in Fig. 3.4. The linear membership functions were chosen because of the advantages it holds, i.e., simple implementation and fast computation.

To clearly see the input-output relationships of the proposed fuzzy logic controllers, the surface views of the fuzzy rule base in Table 3.1 along with the two approaches in Fig. 3.3 are shown in Fig. 3.5 respectively. The surface views show all possible combinations of inputs and its corresponding probability output.

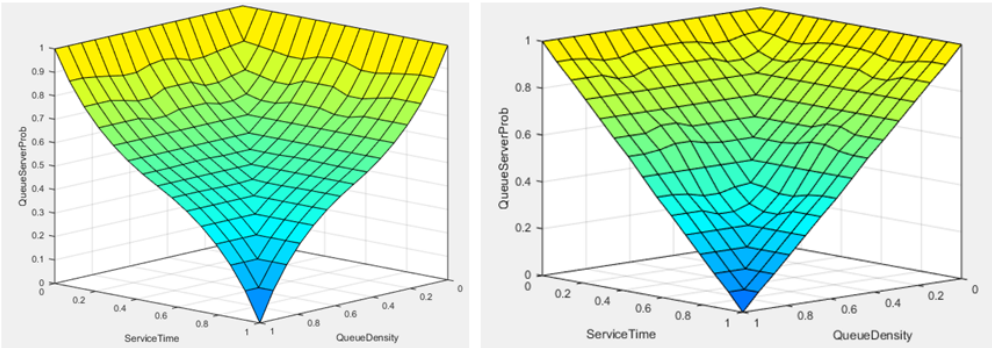

Figure 3.5: Fuzzy surfaces for triangular(left) and trapezoidal(right) MFs.

The implication method of the fuzzy logic controller used the *min* operator, while the aggregation of each rule output was done by the *max* operator. In the implication method, the result (a single number) derived from "*and*-ing" the preconditions is used to reshape the consequent by truncating the output fuzzy set. This is done for all set of rules. On the other hand, the aggregation method combines all truncated output fuzzy sets (inputs) from the implication method into a single fuzzy set by getting the maximum value when comparing all inputs. Finally, defuzzification is implemented by finding the centroid of the aggregated output. The fuzzy logic process is done for every instance a new vehicle joins and chooses a tollgate for its entry/exit.

For example, if the tollgate's current $QD = 0.25$ and $TS_{Ratio} = 0.5$, then its probability of being chosen next when a new car joins is $QP = 0.667$. The process discussed below is shown in Fig. 3.6 for triangular member input functions. The values of both QD and TS_{Ratio} are drawn vertically into its corresponding MFs for all the nine rules. The intersection of this vertical line with the MFs creates the shaded regions (see columns 1 and 2). We compare these two regions point by point and get the minimum of the two. This newly formed region will now be used to reshape the output MFs (see column 3). The aggregation method combines all truncated output MFs by comparing them point by point and getting the maximum value among all compared points (last row and last column text box). Finally, the centroid is determined to arrive at $QP = 0.667$.

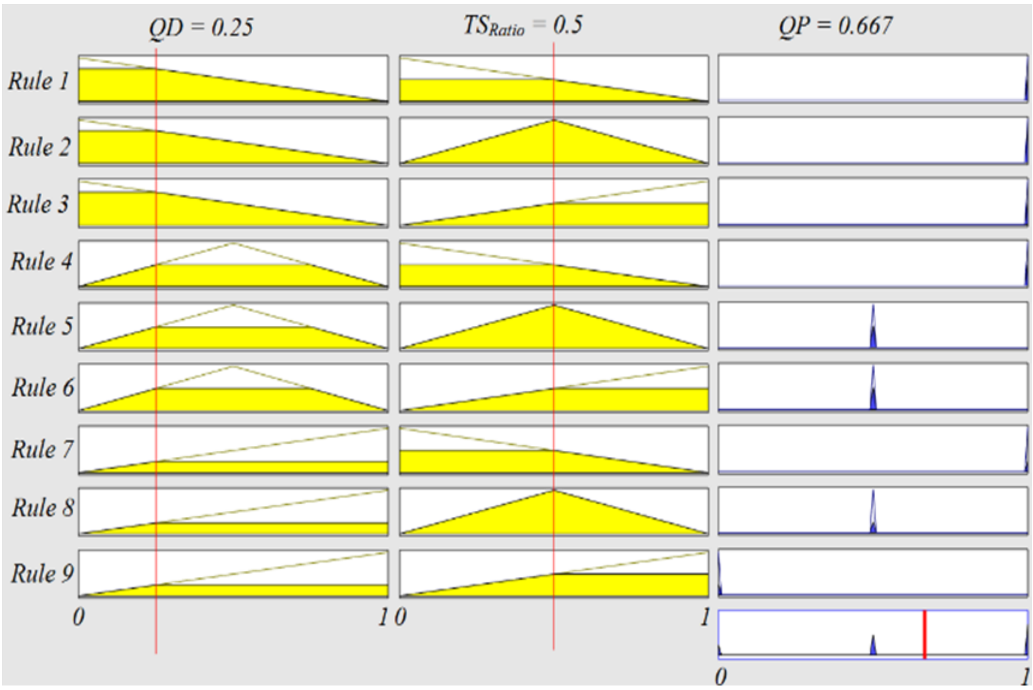

Figure 3.6: Fuzzy Logic Process.

All server-queue pairs have their own probability of being chosen. A Matlab function then selects the one with the highest probability. If two or more server-queue pairs are possible, then it chooses randomly following a uniform distribution.

3.2 Simulation Results

The tollgate symmetric configuration shown in Fig. 3.1 is used to simulate and compare the SQ and FLCQ policies. There are four ($N = 4$) highway lanes and seven ($i = 7$) tollgate servers/queues. The performance metrics used are: (1) Average Queue Length (QL_{Ave}), (2) Average Queue Waiting Time (QW_{Ave}) and (3) Server Utilization ($Util$). The average queue length is defined as the time average of the number of vehicles. The server utilization is defined as ratio of time spent servicing a vehicle over the total simulation time. In queueing theory, it is defined by the expression below.

$$U_i = \frac{\lambda_i}{\mu_i} = \frac{\frac{n_i}{T_{Total}}}{\frac{1}{T_{S_i}}} = \frac{n_i T_{S_i}}{T_{Total}} \tag{3.4}$$

where: λ_i = arrival rate at a tollgate server i, μ = tollgate i service rate, n_i = number of cars in queue i, T_{S_i} = tollgate server i service time and T_{Total} = total simulation time.

The highway lane arrival rate of each lane follows either a homogeneous or non-homogeneous Poisson distribution. The queue capacities (in number of vehicles) are set to $l_1 = l_7 = 50$, $l_2 = l_6 = 40$, $l_3 = l_5 = 30$ and $l_4 = 20$ while the service times are modeled exponentially. Table 3.2 summarizes the different arrival rates used.

Table 3.2: Simulation Parameters.

Sim Trial	λ_1	λ_2	λ_3	λ_4
1	12	10	8.57	7.5
2	30	20	15	12
3	30, 20	15, 12	10, 8.57	7.5, 6.67

The third simulation trial represents the non-homogeneous arrival rates at the highway lanes. For Trials 1 and 2, the simulation time was set to 7200 sec (2 hrs) while for Trial 3, simulation time was set to 14400 sec (4 hrs). For example, the arrival rate of $[x, y]$ in Table 3.2 Trial 3 signifies that during the first hour of simulation, there are x cars/min, for the second hour, there are y cars/min and then becomes periodic.

Fig. 3.7 shows the combined RMS results of all simulation trials of all queues/servers. The average queue length, waiting time and server utilizations of all queue/servers were observed under varying but homogeneous servers' service times with values from 3 – 15 seconds. Note that the y-axis is in semilog scale to highlight the differences between the policies at various traffic conditions.

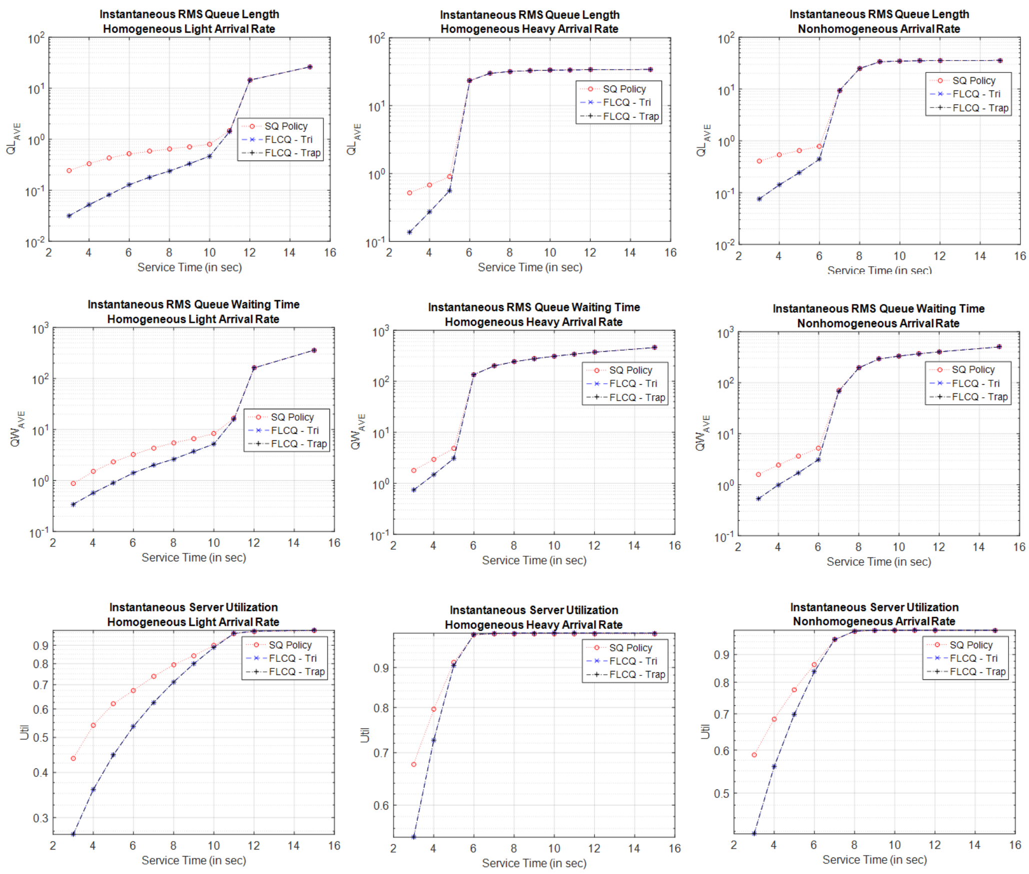

Figure 3.7: Simulation Results for Light (left), Heavy (center) and (c) Non-homogeneous Arrival Rates (right).

Regardless of traffic condition, the service times creates a knee in the performance metrics that separates the interval of fast and slow service times. During fast service times, the FLCQ policies provide shorter average queue length and waiting time for a vehicle approaching a tollgate compared to the SQ policy. It introduced a maximum approximate improvement of 50%. There is also a decrease in the server utilization that practically translates to the servers not tiring out quickly. During the interval of slow service times, i.e., after the knee, the performance metrics are generally the same for all policies. This means that the vehicular arrival rate is faster than how the tollgate servers provide service, therefore, creating more build-ups.

The maximum RMS queue capacity of 38.54 vehicles is reached and the average waiting time is at its maximum and is equivalent to the instantaneous RMS queue length multiplied by the service time. Finally, the server utilization can be seen to be fully utilized. (3.4) is used to obtain the numerical value of the instantaneous server utilization.

Between the two FLCQ policies, the triangular MFs offer a better response than the trapezoidal MFs just before the response's knee. From the surface views of the two FLCQ policies, we note that trapezoidal MFs provide a constant change on the queue/server's probability to be chosen. This means that all queues/servers have an equiprobable chance of selection. On the other hand, examining the triangular MFs surface view reveals that the shorter lane densities and faster service times are given a higher range of probabilities over the half-filled and "OK" service time ratios. Also, if the traffic condition is heavy, a very small chance of being selected is given to a certain queue/server. In this sense, it can be taken that triangular MFs lead to a more reactive policy than the trapezoidal MFs if following the 9-rule base defined in Table 3.1.

Fig. 3.8 shows the response of the two FLCQ policies over the instantaneous RMS queue length performance metric for each queue/server. It can also be seen in Fig. 3.8 that the queue length has been equalized. It follows here

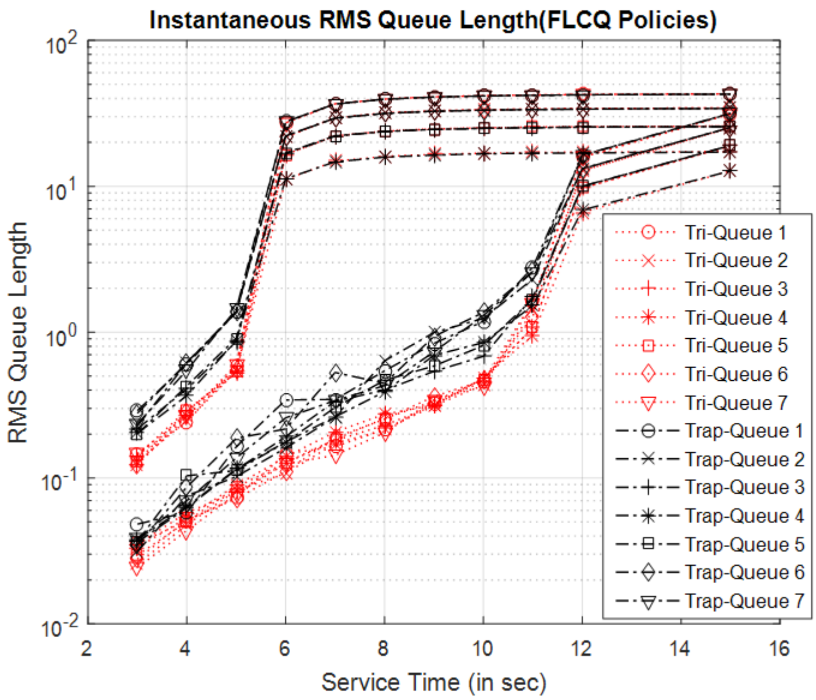

Figure 3.8: Comparison between the Triangular and Trapezoidal MFs.

that the average queue waiting time is also the same as well as the server's utilization. This is not the case in the Shortest Queue policy.

To further enhance the advantage of the FLCQ over the SQ policy especially during longer service times, the server is provided an early warning capability once the lane density is approaching its maximum. In this simulation run, it is assumed that after the FLCQ signal has been given, the servers reduced its service time to half. Fig. 3.9 shows there is a decrease in the RMS server utilization after the addition of this capability. In practice, road side units (RSUs) can be installed along the highway to monitor the traffic density. The FLCQ constantly communicates with these RSUs to update the current traffic density information and decides if there is a need to inform the server to hasten its service time.

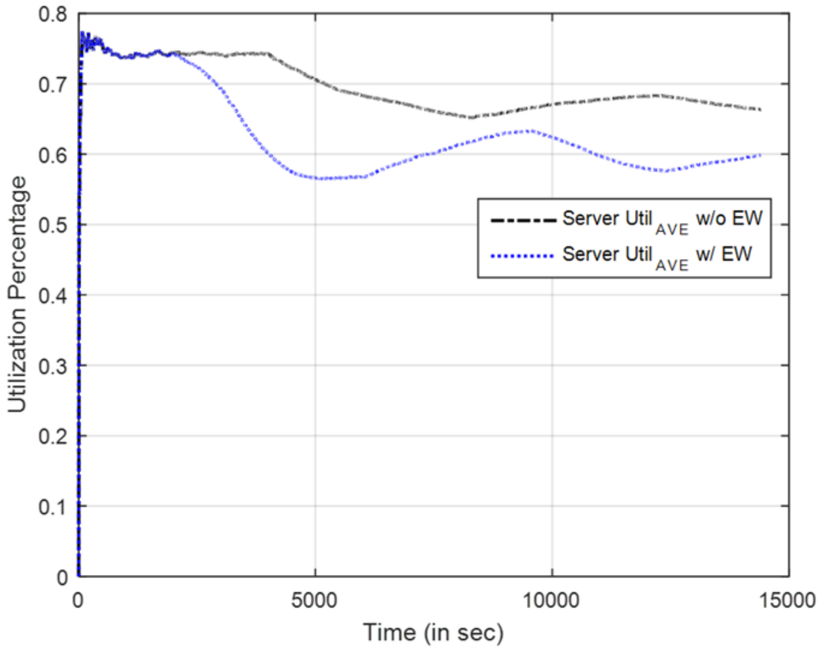

Figure 3.9: Comparison of FLCQs Average Server Utilization with and without early warning (EW) capability.

3.3 Summary

It has been successfully shown that the incorporation of a fuzzy logic controller in a highway tollgate system has effectively decreased a vehicle's average waiting time and average queue length, and improved the server's utilization. This is true for both homogeneous and non-homogeneous vehicular arrivals in the highway lanes especially with tollgates of fast service time. As an added feature to the FLCQ selector, an early warning signal is introduced to allow the server to reduce its service time when the traffic is building up due to relatively long service times.

Chapter 4

Efficient 3D Road Map Data Exchange for Intelligent Vehicles in Vehicular Fog Networks

The basic control data dissemination for implementing an efficient and intelligent highway tollgate system has been previously discussed. Control data are sent to the intelligent connected vehicles by a centralized infrastructure. This control data dictates where a vehicle will enter/exit a tollgate, thus, effectively reducing the vehicle's waiting time and queue length while at the same time improving the server's utilization.

In this chapter, we investigate the availability and demand of environment data to be used in a more useful way to alleviate road accidents, promote driving convenience, and provide road safety to all users. [4] studied a single junction scenario, and we extend the work to a citywide setup where there are multiple intersections involved.

The chapter is organized as follows. The first section describes our proposed information dissemination system at the fog layer. In the second section, we formally define the information dissemination problem and discuss the downloading and uploading operations of 3D road map data. The third section presents motivating examples of utilizing index coding for vehicular data exchange. It derives the optimal index coding scheme for both the single road junction and multi-junction scenarios. The fog-based opportunistic scheduling problem is tackled in the fourth section, while the techniques for efficient uploading of 3D LIDAR point cloud data from vehicles are covered in the fifth section. Sections 6 and 7 present experimental and simulation results obtained based on our multi-robotic vehicle testbed and empirical mobility traces, respectively. Finally, the last section concludes the chapter.

4.1 Information Exchange of 3D Road Map Data in V2X Networks

To implement efficient road map data dissemination in a vehicular fog network, we propose the 3D MAp Dissemination System (3D-MADS). The general operation of 3D-MADS includes intelligent vehicles, roadside units, local controllers and databases, which are all within the fog layer in Fig. 4.1. Overall, the system distributes map data among the parties promptly, taking into account the characteristics of long-range unicast and short-range broadcast transmissions. Short-range local broadcast achieved by Dedicated Short-Range Communications (DSRC) normally has limited available spectrum resources, short transmission range, and restricted data transmission rate, but at lower transmission cost. On the other hand, long-range unicast via cellular networks has larger bandwidth capacity at higher transmission cost. However, it may be inefficient to share common data among nearby transmitters, such as map data for vehicles in the vicinity. We aim at optimizing these transmission options while satisfying the dynamic data demand of respective vehicles. By referring to Fig. 4.1, each component or module in 3D-MADS and its corresponding tasks are explained as follows.

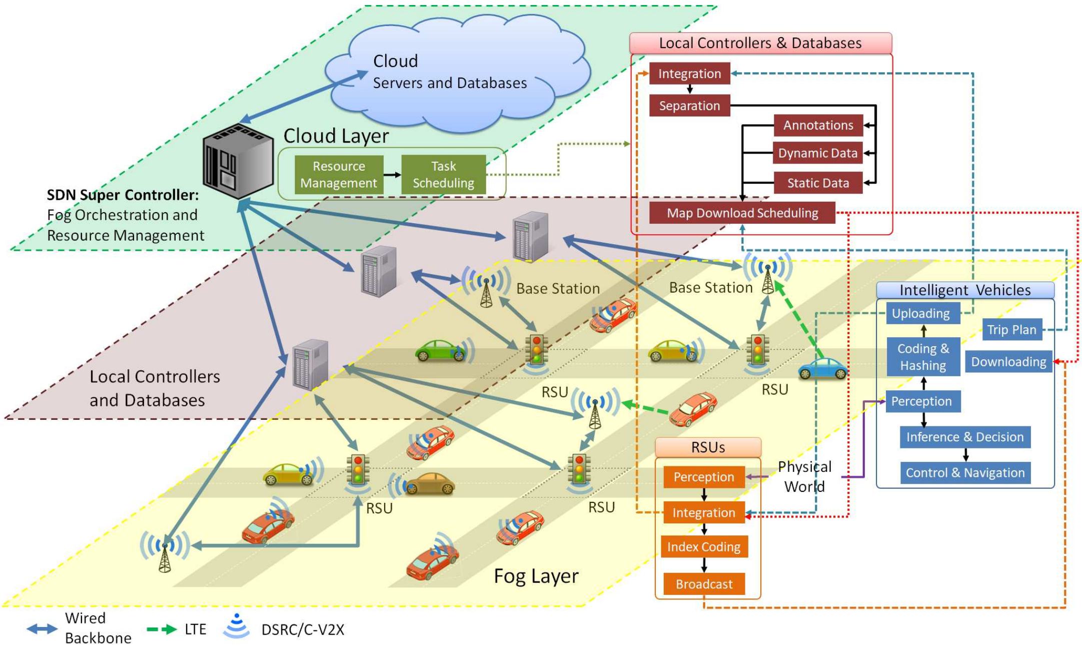

Figure 4.1: The vehicular fog computing architecture. Most information exchange and computation take place in the fog layer.

- **Intelligent Vehicles**

  - *Uploading* enables sharing of on-board LIDAR data among vehicles via the vehicular fog network.

  - *Coding & Hashing* encodes and identifies differentiated data and redundant map information.

  - *Downloading* delivers the most updated 3D road map data from local databases to intelligent vehicles via either cellular network or local broadcast at RSU fog nodes.

  - *Perception* utilizes on-board sensors, e.g., LIDAR and GPS, to perceive the surrounding road environment as 3D point cloud data, from which we can detect and recognize objects and obstacles in the environment. The locally processed 3D point cloud data will be uploaded to RSUs or local controllers for further integration with the data from other vehicular and roadside nodes.

- *Inference & Decision* allows intelligent vehicles to predict their movements for autonomous navigation and control based on the perceived and downloaded 3D road map data as well as position information.

- *Control & Navigation* relies on driving feedback and manages the intelligent vehicles to move safely and appropriately in the environment.

- **Roadside Unit (RSU) Fog Nodes**

  - *Perception* provides blind-spot views that cannot be detected by intelligent vehicles via the local sensors.

  - *Integration* combines downloaded 3D road map data from the cloud with the local LIDAR sensor data before sending them to nearby intelligent vehicles.

  - *Index Coding* encodes 3D road map data according to the data demand and availability of nearby vehicles to improve the transmission efficiency.

  - *Broadcast* is the periodic transmission of index-coded data to nearby vehicles via local short-range broadcast.

- **Local Controllers and Databases (LCD)**

  - *Integration* coordinates the data exchanged among intelligent vehicles and RSU by setting the locations and boundaries of each region of map data. It can also correct and realign the LIDAR data from different vehicles that may contain drifting inaccuracy.

- *Separation* differentiates static and dynamic objects in the integrated 3D road map data via segmentation. Additional annotations can be generated based on machine learning techniques [105] to label the objects in the map data. Different coding and transmission schemes can be applied to data with different characteristics.

- *Scheduling* organizes the download and upload transmissions based on the trip plans of vehicles, given the options of using either the cellular network unicast or the short-range local broadcast transmissions.

With respect to Fig. 4.1, we can see that 3D-MADS is an interdisciplinary system that requires the joint effort from multiple fields (e.g., communications, signal processing, computing, navigation and control, transportation engineering, etc.), which is our long-term goal. In this paper, we focus on investigating and discussing data exchange related modules (which include index coding, map download scheduling, and coding and hashing) to kick start the development of such system.

4.2 Formulation and Definitions

In this section, we formally define the 3D road map data dissemination problem for intelligent vehicles. Consider a set of discrete time slots $t \in \mathcal{T}$, where $|\mathcal{T}| = T$, and a network of roads that is represented by graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where each node $v \in \mathcal{N}$ represents a junction and each undirected edge $e \in \mathcal{E}$ represents a road segment. For each edge e at time t, a set of map data is associated and denoted by $m_e(t)$. $m_e(t)$ consists of both static data set m_e^{s} and dynamic data set $m_e^{\mathsf{d}}(t)$, such that $m_e(t) = m_e^{\mathsf{s}} \cup m_e^{\mathsf{d}}(t)$. We consider an abstract representation, without specifying the elements in $m_e(t)$. That is, one may consider an element in $m_e(t)$ as a map data file. The dynamic data may be generated from roadside sensors, and perception from other vehicles. For practicality, we consider the dynamic data within a certain time window τ from the current time t, namely $m_e^{\mathsf{d}}(t')$ where $t' \in [t-\tau, t]$.

There is a set of vehicles $\mathcal{C}$ where each vehicle $c \in \mathcal{C}$ is associated with a trip plan P^c, which is a path in $\mathcal{G}$. We represent P^c by a set of edges in $\mathcal{E}$, or a sequence of nodes in $\mathcal{N}$. Let the time of vehicle c entering edge (i.e., road segment) $e \in \mathsf{P}^c$ be t_e^c, and the time of entering node (i.e., junction) $v \in \mathsf{P}^c$ be t_v^c.

4.2.1 Downloading

Each vehicle $c \in \mathcal{C}$ downloads both static data m_e^{s} before t_e^c, and dynamic data $m_e^{\mathsf{d}}(t')$, for some $t' \in [\mathsf{t}_e^c - \tau, \mathsf{t}_e^c]$, at some time between t' and t_e^c. The options for downloading are either using short-range broadcast transmissions at RSUs, or unicast transmissions via cellular networks. We assume that LTE cellular network transmissions have much larger capacity, whereas short-range broadcast transmissions are limited by local spectrum allocation. On the other hand, the short-range broadcast transmissions incur no or very low costs, whereas cellular network transmissions incur higher costs.

We assume that RSU fog nodes are only located at a subset of nodes in $\mathcal{G}$, denoted by $\mathcal{R} \subseteq \mathcal{N}$. Denote the set of edges connecting to RSU fog node $r \in \mathcal{R}$ by $\mathcal{E}_r \subseteq \mathcal{E}$. A vehicle c can receive data from r, when entering edge $e \in \mathcal{E}_r$. At each RSU fog node $r \in \mathcal{R}$, there is a download capacity of $\mathrm{C}_r^{\downarrow}$ at r, whereas there is no capacity limit via cellular networks.

Let the data transmitted by RSU fog node r using short-range broadcast at time t be $x_r(t)$. A vehicle c can also download data via cellular networks, which is denoted by $y^c(t)$. At the time t, let $X^c(t)$ be the union of all data that c has received from the visited RSU fog nodes on its path before time t, namely,

$$X^c(t) \triangleq \bigcup_{e \in \mathsf{P}^c \wedge e \in \mathcal{E}_r \wedge \mathsf{t}_e^c \leq t} \left\{ x_r(\mathsf{t}_e^c) \right\} \tag{4.1}$$

Also, let $Y^c(t)$ be the union of data that c has received from cellular network transmissions before time t, namely,

$$Y^c(t) \triangleq \bigcup_{t' \leq t} \left\{ y^c(t') \right\} \tag{4.2}$$

We denote a decoding function by $\mathsf{Dec}[\cdot]$, which decodes all the downloaded data to a set of map data, $M^c(t) = \mathsf{Dec}[X^c(t), Y^c(t)]$.

We aim to minimize the number of cellular network transmissions, subject to the constraints of timely delivery of static and dynamic data:

$$\min_{\{x_r(t), y^c(t) | t \in \mathcal{T}, c \in \mathcal{C}, r \in \mathcal{R}\}} \sum_{c \in \mathcal{C}} \left| Y^c(T) \right| \tag{4.3}$$

$$\text{subject to } |x_r(t)| \leq \mathrm{C}_r^{\downarrow}, \text{ for all } t \in \mathcal{T}, r \in \mathcal{R}, \tag{4.4}$$

$$m_e^{\mathsf{s}} \in M^c(\mathsf{t}_e^c), \text{ for all } c \in \mathcal{C}, e \in \mathsf{P}^c, \tag{4.5}$$

$$m_e^{\mathsf{d}}(t) \in M^c(\mathsf{t}_e^c), \text{ for all } c \in \mathcal{C}, e \in \mathsf{P}^c, \text{ for some } t \in [\mathsf{t}_e^c - \tau, \mathsf{t}_e^c]. \tag{4.6}$$

In this problem, we assume that the trip plans of all vehicles are given a-priori. However, the online version is also discussed in Sec. 4.4. Cons. 4.4 represents the capacity constraint of local broadcast, whereas Cons. 4.5 and Cons. 4.6 represent the download constraint of static data and dynamic data, respectively.

4.2.2 Uploading

The previous section considers downloading map data from fog units (e.g., RSUs and base stations). In practice, intelligent vehicles are equipped with various sensors (e.g., LIDAR, RADAR, camera, inertial measurement unit (IMU), GPS unit, etc.), whose data can be uploaded to LCD via RSUs or base stations for sharing with other vehicles.

We consider the uploading of processed 3D LIDAR point cloud data in this paper, in which the operations can be optimized by uploading hash files of the perception data and differentially coded data to reduce the redundant data load to the network, as described in Sec. 4.5.

4.3 Index Coding for Local Broadcast at RSU Fog Nodes

To reduce latency in the presence of numerous intelligent vehicles, the local broadcast operations at RSU fog nodes can be improved by index coding. Index coding is a variant of network coding [19, 70] applied to wireless communications. Nearby vehicles will likely receive common information by local broadcast, which also possess certain prior information (i.e., information received from other RSU fog nodes at previously traversed road segments). We show that smart data dissemination considering prior information can significantly reduce the number of broadcast transmissions needed.

This section only considers the dissemination of static data, without capacity constraint. In the next section, we will develop heuristics for the settings with capacity constraint and dynamic data. It is assumed that the local broadcast transmissions incur a very low cost, which is negligible.

4.3.1 Motivating Examples

We first present some motivating examples of index coding. The basic idea of using index coding to optimize transmissions at RSU fog nodes is by mixing the transmitted packets with prior information previously received.

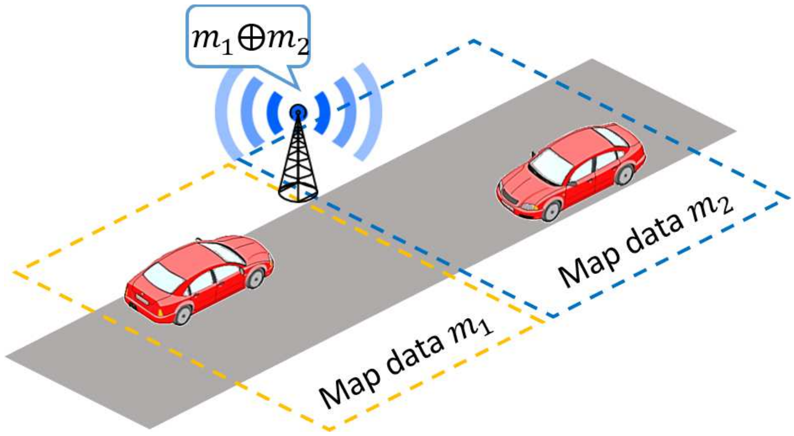

Figure 4.2: An example of index coding for map data dissemination with two opposite traveling intelligent vehicles.

Example 1: We illustrate a simple example using index coding for map data dissemination in Fig. 4.2. There are two intelligent vehicles traveling on opposite directions. Consider the static map data for two road segments, denoted by m_1 and m_2 in bit string representation. Both vehicles are now within the transmission range of a common RSU fog node and had obtained map data m_1 and m_2 correspondingly, before entering their respective road segments. The common RSU fog node can broadcast a coded packet $m_1 \oplus m_2$, where $\oplus$ is a bitwise XOR operator, thereby, reducing the number of broadcast transmissions. To obtain the required map data, the vehicles can decode using the received data as follows: $m_1 \oplus (m_1 \oplus m_2) = m_2$ and $m_2 \oplus (m_1 \oplus m_2) = m_1$.

Example 2: We next consider an example of index coding for a four-way junction in Fig. 1.2. There are three vehicles: c_1 moving from m_1 to m_3, c_2 moving from m_2 to m_1, and c_3 moving from m_3 to m_2. Note that we use m_i to denote road segment i as well as the map data of road i for notation simplicity here. In this case, the RSU fog node only needs to broadcast two packets: $m_1 \oplus m_3$ and $m_2 \oplus m_1$. c_1 can obtain $m_3 = m_1 \oplus (m_1 \oplus m_3)$, c_2 can obtain $m_1 = m_2 \oplus (m_2 \oplus m_1)$, and c_3 can obtain $m_2 = m_3 \oplus (m_2 \oplus m_1) \oplus (m_1 \oplus m_3)$.

In the preceding examples, the vehicles are able to decode the required packets by bitwise XOR operation $\oplus$. Note that the bitwise XOR operator is a linear operator over the binary number field. Applying index coding in these scenarios can improve network throughput and reduce latency. One mild drawback is that it generates overheads in the network. However, since only binary-coded packets are employed in our scheme, it can still be solved within polynomial time. The reader is referred to Section 4.7.4 for the overall delay analysis based on the processing overheads and transmission delay in the proposed index coding scheme.

4.3.2 Optimal Index Coding for Single Junction

In this section, we derive the general theories for constructing index coding schemes for a road network with a-priori trip plans of vehicles. We only consider linear index coding, i.e., the coding/decoding schemes only rely on bitwise XOR operator. In linear index coding, the encoding/decoding operations can sometimes be interpreted as unions and complemented intersections on a set of packets. For example, coding by $m_1 \oplus m_3$ can be interpreted as union $m_1 \cup m_3$, whereas decoding by $(m_1 \oplus m_3) \oplus m_1 = m_3$ can be can be interpreted as complemented intersection $(m_1 \cup m_3) \cap (m_1 \cup m_3 \backslash m_1) = m_3$.

In general, a good index coding scheme for multiple junctions is a hard problem, because it is related to the multi-source network coding problem, which is an open problem [19, 70]. Instead, we focus on one single junction first, and then extend the single-junction scheme as a heuristic for multiple junctions. In fact, under the assumption of 'single meeting' as depicted in the next subsection, this is an optimal solution. Note that we ignore the download capacity in this section, which will be considered in the general schemes in the next section.

To construct a good index coding scheme, we consider a particular RSU fog node at a single n-way junction, labeled as $r \in \mathcal{R}$. We represent the demands for map data by a directed graph (called *demand graph*) $\mathcal{D}_r$ with a set of n nodes representing the set of connected road segments to r. Denote the map data for the k-th road segment by m_k, where $k \in \{1, ..., n\}$. There is a directed edge $(m_{k_1} \to m_{k_2})$ in $\mathcal{D}_r$, if there is a vehicle moving from the k_1-th road segment to the k_2-th road segment, which needs to obtain m_{k_2}, given m_{k_1} as prior information. The destination nodes in $\mathcal{D}_r$ (i.e., those with at least one in-coming directed edge) are called the *demanded packets*. Two examples of $\mathcal{D}_r$ for a four-way junction are shown in Fig. 4.3.

The uncoded packets $\{m_1, ..., m_n\}$ are called *source packets*. A packet consists of K source packets combined by bitwise XOR operator is called a K*-ary coded packets*. For example, $m_2 \oplus m_1$ is a binary-coded packet. An index coding scheme, denoted by $\mathcal{I}$, is a set of coded or source packets. For convenience of analysis, we assume that each packet of map data has a uniform size. If packets have different sizes, padding will be used.

Given a set of demanded packets, we aim to construct an optimal index coding scheme using the minimal number of transmitted packets that can be decoded into the required information (i.e., destination nodes in $\mathcal{D}_r$). Note that the construction of a decodable index coding scheme is similar to a generalization of the set cover problem. Each coded packet is a cover, while the demanded packets are items to be covered by some coded packets. The decodability of coded packets requires that a combination of complemented intersections (i.e., XOR operations) of the received coded packets can generate the demanded packets.

Theorem 4.3.1. *Given the demand graph $\mathcal{D}_r$, an optimal index coding scheme $\mathcal{I}$ can be constructed using source packets and binary-coded packets. In partic-*

ular, each demand $(m_{k_1} \rightarrow m_{k_2})$ *in* $\mathcal{D}_r$ *can be decoded by one of the following ways:*

1. *A source packet (i.e.,* $m_{k_2} \in \mathcal{I}$*).*

2. *Or a sequence of connected binary coded packets, say* $\{m_{k_1} \oplus m_{k_r}, m_{k_r} \oplus m_{k_{r-1}}, ..., m_{k_3} \oplus m_{k_2}\} \subseteq \mathcal{I}$*, such that the required packet can be decoded by* m_{k_1} *and such a sequence of binary-coded packets.*

See two examples of optimal index coding schemes in Fig. 4.3, where an arrow represents a demand, and a dashed enclosure represents a coded or source packet.

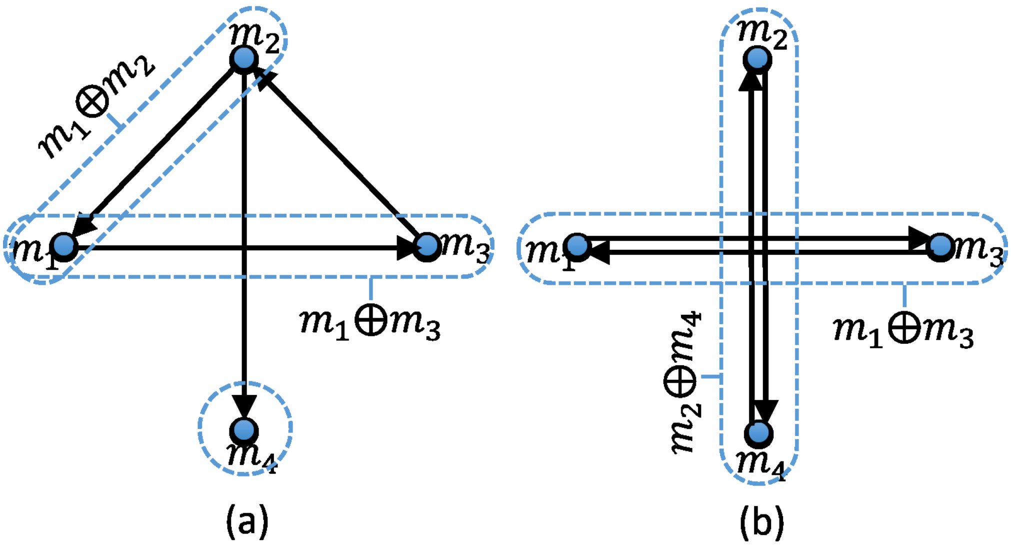

Figure 4.3: Two examples of demand graph $\mathcal{D}_r$ and their optimal index coding schemes for a four-way junction. (a) Four vehicles with directions: $(m_1 \rightarrow m_3)$, $(m_3 \rightarrow m_2)$,$(m_2 \rightarrow m_1)$, $(m_2 \rightarrow m_4)$, and an optimal index coding scheme is $\{m_1 \oplus m_2, m_1 \oplus m_3, m_4\}$. (b) Four vehicles with directions: $(m_1 \rightarrow m_3)$, $(m_3 \rightarrow m_1)$, $(m_2 \rightarrow m_4)$, $(m_4 \rightarrow m_2)$, and an optimal index coding scheme is $\{m_1 \oplus m_3, m_2 \oplus m_4\}$.

By Theorem 4.3.1, it suffices to consider binary-coded packets. We next present a polynomial-time algorithm 1J-IdxCd to identify the optimal index coding scheme, which first adds any demand $(m_{k_1} \rightarrow m_{k_2})$ as a coded packet, and then removes redundant packets in any cycles of coded packets, while ensuring the decodability of demanded packets.

Pseudo-codes 2 $\mathsf{1J\text{-}IdxCd}[\mathcal{D}_r]$

1: $\mathcal{I} \leftarrow \varnothing$
2: **for** $(m_{k_1} \rightarrow m_{k_2}) \in \mathcal{D}_r$ **do**
3:     $\mathcal{I} \leftarrow \mathcal{I} \cup \{m_{k_1} \oplus m_{k_2}\}$ ▷ *Flag* lock_{k_1,k_2} *prevents* $m_{k_1} \oplus m_{k_2}$ *to be removed from* $\mathcal{I}$
4:     $\mathsf{lock}(k_1, k_2) \leftarrow \mathsf{False}$
5: **end for**
6: **while** there exists cycle $\{m_{k_1} \oplus m_{k_2}, m_{k_2} \oplus m_{k_3} \ ..., m_{k_r} \oplus m_{k_1}\} \subseteq \mathcal{I}$ **do**
7:     **for** $m_{k_t} \oplus m_{k_{t+1}} \in \{m_{k_1} \oplus m_{k_2}, m_{k_2} \oplus m_{k_3} \ ..., m_{k_r} \oplus m_{k_1}\}$ **do**
8:         **if** $\mathsf{lock}(k_t, k_{t+1}) = \mathsf{False}$ **then**
9:             $\mathcal{I} \leftarrow \mathcal{I} \backslash \{m_{k_t} \oplus m_{k_{t+1}}\}$
10:             **for** $(m_{k'_1} \rightarrow m_{k'_2}) \in \mathcal{D}_r$ **do**
11:                 **if** there exists no path $\{m_{k'_1} \oplus m_{k'_r}, ..., m_{k'_3} \oplus m_{k'_2}\}$
12:               $\subseteq \mathcal{I} \backslash \{m_{k'_1} \oplus m_{k'_2}\}$ **then**
13:                     $\mathsf{lock}(k'_1, k'_2) \leftarrow \mathsf{True}$
14:                 **end if**
15:             **end for**
16:         **end if**
17:     **end for**
18: **end while**
19: **return** $\mathcal{I}$

Theorem 4.3.2. *Algorithm* $\mathsf{1J\text{-}IdxCd}$ *produces an optimal index coding scheme for a single junction.*

Let us apply Algorithm $\mathsf{1J\text{-}IdxCd}$ to the example in Fig. 4.3(a). The initial index coding scheme is $\mathcal{I} \leftarrow \{(m_1 \oplus m_2), (m_2 \oplus m_3), (m_3 \oplus m_1), (m_2 \oplus m_4)\}$ and their corresponding locks, $\mathsf{lock}(k_1, k_2)$ are set to False. These are defined by lines 2–5. Lines 6–18 remove the redundant and unnecessary coded packets. Coded packet $(m_2 \oplus m_4)$ will be the first to be removed since $\mathsf{lock}(2,4)$ is False and there is no cycle that includes it. On the other hand, the combination of the coded packets $\{(m_1 \oplus m_2), (m_2 \oplus m_3), (m_3 \oplus m_1)\}$ forms a cycle. Any one of the three coded packets can be removed since all their corresponding locks are false. For instance, we remove $(m_2 \oplus m_3)$, then the locks $\mathsf{lock}(1,2)$ and

$\mathsf{lock}(1,3)$ become True. Therefore, Algorithm $\mathsf{1J\text{-}IdxCd}$ returns the final index coding scheme as $\mathcal{I} \leftarrow \{(m_1 \oplus m_2), (m_3 \oplus m_1)\}$.

Note that Algorithm $\mathsf{1J\text{-}IdxCd}$ produces an index coding scheme with coded packets only. $\mathsf{1J\text{-}IdxCd}$ can be improved by replacing some coded packets by source packets, because the source packets are immediately decodable. One can replace any binary-coded packet that is only used to produce just one demanded packet by the corresponding source packet, although the size of $\mathcal{I}$ will remain the same. For the example in Fig. 4.3(a), we add the source packet m_4 to the index coding scheme $\mathcal{I}$, making the optimal index coding scheme as $\mathcal{I} = \{m_1 \oplus m_2, m_1 \oplus m_3, m_4\}$.

The reader is referred to the Appendix in [106] for the proofs of Theorems 1 and 2.

4.3.3 Extension for Multiple Junctions

We next present an extension for multiple junctions. The basic idea is to adopt $\mathsf{1J\text{-}IdxCd}$ as a basis for multiple junctions. We assume that all trip plans, $\{\mathsf{P}^c\}_{c\in\mathcal{C}}$, are given a-priori. We define a *meeting relation graph* among the vehicles by $\mathcal{G}_{\rm meet} = (\mathcal{N}_{\rm meet}, \mathcal{E}_{\rm meet})$, where the set of nodes $\mathcal{N}_{\rm meet}$ are subsets of vehicles ($\subseteq \mathcal{C}$) having intersected trip plans, and the set of directed edges $\mathcal{E}_{\rm meet}$ are the temporal ordering between meetings with a common vehicle. Namely,

$$\mathcal{N}_{\rm meet} \triangleq \Big\{(c_1, ..., c_r) \subseteq \mathcal{C} \mid \exists v \in \bigcap_{i=1}^{r} \mathsf{P}^{c_i} \text{ and } \mathsf{t}_v^{c_1} = ... = \mathsf{t}_v^{c_r}\Big\} \tag{4.7}$$

$$\begin{aligned}\mathcal{E}_{\rm meet} \triangleq \Big\{&(c_1, c_2, ..., c_r) \rightarrow (c_1, c'_2, ..., c'_s) \in \mathcal{N}_{\rm meet} \times \mathcal{N}_{\rm meet} \\ &\mid \exists v_1 \in \bigcap_{i=1}^{r} \mathsf{P}^{c_i} \text{ and } \exists v_2 \in \bigcap_{i=1}^{s} \mathsf{P}^{c'_i} \text{ and } \mathsf{t}_{v_1}^{c_1} < \mathsf{t}_{v_2}^{c_1}\Big\}\end{aligned} \tag{4.8}$$

If two vehicles meet in their trip plans, then there are two cases: (1) traveling in different directions (e.g., meeting at a junction), or (2) traveling along with each other. Case (1) is utilized in index coding to broadcast mixed information (via bitwise XOR) to vehicles, and then the vehicles decode the mixed information using different prior information they received previously. However, there will be no impact by index coding for case (2).

Assumption 1. (**Single Meeting**) We assume that the trip plans of every pair of vehicles intersect at most *once*, namely, if they meet and depart, then they will never meet again. In practice, if the autonomous vehicles always follow the shortest paths and employ deterministic tie-breaking for the paths of equal distance, then the meeting with another autonomous vehicles of different source or destination is only at most once. Otherwise, this will contradict to the property of shortest paths. Since they meet at most once, each meeting event can be uniquely identified as a node in $\mathcal{N}_{\text{meet}}$, and $\mathcal{G}_{\text{meet}}$ is also a directed acyclic graph.

Theorem 4.3.3. *If the single meeting assumption (Assumption 1) holds, then applying Algorithm* 1J-IdxCd *independently at each junction will produce an optimal index coding scheme for multiple junctions.*

The reader is referred to the Appendix in [106] for the proof of Theorem 3. Note that even if the vehicles meet more than once, Theorem 4.3.3 still provides a heuristic to construct a good index coding scheme for multiple junctions with limited meetings among vehicles.

4.4 Fog-based Opportunistic Scheduling of Heterogeneous V2X Networks

The previous section considered the basic setting with static map data and the absence of capacity constraint, under the single meeting assumption. In this section, we present scheduling schemes that decide the transmission options for both static and dynamic map data from the LCD to intelligent vehicles, considering capacity constraint at RSU fog nodes. The scheduling schemes heuristically apply Algorithm 1J-IdxCd at each junction.

4.4.1 Downloading

First, denote the starting and ending time of vehicle c's trip plan by t_c^{s} and t_c^{d} respectively. There are two modes of scheduling:

1. **Offline Mode**: All the trip plans of intelligent vehicles $\{\mathsf{P}^c\}_{c\in\mathcal{C}}$ are known in advance.

2. **Online Mode**: Not all trip plans are known. Only the trip plans of intelligent vehicles started at the current time t_{now} or before (i.e., $\{\mathsf{P}^c \mid t_c^{\mathsf{s}} \leq t_{\mathsf{now}}\}_{c\in\mathcal{C}}$) are known.

As illustrated in Fig. 4.1, the LCD scheduler decides the download operations of map data to individual vehicles according to their GPS locations and trip plans. The map data are first downloaded via nearby RSUs (via short-range broadcast). In case of insufficient capacity at the RSU fog nodes, cellular network transmissions will be utilized.

Recall that static map data is denoted by m_e^{s} and dynamic data by $m_e^{\mathsf{d}}(t)$ for each road segment $e \in \mathcal{E}$. m_e^{s} should be downloaded to vehicle c before or at time t_e^c, and $m_e^{\mathsf{d}}(t)$ should be downloaded to c at some time between $\mathsf{t}_e^c - \tau$ and t_e^c. For each RSU $r \in \mathcal{R}$, let $\mathcal{C}_r(t) = \{c \in \mathcal{C} \mid t = \mathsf{t}_r^c\}$ be the set of vehicles that meet at junction r at time t, and $\mathcal{D}_r(t)$ be the demand graph considering the vehicles in $\mathcal{C}_r(t)$.

Let $x_r^{\mathsf{d}}(t,e)$ and $x_r^{\mathsf{s}}(t,e)$ be the decisions of the scheduler to broadcast static and dynamic data respectively at RSU fog node r, for road segment e at time t.

Similarly, let $y^c_{\mathsf{d}}(t,e)$ and $y^c_{\mathsf{s}}(t,e)$ be the decisions of the scheduler to download static and dynamic data via cellular networks to vehicle c. Let $\mathcal{I}^{\mathsf{s}}_r(t)$ and $\mathcal{I}^{\mathsf{d}}_r(t)$ be the index coding schemes for static and dynamic data respectively at time t. Also, let us denote the single-junction scheme applied to RSU fog node r for static data by $\mathsf{1J\text{-}IdxCd}^{\mathsf{s}}[\mathcal{D}_r(t)]$. Similarly, $\mathsf{1J\text{-}IdxCd}^{\mathsf{d}}[\mathcal{D}_r(t)]$ for the scheme applied to dynamic data in $[t-\tau,t]$ for a given time window τ. We denote the size of the code by $|\cdot|$ (e.g., $|m^{\mathsf{s}}_e|$ and $|\mathsf{1J\text{-}IdxCd}^{\mathsf{s}}[\mathcal{D}_r(t)]|$).

By the single meeting assumption, myopic scheduling of static data at the respective junction in an on-demand manner is optimal. For dynamic data, the latest information is always more useful. Hence, myopic scheduling is also desirable. However, in the presence of capacity constraint, it may not be possible to schedule all required transmissions in an on-demand manner. In this case, we have to greedily pick a subset of vehicles at each junction to maximize the efficiency of transmissions. Formally, given a demand graph $\mathcal{D}_r(t) = (\mathcal{N}[\mathcal{D}_r(t)], \mathcal{E}[\mathcal{D}_r(t)])$, we define subgraph $\mathcal{H}_r = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is a subset of $\mathcal{N}[\mathcal{D}_r(t)]$ and $\mathcal{E}$ is the induced subset of edges of $\mathcal{E}[\mathcal{D}_r(t)]$. Let $\mathcal{N}_{\rm src}[\mathcal{D}_r(t)]$ be the set of source nodes (i.e., nodes with at least one in-coming directed edge).

For such a subgraph $\mathcal{H}_r$, we define $W(\mathcal{H}_r)$ as the number of vehicles that can be satisfied by performing index coding on $\mathcal{H}_r$. We aim to find the best subgraph $\mathcal{H}_r$ that maximizes $W(\mathcal{H}_r)$ subject to the capacity constraint $\mathsf{1J\text{-}IdxCd}^{\mathsf{s}}[\mathcal{H}_r] \leq \mathrm{C}^{\downarrow}_r$. Since the number of roads connecting a junction is small, this process can be performed efficiently. For each demand packet that cannot be accommodated by local broadcast, the scheduler will download it via the cellular network.

First, a greedy online opportunistic scheduling scheme is presented in Algorithm $\mathsf{ONLSchd}$, which schedules the local broadcast transmissions at RSU based on the arrival of vehicles in an online manner. Static map data will be scheduled before dynamic map data. If there is insufficient capacity at the RSU fog nodes, then the scheduler will download the remaining map data via the cellular network.

The greedy offline opportunistic scheduling scheme is presented in Algorithm OFLSchd. At each RSU fog node, optimal single-junction index coding is employed, considering all autonomous vehicles that approach the junction at the current time. If there is any spare capacity at RSUs, the scheduler will download the undelivered static map data in advance at any RSU fog nodes with spare capacity. Finally, undelivered map data will be downloaded via the cellular network.

Pseudo-codes 3 $\mathsf{ONLSchd}[\mathcal{D}_r(t_{\mathsf{now}})_{r\in\mathcal{R}}]$

for $r \in \mathcal{R}$ **do**
  $\mathsf{c}_r^{\downarrow}(t_{\mathsf{now}}) \leftarrow \mathrm{C}_r^{\downarrow}$
    ▷ **Initialize RSU r download capacity**
    ▷ *Download static map data via local broadcast by index coding*
  $\mathcal{H}_r \leftarrow \mathrm{argmax}_{\mathcal{H}} W(\mathcal{H})$
  subject to $\mathsf{1J\text{-}IdxCd}^{\mathsf{s}}[\mathcal{H}] \le \mathsf{c}_r^{\downarrow}(t_{\mathsf{now}})$ and $\mathcal{H}$ is subgraph of $\mathcal{D}_r(t_{\mathsf{now}})$
    ▷ **Get max # of vehicles whose demands can be satisfied**
  $\mathcal{I}_r^{\mathsf{s}}(t_{\mathsf{now}}) \leftarrow \mathsf{1J\text{-}IdxCd}^{\mathsf{s}}[\mathcal{H}_r]$
    ▷ **Perform Optimal Index Coding on $\mathcal{H}_r$**
  $\mathsf{c}_r^{\downarrow}(t_{\mathsf{now}}) \leftarrow \mathsf{c}_r^{\downarrow}(t_{\mathsf{now}}) - |\mathcal{I}_r^{\mathsf{s}}(t_{\mathsf{now}})|$
    ▷ **Update RSU r download capacity**
  for $c \in \mathcal{C}_r(t_{\mathsf{now}}), e \in \mathsf{P}^c \cap \mathcal{E}_r$ **do**
    if $m_e^{\mathsf{s}} \in \mathcal{N}_{\mathrm{src}}[\mathcal{H}_r]$ **then**
      $x_r^{\mathsf{s}}(t_{\mathsf{now}}, e) \leftarrow 1$
        ▷ **Download static map data via local broadcast**
    else
      $y_{\mathsf{s}}^{c}(t_{\mathsf{now}}, e) \leftarrow 1$
        ▷ **Download undelivered static map data via cellular networks**
    end if
  end for
    ▷ *Download dynamic map data via local broadcast by index coding*
  $\mathcal{H}'_r \leftarrow \mathrm{argmax}_{\mathcal{H}'} W(\mathcal{H}')$
  subject to $\mathsf{1J\text{-}IdxCd}^{\mathsf{d}}[\mathcal{H}'] \le \mathsf{c}_r^{\downarrow}(t_{\mathsf{now}})$ and $\mathcal{H}'$ is subgraph of $\mathcal{D}_r(t_{\mathsf{now}})$
    ▷ **Get max # of vehicles whose demands can be satisfied**
  $\mathcal{I}_r^{\mathsf{d}}(t_{\mathsf{now}}) \leftarrow \mathsf{1J\text{-}IdxCd}^{\mathsf{d}}[\mathcal{H}'_r]$
    ▷ **Perform Optimal Index Coding on $\mathcal{H}'_r$**
  $\mathsf{c}_r^{\downarrow}(t_{\mathsf{now}}) \leftarrow \mathsf{c}_r^{\downarrow}(t_{\mathsf{now}}) - |\mathcal{I}_r^{\mathsf{d}}(t_{\mathsf{now}})|$
    ▷ **Update RSU r download capacity**
  for $c \in \mathcal{C}_r(t_{\mathsf{now}}), e \in \mathsf{P}^c \cap \mathcal{E}_r$ **do**
    if $m_e^{\mathsf{d}} \in \mathcal{N}_{\mathrm{src}}[\mathcal{H}'_r]$ **then**
      $x_r^{\mathsf{d}}(t_{\mathsf{now}}, e) \leftarrow 1$
        ▷ **Download static map data via local broadcast**
    else
      $y_{\mathsf{d}}^{c}(t_{\mathsf{now}}, e) \leftarrow 1$
        ▷ **Download undelivered dynamic map data via cellular networks**
    end if
  end for
end for

Pseudo-codes 4 $\mathsf{OFLSchd}[\mathcal{D}_r(t)_{t\in\mathcal{T},r\in\mathcal{R}}]$

for $t \in \mathcal{T}, r \in \mathcal{R}$ **do**
  $\mathsf{c}_r^{\downarrow}(t) \leftarrow \mathrm{C}_r^{\downarrow}$
  ▷ **Initialize RSU r download capacity** ▷ *Download static map data via local broadcast by index coding*
  $\mathcal{H}_r \leftarrow \operatorname{argmax}_{\mathcal{H}} W(\mathcal{H})$
  subject to $\mathsf{1J\text{-}IdxCd}^{\mathsf{s}}[\mathcal{H}] \leq \mathsf{c}_r^{\downarrow}(t_{\mathsf{now}})$ and $\mathcal{H}$ is subgraph of $\mathcal{D}_r(t_{\mathsf{now}})$
  ▷ **Get max # of vehicles whose demands can be satisfied**
  $\mathcal{I}_r^{\mathsf{s}}(t) \leftarrow \mathsf{1J\text{-}IdxCd}^{\mathsf{s}}[\mathcal{H}_r]$
  ▷ **Perform Optimal Index Coding on $\mathcal{H}_r$**
  $\mathsf{c}_r^{\downarrow}(t) \leftarrow \mathsf{c}_r^{\downarrow}(t) - |\mathcal{I}_r^{\mathsf{s}}(t)|$
  ▷ **Update RSU r download capacity**
  for $c \in \mathcal{C}_r(t), e \in \mathsf{P}^c \cap \mathcal{E}_r$ **do**
    if $m_e^{\mathsf{s}} \in \mathcal{N}_{\mathrm{src}}[\mathcal{H}_r]$ **then**
      $x_r^{\mathsf{s}}(t,e) \leftarrow 1$
      ▷ **Download static map data via local broadcast**
    end if
  end for
  ▷ *Download dynamic map data via local broadcast by index coding*
  $\mathcal{H}'_r \leftarrow \operatorname{argmax}_{\mathcal{H}'} W(\mathcal{H}')$
  subject to $\mathsf{1J\text{-}IdxCd}^{\mathsf{d}}[\mathcal{H}'] \leq \mathsf{c}_r^{\downarrow}(t_{\mathsf{now}})$ and $\mathcal{H}'$ is subgraph of $\mathcal{D}_r(t_{\mathsf{now}})$
  ▷ **Get max # of vehicles whose demands can be satisfied**
  $\mathcal{I}_r^{\mathsf{d}}(t) \leftarrow \mathsf{1J\text{-}IdxCd}^{\mathsf{d}}[\mathcal{H}'_r]$
  ▷ **Perform Optimal Index Coding on $\mathcal{H}'_r$**
  $\mathsf{c}_r^{\downarrow}(t) \leftarrow \mathsf{c}_r^{\downarrow}(t) - |\mathcal{I}_r^{\mathsf{d}}(t)|$
  ▷ **Update RSU r download capacity**
  for $c \in \mathcal{C}_r(t), e \in \mathsf{P}^c \cap \mathcal{E}_r$ **do**
    if $m_e^{\mathsf{d}} \in \mathcal{N}_{\mathrm{src}}[\mathcal{H}'_r]$ **then**
      $x_r^{\mathsf{d}}(t,e) \leftarrow 1$
      ▷ **Download dynamic map data via local broadcast**
    end if
  end for
end for
▷ *Download static map data via local broadcast in advance, if sufficient capacity*
for $t \in \mathcal{T}, c \in \mathcal{C}, r \in \mathsf{P}^c$ **do**
  if $\exists e \in \mathsf{P}^c$ and $\exists r \in \mathsf{P}^c$ and $\mathsf{t}_r^c < \mathsf{t}_e^c$ and $\prod_{t' \leq \mathsf{t}_e^c}(1 - x_r^{\mathsf{s}}(t',e)) = 0$ and $\mathsf{c}_r^{\downarrow}(\mathsf{t}_r^c) \geq |m_e^{\mathsf{s}}|$ **then**
    $x_r^{\mathsf{s}}(\mathsf{t}_r^c, e) \leftarrow 1$
    ▷ **Download static map data via local broadcast**
    $\mathcal{I}_r^{\mathsf{s}}(\mathsf{t}_r^c) \leftarrow \mathcal{I}_r^{\mathsf{s}}(\mathsf{t}_r^c) \cup \{m_e^{\mathsf{s}}\}$
    ▷ **Advanced static map data to be downloaded**
    $\mathsf{c}_r^{\downarrow}(t) \leftarrow \mathsf{c}_r^{\downarrow}(t) - |m_e^{\mathsf{s}}|$
    ▷ **Update RSU r download capacity**
  end if
end for
▷ *Download undelivered map data via cellular networks*
for $t \in \mathcal{T}, r \in \mathcal{R}$ **do**
  for $c \in \mathcal{C}_r(t), e \in \mathsf{P}^c \cap \mathcal{E}_r$ **do**
    if $x_r^{\mathsf{s}}(t,e) \neq 1$ **then**
      $y_{\mathsf{s}}^c(t,e) \leftarrow 1$
      ▷ **Download undelivered static map data via cellular networks**
    end if
    if $x_r^{\mathsf{d}}(t,e) \neq 1$ **then**
      $y_{\mathsf{d}}^c(t,e) \leftarrow 1$
      ▷ **Download undelivered dynamic map data via cellular networks**
    end if
  end for
end for

4.5 Uploading 3D LIDAR Point Cloud Data

This section focuses on the discussion of 3D LIDAR point cloud data, and a common representation called Octree. We present differential coding and hashing schemes especially for uploading 3D LIDAR point cloud data. Since vehicles individually upload their sensed data to the RSU fog node, differentiation is done at the vehicular level to reduce bandwidth consumption and redundant information to be sent.

4.5.1 Octree Representation

3D point cloud depicts objects and surfaces as a set of 3D points in the Cartesian coordinate system within a bounded region [107]. A common approach to encode 3D point cloud is using *Octree*, by which the 3D space is recursively partitioned into 8 cells (*voxels*) and a binary number is used to indicate the presence of an object in each cell. See an illustration of Octree representation of 3D point cloud in Fig. 4.4.

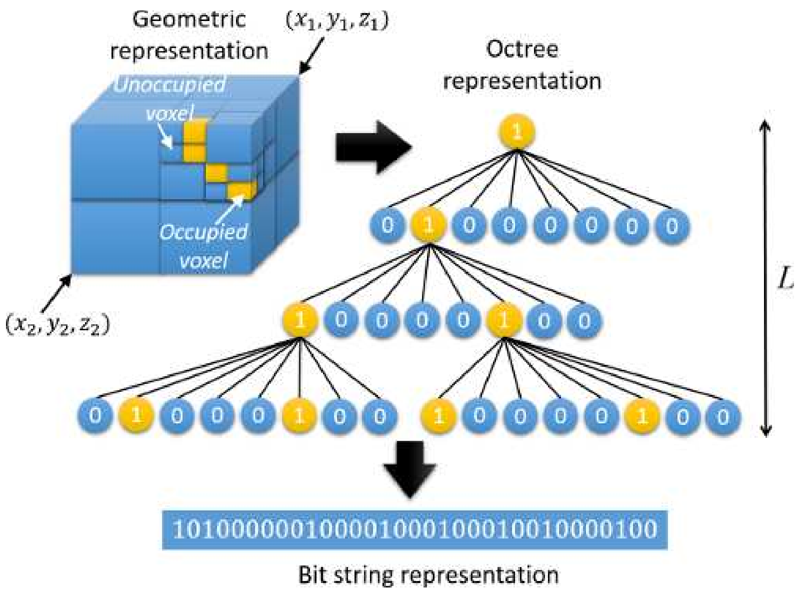

Figure 4.4: An illustration of Octree representation of 3D point cloud.

Octree is a tree-based data structure suitable for sparse 3D point data, where each node represents a cell or volume element (voxel). From the root, it is iteratively divided into eight children until a certain depth or level L is achieved [108] or if there is no more 3D point cloud to be partitioned. An occupied voxel contains a point or a set of points, and is labeled by '1', otherwise by '0'. A node labeled by '1' can be further decomposed into eight more child nodes, whereas there is no need to expand a node labeled by '0'. Accordingly, the larger the depth (i.e., higher value of L), the higher the resolution of the 3D object.

Two reference corners for the boundary of region of an Octree are denoted by (x_1, y_1, z_1) and (x_2, y_2, z_2) (see Fig. 4.4). One can represent an Octree by a bit string representation that encodes its contents by a fixed traversal order in the voxels of each layer. We can apply further coding schemes on the bit string representation. Note that different LIDAR sensors may produce different sets of 3D LIDAR point cloud data on the same objects in the region because of different sensing specifications. But the Octree representations can approximate closely with each other, under a suitable value of L. Hence, it is possible to compare different sets of 3D LIDAR point cloud data in Octree representations.

There are several proposals for point cloud compression [109, 110]. These techniques can be applied to our system, but note that they are mainly for storage and are not optimized for communication systems.

4.5.2 Differentiation and Differential Coding

Autonomous vehicles can identify and upload the necessary dynamic map data to LCD using differentiation. Since dynamic map data is only detectable at the moment of departing from a road segment, the upload transmissions take place immediately through the nearby RSU fog node (in short-range broadcast), whenever possible. Otherwise, cellular network transmissions are employed.

Differentiation is particularly useful for identifying the dynamic components in 3D LIDAR point cloud data. We denote the differentiated data between observed point cloud $x_c(t)$ and reference point cloud $m_e(t-1)$ by:

$$\mathsf{Diff}_c(t) = \Big(x_c(t)\backslash m_e(t-1)\Big) \cup \Big(m_e(t-1)\backslash x_c(t)\Big) \tag{4.9}$$

where $t = \mathsf{t}_e^c$ and $e \in P_c$.

To encode the differentiated data, we employ *differential coding* on Octree. Octree allows efficient identification of the differences by enumerating the voxels along the tree. Once the differences are identified, we can employ another Octree to encode the differentiated parts. However, the meanings of voxels are now different: '0' means no difference with respect to the reference 3D LIDAR point cloud data, whereas '1' means the binary content in the respective voxel should be flipped. See an illustration in Fig. 4.5.

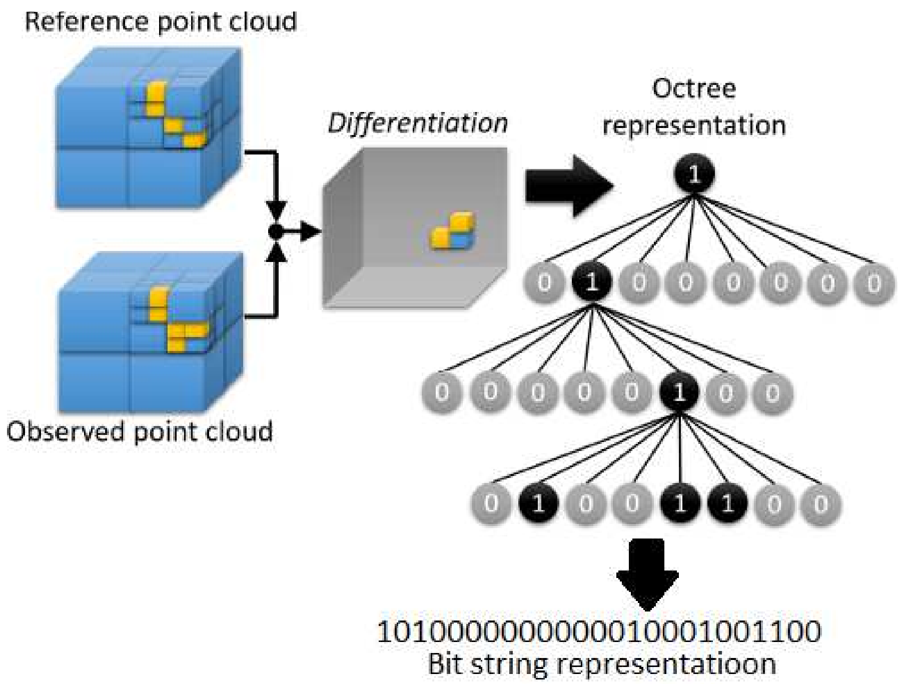

Figure 4.5: An illustration of differential coding on 3D point cloud.

4.5.3 Hashing 3D LIDAR Data

Comparison through the hash files associated with 3D LIDAR point cloud data is more efficient than using the whole data set. The hash files should have certain desirable properties. For example, one can compare two hash files to identify which point cloud data consists of more contents (e.g., more observed objects). Second, one can check if the point cloud data contains certain known objects, without looking at the whole data set. A simple solution is to use a Bloom filter [111], a compact lossy data structure representing the membership of a set of elements. The basic operations of a Bloom filter involve adding an element to the set and querying the membership of an element. It does not support element removal, therefore, upon query of an element membership, the Bloom filter output may only result in false positives, which can be minimized through parameter setting. In our system, each vehicle first communicates with RSU and LCD using Bloom filters before uploading the whole perception data.

Recall that $x_c(t) = \{p_1, p_2, ...\}$ is a set of 3D points. Note that each p_i has a unique octary representation, such that each digit in the octary representation represents the order of the respective occupied voxel at each layer in Octree. We denote index '0' to represent the first voxel. For example, the four 3D points in the Octree of Fig. 4.4 can be represented in octary representation as $\{101, 105, 150, 155\}$. Next, we map each point in octary representation by a set of K binary hash functions: $f_k(p_i) \mapsto \{0, 1\}$, where $k = 1, ..., K$. Let $f_k(x_c(t)) = f_k(p_1) \vee f_k(p_2) \vee ...$ be the bitwise disjunction of all the points in $x_c(t) = \{p_1, p_2, ...\}$. The K output bits $\left(f_k(x_c(t))\right)_{k=1}^{K}$ will be a Bloom filter for $x_c(t)$, denoted by $\mathsf{BF}(x_c(t))$. Bloom filters have some desirable properties. If a 3D point cloud has more contents, then its Bloom filter contains more 1's. One can check if a 3D point cloud contains a set of known 3D points, by checking if its Bloom filter contains the corresponding hash values.

4.6 Robotic Testbed Evaluation

We implemented the single junction scenario and evaluated our proposed system in a practical testbed. In this set-up, as depicted in Fig. 4.6, two cases are studied:

- *Scenario 1*: Car A on Road A intends to turn into Road B with Car B. There is no time-sensitive data.
- *Scenario 2*: Similar to scenario 1, but there is a moving object in front of Car B on Road B.

The robotic vehicles used in the testbed are shown in Fig. 4.6(a). They represent the intelligent vehicles equipped with suite of sensors, including a Kinect camera and LIDAR for capturing its environment's 3D point cloud data and proximity sensors for collision detection. The experimental set-up is shown in Fig. 4.6(b). The cardboard boxes represent buildings (see Fig. 4.6). Typical Kinect image data file size is approximately 10 MB. The 3D point cloud data is compressed to a 1 cm^3 resolution to achieve at least a 60% compression rate before being transmitted to the RSU fog node. Such Octree resolution offers a significant compression rate while maintaining an accurate representation of the sensed environment. The RSU fog node and robotic vehicles exchange information by using the IEEE 802.11 standard (WiFi).

In Scenario 1, at every five seconds, both vehicles captured their respective environment in form of 3D point cloud data, and performed Octree compression. The data are then transmitted to the RSU fog node along with their requests of road map data. Upon reception, the RSU fog node performs the encoding $m_A \oplus m_B$, where m_i is the map data for road segment i, and broadcasts

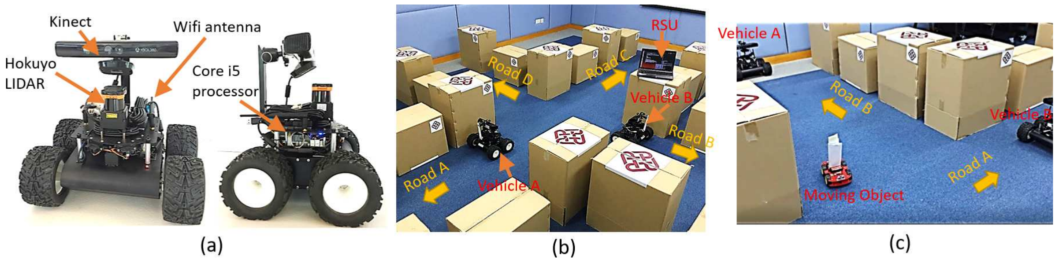

Figure 4.6: (a) Robotic vehicles in (b) a single 4-way road junction scenario. (c) A moving object is introduced on Road B

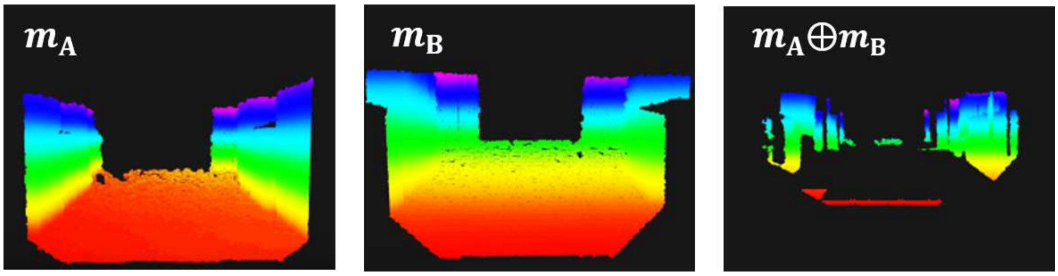

Figure 4.7: The 3D point cloud road map data captured by (a) car A, (b) car B and (c) the XOR-ed result of maps A and B.

the encoded packets. The 3D point cloud data perceived by individual vehicle and the corresponding encoded 3D point cloud data are shown in Fig. 4.7. After receiving the encoded packets ($m_A \oplus m_B$), car A decodes it via ($m_A \oplus m_B$) $\oplus$ m_A to obtain its desired information regarding road segment B. Car B does the same to acquire information regarding road segment A. Since both road segments have no obstacles detected, each vehicle immediately turns to its desired road without the need of reducing its speed.

In Scenario 2, a small programmable mobile robot is added in front of car B to introduce a dynamic object to the environment. Such set-up is depicted in Fig. 4.6(b). In order to detect obstacles that present on the road, we integrated a map filter for object extraction after the decoding process, and we search for the blocked information on the ground to determine the location of the object. Fig. 4.6(c) illustrates the detected dynamic data by car A after map filtering and object detection. From the gathered information, car A reduces its speed and waits until the small robot moves past the junction before turning into

Road B.

In summary, our robotic testbed manages to achieve cooperative autonomous driving through transmitting road map data between the two robot vehicles. It also experimentally demonstrates that an efficient 3D road map data dissemination based on the proposed index coding scheme is feasible in practice, especially when dealing with moving dynamic objects on road.

However, the following limitations may be observed from the robotic testbed and can be considered in future experiments. First, due to the lack of available mobile robots to represent intelligent connected vehicles, the efficiency of the index coding-based opportunistic download scheme has not been thoroughly tested in extensive scenarios, e.g., where there are various vehicles approaching the junction at varying speeds. This has also limited the testing to a single junction experiment. The vehicular environment also depicted a less dynamic testing situation and covered only the simplest situation. Second, since there were only two intelligent connected vehicles in our setup, the uploading of 3D LIDAR point cloud data from the vehicles to the RSU is always successful. The RSU is also unburdened in the collection and storage of the uploaded vehicular data. Lastly, in determining the performance of the proposed opportunistic upload and download schemes, throughput, delay, and processing time should be measured.

4.7 Simulation Studies

In the previous section, we have demonstrated the feasibility of employing index coding in the dissemination of road segment data to nearby vehicles at a road junction. In this section, we present further evaluation of our proposed system by simulation studies using real-world 3D point cloud data of city streets and GPS mobility traces of vehicles. We consider both scenarios of single and multiple road junctions for analyzing the effectiveness of the proposed schemes.

4.7.1 Local Broadcast by Index Coding for Single Junction

First, we consider the single-junction scenario. The simulation set-up is described as follows.

Simulation Set-up

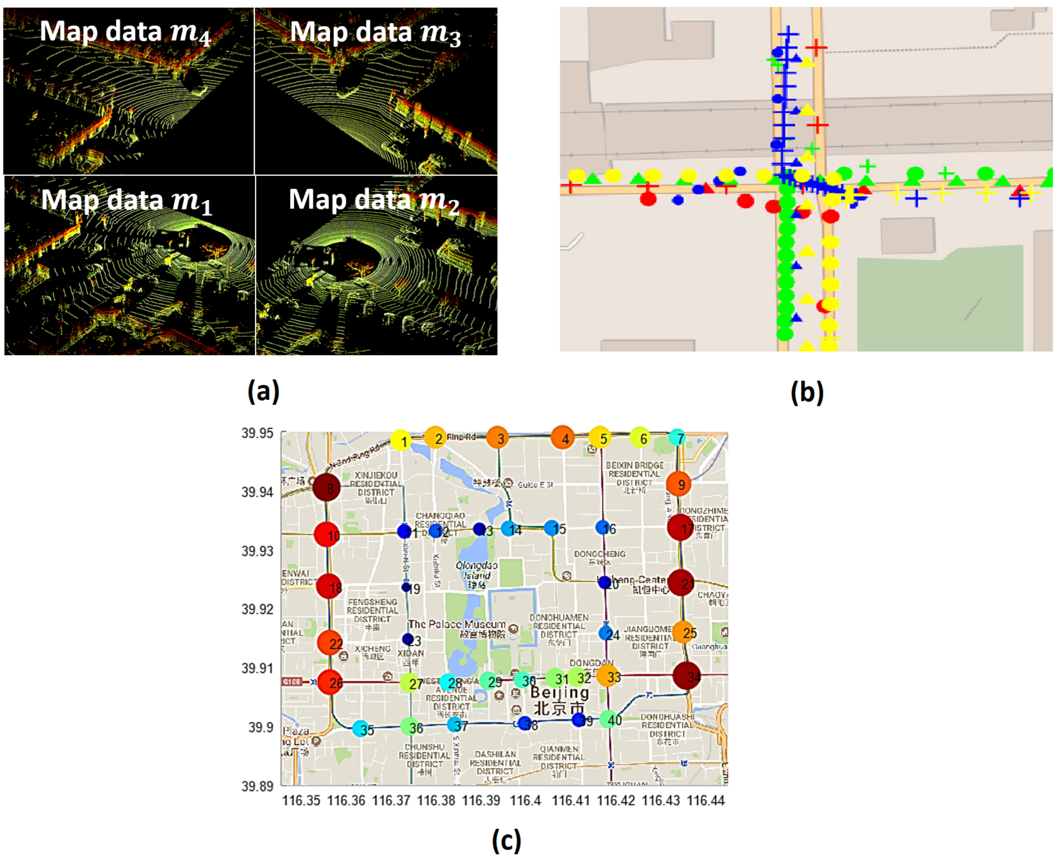

Figure 4.8: (a) Partitioned 3D point cloud map data of a real-world junction. (b) Empirical GPS mobility traces for a particular single junction in Beijing City. (c) Average mobility patterns of 40 selected junctions in Beijing City.

To study the performance on realistic 3D point cloud data, we consider the

3D point cloud static map data of a real-world junction depicted in Fig. 4.8 (a), which is obtained by Ford Research campus in downtown Dearborn, Michigan [11]. It is partitioned into four separate views as perceived by the vehicles in each road segment connecting to the junction. In the dissemination process, the 3D point cloud data is compressed using Octree compression [109]. The sizes of compressed 3D point cloud data packets of each road segment and binary-coded packets are shown in Table 4.1.

Table 4.1: Sizes of compressed 3D point cloud data for the static map data shown in Fig. 9 (a).

Map Data	Number of Points	Data Size (MB)
m_1	104,255	5.838
m_2	95,537	5.254
m_3	69,200	2.763
m_4	73,168	3.184
$m_1 \oplus m_2$	63,607	2.126
$m_1 \oplus m_3$	65,920	2.812
$m_1 \oplus m_4$	61,738	2.630
$m_2 \oplus m_3$	67,806	2.631
$m_2 \oplus m_4$	66,072	2.363
$m_3 \oplus m_4$	64,025	2.126

To incorporate realistic vehicle mobility patterns, we consider the dataset of Beijing taxi GPS mobility traces [112] to simulate the mobility traces of autonomous vehicles at a junction. The Beijing taxi dataset contains seven days of GPS mobility traces (including longitude and latitude positions), timestamps of recorded positions, and vehicle IDs of 28,590 taxis traveling in Beijing City. Beijing City resembles a grid network geographically, consisting of mostly four-way junctions. In particular, we consider the junction between the *East 3rd Ring Road Middle* and *Jianguo Road*, as shown in Fig. 4.8(b). There are 8,663 taxis on average traversing it daily. Fig. 4.8(b) depicts the empirical

GPS mobility traces of 12 taxis. We assume that the RSU fog node is deployed near the junction center with a transmission range of 200 meters.

Evaluation of Download Operations

To perform the download operations, a RSU fog node r first scans the nearby vehicles in every sampling time T_S. Once the vehicles reach within the proximity of r, it determines the vehicles' map data demands and constructs the demand graph $\mathcal{D}_r$. Next, RSU fog node r applies 1J-IdxCd$[\mathcal{D}_r]$ to perform local broadcast based on index coding.

To study the performance of 1J-IdxCd$[\mathcal{D}_r]$, we consider two benchmarks:

1. Random Broadcast (Rand): It broadcasts all source packets in a random fashion.

2. Index Coding with Prior Information (1J-IdxCd-PI): It explores the scenario that some vehicles may have extra prior knowledge of a certain road segment. For example, a particular road segment is popular among all vehicles. The map data is likely to be pre-downloaded to the vehicles in advance.

The evaluation results are depicted in Fig. 4.9, which shows the daily total number of transmissions and sizes of transmitted packets for Rand and 1J-IdxCd for seven days based on GPS mobility traces. The sizes of each transmitted packets are set according to Table 4.1.

It is observed that 1J-IdxCd can effectively reduce the total number of transmissions by around 500 transmissions less when compared to the benchmark Rand. For downloading static data, the benchmark requires a number of 7.75 transmissions on average to satisfy all vehicles' demands as compared to 1J-IdxCd that requires only 5.94 transmissions on average. The average daily

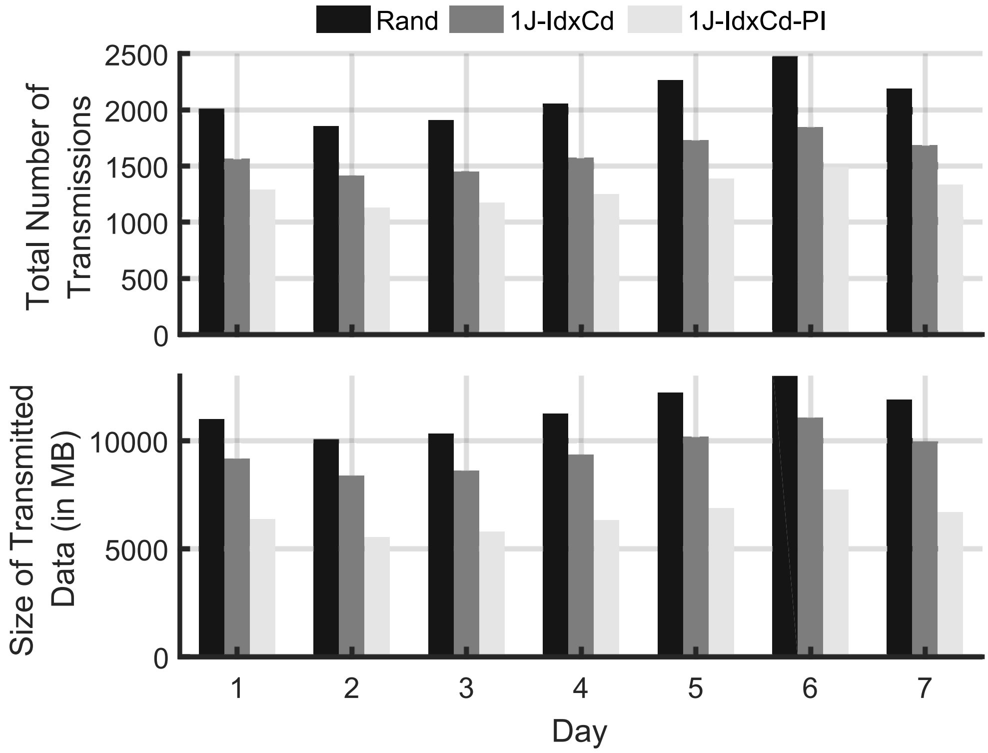

Figure 4.9: Daily total number of transmissions and sizes of transmitted packets for Rand, 1J-IdxCd, 1J-IdxCd-PI.

sizes of transmitted data for random transmission is 12.18 GB while that for 1J-IdxCd is only 10.24 GB. 1J-IdxCd transmits 6.00 MB on average within a period of T_{S}, while Rand transmits 5.46 MB on average. Overall, employing 1J-IdxCd enables higher data rate with the fewest number of transmissions.

Next, we evaluate the effectiveness with extra prior information. 1J-IdxCd-PI considers extra prior information for road segment 2. In this case, 1J-IdxCd-PI preforms like 1J-IdxCd, but it assumes every vehicle already has map data of road segment 2, and performs index coding incorporating such prior information. We observe that the availability of extra prior information considerably reduces the number of transmissions, and thus the required bandwidth, transmitting 5.01 MB on average by 4.49 transmissions.

Usefulness of Sharing Dynamic Data

In this section, we study the usefulness of sharing dynamic data among vehicles. Vehicles equipped with perception sensors can capture the dynamic objects and detect time-varying information in road environments. Hence, sharing such information can assist the autonomous driving decisions. One way to characterize the availability of dynamic data is by examining the frequency of passing-by vehicles. The more frequent are the passing-by vehicles, the more time-sensitive information can be detected and captured.

On the other hand, the usefulness of dynamic data also depends on the demands from other vehicles. Consider the example of map data demands at a junction in Fig. 4.3(a). Road segments 1, 2, and 3 have captured dynamic data for the vehicles coming to the road segments, because there are vehicles departing from these road segments. However, there is no captured dynamic data of road segment 4 from other vehicles . In this example, there are total number of demands for dynamic data is 4 (for each road segment), while the number of demands can be met by captured dynamic data from other vehicles is only 3.

Let TN_r be the total number of demands for dynamic data at a junction r, and SN_r be the number of satisfied demands for captured dynamic data from other vehicles. The ratio $\frac{\mathsf{SN}_r}{\mathsf{TN}_r}$ allows us to characterize the usefulness of sharing information among vehicles. Hence, for the scenario in Fig. 4.3(a), $\frac{\mathsf{SN}_r}{\mathsf{TN}_r}$=75% . For the effective sharing of dynamic data among vehicles, the ratio $\frac{\mathsf{SN}_r}{\mathsf{TN}_r}$ should be high.

We empirically evaluated the quantities SN_r and TN_r for the junction in Fig. 4.8(b) from real-world GPS mobility traces over seven days. The results are plotted in Fig. 4.10. We observe that the aggregate ratio $\frac{\mathsf{SN}_r}{\mathsf{TN}_r}$ for one day is more than 80%. Hence, sharing dynamic data can benefit individual vehicles

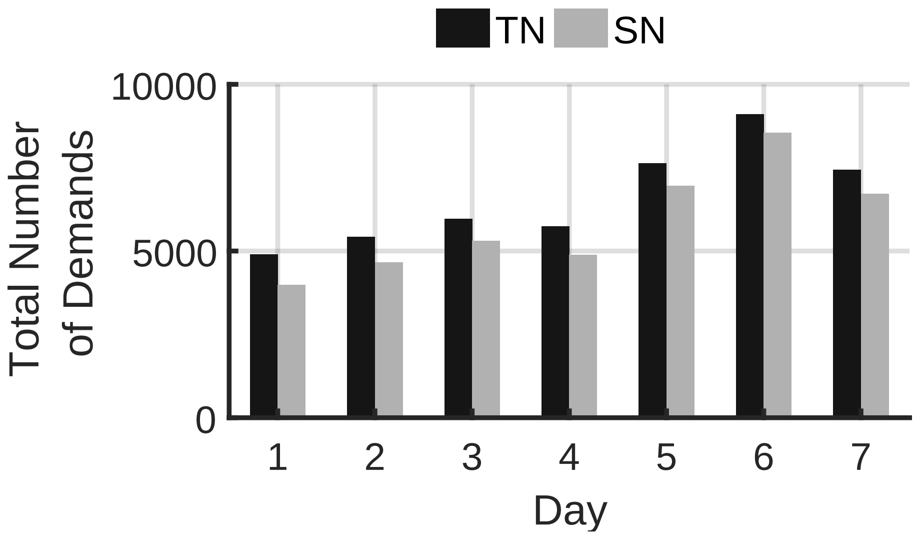

Figure 4.10: TN_r (the number of demands for dynamic data at the junction in Fig. 4.8(b)) and corresponding SN_r (the number of satisfied demands for captured dynamic data from other vehicles).

significantly in real-world mobility patterns.

4.7.2 Applying Index Coding to Multiple Junctions

After evaluating the performance of single-junction index coding, we consider index coding for multiple junctions.

Simulation Set-up

We selected 40 junctions in Beijing, as depicted in Fig. 4.8(c), and use the corresponding GPS mobility traces of taxis traversing these junctions to simulate the mobility patterns. The simulation parameters are listed in Table 4.2. In Fig. 4.8(c), we visualize the average mobility patterns of these 40 junctions by circles of different sizes. The bigger the circle, the more number of taxis traversed the respective junction.

Table 4.2: Simulation Parameters for Multiple Junctions in Fig. 4.8(c).

Simulation Attribute/Parameter	**Value**
Total area (in $\approx$ km^2)	50
Number of observed days	7
Number of road segments per junction	4
Number of RSUs	40
RSU transmission range (meters)	200
Total number of taxis	24,845
Daily average number of taxi trips	79,012
Hourly average number of taxis in each junction	466
Total number of recorded time each day (hrs)	24
Sampling time of GPS traces (mins)	2

Evaluation Results

Fig. 4.11 depicts the average number of transmissions and sizes of transmitted data when 1J-IdxCd is applied independently at the 40 junctions. Note that the number of visits is not directly proportional to the average number of transmissions. In particular, RSU fog nodes 19 and 24 have relatively low volume of visits, whereas RSU fog nodes 9 and 25 have a relatively high number. However, RSU fog node 19 has fewer number of transmissions than RSU fog node 24. This is because the vehicles in RSU 19 arrive more regularly than those at RSU 24, hence more significant performance gain can be found in terms of the number of transmissions and sizes of transmitted data by 1J-IdxCd. A similar phenomenon is observed at the high-volume RSU fog node 25.

Fig. 4.12 shows the RSU fog nodes located on *West 2nd Ring Road* (i.e., RSUs 8, 10, 18, 22, 26) and depicts the hourly performance of each RSU. The average volume of visits through these road sections are similar. The curve labeled by 'Avg' indicates the average value of the 40 RSU fog nodes over a day. We observe that the information dissemination by the RSU fog nodes increased starting from 08:00h, because the peak traffic hours occur at 08:00h.

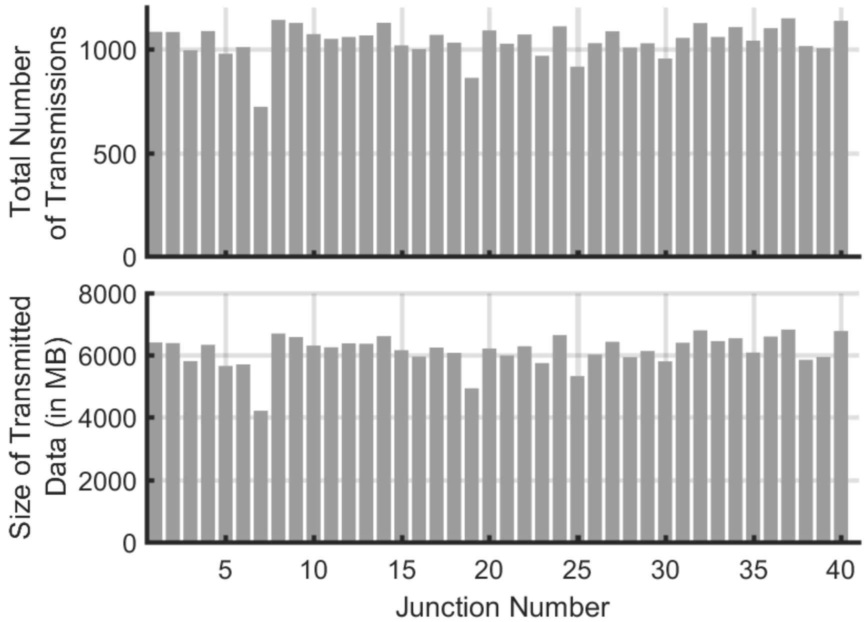

Figure 4.11: Average total number of transmissions and sizes of transmitted data for each of the 40 RSU fog nodes situated in Beijing.

From midnight to 06:00h, the traffic is relatively low.

4.7.3 Scheduling over Multiple Junctions

In this section, we evaluate the performance of the proposed scheduling schemes over the selected 40 junctions. We employ both the online (ONLSchd) and offline (OFLSchd) opportunistic scheduling schemes for disseminating map data to vehicles.

Applying the Opportunistic Scheduling

We assign each RSU fog node the same download capacity ($C^{\downarrow}$). The performance of various scheduling schemes are shown in Fig. 4.13, under various download capacity ($C^{\downarrow}$) based on the GPS mobility traces of taxis traversing the 40 junctions in Beijing in Fig. 4.8(c).

We observe that the RSU fog nodes reach the download capacity at a much

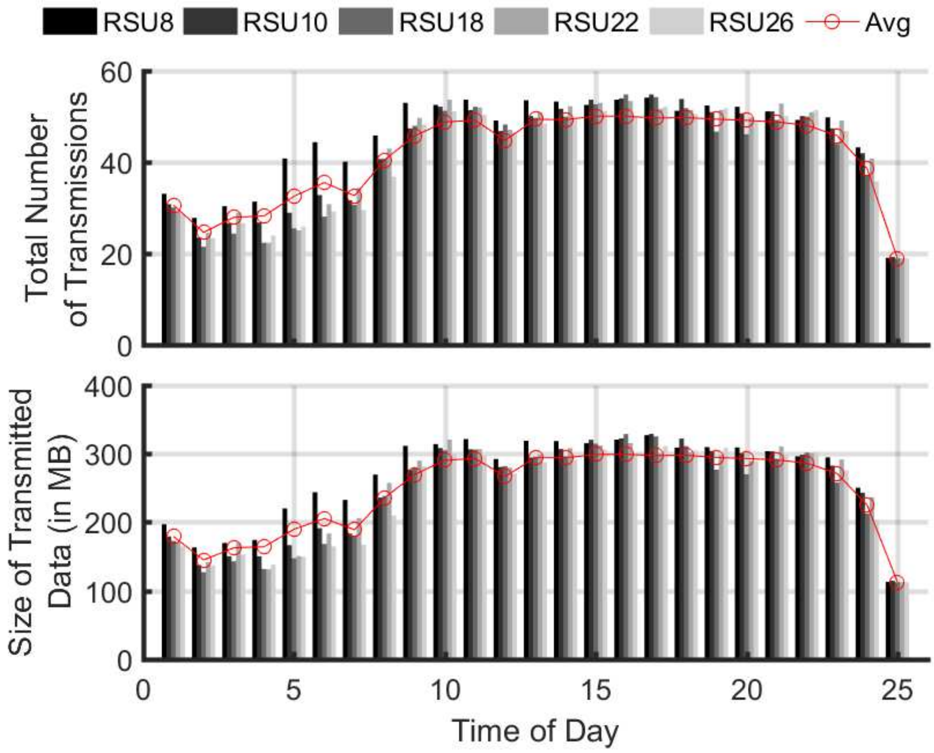

Figure 4.12: Hourly average total number of transmissions and sizes of transmitted data for each RSU fog nodes located on *West 2nd Ring Road*.

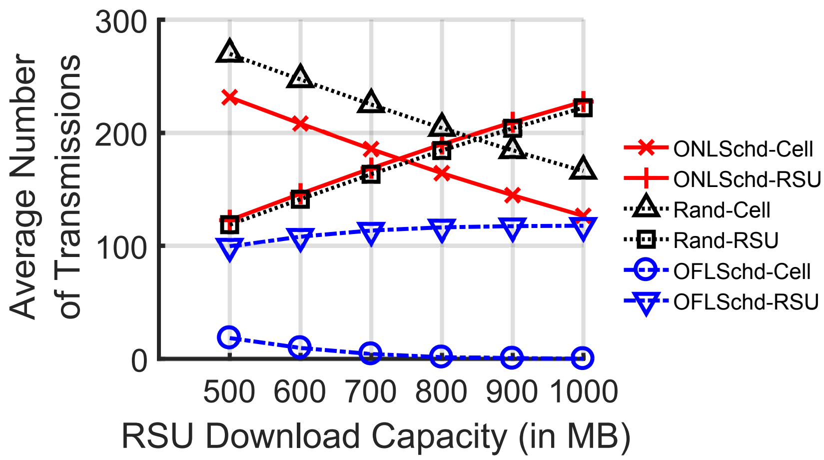

Figure 4.13: Performance of various scheduling schemes against download capacity ($C^{\downarrow}$).

faster rate by Rand, because Rand broadcasts a large amount of data (equal to the sum of all source packets of map data in a trip per vehicle), as compared to the opportunistic scheduling schemes per sampling time. This leads to heavier load on the cellular network when Rand is used in scenarios with low download capacity ($C^{\downarrow} \leq 700$ MB). Although ONLSchd reduces the required cellular transmissions, it is still evident that it exhibits the same effect during low download capacity, i.e., more cellular transmissions than local broadcast transmissions. For a given set of map data, increasing the local broadcast download capacity can reduce the need for cellular network transmissions. This presents a design trade-off for the network administrator to balance the loads between local broadcast and cellular unicast.

Among the three schemes, OFLSchd employs considerably less cellular network bandwidth even when the local broadcast download capacity is low. It relies almost totally on local broadcast transmissions as the download capacity is over 700 MB. This is because all of the vehicles' trip plans are known in advance, thus, enabling the RSU fog nodes to schedule map data dissemination more efficiently, and less rely on the cellular download of road map data.

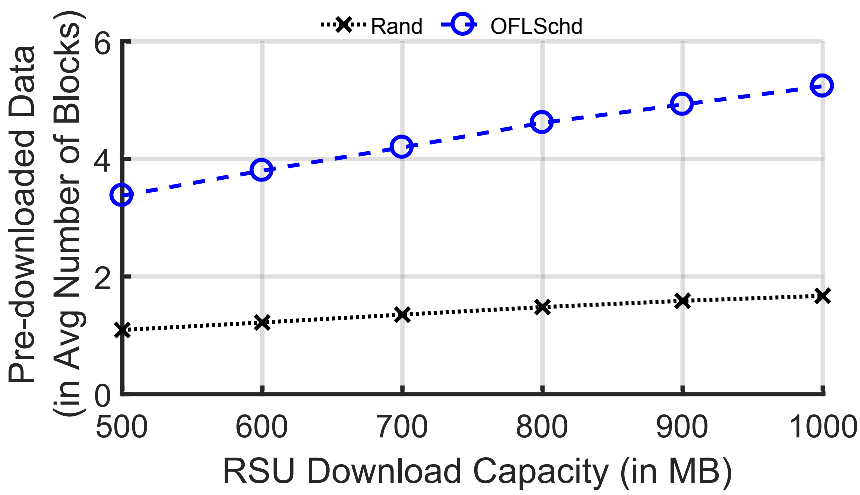

Figure 4.14: Comparing the OFLSchd and Rand methods on the distance of pre-downloaded remote data to requesting vehicles.

Fig. 4.14 shows the distance (in terms of the number of blocks away) of pre-downloaded remote data to a requesting vehicle at an RSU fog node. Since OFLSchd knows the planned trips of the vehicles, it reduces the number of transmissions required to satisfy all vehicles, while providing pre-downloaded remote data up to five blocks away (when the RSU fog node download capacity is $\geq$ 900 MB). For example, RSU fog node 14 can transmit road segment data from remote RSU fog nodes such as 1, 10, 17, 32, etc. Such amount of advanced data will allow a vehicle to update its planned trip and alter its route if necessary. On the other hand, Rand can only deliver data up to an average of 1.66 RSU blocks away from the requesting RSU fog node.

Meeting Frequency of Vehicles

Theorem 4.3.3 shows that applying Algorithm 1J-IdxCd independently at each junction produces an optimal index coding scheme for multiple junctions, under the single meeting assumption. In this section, we empirically examine the meeting frequency of vehicles based on GPS mobility traces.

We define that a meeting occurs when the taxis' routes (according to GPS traces) are within 200 m of each other. Given an observation window during the day, these frequencies of meetings are stored in the adjacency matrix **TM**, defined below in (4.10):

$$\mathbf{TM} = \begin{bmatrix} 0 & x_{1,2} & \cdots & x_{1,n-1} & x_{1,n} \\ x_{2,1} & 0 & \cdots & \vdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{n-1,1} & \cdots & \cdots & \ddots & x_{n-1,n} \\ x_{n,1} & x_{n,2} & \cdots & x_{n,n-1} & 0 \end{bmatrix} \tag{4.10}$$

where $x_{i,j} \in \{0, 1, 2, ...\}$ is the number of times that vehicles i and j met and $x_{i,j} = x_{j,i}$. The observation window is equal to $t_{\text{stop}} - t_{\text{start}}$. In this case, $t_{\text{stop}} - t_{\text{start}} = 24$ hours, having a 2-min sampling interval.

The results are illustrated in Fig. 4.15. Any two given taxis from the mobility traces only met each other once 86% of the time, while the remaining 14% met more than once. From Fig. 4.15, we know that any pair of taxis met at most once within a moderate time window. This shows limited meetings among vehicles with different trips in a city in practice. Hence, applying Algorithm 1J-IdxCd independently at each junction still provides a heuristic to construct a good index coding scheme for a city-wide multiple junction scenario.

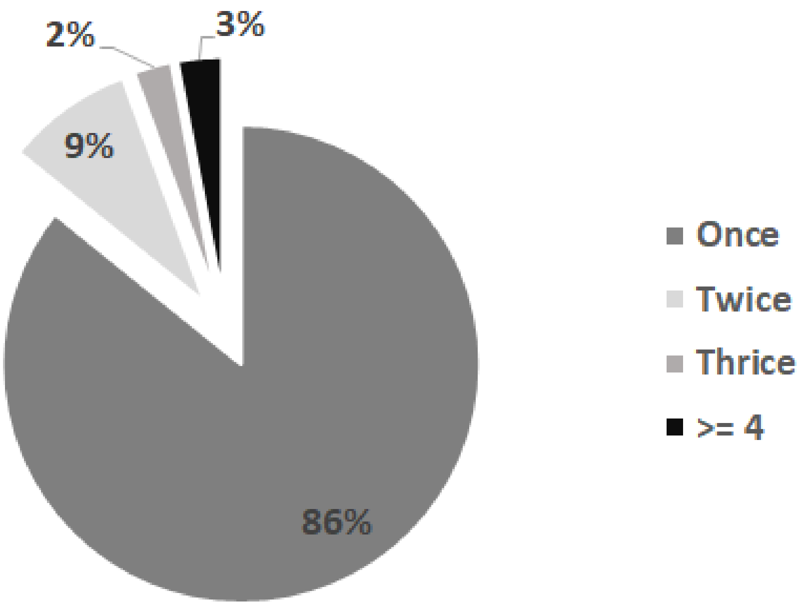

Figure 4.15: Meeting frequency between any pair of taxis.

Shown in Table 4.3 are the frequencies of taxi meetings employing other empirical datasets expressed in percentage.We can notice that any taxis meeting once in a 24-hour window is still very high ($\geq$ 80%), with the exemption of the Rome dataset with only 61.52%. However, this result for the Rome dataset is still above 50% and can still be considered that reliance of intelligent connected vehicles for up-to-date road map data is still well-served by the RSU fog nodes.

Finally, we study the scenarios that the majority of the number of meetings in the adjacency matrix **TM** is more than one (i.e., $x_{i,j} \geq 2$). Note

Table 4.3: Frequency of taxi meetings (in %).

Taxi Mobility Trace		Meeting Frequency (in %)			
City (Location)	Number of Vehicles	1	2	3	≥ 4
Beijing [112]	28590	85.80	8.62	2.57	3.01
Shanghai [113]	2300	99.65	0.35	0.00	0.00
San Francisco [114]	536	98.35	1.56	0.09	0.00
Shenzhen [115]	350	80.16	18.73	0.93	0.19
Rome [116]	320	61.52	19.98	8.52	9.98
Singapore [117]	28000	76.94	19.64	0.98	2.45
Jakarta [117]	28000	79.56	18.20	1.89	0.35
New York* [118]	23930	91.58	7.55	0.74	0.13
*considering Pick-up and Drop-off points only					

that the proposed scheme produces the best performance when $x_{i,j} = 1$. The simulation results for 3,000 runs are shown in Fig. 4.16. We also compare the performance with a benchmark scheme, OnDemand [119], which transmits the source packets of the most demanded road segment first, and then transmits binary-coded packets of the most demanded road segment data until all demands are satisfied.

We observe that as the number of meetings between any pair of vehicles increases, the total number of transmissions increases in all scheduling schemes (see the top figure of Fig. 4.16). We note that both Rand and OnDemand, having equal number of transmissions, increase at a higher rate than 1J-IdxCd. However, OnDemand transmits fewer data than Rand to satisfy all requesting vehicles. Overall, considering both performance metrics, 1J-IdxCd outperforms the two benchmarks in the multiple junction scenario.

4.7.4 Processing Overhead Analysis

We analyze in this subsection the processing overheads of the proposed index coding algorithm based on the overall data dissemination delay (including

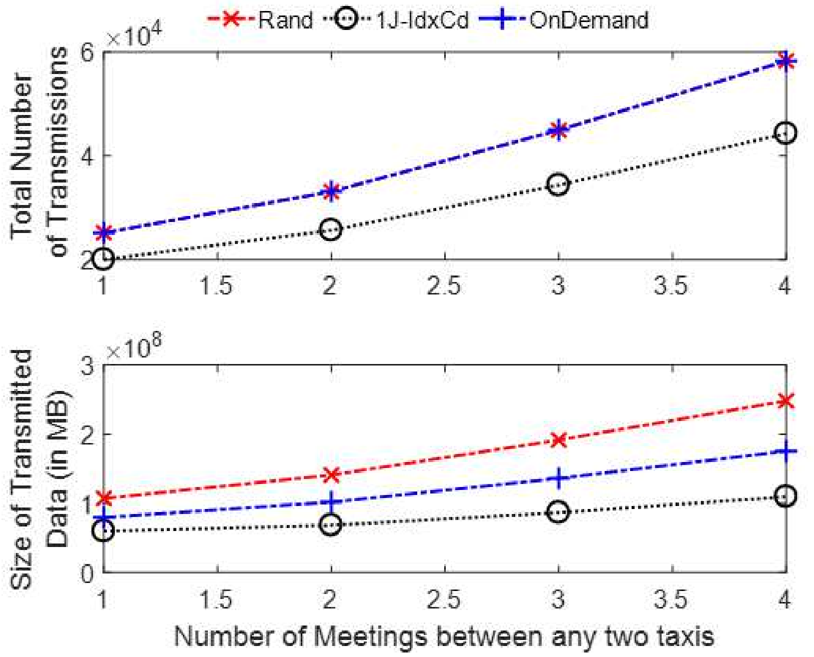

Figure 4.16: Comparison of the three schemes in terms of the total number of transmissions (top) and total sizes of transmitted data (bottom) against the meeting frequencies between any pair of vehicles.

both the processing/encoding delay and transmission delay) from the RSU to the nearby vehicles. For the Rand method, the overall delay only contains the transmission delay, while the 1J-IdxCd scheme includes also the processing delay due to the XOR encoding of relevant road map data, which is proportional to the number of encoded packets generated. To compute for the transmission delay, we assume that the packet size is 1024 bytes and the data rate is 6 Mbps. On the other hand, the encoding processing delay is assumed to be fixed. For a given RSU fog node, the overall delay is computed every sampling time T_S = 2 min.

Given an assumed processing delay of 1 ms, Fig. 4.17 illustrates the overall delay averaged over seven days for RSUs 8, 10, 18, 22, and 26. These five RSU fog nodes have a daily average of 8,100 taxis passing through, and there are 11–12 taxis connected to each RSU per T_S on average. We can observe that even if there is an additional processing time introduced by the 1J-IdxCd method, its daily average overall delay is still less than that of the Rand method by about 34%. This is because the 1J-IdxCd scheme has a much shorter transmission delay than the Rand method by reducing the total number of required packets and the number of road segments at each intersection is limited.

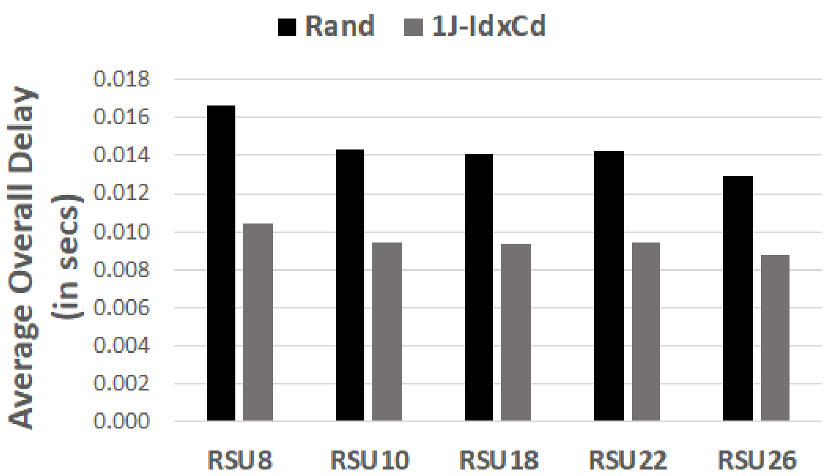

Figure 4.17: Average overall delay of each RSU fog node under the two transmission schemes.

4.8 Summary

In this chapter, we have presented an efficient information dissemination system of 3D point cloud road map data (3D-MADS) for intelligent vehicles and roadside infrastructure integrated in a vehicular fog computing architecture. Our system minimized the amount of cellular network unicast while maximizing the utility of short-range local broadcast transmissions by implementing fog-based opportunistic schedulers. We have also optimized the performance of 3D point cloud data dissemination and update by utilizing techniques such as index coding at roadside unit fog nodes and hashing of 3D point cloud data at vehicular nodes. The overall system was validated with empirical mobility traces, 3D LIDAR data, and an experimental multi-robotic testbed.

Chapter 5

Roadside Unit Allocation for Fog-based Information Sharing in Vehicular Networks

In this chapter, we consider a set of intersections in Beijing City as potential locations for strategically allocating fog computing hotspots to maximize the information shared among vehicles and fog nodes. Using empirical findings from mobility traces such as vehicular density, the total daily number of transmissions, transmitted data size, and space mean speed, we propose the Information Sharing via Roadside unit Allocation (ISRA) strategy to determine the optimal locations for deploying RSU fog computing nodes. The developed optimal index coding transmission scheme from Chapter 4 will be used by ISRA to locate information-rich and energy-efficient intersections. Then, as an application of ISRA, the problem in Chapter 3 is re-visited and extended to include highway sections before the tollgate in the discussion. ISRA-based deployment method has been implemented to manage the density and speed of vehicles in highway sections before the tollgate section.

The chapter is organized as follows: Section 1 presents the urban citywide scenario and assumptions. Section 2 discusses the proposed Information Sharing via Roadside Unit Allocation (ISRA) strategy for allocating RSU fog nodes in any of the city's intersections. Section 3 shows the various performance metrics to evaluate ISRA. Finally, Section 4 concludes this research study and provides future research undertakings.

5.1 Urban City-Wide Scenario

Fig. 5.1 illustrates a section of a city urban grid where there are six junctions and vehicles traversing the city roads. Each colored vehicle corresponds to a specific vehicular density, at a specific sampling time T_S, occupying the road segments connected to the junction. These vehicles have on-board sensors to measure their current surroundings. Road intersections (according to [95]) are the most probable places for situating RSU fog nodes to be used for information dissemination/exchange. The instantaneous V2I contact density (number of transmitting vehicles within the RSU fog node's transmission range) is described in Table 5.1. It is observed that candidate RSU fog node locations r_1 and r_6 have an increasing number of transmitting vehicles, while r_2 and r_5 experience the opposite. r_4 always has nearby transmitting vehicles, and r_3 has time intervals with and without transmitting cars.

Table 5.1: V2I contacts at each candidate RSU fog node location in Fig. 5.1, sampled at each sampling time.

r_j/T_S	t_0	t_1	t_2
r_1	0	20	30
r_2	30	10	0
r_3	20	0	40
r_4	20	30	40
r_5	20	10	0
r_6	0	10	30

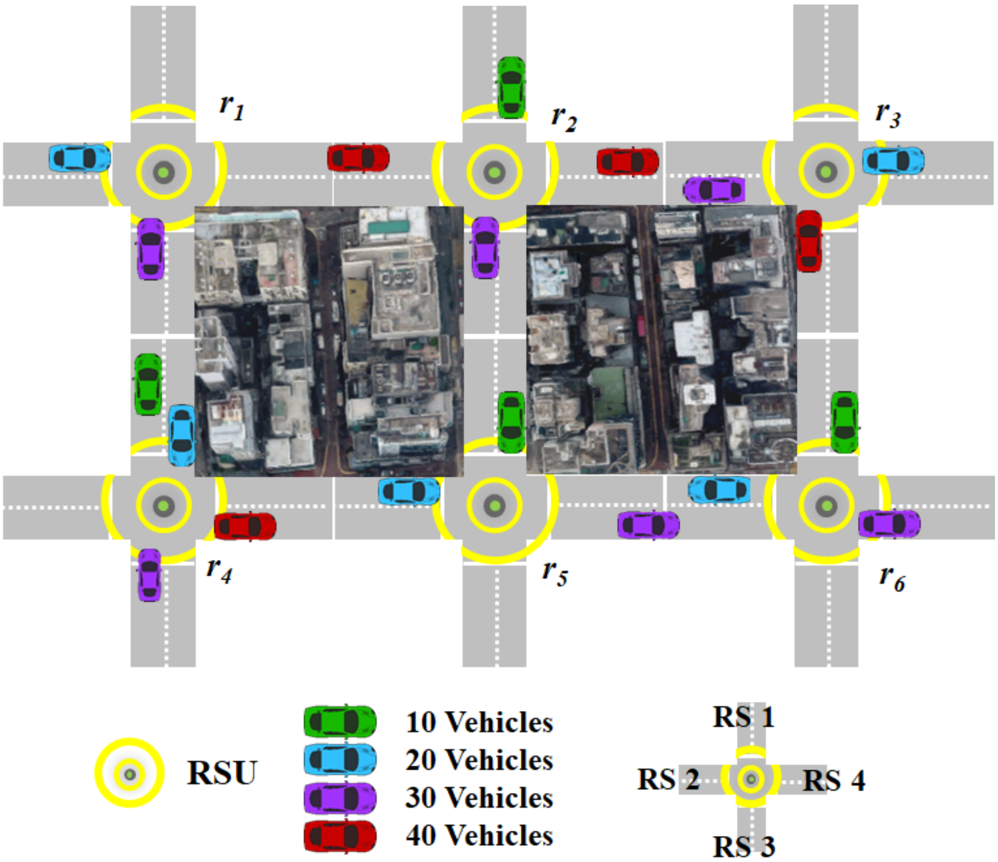

Figure 5.1: Section of a city-wide scenario having six junctions with various V2I and I2V contact densities.

We consider all junctions as candidate locations for allocating RSU fog nodes, and each junction has four road segments. The deployed RSU fog nodes are assumed to be identical, located at the junction's center, and have a transmission range of T_x. Every RSU fog node (r_j) samples its surroundings within every sampling period T_S to check if there are nearby vehicles ready for information exchange. For r_j's with V2I contacts at each sampling time T_S, vehicles send their measurements to the RSU fog node following a scheduled procedure, i.e., according to their arrival. They also send their instantaneous position, speed, and road segment measurement request to the RSU fog node.

After reception, the RSU fog node does the following:

1. From the instantaneous speed readings, it computes the region's space

mean speed to monitor if there are slow moving vehicles staying on the road for a long time [120].

2. From the vehicular demands, it encodes required road segment measurements for packet transmission.

3. It broadcasts encoded packets to satisfy all vehicular requests.

The measured environment data to be shared, prior to being sent out by the vehicles to the RSU fog nodes and vice versa, are already compressed, e.g., via Octree compression [108].

There are two modes of broadcast transmission employed by the RSU fog node: 1) Random transmission (RandTrans), and Optimal Index Coding transmission (OptTrans) [106]. RandTrans sends out data from the most to the least demanded road segment information, or based on a uniform distribution if there are equal number of requests of a particular road segment. On the other hand, the OptTrans scheme sends out either the source or encoded packets to satisfy all vehicular demands. Such transmission mode is also able to reduce the file size of the information to be sent, and the number of packet transmissions needed.

Consider sampling time $T_S = t_2$ and road junction r_3. Assume that there are 10 vehicles on each of the four road segments (RS) carrying RS maps m_1, m_2, m_3, and m_4 respectively. The vehicles on RS 1 request information of RS 2, vehicles on RS 2 require RS 3 information, RS 3 vehicles want measurements of RS 4, and vehicles found on RS 4 demand the information of RS 1. Based on the RandTrans method, the RSU fog node transmits vehicular requests randomly based on a uniform distribution. In OptTrans, the RSU fog node encodes and sends out the following: $m_1 \oplus m_2, m_2 \oplus m_3$, and $m_3 \oplus m_4$. The $\oplus$ symbol is the exclusive OR (XOR) operator. Vehicles on RS 1, upon receiving $m_1 \oplus m_2$, will perform $m_1 \oplus (m_1 \oplus m_2)$ to recover its desired road segment measurement m_2. This is also done by the other vehicles located on the other road segments to obtain their desired road segment environment information.

5.2 Information Sharing via Roadside Unit Allocation

Transmitted compressed road segment measurements from the surrounding vehicles, such as $m_1, \ldots, m_4$ are called source information, while $(m_1 \oplus m_2)$, $\ldots$, $(m_3 \oplus m_4)$ are labeled as encoded information. Source information will be transmitted by both vehicles and RSU fog nodes, but encoded information will only be transmitted by the RSU fog nodes.

The amount of information shared (I_{sh}) in the vehicular network depends on the amount of information transmitted by the vehicles (I_v) and the RSU fog nodes (I_j). Therefore, I_v and I_j indicate the amount of V2I and I2V information shared, respectively. This is denoted by

$$I_{sh} = \sum_{v \in V} I_v + \sum_{j \in J} \alpha_j \beta_j I_j \tag{5.1}$$

In (5.1), the scheduled uploading of V2I information takes place before the downloading of I2V information. With this scenario, road information at the RSU is first updated by the vehicles, and then the RSU updates the other surrounding vehicles. This information exchange happens within the designated sampling period T_S.

The information sent by the RSU fog node (I_j), depending on the mode of transmission, is either a source packet or an index-coded packet. A transmitted packet has a uniform size and contains one road segment measurement ($\beta_j = 1$), e.g., m_1, or two road segment measurements ($\beta_j = 2$), e.g., $m_1 \oplus m_2$. α_j

denotes the number of vehicles in contact with the RSU fog node that received the broadcasted information. We assume all vehicles in contact with the RSU, at a specific sampling time, will share information. Therefore, the amount of information shared increases with the number of V2I contacts within a transmission period.

We propose an Information Sharing via Roadside Unit Allocation (ISRA) strategy to optimally determine the RSU locations that will maximize the amount of information shared between vehicles and RSU fog nodes subject to various constraints. The maximization problem is given by (6.4a) subject to constraints (6.4b), (5.2c), and (6.4c).

$$\text{maximize} \quad I_{sh} \tag{5.2a}$$

$$\text{subject to} \quad \sum_{j=1}^{J} x_j \leq R, \; x_j \in \{0,1\} \tag{5.2b}$$

$$v(x_j) \geq \tau_s, \; \forall j \text{ s.t. } x_j = 1 \tag{5.2c}$$

$$\frac{N_{T_j}}{VC_{tot_j}} \leq \tau_p, \; \forall j \text{ s.t. } x_j = 1 \tag{5.2d}$$

In constraint (6.4b), R is the maximum number of roadside units to be deployed. If a candidate intersection r_j is chosen, $x_j = 1$, otherwise, $x_j = 0$.

In constraint (5.2c), the function $v(x_j)$ computes the RSU r_j ($\forall x_j = 1$) region's space mean speed derived from the surrounding instantaneous vehicular speeds, v_t. Based on the computed value, the junction that has a space mean speed equal to or over a threshold speed limit, τ_s, is selected. This is important especially when there are too many cars willing to share information to the RSU fog node, and the RSU fog node's memory capacity or computational power is exceeded. In case a chosen candidate location runs out of computational power or storage, vehicles in the vicinity will be able to transfer to other RSU fog nodes for information dissemination. If the space mean speed is above the threshold, then, it means that the vehicle is able to reach the next RSU for information exchange. This constraint, thus, effectively balances the communication load of each selected RSU.

Finally, constraint (6.4c) discriminates selected intersections based on a junction's transmission density threshold τ_p. Transmission density is defined as the total number of transmissions needed by RSU r_j (N_{T_j}) to satisfy a set of demands by a given number of V2I contacts (VC_{tot_j}) per sampling period. This constraint restricts the energy consumption for delivering information to the vehicles.

A higher(lower) threshold value for constraint (5.2c)((6.4c)) dramatically reduces the number of possible locations where the hotspots can be allocated.

5.3 Simulation Studies and Discussion

In this section, we present how useful information from empirical taxi mobility traces are extracted and used by ISRA to allocate hotspots for maximum information sharing between vehicles and RSU fog nodes. We use realistic 3D point cloud data found in [121] to represent the static information of all road segments of a junction shared by the vehicles and RSU fog nodes.

5.3.1 Empirical Findings from Mobility Traces

We investigated a seven-day dataset of mobility traces of 28,590 taxis plying the City of Beijing. The dataset contains the taxi's ID number, location's GPS coordinates, and its timestamp [112].

We studied 40 junctions from the first two inner rings of Beijing City and these are shown in Fig. 5.2. The separation between two adjacent road intersections is at least 400 m. The light and small colored circles (colors ap-

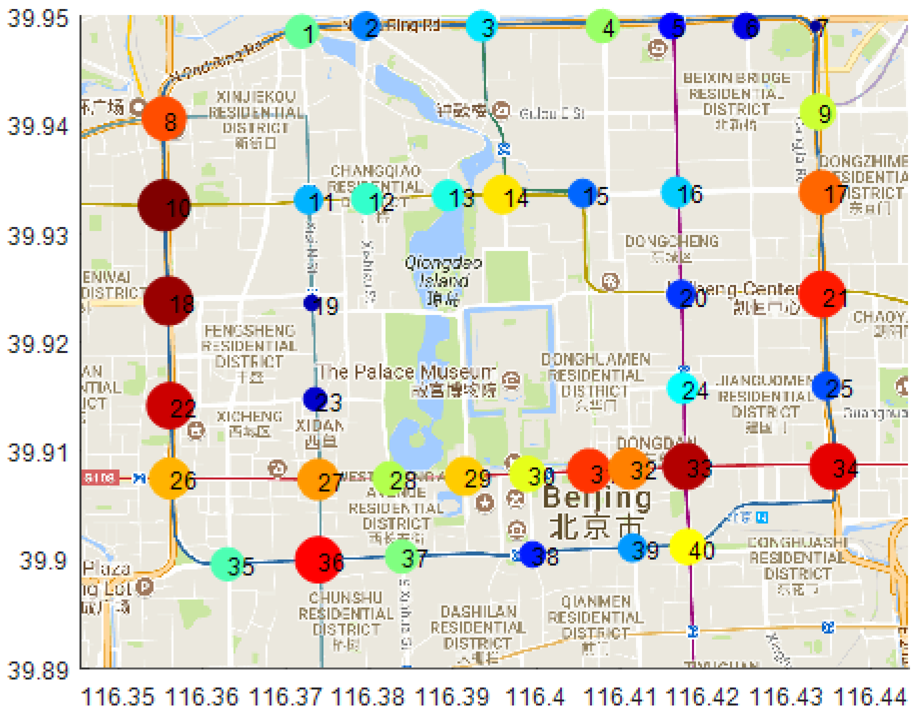

Figure 5.2: The 40 candidate RSU fog node allocations. The color and size of the circle highlights the average V2I contact density at each junction.

proaching blue) depict a low volume of V2I contacts, while large dark colored circles represent the opposite. The V2I contacts of each RSU r_j, VC_j, are sampled every $T_S = 2$ min with the transmission range T_x set to be 200 m. $j = 1, 2, \ldots, 40$. The numerical values of the V2I contacts of each junction are seen in the top portion of Fig. 5.3.

To compute for the junction's space mean speed, we first calculated the instantaneous speed, v_t by:

$$v_t = \frac{GPS_t - GPS_{t-1}}{t_s} \tag{5.3}$$

where GPS_t and GPS_{t-1} are the vehicle's current and previous GPS locations, respectively, converted to distance using the Haversine formula [122]. Each GPS update is taken every $t_s = 10$ s. The space mean speed for each junction, using v_t, is computed according to [120] and are shown in the bottom part of Fig. 5.3. It is noticeable that the daily space mean speed value for each considered junction is below the normal speed limits for most urban roads. This observation highlights that there is mostly traffic congestion throughout the day, even if there are fewer vehicles on the road, e.g., junction 23.

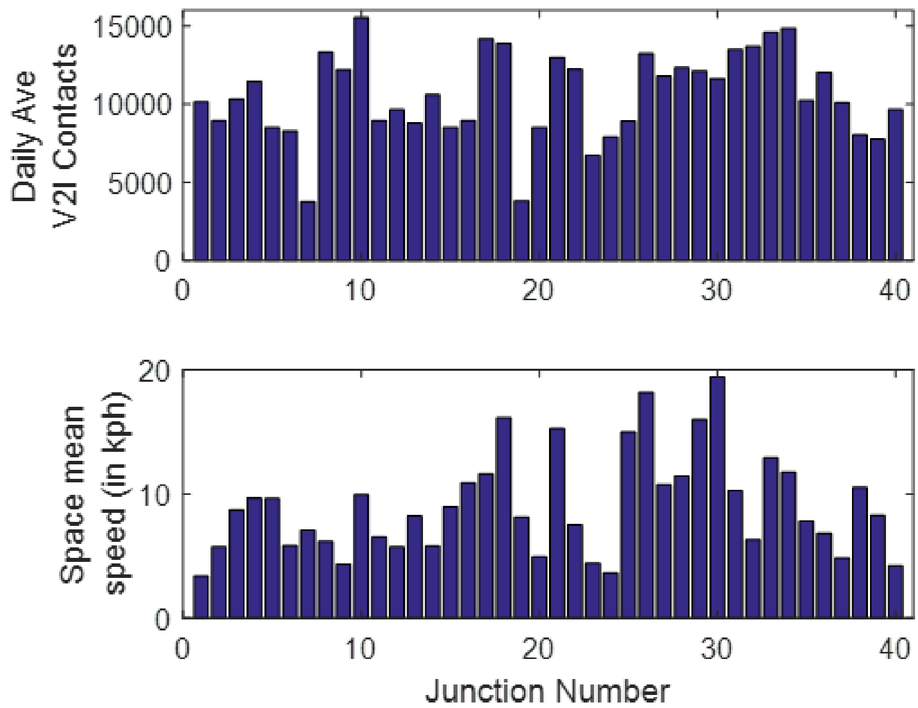

Figure 5.3: Empirical findings for the chosen 40 possible roadside unit locations around the first and second rings of Beijing City averaged over the 7-day period.

To obtain the candidate RSU fog nodes r_j's total number of transmitted packets (NT_j), the OptTrans scheme is employed [106] and compared to the RandTrans. The transmitted packet size is 1024 bytes, where 1000 bytes correspond to the payload and the remaining 24 bytes are for the overhead. The benefit of using OptTrans over RandTrans is illustrated in Fig. 5.4. It is evident that the optimal index coding transmission scheme guarantees that the total number of packet transmissions and transmitted data size are minimized while satisfying the demands of all nearby vehicles. Given these, the number of packet transmissions employing the OptTrans scheme will be used in constraint

(6.4c).

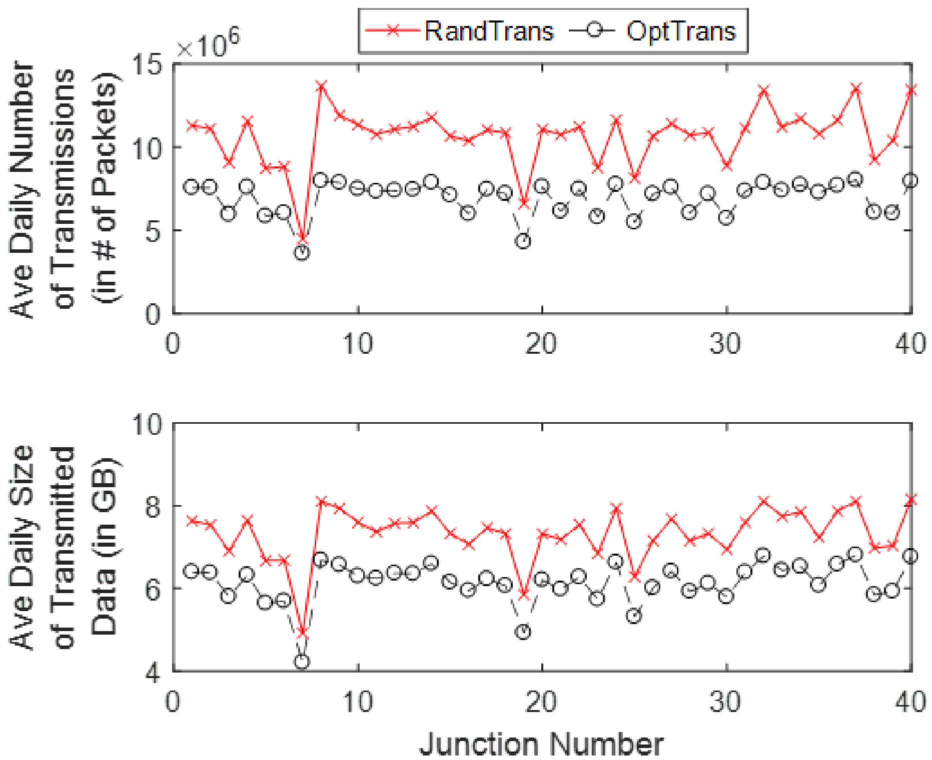

Figure 5.4: Total number of transmitted packets (top part) and transmitted data size (bottom part) for the chosen 40 possible roadside unit locations around the first and second rings of Beijing City employing the RandTrans and OptTrans schemes.

5.3.2 Performance Evaluation

During the simulations, the following threshold values are set. R is set to 10 to limit the number of deployed RSU fog nodes around the two rings of Beijing City to 25%. τ_s is set to 10 kph, which is approximately half the maximum space mean speed allowed by any candidate location. Finally, we make τ_p equal to 1, such that on average, the RSU fog node satisfies one vehicle per transmission.

We first compare how the mode of transmission affects the amount of I2V information shared, since the number of packet transmissions required by the OptTrans scheme to satisfy the vehicular demands is approximately equal to two thirds of that required by the RandTrans method according to Fig. 5.4. Fig. 5.5 shows that OptTrans allows more sharing of information when compared to RandTrans, even if there is less packet transmission and fewer transmitted data size. This is due to the fact that OptTrans often sends encoded packets with doubled amount of information. Given the results presented in Figs. 5.4 and 5.5, we decide to employ OptTrans with the proposed ISRA strategy in the following.

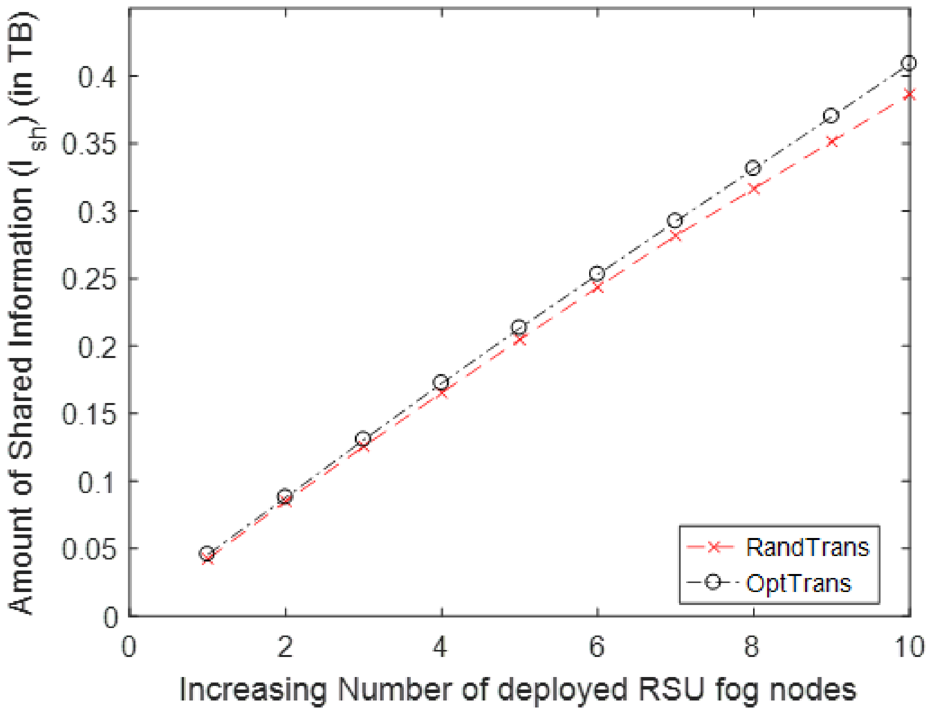

Figure 5.5: Comparing the amount of I2V information shared when using two modes of transmission.

The ISRA strategy is then compared to three other deployment methods described as follows:

1. Downtown-based Deployment (DRSU) [123]: More RSUs are deployed in low-density areas and less in high-density places. 60% of the total number of RSUs to be deployed should be located in low-density areas.

2. Critical Intersections Deployment (CritInt) [124]: This uses a cross-road rank algorithm to determine critical intersections in Beijing City. It ranks the candidate junctions based on the eigenvector centrality measure that factors the effects and relationships of the origin-destination pairs, irreplaceable paths and the junctions involved. The RSU fog nodes, according to their findings, are deployed at junctions 2, 4, 8, 10, 18, 21, 26, 27, 33, and 34.

3. Distinct Vehicles Deployment (DistVeh) [95]: Candidate locations are chosen based on the number of unique taxi IDs in contact with an RSU, i.e., the 10 RSUs with the most number of unique taxi IDs are selected as the candidate locations. The RSUs are deployed to maximize the number of distinct vehicles having at least a single V2I contact based on vehicular trajectories. This is done for collection and dissemination of unique traffic announcements. In this study, these are found at candidate locations 10, 17, 18, 21, 22, 31, 32, 33, 34, and 36.

The top figure of Fig. 5.6 illustrates that ISRA outperforms the other three allocation schemes to maximize the amount of shared information between vehicles and infrastructure nodes. Notice that all allocation methods have the same monotonically increasing trend as the number of RSUs increases. ISRA captures both the characteristics of CritInt and DistVeh, i.e., (1) identifying most of the critical junctions even by only considering the space mean speed and transmission density, and (2) locating junctions that maximizes the number of V2I contacts coming from distinct vehicles. ISRA also avoids the disadvantage provided by DRSU in terms of sharing information in the vehicular network by allocating a fixed amount of RSU fog nodes in certain areas.

We also compare in terms of the percentage change of information shared when ISRA is applied in the simulated city-wide scenario. This is shown in the

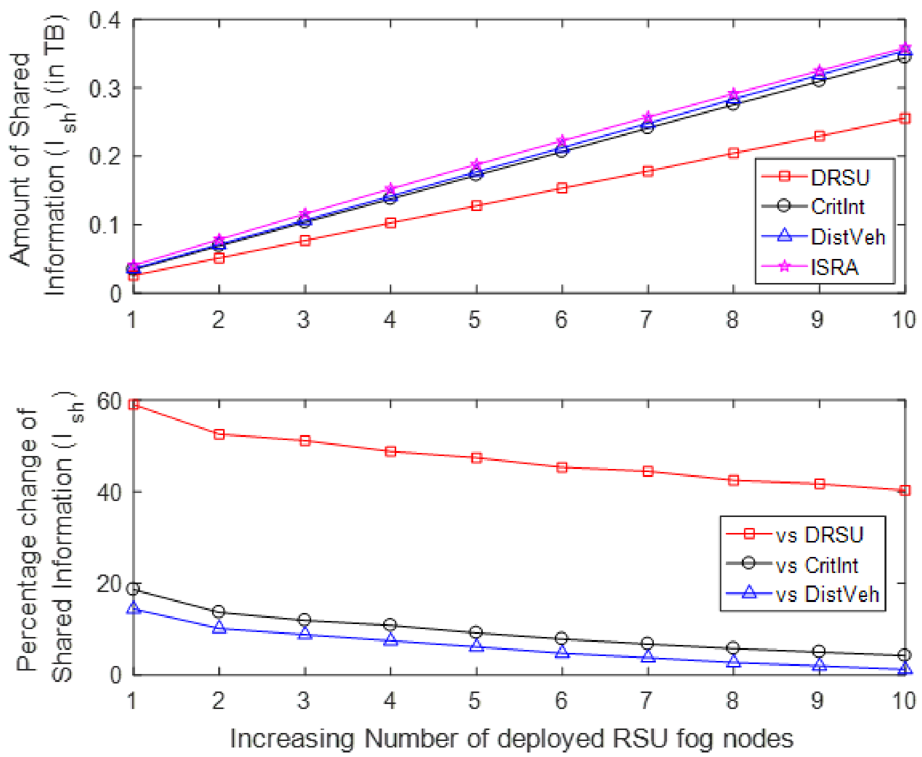

Figure 5.6: The amount of information shared (I_{sh}) using four deployment methods (top figure) and the percentage change of information shared (I_{sh}) provided by ISRA against the other three deployment methods (bottom figure).

bottom figure of Fig. 5.6. On average, ISRA has approximately 6%, 10%, and 47% more shared information than DistVeh, CritInt, and DRSU, respectively. As more RSU fog nodes are being deployed, the behaviors of CritInt and DistVeh approach that of the ISRA strategy.

The energy efficiency (EE) of each of the deployment methods are also evaluated. We define EE in (5.4).

$$EE = \left[1 - \gamma\left(\frac{\sum_{j=1}^{J} NT_j}{I_{sh}}\right)\right] * 100\%, \forall x_j = 1. \tag{5.4}$$

where $\gamma = \frac{\text{Byte}}{\text{\# of packets}}$ is a unit correction factor.

A deployment method is energy-efficient if the chosen RSU fog node locations are able to share the most information using the least number of packet

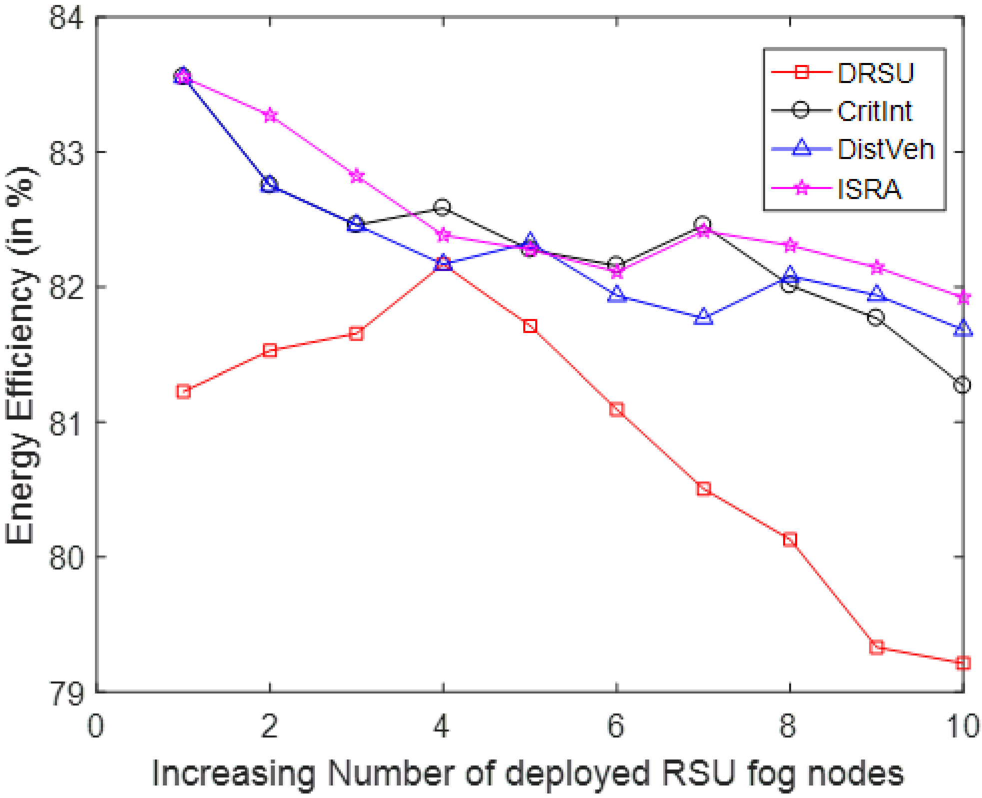

Figure 5.7: The energy efficiency of the four deployment methods.

transmissions. The energy efficiency for each deployment method is shown in Fig. 5.7. Generally, ISRA has a higher energy efficiency (average of 83%) compared to the other three deployment methods. For DRSU, the decrease of its energy efficiency is affected by the deployment of RSU fog nodes at low-density junctions.

Finally, Fig. 5.8 shows how much information is shared by each selected RSU fog node in each deployment scheme. It is evident that ISRA balances the information sharing between the deployed RSUs. ISRA, by considering the region's space mean speed, allows the offloading of vehicles to nearby RSU fog nodes such that the system's computational power and memory capacity will not be exceeded. Such capability allows ISRA to virtually interconnect all deployed RSU fog nodes and maintain RSU fog nodes in the vicinity to operate at roughly the same rate.

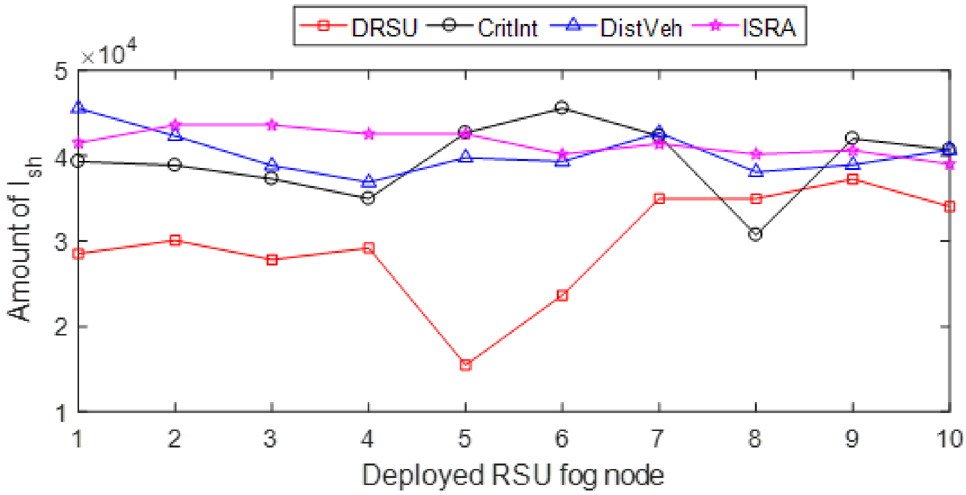

Figure 5.8: ISRA balances the information shared between the deployed RSU fog nodes. (For ISRA, deployed RSU fog nodes 1-5 and 6-10 are groups of adjacent RSU locations.)

Therefore, supported by Figs. 5.6 - 5.8, ISRA outperforms the other three deployment methods by being able to allocate RSU fog nodes in energy-efficient and information-rich junctions, while at the same time, able to balance the load (shared information processing/storage) among all deployed RSU fog nodes.

5.3.3 ISRA-based Highway Control Application

The highway tollgate section in Fig. 3.1 road is extended to include i sections of the highway to consider the flow of intelligent connected vehicles before they approach the tollgate servers as shown in Fig. 5.9. We model each highway section by studying its vehicular density (5.5), traffic flow (5.6), the empirical relationship between speed and density (5.7), and space mean speed (5.8) [125].

$$\rho_i(k+1) = \rho_i(k) + \frac{T}{L_i}\Big[q_{i-1}(k) - q_i(k)\Big] \tag{5.5}$$

$$q_i(k) = \rho_i(k)v_i(k) \tag{5.6}$$

$$V[\rho_i(k)] = v_f\left(1 - \left[\frac{\rho_i(k)}{\rho_{cr}}\right]^l\right)^m \tag{5.7}$$

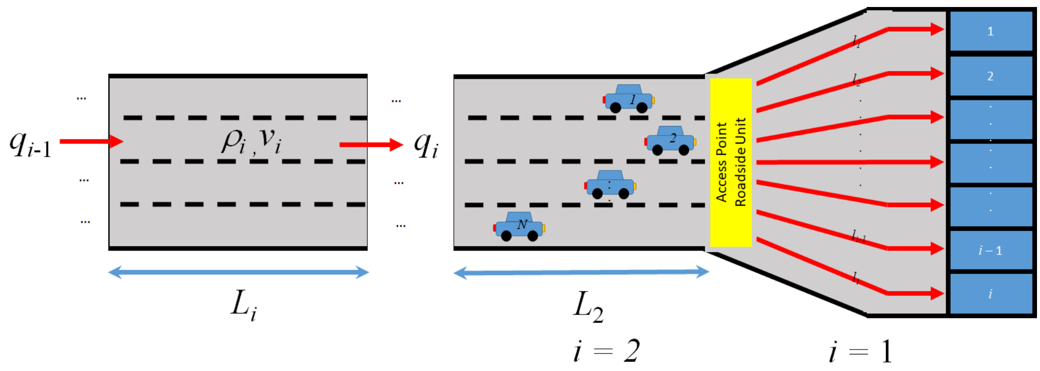

Figure 5.9: Highway section and its traffic parameters.

$$\begin{aligned} v_i(k+1) = v_i(k) &+ \frac{T}{\tau}\Big[V\rho_i(k) - v_i(k)\Big] \\ &+ \frac{Tv_i(k)}{L_i}\Big[v_{i-1}(k) - v_i(k)\Big] \\ &- \frac{aT}{\tau L_i}\left[\frac{\rho_{i+1}(k) - \rho_i(k)}{\rho_i(k) + \kappa}\right] \end{aligned} \tag{5.8}$$

where T is the sampling interval expressed in hours and i is the highway section from $1, 2, ..., N$ with length L_i in km. v_f is the maximum allowable highway section space mean speed expressed in kph and ρ_{cr} is the maximum vehicular density (in veh/km). The constants τ, v, κ, l, m are parameters characterizing the highway. We adopt these parameter values from [126].

$\rho_i(k), v_i(k)$, and $q_i(k)$ are the highway section i average vehicular density (in veh/km), space mean speed (in kph), and average traffic flow (in veh/hr) at sampling time k, respectively. $\rho_i(k) = E[P_i(k)]$ and $q_i(k) = E[Q_i(k)]$, where $P_i(k)$ and $Q_i(k)$ are the remaining and passing intelligent connected vehicles in highway section i, respectively. $E[\cdot]$ is the expectation operator.

The boundary conditions at the start ($i = 1$) and end ($i = N$) of the highway are given in (5.9)–(5.12).

$$v_0(k) = v_1(k) \tag{5.9}$$

$$\rho_0(k) = \frac{q_0(k)}{v_i(k)} \tag{5.10}$$

$$v_{N+1}(k) = v_N(k) \tag{5.11}$$

$$\rho_{N+1}(k) = \rho_N(k) \quad \forall k \tag{5.12}$$

The number of cars remaining, $P_i(k)$ and passing $Q_i(k)$ in section i follows a Poisson distribution with means $E[P_i(k)] = \sum_{\sigma(k)}^{k} \alpha(k)$ and $E[Q_i(k)] = \sum_{-\infty}^{\sigma(k)} \alpha(k)$, respectively, where $\alpha(k)$ is equal to the arriving vehicle that remains at section i at time k. Additionally, the numbers of vehicles, P_i and P_{i+1}, for two non-overlapping sections i and $i+1$ are also Poisson-distributed and independent. [127].

We assume that the highway section lengths are $L_1 = L_2 = L_4 = 500$ meters [126,128]. $L_3 = 1000$ meters. The simulation parameters for controlling the highway section vehicular densities and speeds are given in Table 5.2.

Table 5.2: Simulation Parameters for Highway Section Vehicular Density and Speed Control

Parameter	Value	Parameter	Value
ρ_{cr}	3000 $\frac{\text{veh}}{\text{km}}$	l	1.8
T	10 sec	m	1.7
v_f	80 kph	$v_{i0}(0)$	15 kph
τ	0.01 hr	$\rho_{i0}(0)$	50 $\frac{\text{veh}}{\text{km}}$
υ	35 $\frac{\text{km}^2}{\text{hr}}$	N	4
κ	13 $\frac{\text{veh}}{\text{km}}$	K_p	1

Fig. 5.10 illustrates the space mean speed and average vehicular density profiles of each section when the central Access Point (AP) is not communicating with the RSUs installed on the highway sections, therefore, AP control is within section 1 only. At steady state, the Tollgate region (Section 1) allows the most vehicular density and the slowest space mean speed since these vehicles are now exiting the highway. This coincides with the findings in Fig. 3.7 in Chapter 3 for a given homogeneous heavy traffic flow. All the other

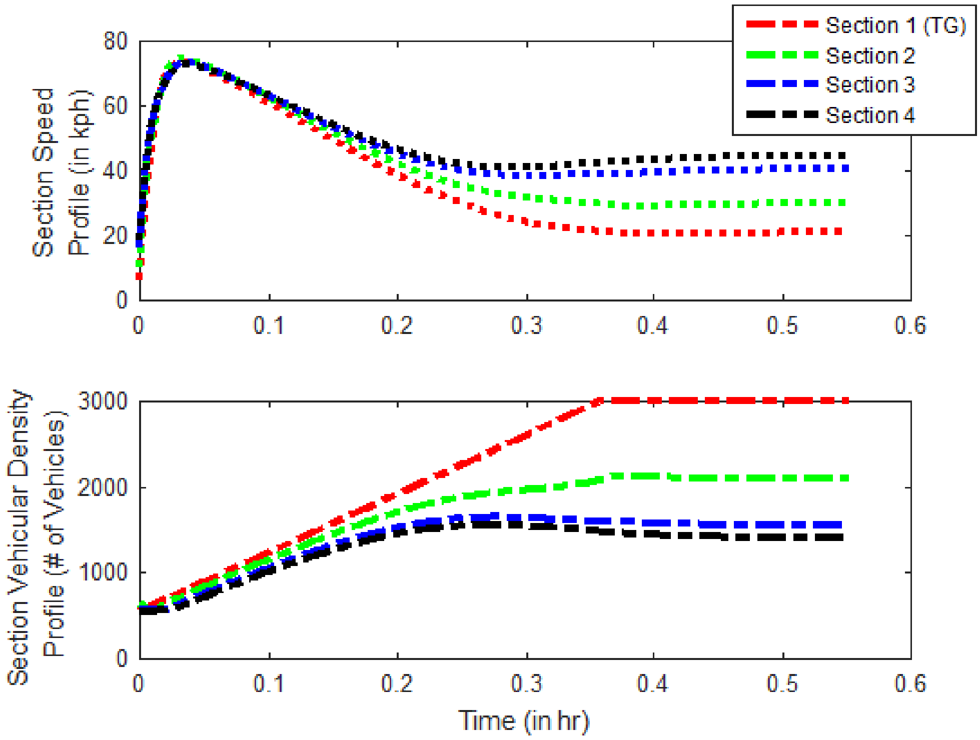

Figure 5.10: Traffic space mean speed and density responses along the highway divided into four sections. The centralized AP is not communicating to any deployed RSUs.

sections' vehicular densities will populate accordingly, i.e., the section nearer to tollgate region (Section 1), the more intelligent connected vehicles are seen with slower space mean speed.

In Chapter 3, the developed centralized fuzzy logic controller can provide early warning signal to the tollgate servers to hasten their service. This early warning signal is now also transmitted to the other RSUs on the highway to inform vehicles of their corresponding speed, so that they will not have to wait longer when they reach the tollgate section. We implement an error controller for controlling a section's average vehicular density and space mean speed, as illustrated in Fig. 5.11. The response is depicted in Fig. 5.12.

For example, as the central AP recognizes the current low(high) vehicular density in its region, it allows the other RSUs to inform its intelligent connected vehicles to speed up(down), accordingly. Notice that all sections' speed

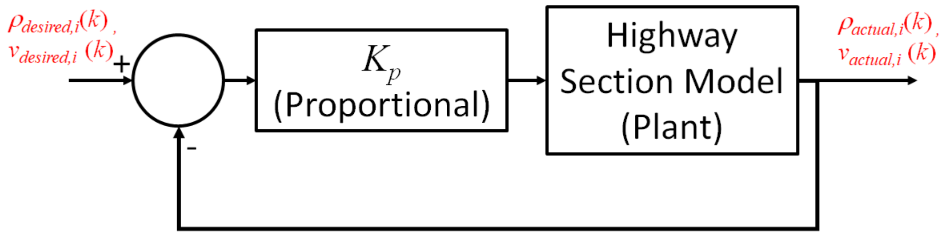

Figure 5.11: Proportional controller for improving the highway section's average vehicular density and space mean speed.

and density profiles are maintained to be approximately be the same, while not allowing each vehicle to approach full stop ($v_i(k) \neq 0$). With the combination of faster tollgate service times and coordinated traffic flow, intelligent connected vehicles experience more travel comfort and convenience. This will also allow better fuel efficiency [129].

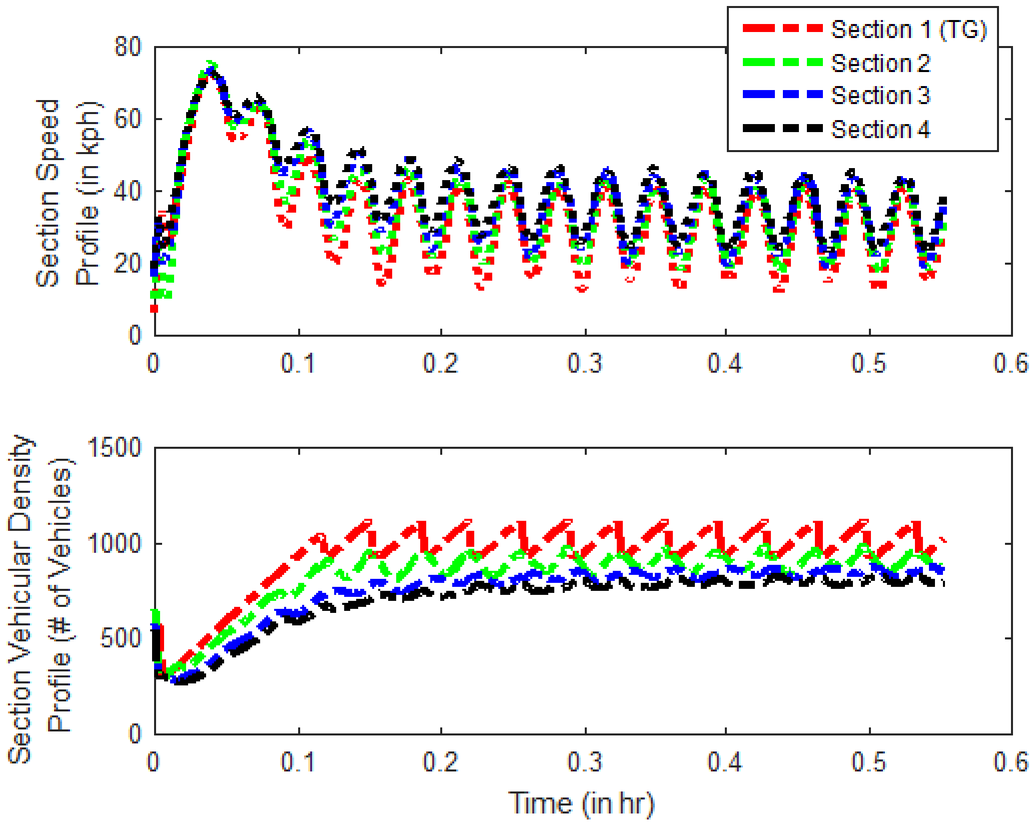

Figure 5.12: Highway sections' space mean speed and average vehicular density profiles when the early warning signal from the central AP is also transmitted to deployed RSUs at Sections 2 and 4.

5.4 Summary

In this study, we have proposed an Information Sharing via Roadside Unit Allocation (ISRA) strategy for deploying RSU fog nodes in a city-wide context to maximize the amount of information being shared among vehicles and RSU fog computing hotspots. To do this, empirical findings from taxi mobility traces plying the City of Beijing are used. ISRA allocates RSU fog nodes to city intersections that are information-rich and energy-efficient. Given a constraint of deploying 10 RSU fog nodes in Beijing City, ISRA enjoys a 6% increase in the amount of information being shared compared with the best conventional scheme, which is equivalent to about 4 GB more of shared information. Also, ISRA achieves an 83% energy efficiency that translates to fewer packet transmissions needed for sharing more information to the surrounding vehicles. ISRA is able to balance the information sharing among deployed RSU fog nodes, such that in practice, all deployed RSU fog nodes can operate at similar computational power and memory consumption. Finally, a control data dissemination application has verified the effectiveness of deploying RSUs at optimal locations by using ISRA. By this method, the space mean speed and vehicular density of each of the highway sections have been equalized and responsive to the central AP's early warning signal.

Chapter 6

Enhanced Information Sharing via Roadside Unit Allocation Scheme

In the previous chapter, ISRA chose locations from a set of intersections to deploy RSUs that will maximize the amount of shared information. In this work, we develop an Enhanced Information SHAring via RSU Allocation (EISHA-RSU) algorithm that will consider all spatial locations as possible candidate RSU locations. Unlike ISRA, EISHA-RSU irregularly partitions an urban area to discriminate regions according to vehicular capacity. This irregular partitioning prioritizes locations where to allocate RSUs for obtaining maximum V2I and I2V information while addressing the issues of coverage and connectivity among vehicles and infrastructure.

The chapter is outlined as follows. Section 1 discusses the definitions, assumptions, and setup considered to solve the allocation problem. Section 2 presents the novel EISHA-RSU algorithm that utilizes the concepts of Effective Regions of Movement and Effective Position. Section 3 discusses the results derived from extensive simulation employing three urban empirical mobility traces. The summary and conclusion are stated in Section 4.

6.1 Effective Regions of Movement

In this section, we discuss the irregular partitioning of an urban map according to its vehicular distribution, thus, automatically removing unnecessary city details and optimally form homogeneous effective regions of movement (ERMs).

6.1.1 Spatiotemporal Stable Network Characteristics

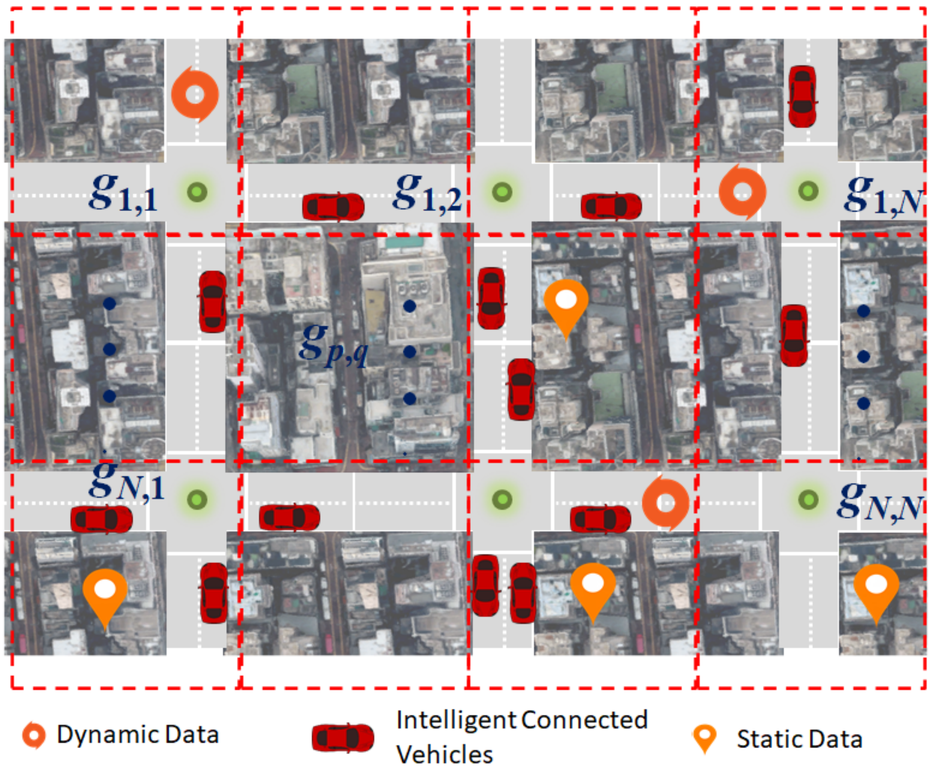

Figure 6.1: Uniform partitioning of an urban map revealing various $g_{p,q}$ and its corresponding utility of network parameters (e.g., dynamic and static data, and the number of intelligent connected vehicles) at sampling time $t = iT_S$.

Consider the uniform partitioning of an urban area under study in Fig. 6.1 into $N \times N$ map grids, $g_{p,q}$. Each map grid, $g_{p,q}$, is characterized by its utility function, $\zeta_{p,q}$, at time t and depends on the map grid's longitude and latitude location. The indexes p and $q \in \{1, \ldots N\}$. We define $\zeta_{p,q}$ as:

$$\zeta_{p,q} = \mathbf{E}\big[\eta_{p,q}\big]$$

where $\mathbf{E}\big[\bullet\big]$ is the expectation of $\big[\bullet\big]$. $\eta_{p,q}$ describes the map grid's current spatial network characteristic, such as the grid's dynamic data, $\delta_{p,q}$, static data, $\gamma_{p,q}$, vehicular capacity, $c_{p,q}$, vehicular density, connectivity, accident rate, etc. Each network characteristic is assumed to be independent from each other and the utility function can be a combination of any of these $\eta_{p,q}$'s. In this work, $\eta_{p,q} = c_{p,q}$.

If vehicular movements and other dynamic network characteristics are disregarded, the map grid's current spatial network characteristic is constant, i.e., $\zeta_{p,q} = K$. However, when dynamic map data, sources, and movement of vehicles are considered, $\zeta_{p,q}$'s vary in both space and time. Therefore, to implement a consistent and reliable grid partitioning, the map grid's spatiotemporal stable $\zeta_{p,q}$, must be determined from available sampling times. A map grid $g_{p,q}$'s spatiotemporal stable network characteristic, $\zeta_{p,q,STS}$, is established according to (6.1).

$$\zeta_{p,q,STS} = \sum_{i=0}^{I} \alpha(iT_S)\omega(iT_S) \tag{6.1}$$

where

$$\alpha(iT_S) = \frac{\zeta(iT_S) - \min\big[\zeta(iT_S)\big]}{\max\big[\zeta(iT_S)\big] - \min\big[\zeta(iT_S)\big]}$$
$$\omega(iT_S) = \frac{\zeta(iT_S)}{\max\Big[\zeta(i = 0, \ldots, IT_S)\Big]}$$

$\alpha(iT_S)$ is the feature scaling parameter at time $t = iT_S$, while $\omega(iT_S)$ is the weight correlating all the $\zeta_{p,q}$'s, respectively. Fig. 6.2 illustrates an example of how the spatiotemporal stable network characteristics $\zeta_{p,q,STS}$'s are generated as defined in (6.1). $\zeta_{p,q}$ is characterized by the map grid's vehicular capacity, $c_{p,q}$.

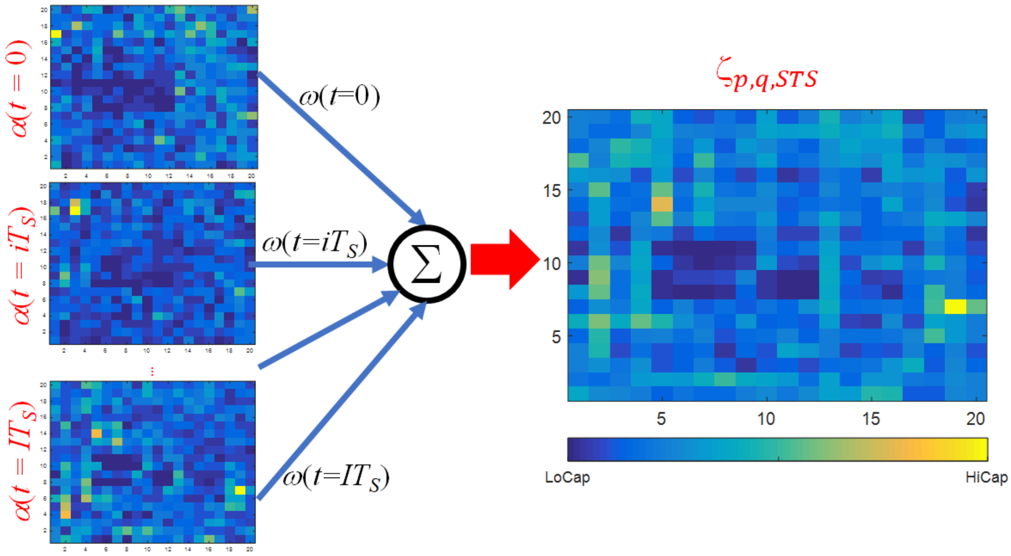

Figure 6.2: An example of how $\zeta_{p,q,STS}$ of each map grid $g_{p,q}$ is formed. Each $\zeta(t = iT_S)$ is characterized by its current vehicular capacity, $c_{p,q}$. Darker map grids have lower vehicular capacities (LoCap) over lighter grids (HiCap).

6.1.2 Forming Effective Regions of Movement

To divide a geographical area into various sections and determine the possible candidate locations for deploying roadside units, we introduce the concept of the Effective Regions of Movement (ERMs). An ERM is a grouping of edge-adjacent map grids with its spatiotemporal stable network characteristic, $\zeta_{p,q,STS}$, that possess a unifying characteristic, such as vehicular capacity, density, etc. The merging [130] of spatiotemporal map grids form an ERM,

ERM_e, is governed by (6.2a) below.

$$ERM_e \equiv g_{p,q} \cup g_{p+\Delta p,q+\Delta q} \tag{6.2a}$$

$$\text{subject to} \qquad |\{c_{g_{p,q}}\} \cup \{c_{g_{p+\Delta p,q+\Delta q}}\}| \leq \tau_c \tag{6.2b}$$

$$\min(\rho_{g_{p,q}}, \rho_{g_{p+\Delta p,q+\Delta q}}) \geq \rho_0 \tag{6.2c}$$

where $\Delta p, \Delta q \in \{-1, 0, 1\}$. $c_{g_{p,q}}$ is the expected vehicular capacity found in $g_{p,q}$, while τ_c is the vehicular capacity threshold of each formed ERM_e. Note that only edge-adjacent grids are considered during the merging. $\rho_{g_{p,q}}$ and $\rho_{g_{p+\Delta p,q+\Delta q}}$ denote the outbound and inbound vehicular flow from and to $g_{p,q}$, respectively. If the minimum vehicular flow between two map grids is $\geq \rho_0$, then merging proceeds; otherwise, $g_{p+\Delta p,q+\Delta q}$ is dropped.

The algorithm to determine various ERMs of an area under study is illustrated in Algorithm 5. Fig. 6.3 shows an illustration of formed ERMs derived from Fig. 6.2 containing single- and multiple-grid ERMs.

Pseudo-codes 5 Determining Effective Regions of Movement (ERMs)

INPUT:
$g_{p,q}$'s – List of spatiotemporal stable geographical grids
$c_{p,q}$ – vehicular capacity of $g_{p,q}$
τ_c – vehicular quantity threshold,
N^2 - number of grids
OUTPUT: ERM, List containing ERMs.

Δp = [-1,0,1,0]; and Δq = [0,1,0,-1];
$ERM = 0_{N\times N}$;
region = 1; ▷ **Initialize first ERM region to 1.**
$L_c = \text{sort}(c_{p,q})$ in descending order;
for $i = 1$ to length of L_τ **do**
    $[p, q] = \text{ind2sub}(\text{index}(L_\tau(i)))$;
    ▷ **Convert linear indices to subscripts**
    $ERM(p, q)$ = region;
    $L_c = L_c \setminus L_c(g_{p,q})$;
    while $L_c \neq \emptyset$ **do**
        for $k = 1$ to 4 **do** ▷ **Get edge-adjacent map grids.**
            if Constraints (6.2b) AND (6.2b) == TRUE **then**
                if $g_{p+\Delta p(k),q+\Delta q(k)}$ is not yet visited **then**
                    $ERM(p + \Delta p(k), q + \Delta q(k))$ = region;
                    ▷ **Assign $g_{p+\Delta p(k),q+\Delta q(k)}$ to ERM region**
                    $L_c = L_c \setminus L_c(g_{p+\Delta p(k),q+\Delta q(k)})$;
                end if
            end if
        end for
    end while
    region = region + 1; ▷ **Move to next ERM**
end for

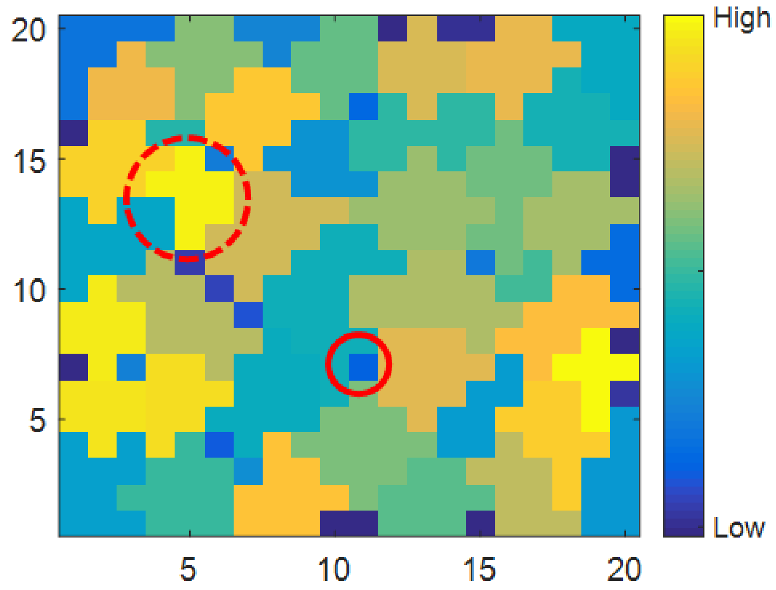

Figure 6.3: An example of various ERMs formed by following Algorithm 5. Dark and light colors depict low and high ERM priorities, respectively. In the formation of ERMs, single-grid (bold circle) and multi-grid (dotted circle) ERMs are created.

6.2 Enhanced Information Sharing RSU Allocation (EISHA-RSU) Scheme

In this section, we discuss the novel Enhanced Information SHAring RSU (EISHA-RSU) allocation technique. EISHA-RSU maximizes information sharing and vehicular connectivity in ERMs by allowing relevant and on-time information exchange among the largest number of vehicles and infrastructure. In essence, by considering the proper spacing between deployed RSUs, EISHA-RSU, can maximize the urban coverage area and achieve its goals by utilizing the effective position (EP) concept for locating ideal RSU deployment position. We define an EP as a physical urban location where we can deploy an RSU for environment information collection, such as an intersection, certain location along a road segment, a combination of both, or any landmark.

6.2.1 Problem Formulation

The main goal of EISHA-RSU is to locate RSUs in spots where maximized information is shared between the infrastructures and vehicles. These road and environment data can be either static (landmarks and roads) or dynamic (pedestrian, road accidents, events, etc.) in nature. Given a constraint in the number of RSUs to be deployed, EISHA-RSU prioritizes each effective region of movement to ensure that even the least prioritized map grids still has a chance to obtain relevant and on-time information from high-priority ERMs. EISHA-RSU also assures on-time delivery and storage of dynamic environment data collected from the surrounding vehicles.

In each ERM, the amount of information shared, I_{Sha}, is given in (6.3).

$$\begin{aligned} I_{Sha} &= U_{\gamma} + \beta U_{\delta} \\ &= \sum_{p=1}^{N}\sum_{q=1}^{N} \gamma_{g_{p,q}} + \beta \sum_{p=1}^{N}\sum_{q=1}^{N} \delta_{g_{p,q}}(t) \end{aligned} \tag{6.3}$$

where $\gamma_{g_{p,q}}$ and $\delta_{g_{p,q}}(t)$ represent the amount of static and dynamic environment road map data, respectively. We note again that each $g_{p,q}$ is dependent on its corresponding longitude (x) and latitude (y) coordinates. β is an importance factor we assign to dynamic road data to signify the repetitive occurrence of instantaneous events, such as accident-prone areas in a $g_{p,q}$, where $1 \leq \beta \leq \frac{1}{\xi}$. ξ denotes the proportionality constant between static and dynamic environment data such that $\delta_{g_{p,q}} = \xi \gamma_{g_{p,q}}$, where $0 < \xi \leq 1$, i.e., dynamic environment data are much less than static environment data. When $\beta = \frac{1}{\xi}$, it implies that the dynamic environment data $\delta_{g_{p,q}}$ have high importance and are treated equally as static environment data.

EISHA-RSU addresses the maximization problem given by (6.4a) subject to constraints (6.4b) and (6.4c).

$$\begin{aligned} &\text{maximize} && I_{Sha} && (6.4a)\\ &\text{subject to} && \sum_{l=1}^{N^2} EP_l \leq \Omega_R,\ EP_l \in \{0,1\} && (6.4b)\\ & && \frac{d(EP_l, EP_m)}{v(EP_l, EP_m)} \leq W \quad \forall EP_l, EP_m = 1 && (6.4c) \end{aligned}$$

Constraint (6.4b) assures that there is only the at most Ω_R RSUs to be deployed in each ERM or the urban map under study, located at effective positions, EP_l. $EP_l = 1$ means an RSU can be deployed there; otherwise, $EP_l = 0$. Constraint (6.4c) denotes the network's allowable on-time delivery delay, W. It considers space mean speed, $v(EP_l, EP_m)$, and separation, $d(EP_l, EP_m)$, between EPs such that vehicular data outside RSU coverage can still be valid once it is within RSU range. We note that $d(EP_l, EP_m)$ does not automatically equal to the shortest distance between EP_l and EP_m. It is the travel distance with respect to the road network. In Fig. 6.4, $d_1 = d(EP_l, EP_m) = 4$ and $d_2 = d(EP_l, EP_k) = 3$.

6.2.2 Delay Analysis between two Effective Positions

To achieve on-time and up-to-date delivery of environment data, the distance between two EPs must be minimized, according to the delivery delay W. The calculated separation between EPs will allow vehicles outside an RSU's transmission range to travel and carry valid and relevant road information from one grid to another without any RSU.

The general expression for the total average delivery delay, W, to locate two EPs, EP_l and EP_m found in ERM_e, is given in (6.5).

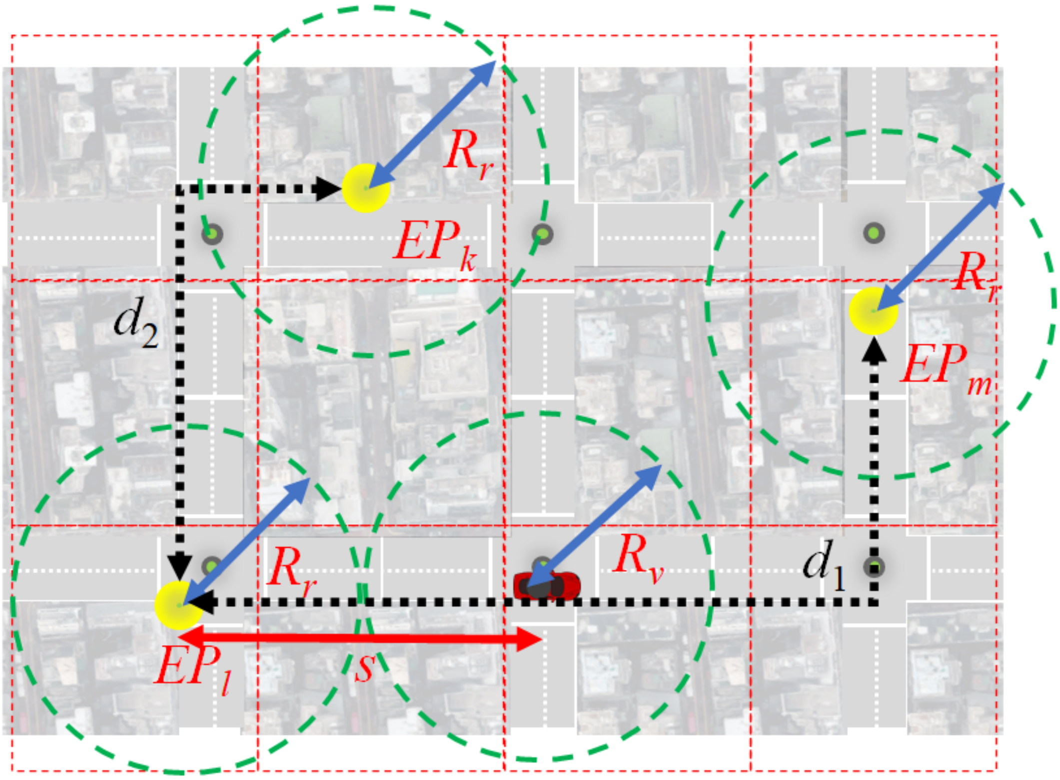

Figure 6.4: Determining the effective positions in an ERM given the allowable network data delivery delay and road parameters of the urban city.

$$W = W_g + W_a \tag{6.5}$$

where W_g is the average time required for a vehicle to deliver its stored data before becoming invalid to an RSU at an effective position while traversing road distance d on g grids along the path. W_a is the average additional stop time the vehicle encounters during its trip, e.g., passing an intersection or encountering accidents. For simplicity, we assume that W_a is constant. (6.5) can be re-written to the expression given in (6.6) to determine how much time a vehicle takes to traverse grids g along a given path found in ERM_e.

$$W_g = W - W_a \tag{6.6}$$

We adopt the delay analysis in [98,131], and apply it to the 2D scenario in Fig. 6.4 to determine how far an effective position should be situated from an initial EP, EP_l. The following are assumed to determine the effective positions where RSUs can be allocated.

1. When outside of an RSU transmission range, intelligent connected vehicles can still sense environment data. These collected vehicular data can only be shared and forwarded to an RSU found at an effective position in the path ahead (V2I operations only). V2V communications are not considered here to model the worst case scenario, i.e., a vehicle must bring its information directly to the RSU.

2. The transmission ranges of a deployed RSU at an effective position and a vehicle are R_r and R_v, respectively.

Upon leaving an RSU at EP_l, the average time for the red vehicle as depicted in Fig. 4 to deliver its newly collected data to a nearby effective position found in another grid, while still being valid, is (6.7), given that the conditional probability of s (the location of the vehicle with information) in $[0, d]$ is $f(s)$.

$$W_g = \int_0^d WT_g f(s) ds \tag{6.7}$$

Given the assumptions above, the time needed for the red vehicle (source vehicle) in Fig. 6.4 to deliver its valid data along a road of length d, WT_g, is given by (6.8).

$$WT_g = pT \tag{6.8}$$

$$p = 1 - (1 - e^{-\lambda R_v})^{\kappa}$$

$$T = \frac{d - s - R_r + R_v - E_x}{v}$$

where

$$\kappa = \frac{2(d - s - R_r + R_v)}{\mathrm{E}[d_V]}$$
$$E_x = \frac{\mathrm{E}[d_V]\big[1 - (\kappa+1)(1 - e^{-\lambda R_v})^{\kappa} + \kappa(1 - e^{-\lambda R_v})^{\kappa+1}\big]}{2[1 - (1 - e^{-\lambda R_v})^{\kappa}]e^{-\lambda R_v}}$$

p is the probability that the RSU at EP_m is beyond the range of the source vehicle with transmission range R_v. T is the delivery delay from position s to $d - R_r$, i.e., out of RSU's communication range at EP_m.

λ is the departing rate of vehicles from an effective position that can overtake the source vehicle and can become a forwarding node. v is the space mean speed of the road segment d. $\mathrm{E}[d_V]$ is the average distance of vehicles found between the two effective positions, d is the maximum separation between two effective positions and is not necessarily the shortest distance but the distance defined by the road topology. s is the location (distance traveled from EP_l) of a source vehicle having new environment data to be shared.

The worst-case scenario happens when a vehicle has real-time environment data and has no immediate RSU to offload its contents. This scenario also occurs when it has no leading vehicle(s) within its transmission range, R_v, to which it can forward its information. As such, the worst case scenario happens where $s = R_r + R_v$, $\lambda = 0$, and $\mathrm{E}[d_V] = d - s - R_r$.

Given the values of W and W_a, the maximum allowable separation, d, between two effective positions is given in (6.9). Note here that the conditional probability of s in $[0, d]$ follows a uniform distribution.

$$d = 2W_g v + R_r \tag{6.9}$$

6.2.3 EISHA-RSU Algorithm

The EISHA-RSU scheme allocates RSUs to effective positions found in each ERM by satisfying the maximization problem in (6.4). Its detailed operation is illustrated in the pseudo-code in Algorithm 6.

ERMs can be categorized into two configurations, namely: (1) single-grid and (2) multiple-grid. We discuss for each configuration how the EPs are identified.

Single-Grid ERMs

For single-grid ERMs, an example is shown in Fig. 6.5, where its static environment data, $\gamma_{g_{p,q}}$, is represented by the blue shade. The grid is further sub-divided into $k = 1, 2, \ldots, K$ sub-grids to introduce additional dynamic data, $\delta_{g_{p,q}} = \sum_{k=1}^{K} \mu_k \delta_{k,g_{p,q}}$, where $\mu_k \in \{0, 1\}$ and is used to reduce the computation cost for dynamic data. A sub-grid k contributes dynamic data when $\mu_k = 1$, else zero.

(6.4a) reduces to an optimization problem requiring only one RSU to be deployed, since this is the least deployable number of RSUs. The effective position where the RSU is located is at the point where maximum static and dynamic data are shared, as defined in (6.10). We set $\beta = 1$.

$$\begin{aligned} I_{Sha} = & \int_{x_1}^{x_1+\Delta x+R_r} \int_{y_1}^{y_1+\Delta y+R_r} \gamma_{g_{p,q}}(x,y)dxdy \\ & + \sum_{k=1}^{K} \int_{\Delta x_{1,k}}^{\Delta x_{2,k}} \int_{\Delta y_{1,k}}^{\Delta y_{2,k}} \mu_k \delta_{k,g_{p,q}}(x,y)dxdy \end{aligned} \tag{6.10}$$

The center of an RSU with transmission range R_r is moved from the corner point x_1, y_1 by $\Delta x, \Delta y$ until maximum static data are covered. By doing this, the maximum static data are shared when $\Delta x = \frac{x_2 - x_1}{2}$ and $\Delta y = \frac{y_2 - y_1}{2}$, i.e.,

Pseudo-codes 6 EISHA-RSU Algorithm

INPUT:
ERM's – Formed ERMs with its priority
Ω_R – maximum number of RSU's to be deployed
W_g – Waiting time for data to be considered valid
R_r and R_v – transmission ranges of RSU's and vehicles, respectively
K – number of sub-grids

OUTPUT: EP_{List}, List containing EPs of each ERM.

Determine ERM ascending priority list, ERM_{List};
Divide each grid of ERM_e into K sub-grids.
$\Omega_{R_{EP}} = \text{ceil}(\Omega_R / ERM_{List})$; ▷ # of EPs for high-priority ERMs
$EP_{List} = \emptyset$.
$\Omega_{R_{ctr}} = \Omega_R$;
for $e = 1 : \max(ERM_{List})$ **do**
  if ERM_e is single-grid ERM **then**
    if $\Omega_{R_{ctr}} > 0$ **then**
      $EP_{temp} = [x^*, y^*]$;
      $EP_{List} = EP_{List} \cup EP_{temp}$;
      $\Omega_{R_{ctr}} = \Omega_{R_{ctr}} - 1$;
    end if
  else
    Determine how many $\Omega_{R_{EP}}$ for ERM_e.
    if $\Omega_{R_{EP}} > 0$ **then**
      M = Convolve ERM_e with $\mathbf{1}_{\sqrt{K} \times \sqrt{K}}$
      $\mathbf{M_{List}} = \text{nchoosek}(\mathbf{M}, \Omega_{R_{EP}})$;
      $ctr = 1$;
      while $\text{size}(EP_{List}) \neq \Omega_{R_{EP}}$ **do**
        Compute distances in $\mathbf{M_{List}}(ctr)$.
        EP_{temp} = center locations of $\mathbf{M_{List}}(ctr)$.
        if (6.9) is satisfied **then**
          $EP_{List} = EP_{List} \cup EP_{temp}$;
        end if
        $ctr = ctr + 1$;
      end while
    end if
    $\Omega_{R_{ctr}} = \Omega_{R_{ctr}} - \Omega_{R_{EP}}$;
  end if
end for
Output EP_{List}.

the center of the grid. Thus, the location of the effective position EP to cover static data in a single grid is:

$$x_\gamma = x_1 + \Delta x \text{ and } y_\gamma = y_1 + \Delta y \tag{6.11}$$

Likewise, for covering all K sub-grids of an ERM containing dynamic data, the location is at:

$$x_\delta = \sum_{k=1}^{K} \frac{\mu_k \delta_{k,g_{p,q}}(x,y)}{\sum_{k=1}^{K} \mu_k \delta_{k,g_{p,q}}(x,y)} \varepsilon_k \tag{6.12}$$

$$y_\delta = \sum_{k=1}^{K} \frac{\mu_k \delta_{k,g_{p,q}}(x,y)}{\sum_{k=1}^{K} \mu_k \delta_{k,g_{p,q}}(x,y)} \zeta_k \tag{6.13}$$

ε_k and ζ_k are the centroid coordinates of sub-grid k with available dynamic data, where $\varepsilon_k = \frac{\Delta x_{2,k} - \Delta x_{1,k}}{2}$ and $\zeta_k = \frac{\Delta y_{2,k} - \Delta y_{1,k}}{2}$.

$x_1 \leq x_\gamma, x_\delta \leq x_2$ and $y_1 \leq y_\gamma, y_\delta \leq y_2$. Given these two possible EP locations, the maximum static and dynamic information shared is achieved when the RSU is situated at an EP having coordinates in (6.14), i.e., at the center of the single-grid ERM.

$$x^* = x_\gamma \text{ and } y^* = y_\gamma \tag{6.14}$$

under the constraint that $x_2 - x_1 = y_2 - y_1 \leq \sqrt{2} R_r$.

Multiple-Grid ERMs

There are two cases for multiple-grid ERMs, 1) the number of map grids is equal to Ω_R, and 2) the number of map grids is greater than Ω_R. For case 1),

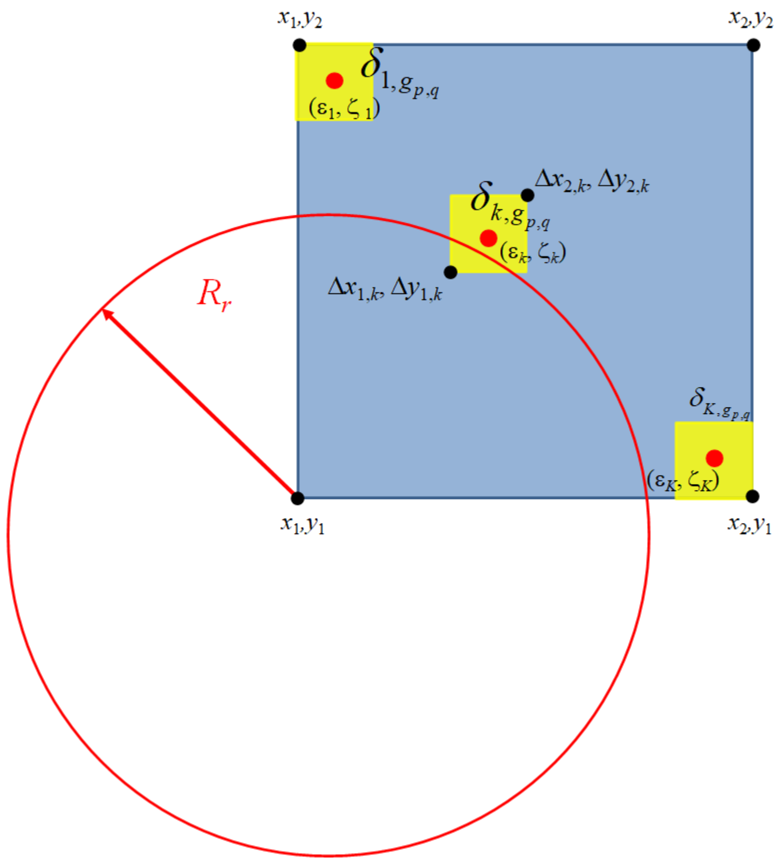

Figure 6.5: Determining the effective position in a single-grid ERM, given the grid's static and dynamic data.

locating EPs follows the single-grid ERM deployment, where each map grid of the ERM has an effective position at the center. However, if the number of grids is higher than the desired number of deployable RSUs, then the EPs are heuristically searched.

Given Ω_R RSUs to be deployed in an urban setup, EISHA-RSU allocates an RSU to all ERMs by following a round-robin procedure. If Ω_R is higher than the lowest priority ERM, then all ERMs are guaranteed to have at least one effective position where an RSU can be deployed. Another round robin deployment ensues and ends until the target number Ω_R has been reached.

After calculating the number of RSUs needed to be deployed in each ERM, $\Omega_{R_{EP}}$, then EISHA-RSU follows a greedy heuristic method for finding these

$\Omega_{R_{EP}}$ EPs. When Ω_R is not exactly divisible by the number of ERMs, higher priority ERMs will have $\Omega_{R_{EP}}$ EPs, while lower priority ERMs will have $(\Omega_{R_{EP}} - 1)$ EPs. A map grid is further divided into K sub-grids to accommodate the presence of dynamic data, if any. $K = 9$ sub-grids is considered for the results in this paper. The ERM is then convolved with a $\mathbf{1}_{\sqrt{K}\times\sqrt{K}}$ filter to determine how much information can be shared when one vehicle travels from one sub-grid to another in the ERM. With respect to Algorithm 2, the convolution results are stored in $\mathbf{M}$. From $\mathbf{M_{List}}$, EISHA-RSU selects the first $\Omega_{R_{EP}}$ locations with maximum shared information. Their distances should be equal to (6.9) to avoid overlapping between RSU coverage and prevent invalid data delivery. If their separations do not satisfy (6.9), then the next maximum information location combination is considered until the optimization problem in (6.4a) is satisfied. The discussion of this is seen in lines 13–25 of Algorithm 6.

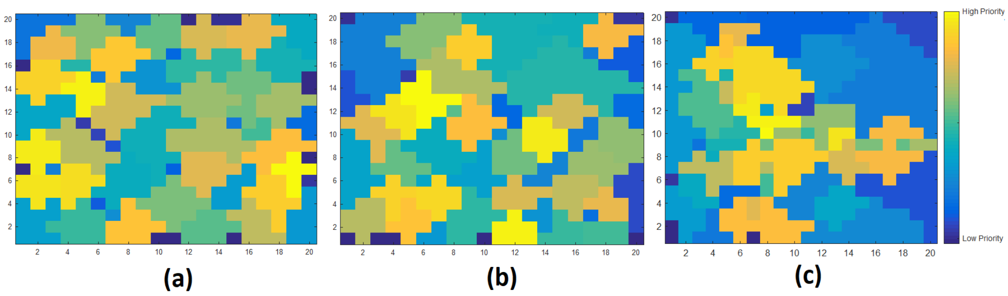

Figure 6.6: Spatiotemporal stable ERMs with $N = 20$ sampled at $T_S = 10$ min for (a) Beijing, (b) Jakarta, and (c) Singapore. Dark color ERMs have lower priorities over lighter ERMs.

6.3 Simulation Results and Discussion

In this section, we present extensive simulation results employing empirical mobility traces and city locations to evaluate the performance of the proposed EISHA-RSU allocation scheme.

6.3.1 Simulation Setup

We utilize three public transport mobility trace datasets, namely, (1) Beijing (BJS) [112], (2) Singapore (SIN) [117], and (3) Jakarta (JKT) [117]. The statistics of these three mobility traces, as well as the simulation parameters, are summarized in Table 6.1.

Table 6.1: Empirical Mobility Traces Attributes and Simulation Parameters

Urban City Parameter	**BJS**	**JKT**	**SIN**
Total Area (in $\approx$ km^2)	51	51	51
Grid Area (in $\approx$ m^2)	125,000	125,000	125,000
Total number of vehicles	24,845	28,000	16,174
$R_v = R_r$ (in m)	250	250	250
τ_c (#of vehicles)	248	280	161
Sampling Time T_S (in min)	10	10	10
ρ_0	0.25	0.25	0.25
W_g (in min)	2.8070	2.8070	2.8070
v (in m/s)	5.5556	5.5556	5.5556

By implementing Algorithm 6 in the three empirical mobility traces, the ERMs formed per city is shown in Fig. 6.6. By appropriately selecting the values of τ_c and ρ_0, there are approximately 50 spatiotemporal ERMs found in each urban map, both having single- and multiple-grid ERMs. If these values (τ_c and ρ_0) are too small, more single-grid ERMs are present. Increasing these values will form larger ERMs allowing less distinctions in the given urban map.

The generation of the static environment data (in MB) for each $g_{p,q}$ is governed by:

$$\gamma_{g_{p,q}} = 5000\big(\sin(150y + 15)\cos(150x + 20) + 1\big) \tag{6.15}$$

where x and y are the corresponding longitude and latitude coordinates of each $g_{p,q}$, respectively. On the other hand, we generate dynamic data by further subdividing a map grid into nine smaller grids. We then perform uniform selection across all sub-grids to randomly select locations where 'accidents' happen, since we do not have enough accident data of BJS, JKT, or SIN. Thus, the generation of additional dynamic environment data is governed by:

$$\delta_{g_{p,q}} = \xi \gamma_{g_{p,q}} \tag{6.16}$$

To represent the dynamic environment data available in each $g_{p,q}$, we let $\xi = 0.01$, thus, the importance factor $1 \leq \beta \leq 100$.

6.3.2 Stationarity of ERMs

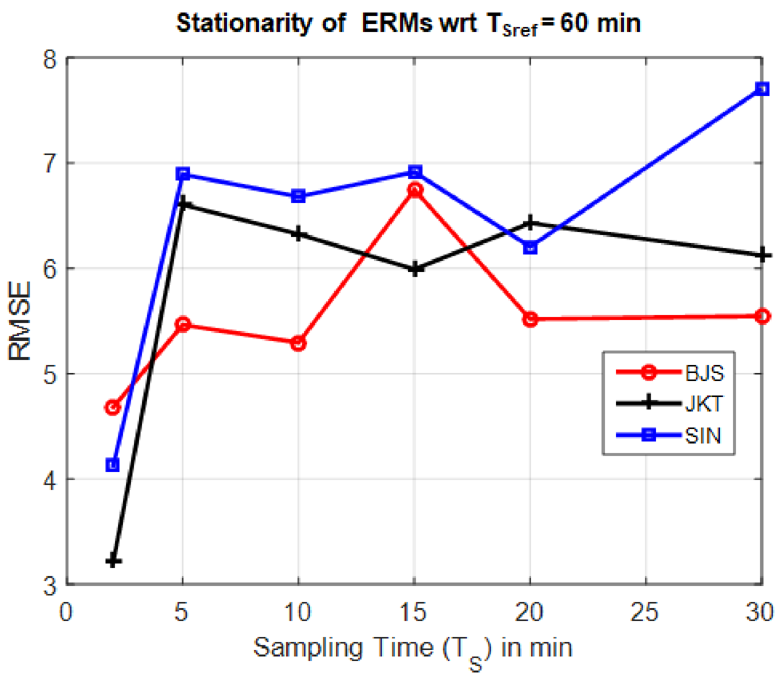

Figure 6.7: Determining stationarity of ERMs by varying the sampling time where each $g_{p,q,STS}$ is determined. $T_{S_{ref}} = 60$ min is the reference to test stationarity.

Determining the appropriate sampling time in establishing the ERMs is the first step. Faster sampling time yields a higher number of mobility traces and can depict the movement within the area under study, at the expense of high computational cost. On the other hand, slower sampling time reduces the mobility traces under study and may reduce vehicular network information necessary to provide valid results. To measure ERM stationarity, the root-mean-squared-error (RMSE) (6.17) of a formed ERM at sampling time T_S is compared to the ERM formed at a reference sampling time, $T_{S_{ref}} = 60$ min.

$$RMSE = \sqrt{\frac{\sum_{p=1}^{N}\sum_{q=1}^{N}\left[ERM_{p,q}(T_{S_{ref}}) - ERM_{p,q}(T_S)\right]^2}{N^2}} \tag{6.17}$$

Fig. 6.7 illustrates the effect of varying the sampling time from 2 to 30 min. Notice that as we decrease the value of the sampling time, there is an approximate flat response. If the ERMs are dynamic, the RMSE value should approach 400 (i.e., $N \times N$, $N = 20$), signifying that the vehicular trajectories change hastily. However, from Fig. 6.7, the average RMSE value is only six. The small average RMSE value implies that formed ERMs with sampling times $T_S = 5$, 10, 15, 20, and 30 min have minimal differences with $T_{S_{ref}} = 60$ min and can be regarded as stationary. With this finding, we can use any of these sampling times, and still be able to form approximately the same ERMs. We note that $T_S = 5$, 10, 15, 20, and 30 min are practical sampling times to sense the environment data.

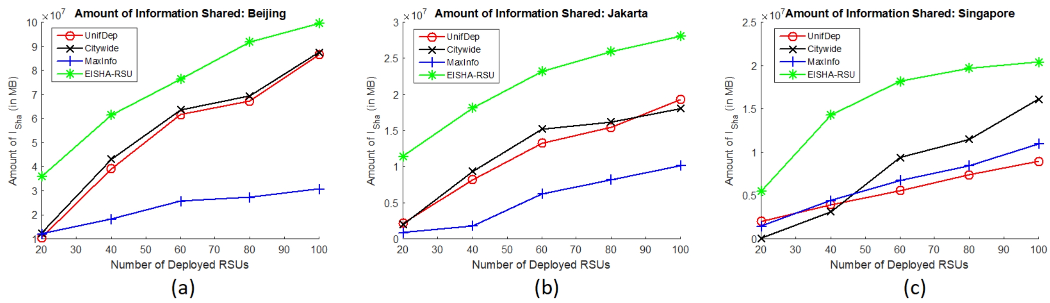

Figure 6.8: The amount of information shared in (a) Beijing, (b) Jakarta, and (c) Singapore by utilizing UnifDep, CityWide, MaxInfo, and EISHA-RSU deployment schemes.

6.3.3 Deployment Performance Evaluation

We compare the performance of the EISHA-RSU allocation scheme with the following benchmarks, as described below.

1. Uniform Deployment (UnifDep) [123]: The allocation of Ω_R effective positions follows the uniform distribution, where each of the $N \times N$ map grids has an equal probability of being selected. For this deployment strategy, simulations are run for 1000 times to capture uniformity.

2. Citywide Deployment (CityWide) [132]: The chosen Ω_R effective positions ensure maximum urban area coverage. The urban area is divided by Ω_R, and the effective positions are placed in linearly-spaced locations.

3. Maximum Information Deployment (MaxInfo) [133]: Effective positions are placed at Ω_R map grids where there is maximum information.

The amount of information shared by the four allocation schemes is illustrated in Fig. 6.8. This covers information within the RSU coverage area and those carried by vehicles beyond the RSU transmission range but within the allowable distance dictated by (6.9). For all deployment schemes, it is noticeable that as more effective positions are selected, more information is gathered and exchanged in the vehicular network. EISHA-RSU also performs the best by correctly placing effective positions in the urban area, therefore, enhancing the amount of information shared.

One may argue that the MaxInfo strategy should collect the most data, as its name implies. However, we note that once we placed the RSUs at locations where there is maximum information shared, we observed that the RSUs' transmission ranges are overlapping or at least adjacent to one another, thus, there is redundancy in shared information among RSUs in MaxInfo. This type of deployment allows EPs to be very close to each other, e.g., several meters, leading to the reduced coverage area and vehicular connectivity. Therefore, there will be less or no information-carrying vehicles found within a distance $\leq d$ that will arrive at an EP to deliver additional contents. On the other hand, the other three allocation schemes, especially EISHA-RSU, are able to accurately discriminate grid locations as possible effective positions to sense more environment data.

For all allocation methods, there is an assumption that RSUs are not connected. However, if all RSUs have a wireless or wired connection, then, EISHA-RSU will still be the scheme with the highest amount of shared information in the network. This is attributed to fact that the RSU allocation done by EISHA-RSU is well-positioned in the urban area to collect more base information, when compared to the other three benchmarks.

In addition, the fairness of the network's starvation is measured. To measure how fairly the installed RSUs collect information over the urban map, we relate the proportional fairness [135] to the average throughput, S_i, for each RSU assuming equal data upload and download rate. According to [135], proportional fairness happens when the deployed RSUs receive an equal amount of environment information. We then calculate the Jain's Network Starvation Fairness Index (6.18) to evaluate the performance of the deployment scheme as the number of deployed RSUs increases [136]. Ω_R is equal to the total number of deployed RSUs.

$$J(S_i) = \frac{\left(\sum_{i=1}^{\Omega_R} S_i\right)^2}{\Omega_R \sum_{i=1}^{\Omega_R} S_i^2} \tag{6.18}$$

A higher value of the Jain's starvation fairness index implies that there is almost an equal average amount of information collected by the RSUs al-

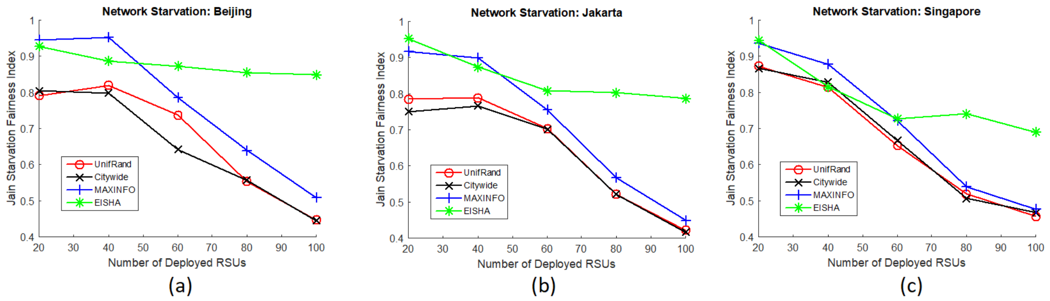

Figure 6.9: The Jain's network starvation fairness index indicates how much equal the average throughput there is per deployed RSU by each deployment scheme for (a) Beijing, (b) Jakarta, and (c) Singapore.

located on EPs across the urban map under study. Interpreting this index reveals that EISHA-RSU fairly allocates effective positions to allow RSUs to capture approximately equal amount of environment information, as shown in Fig. 6.9. As the number of deployed RSUs ($>$ 40 deployed RSUs) increases, it is noticeable that the other three deployment schemes have a fast rate of decreasing Jain's starvation fairness index when compared to EISHA-RSU. This decline in index value highlights that there is a huge discrepancy in the collected data among deployed RSUs. Though EISHA-RSU also experiences the index decline, the deployment method still achieves a higher value compared to the three, signifying a more balanced data collection.

We compare the performance of the deployment schemes according to its *Effectiveness* [137]. Let $I(D_i)$ and $C(D_i)$ denote the amount of information shared and coverage area of a deployment scheme i for a given number of deployed RSUs, respectively. The effectiveness of a deployment scheme i is defined by $E(D_i) = \big[I(D_i), C(D_i)\big]$. If $E(D_1) \succ E(D_2)$, then we say that any of these three conditions is true: 1) $\big[I(D_1) > I(D_2)$ and $C(D_1) > C(D_2)\big]$, 2) $\big[I(D_1) > I(D_2)$ and $C(D_1) = C(D_2)\big]$, or 3) $\big[I(D_1) = I(D_2)$ and $C(D_1) > C(D_2)\big]$.

Fig. 6.10 shows the *Effectiveness* plot against the amount of information shared and coverage area of all the deployment strategies for a given RSU deployment density. From Fig. 6.10, MaxInfo is the least effective in terms of the amount of information shared and coverage area. Given any RSU deployment density, i.e., 20, 40, or 60 deployed RSUs, it is evident that EISHA-RSU is the most effective since it captures the most amount of information with the widest coverage urban area. Hence, I(EISHA-RSU)$>$ I(UnifDep) and C(EISHA-RSU)$>$ C(UnifDep), I(EISHA-RSU)$>$ I(CityWide) and C(EISHA-RSU)$>$ C(CityWide), and I(EISHA-RSU)$>$ I(MaxInfo) and C(EISHA-RSU)$>$ C(MaxInfo). This result implies that EISHA-RSU properly allocates RSUs in effective positions that will both capture directly-sensed (either by RSU or vehicle) and single-hop environment data. In some cases, a lower deployment density while employing EISHA-RSU has more shared information than other deployment schemes at a higher deployment density, e.g., RSU20 vs RSU40 in the Jakarta and Singapore datasets.

However, the CityWide and UnifDep allocation methods provide contrasting results between each other. As noticed, CityWide offers higher amount of information shared but at a less coverage area than UnifDep. Though, it can be said that these small changes may denote that these two allocation techniques are interchangeable.

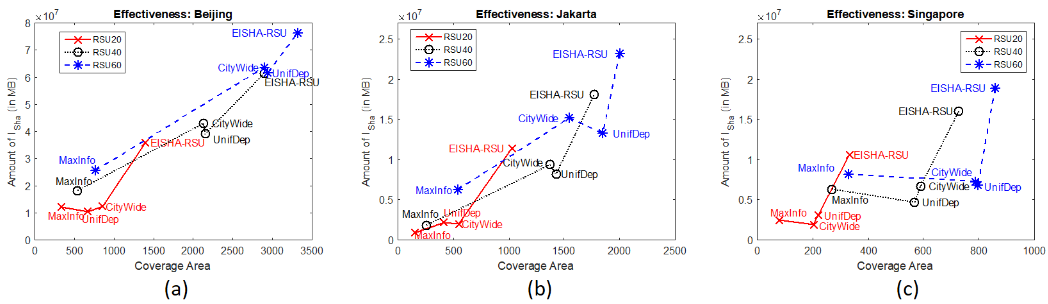

Figure 6.10: Performance comparison of each deployment scheme on the cities of (a) Beijing, (b) Jakarta, and (3) Singapore based on Effectiveness.

Assume that we want to have a coverage area of 2500 sub-grids for BJS ($\approx$ 35km^2), 2000 sub-grids for JKT ($\approx$ 28km^2), and 800 sub-grids for SIN

($\approx$ 12km^2), then the number of RSUs needed to be allocated in each urban map is listed in Table 6.2. For each urban map, EISHA-RSU, on the average, saves up to 16%, 21%, and 113% of RSUs when compared to UnifDep, CityWide, and MaxInfo, respectively.

Table 6.2: Required number of RSUs to be installed by each Allocation Scheme given a desired coverage area and amount of shared information.

RSU Allocation Scheme	**BJS**	**% Diff**	**JKT**	**% Diff**	**SIN**	**% Diff**
UnifDep	50	24.72	60	8.70	59	14.55
CityWide	50	24.72	70	24.00	58	12.84
MaxInfo	222	140.23	202	114.40	125	84.09
EISHA-RSU	39		55		51	

To evaluate network connectivity, we only consider how many vehicles are within the single-hop transmission range of its nearest EP, without the consideration of V2V communication. Fig. 6.11 displays the number of one-hop vehicles for each Ω_R deployment constraint. Because EISHA-RSU considers the appropriate spacing between EPs, the results show that EISHA-RSU captures more single-hop vehicles in the network, when compared to the other deployment schemes for all cities. We emphasize that this spacing between EPs covers the worst-case scenario when there are no leading vehicles. Hence, if V2V is allowed in areas without RSU coverage, data delivery time will be further reduced and would provide fresher and more up-to-date environment data.

In summary and supported by Figs. 6.8–6.11, our extensive simulation involving three empirical mobility traces have demonstrated that the proposed EISHA-RSU has enhanced information sharing by accurately selecting effective positions that can provide broader and fairer coverage, stable network connectivity, and maximum shared information. It is also evident that EISHA-RSU

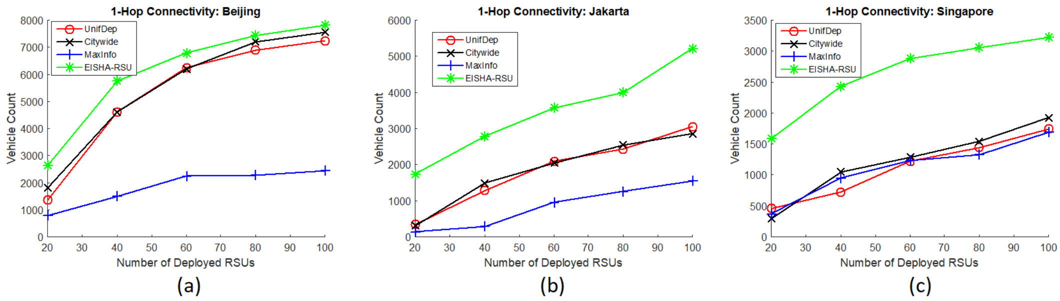

Figure 6.11: The amount of vehicles within single hop from an effective position in (a) Beijing, (b) Jakarta, and (c) Singapore that are still capable of delivering valid and up-to-date environment information.

has outperformed the three presented benchmarks.

6.4 Summary

In this work, we have presented an Enhanced Information SHAring RSU (EISHA-RSU) allocation scheme that targets maximal area coverage and vehicular connectivity, resulting to an enhanced amount of information shared between RSUs and vehicles in a vehicular network. To achieve these objectives, an urban map is partitioned according to its Effective Regions of Movement (ERMs) based on its vehicular capacity. EISHA-RSU then locates the effective positions (EPs) that are separated by an optimal distance where RSUs should be deployed to allow maximum information sharing and delivery of vehicular data. The performance of the proposed RSU allocation scheme has been validated by employing three urban empirical mobility traces. Simulation results have verified the fairness, effectiveness, and efficiency of EISHA-RSU when compared to three other benchmarks. In summary, EISHA-RSU allocates fewer RSUs to maximize information sharing, provides wider coverage, and improves connectivity in urban vehicular networks.

Chapter 7

Conclusion and Future Work

This chapter concludes the work done in this thesis. The main contributions are summarized and then tackles future directions of the work done.

7.1 Conclusion

This thesis investigated the issues and problems to achieve efficient data dissemination for intelligent connected vehicles. Real-time, efficient, and optimal methods for acquiring/broadcasting control and environment data have been proposed to achieve travel convenience and comfort, such as the reduced waiting time when exiting highway tollgates, up-to-date 3D road map data download, and maximized information exchange in vehicular networks. Extensive simulation employing both synthetic and empirical mobility traces suggested the efficiency of our proposed schemes.

The thesis's main contributions are enumerated as follows:

1. In Chapter 3, the practical problem of exiting highway tollgates and the amount of server utilization were tackled. The proposed solution presented a centralized control data dissemination implemented by fuzzy logic. The fuzzy logic controller coordinated movements of the intelli-

gent connected vehicles when selecting their exit tollgates to reduce the total average vehicle waiting time and average tollgate server utilization. These benefits have been achieved while maintaining the highway tollgate infrastructure constant and considering the ever-growing number of vehicles plying the highway. Extensive simulation has verified a 50% improvement in terms of waiting time and a 20% decrease in server utilization while considering both the homogeneous and non-homogeneous vehicular arrival. When the current traffic density exceeds a certain threshold, the centralized controller sent an early warning signal to all tollgate servers to ensure that their current service time is hastened to reduce build-up.

2. In Chapter 4, an optimal index coding transmission scheme was presented to disseminate environment data to numerous vehicles in a single and multi-junction setup by relying heavily on roadside infrastructure instead of unicast cellular communication. Based on data demand and availability, the bandwidth was conserved as more environment data (both static and dynamic) were transmitted with a fewer number of short-range broadcast transmissions due to hashing, octree compression, and index coding. While conserving the bandwidth, the proposed data dissemination scheme has been able to satisfy 100% and 80% of all vehicular static and dynamic data requests, respectively. By proving that most of the vehicles would meet only once in their trips, the proposed opportunistic download scheduling for multi-junction was also proven to be optimal. The processing overhead of the proposed index coding transmission scheme was also found to be on the average 34% below that of the traditional broadcasting technique. Robotic experiments and extensive simulation employing empirical taxi mobility traces have supported

the feasibility of the proposed data dissemination schemes.

3. In Chapter 5, maximum information sharing between intelligent connected vehicles and roadside fog-based infrastructures in a vehicular network was achieved by optimally deploying RSU fog nodes on information-rich and energy-efficient intersections. The proposed ISRA deployment scheme maximized the amount of V2I and I2V information exchanges while considering the infrastructure deployment capacity, the region's space mean speed, and transmission and traffic densities. The transmission density was calculated by employing the index-coding transmission scheme proposed in Chapter 4. Extensive simulations using empirical mobility traces were run to verify ISRA's performance. ISRA achieved an 83% energy efficiency while sharing on the average 20% more information when compared to three other deployment methods. Unique to ISRA was its ability to manage the computational load (processing and transmission) of the chosen optimal locations by considering the region's space mean speed.

   The optimal placement results of RSUs, as computed by ISRA has also been applied to the problem presented in Chapter 3. The deployment of RSUs on some highway sections before approaching the tollgate allows vehicles to run at uniform speed while also maintaining a uniform vehicular density. Without this early warning signal from the tollgate RSU, vehicles in each section run at different speeds, while the majority of the vehicles are jampacked at the tollgate section.

4. In Chapter 6, the RSU deployment problem was approached differently since the urban setup considered a broader set of locations. The EISHA-RSU allocation scheme exploited the concepts of effective regions of movement and effective positions to determine priority and positions

where RSUs can be allocated, respectively. Simulation results from employing three empirical mobility traces indicated that he proposed deployment strategy provided a fairer and effective means of achieving maximum information sharing between vehicles and infrastructure. EISHA-RSU also permitted nearby vehicles to act as sensing nodes to capture their surroundings and still be able to offload its environment data to the nearest RSU.

7.2 Future work

The thesis's results have presented efficient control and environment data dissemination schemes for intelligent connected vehicles. This study can be used as a groundwork for future research ideas, as stated below.

1. The concept of developing intelligent highway tollgates in Chapter 3 can be further extended by including (1) practical modeling of the manual service time and its operation, (2) economic analysis on the installation of manual and electronic tollgate collection booths, and (3) driver maneuver attitude. The modeling of manual service time should provide a range of tollgate service operations, particularly during peak time, while the economic analysis will push vehicles to incorporate electronic readers for toll fee collection and reloading. Also, the economic analysis will dictate how many electronic readers and manual collection toll booths must be installed. The driver maneuver attitude encompasses swerving and acceleration/deceleration behaviors. When considered in the simulation, vehicles found on the right/left/center side of the highway should be given topmost queuing priority in the right/left/center tollgates, respectively. The stopping of vehicles entering manual tollgates surely adds to the delay and effectively reduces the server utilization because of wasted

time. Lastly, the results of this thesis can be used and extended for empirical tollgate analysis, e.g., Philippine expressways. Finally, machine and deep learning can be employed to derive the accurate model of these transportation systems. In particular, the control of highway sections can be done by incorporating reinforcement learning in the feedback loop.

2. The simulation work done in Chapter 4 employs 3D LIDAR data for representing both static and dynamic environments. However, there is less work done in the evaluation of dynamic data, particularly with the presence of pedestrians and other vehicles, and in understanding the dynamic data parameters such as delivery delay, dynamic content annotations, etc. Other data types can also be explored for various use-case scenarios, such as 2D representations for capturing pedestrians crossing the intersection or quick road environment modeling, and sensor measurement for obstacle avoidance or identification. Also, schedule-based or more sophisticated uploading schemes can be utilized to reduce information flooding in the roadside infrastructure during the transfer of vehicular information. An example of such schedule-based data uploading is cluster formation and leader selection. Finally, there is a need to investigate how Bloom filters and other data structures can effectively lower the amount of repeated and redundant uploads and lessen false upload positives.

3. The works in Chapters 5 and 6 can be extended by studying pre-defined public spaces such as bus stops, taxi stands, and old phone booth sites as possible RSU deployment locations. Economically, bus stops, taxi stands, and phone booth sites will already identify which locations have more vehicles and people traffic. This characteristic can then be exploited since people, aside from vehicles, can also have surrounding information

because of their mobile phones. Deployed RSUs are now operating both for people and vehicle convenience. However, security and privacy issues must be thoroughly addressed not to compromise the users. At the same time, schemes that give incentives to vehicles and people for offloading appropriate and up-to-date environment data must also be created to have a comprehensive set of data sources.

Appendices

Appendix A

Extracting Mobility Traces

The taxi mobility trace datasets contain GPS locations of taxis plying an urban city. Each GPS location is sampled every T_S sec from 00:00 to 24:00 daily. The mobility traces dataset follows the format given in Table A.

Table A.1: Data format of the taxi mobility trace datasets

Longitude	Latitude	Time stamp	Taxi ID

Shown in Code Listing A.1 is the Matlab source code to extract information from a Taxi Mobility Traces dataset on a given day **Day**. It extracts taxi GPS traces given the coordinates of the region's bounding box under study. The description of each code part is discussed below.

1. Part 1 loads the two-part daily dataset of the mobility traces and stores it in a single variable **DayData**. (Note: Except for the Beijing mobility traces dataset, the other datasets have only one file to be loaded.)

2. Part 2 ensures that all mobility traces are sampled starting at 00:00 and terminating at 24:00.

3. Part 3 pre-processes the raw data by converting it to its natural form, e.g., longitude and latitude are given in integer format.

4. Part 4 locates the GPS traces, time stamp, and Taxi ID inside the region's bounding box.

Listing A.1: Source code for extracting mobility traces information from the Beijing dataset.

```
function ExtractedDay = ExtractTaxiTraces(Lat1,
Lat2,Long1,Long2,Day)
% ———— Part 1 ———— %
load (['TaxiDay0' num2str(Day) 'Part01.mat']);
load (['TaxiDay0' num2str(Day) 'Part02.mat']);
DayData = vertcat(TaxiDayData1,TaxiDayData2);
clear TaxiDayData1
clear TaxiDayData2
% ———— Part 2 ———— %
day = Day;
if day > 2
factor = (day-1)*86400;
else
factor = 0;
end
Time = DayData(:,3)- factor*ones(length(DayData(:,3)),1);
% ———— Part 3 ———— %
Long = DayData(:,1)/10e4;
Lat = DayData(:,2)/10e4;
TaxiID = DayData(:,4);

% ———— Part 4 ———— %
boundary = [Long1 Long1 Long2 Long2 Long1; Lat1 Lat2
 Lat2 Lat1 Lat1];
[ix] = find(Long>Long1 & Long<Long2);
[iy] = find(Lat(ix)>Lat1 & Lat(ix)<Lat2);
xLong1 = Long(ix(iy));
```

```
yLat1 = Lat(ix(iy));
Time1 = Time(ix(iy));
TaxiID1 = TaxiID(ix(iy));
ExtractedDay = [xLong1 yLat1 Time1 TaxiID1];
end
```

Bibliography

[1] A. Chesterton, "How many cars are there in the world?" *URL https://www.carsguide.com.au/car-advice/how-many-cars-are-there-in-the-world-70629*, 2018.

[2] A. Eskandarian, *Handbook of intelligent vehicles*, 2012.

[3] C. Sommer and F. Dressler, "Information dissemination in vehicular networks," *Vehicular Communications and Networks*, pp. 75–93, 2015.

[4] K. F. Chu, E. R. Magsino, I. W.-H. Ho, and C.-K. Chau, "Index coding of point cloud-based road map data for autonomous driving," *Proceedings of the 85th IEEE Vehicular Technology Conference (VTC Spring)*, pp. 1–7, 2017.

[5] Y. Xie, I. W.-H. Ho, and E. R. Magsino, "The modeling and cross-layer optimization of 802.11 p vanet unicast," *IEEE access*, vol. 6, pp. 171–186, 2017.

[6] A. J. Ghandour, M. Di Felice, H. Artail, and L. Bononi, "Dissemination of safety messages in ieee 802.11 p/wave vehicular network: Analytical study and protocol enhancements," *Pervasive and mobile computing*, vol. 11, pp. 3–18, 2014.

[7] C.-F. Chiasserini, R. Gaeta, M. Garetto, M. Gribaudo, and M. Sereno, "Efficient broadcasting of safety messages in multihop vehicular net-

works," *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium*, pp. 8–pp, 2006.

[8] H. A. Omar, N. Lu, and W. Zhuang, "Wireless access technologies for vehicular network safety applications," *IEEE Network*, vol. 30, no. 4, pp. 22–26, 2016.

[9] M. Caliskan, D. Graupner, and M. Mauve, "Decentralized discovery of free parking places," *Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, pp. 30–39, 2006.

[10] E. Kim and G. Medioni, "Urban scene understanding from aerial and ground lidar data," *Machine Vision and Applications*, vol. 22, no. 4, pp. 691–703, 2011.

[11] "Ford Campus Vision and LIDAR Data Set," *URL http://robots.engin.umich.edu/SoftwareData/Ford*, 2016.

[12] A. Mchergui, T. Moulahi, B. Alaya, and S. Nasri, "A survey and comparative study of qos aware broadcasting techniques in vanet," *Telecommunication Systems*, vol. 66, no. 2, pp. 253–281, 2017.

[13] I. W.-H. Ho and E. R. Magsino, "Performance evaluation of the 802.11p vanet protocol based on realistic vehicular traffic flow," *Vehicular Networks: Applications, Performance Analysis and Challenges*, pp. 1–30, 2019.

[14] O. Chakroun, S. Cherkaoui, and J. Rezgui, "Mudds: Multi-metric unicast data dissemination scheme for 802.11 p vanets," *Proceedings of the 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1074–1079, 2012.

[15] "Velodyne LIDAR HDL-64E," *URL http://velodynelidar.com/hdl-64e.html*, 2016.

[16] Y. Wang, W. Bin, and Z. Hui, "The national freeway control system-further development with yy logic theory," *Proceedings of IEEE International Conference on Industrial Technology-ICIT'94*, pp. 729–733, 1994.

[17] E. R. Magsino and I. W. Ho, "An intelligent highway tollgate queue selector for improving server utilization and vehicle waiting time," *Proceedings of the IEEE Region 10 Symposium (TENSYMP)*, pp. 271–276, 2016.

[18] S. El Rouayheb, A. Sprintson, and C. Georghiades, "On the index coding problem and its relation to network coding and matroid theory," *IEEE Transactions on Information Theory*, vol. 56, no. 7, pp. 3187–3195, 2010.

[19] Z. Bar-Yossef, Y. Birk, T. Jayram, and T. Kol, "Index coding with side information," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1479–1494, 2011.

[20] A. SINGHVI and K. RUSSELL, "Inside the Self-Driving Tesla Fatal Accident," *URL https://www.nytimes.com/interactive/2016/07/01/business/inside-tesla-accident.html*, 2019.

[21] F. E. Sancar, B. Fidan, J. P. Huissoon, and S. L. Waslander, "Mpc based collaborative adaptive cruise control with rear end collision avoidance," *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 516–521, 2014.

[22] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog computing: A taxonomy, survey and future directions," *Internet of everything*, pp. 103–130, 2018.

[23] D. C. Klonoff, "Fog computing and edge computing architectures for processing data from diabetes devices connected to the medical internet of things," 2017.

[24] N. Chen, Y. Yang, T. Zhang, M.-T. Zhou, X. Luo, and J. K. Zao, "Fog as a service technology," *IEEE Communications Magazine*, vol. 56, no. 11, pp. 95–101, 2018.

[25] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on fog computing: architecture, key technologies, applications and open issues," *Journal of network and computer applications*, vol. 98, pp. 27–42, 2017.

[26] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *Journal of Systems Architecture*, vol. 98, pp. 289–330, 2019.

[27] G. A. Association, "The Case for Cellular V2X for Safety and Cooperative Driving," *URL https://5gaa.org/wp-content/uploads/2017/10/5GAA-whitepaper-23-Nov-2016.pdf*, 2020.

[28] K. Liang, L. Zhao, X. Chu, and H.-H. Chen, "An integrated architecture for software defined and virtualized radio access networks with fog computing," *IEEE Network*, vol. 31, no. 1, pp. 80–87, 2017.

[29] Z. Wen, R. Yang, P. Garraghan, T. Lin, J. Xu, and M. Rovatsos, "Fog orchestration for iot services: issues, challenges and directions," *IEEE Internet Computing*, vol. 21, no. 2, pp. 16–24, 2017.

[30] S. Tomovic, K. Yoshigoe, I. Maljevic, and I. Radusinovic, "Software-defined fog network architecture for iot," *Wireless Personal Communications*, vol. 92, no. 1, pp. 181–196, 2017.

[31] C. Huang, R. Lu, and K.-K. R. Choo, "Vehicular fog computing: architecture, use case, and security and forensic challenges," *IEEE Communications Magazine*, vol. 55, no. 11, pp. 105–111, 2017.

[32] C. Lochert, B. Scheuermann, M. Caliskan, and M. Mauve, "The feasibility of information dissemination in vehicular ad-hoc networks," *Proceedings of the Fourth Annual Conference on Wireless on Demand Network Systems and Services*, pp. 92–99, 2007.

[33] L. He, Q. Gao, S. Li, and Z. Peng, "Optimal allocation of operating toll booths at highway toll station," *Proceedings of the Sixth International Conference on Business Intelligence and Financial Engineering*, pp. 468–471, 2013.

[34] K. A. Majid, Z. Yusoff, and A. A. Jemain, "A new mathematical model for traffic systems: the basic traffic unit," *Creative Practices in Language Learning and Teaching (CPLT) Special Issue: Generating New Knowledge through Best Practices in Computing and Mathematical Sciences*, vol. 7, no. 1, pp. 34–41, 2019.

[35] I. W.-H. Ho, K. K. Leung, and J. W. Polak, "Stochastic model and connectivity dynamics for vanets in signalized road systems," *IEEE/ACM Transactions on Networking (TON)*, vol. 19, no. 1, pp. 195–208, 2011.

[36] "100-km traffic jam reappears on N. China highway," *URL http://www.china.org.cn/china/2011-07/20/content_23030293.html*, 2011.

[37] "'Carmageddon': 50-lane traffic jam in China causes chaos," *URL https://www.smh.com.au/world/carmageddon-50lane-traffic-jam-in-china-causes-chaos-20151009-gk4tq9.html*, 2015.

[38] L. C. Edie, "Traffic delays at toll booths," *Journal of the operations research society of America*, vol. 2, no. 2, pp. 107–138, 1954.

[39] D. Gazis and R. Gomory, "Delays at toll booths–why wait in line?" *Transportation Quarterly*, vol. 48, no. 2, 1994.

[40] E. Correa, C. Metzner, and N. Niño, "Tollsim: Simulation and evaluation of toll stations," *International Transactions in Operational Research*, vol. 11, no. 2, pp. 121–138, 2004.

[41] J. Danko and V. Gulewicz, "Operational planning for electronic toll collection: a unique approach to computer modeling/analysis," Institute of Electrical and Electronics Engineers (IEEE), Tech. Rep., 1991.

[42] Y. Du, S. Yu, and Y. Ji, "Allocation of pay-on-foot and pay-at-exit tollbooths based on 3 simulation of queuing at parking exit 4," *Proceedings of the 95th Annual Meeting of the Transportation Research 32 Board and publication in the Transportation Research Record*, pp. 1–13, 2015.

[43] T. Erdelić, T. Carić, and M. Ravlić, "Predicting waiting times at intersections," *Proceedings of the International Symposium ELMAR*, pp. 31–34, 2017.

[44] "AVERAGE DAILY TRAFFIC COUNT," *URL http://www.trb.gov.ph/8-transparency/39-vehicle-classifications.html/average-daily-traffic-count*, 2020.

[45] K. N. Karna, "Artificial intelligence in intelligent transportation systems," 1995.

[46] M. Davarynejad and J. Vrancken, "A survey of fuzzy set theory in intelligent transportation: State of the art and future trends," *Proceedings of*

the IEEE International Conference on Systems, Man and Cybernetics, pp. 3952–3958, 2009.

[47] N. K. Swain, "A survey of application of fuzzy logic in intelligent transportation systems (its) and rural its," *Proceedings of the IEEE SoutheastCon*, pp. 85–90, 2006.

[48] P. Albertos and A. Sala, "Fuzzy logic controllers. advantages and drawbacks," vol. 3, pp. 833–844, 1998.

[49] A. I. Al-Odienat and A. A. Al-Lawama, "The advantages of pid fuzzy controllers over the conventional types," *American Journal of Applied Sciences*, vol. 5, no. 6, pp. 653–658, 2008.

[50] R. Abduljabbar, H. Dia, S. Liyanage, and S. A. Bagloee, "Applications of artificial intelligence in transport: An overview," *Sustainability*, vol. 11, no. 1, p. 189, 2019.

[51] H. Cheng, *Autonomous intelligent vehicles: theory, algorithms, and implementation.* Springer Science & Business Media, 2011.

[52] M. Chaqfeh, A. Lakas, and I. Jawhar, "A survey on data dissemination in vehicular ad hoc networks," *Vehicular Communications*, vol. 1, no. 4, pp. 214–225, 2014.

[53] L. Aparecido, "Data dissemination in vehicular networks: Challenges, solutions, and future perspectives," *Proceedings of the 7th International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–5, 2015.

[54] F. Ye, S. Roy, and H. Wang, "Efficient data dissemination in vehicular ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 4, pp. 769–779, 2012.

[55] A. Nandan, S. Das, G. Pau, M. Gerla, and M. Sanadidi, "Co-operative downloading in vehicular ad-hoc wireless networks," *Proceedings of the Second Annual Conference on Wireless On-demand Network Systems and Services*, pp. 32–41, 2005.

[56] K. Lee and I. Yap, "Cartorrent: A bit-torrent system for vehicular ad-hoc networks," *Los Angeles*, 2006.

[57] I. Leontiadis and C. Mascolo, "Opportunistic spatio-temporal dissemination system for vehicular networks," *Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking*, pp. 39–46, 2007.

[58] F. J. Ros, P. M. Ruiz, and I. Stojmenovic, "Acknowledgment-based broadcast protocol for reliable and efficient data dissemination in vehicular ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 1, pp. 33–46, 2010.

[59] ——, "Reliable and efficient broadcasting in vehicular ad hoc networks," *Proceedings of the IEEE 69th Vehicular Technology Conference*, pp. 1–5, 2009.

[60] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode, "Trafficview: traffic data dissemination using car-to-car communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, no. 3, pp. 6–19, 2004.

[61] D. Wu, Y. Zhang, J. Luo, and R. Li, "Efficient data dissemination by crowdsensing in vehicular networks," *Proceedings of the 22nd International Symposium of Quality of Service (IWQoS)*, pp. 314–319, 2014.

[62] M. Shahverdy, M. Fathy, and S. Yousefi, "Scheduling algorithm for vehicle to road-side data distribution," *High Performance Networking, Com-*

puting, Communication Systems, and Mathematical Foundations, pp. 22–30, 2009.

[63] V. Vijayakumar and K. S. Joseph, "Adaptive load balancing schema for efficient data dissemination in vehicular ad-hoc network vanet," *Alexandria Engineering Journal*, vol. 58, no. 4, pp. 1157–1166, 2019.

[64] G. M. N. Ali, E. Chan, and W. Li, "On scheduling data access with cooperative load balancing in vehicular ad hoc networks (vanets)," *The Journal of Supercomputing*, vol. 67, no. 2, pp. 438–468, 2014.

[65] K. Paridel, J. Balen, Y. Berbers, and G. Martinovic, "Vvid: A delay tolerant data dissemination architecture for vanets using v2v and v2i communication," *Proceedings of the MOBILITY, International Conference on Mobile Services, Resources, and Users*, pp. 151–156, 2012.

[66] J. Zhao, Y. Zhang, and G. Cao, "Data pouring and buffering on the road: A new data dissemination paradigm for vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3266–3277, 2007.

[67] P. Tomar, B. K. Chaurasia, and G. Tomar, "State of the art of data dissemination in vanets," *International Journal of Computer Theory and Engineering*, vol. 2, no. 6, p. 957, 2010.

[68] Y. Li, X. Zhu, D. Jin, and D. Wu, "Multiple content dissemination in roadside-unit-aided vehicular opportunistic networks," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 8, pp. 3947–3956, 2014.

[69] H. Hartenstein and K. Laberteaux, *VANET: vehicular applications and inter-networking technologies.* John Wiley & Sons, 2009, vol. 1.

[70] S. El Rouayheb, A. Sprintson, and C. Georghiades, "On the index coding problem and its relation to network coding and matroid theory," *IEEE Transactions on Information Theory*, vol. 56, no. 7, pp. 3187–3195, 2010.

[71] S.-H. Lee, U. Lee, K.-W. Lee, and M. Gerla, "Content distribution in vanets using network coding: The effect of disk i/o and processing o/h," *Proceedings of the 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pp. 117–125, 2008.

[72] K. Liu, J. K.-Y. Ng, J. Wang, V. C. Lee, W. Wu, and S. H. Son, "Network-coding-assisted data dissemination via cooperative vehicle-to-vehicle/-infrastructure communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1509–1520, 2015.

[73] Q. Wang, P. Fan, and K. B. Letaief, "On the joint v2i and v2v scheduling for cooperative vanets with network coding," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 62–73, 2011.

[74] Cisco, "Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are," *URL https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf*, 2015.

[75] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 416–464, 2017.

[76] C. Perera, Y. Qin, J. C. Estrella, S. Reiff-Marganiec, and A. V. Vasilakos, "Fog computing for sustainable smart cities: A survey," *ACM Computing Surveys (CSUR)*, vol. 50, no. 3, pp. 1–43, 2017.

[77] J. Chi, Y. Jo, H. Park, T. Hwang, and S. Park, "An effective rsu allocation strategy for maximizing vehicular network connectivity," *International Journal of Control and Automation*, vol. 6, no. 4, pp. 259–270, 2013.

[78] M. Rezaei, H. Noori, D. Rahbari, and M. Nickray, "Refocus: A hybrid fog-cloud based intelligent traffic re-routing system," *Proceedings of the 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, pp. 0992–0998, 2017.

[79] J. Liu, J. Li, L. Zhang, F. Dai, Y. Zhang, X. Meng, and J. Shen, "Secure intelligent traffic light control using fog computing," *Future Generation Computer Systems*, vol. 78, pp. 817–824, 2018.

[80] Y. Lai, F. Yang, J. Su, Q. Zhou, T. Wang, L. Zhang, and Y. Xu, "Fog-based two-phase event monitoring and data gathering in vehicular sensor networks," *Sensors*, vol. 18, no. 1, pp. 82–104, 2017.

[81] S.-I. Sou and O. K. Tonguz, "Enhancing vanet connectivity through roadside units on highways," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 8, pp. 3586–3602, 2011.

[82] A. Ahmad, S. Din, A. Paul, G. Jeon, M. Aloqaily, and M. Ahmad, "Real-time route planning and data dissemination for urban scenarios using the internet of things," *IEEE Wireless Communications*, vol. 26, no. 6, pp. 50–55, 2019.

[83] X. Lin, R. Lu, C. Zhang, H. Zhu, P.-H. Ho, and X. Shen, "Security in vehicular ad hoc networks," *IEEE Communications Magazine*, vol. 46, no. 4, 2008.

[84] Y. Liang, H. Liu, and D. Rajan, "Optimal placement and configuration of roadside units in vehicular networks," *Proceedings of the 75th Vehicular Technology Conference (VTC Spring)*, pp. 1–6, 2012.

[85] J. Chi, Y. Jo, H. Park, and S. Park, "Intersection-priority based optimal rsu allocation for vanet," *Proceedings of the Fifth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 350–355, 2013.

[86] C. Lochert, B. Scheuermann, C. Wewetzer, A. Luebke, and M. Mauve, "Data aggregation and roadside unit placement for a vanet traffic information system," *Proceedings of the Fifth ACM international workshop on VehiculAr Inter-NETworking*, pp. 58–65, 2008.

[87] J. F. Sarubbi, F. V. Martins, and C. M. Silva, "A genetic algorithm for deploying roadside units in vanets," *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pp. 2090–2097, 2016.

[88] C. Wang, X. Li, F. Li, and H. Lu, "A mobility clustering-based roadside units deployment for vanet," *Proceedings of the 16th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 1–6, 2014.

[89] O. Trullols, M. Fiore, C. Casetti, C.-F. Chiasserini, and J. B. Ordinas, "Planning roadside infrastructure for information dissemination in intelligent transportation systems," *Computer Communications*, vol. 33, no. 4, pp. 432–442, 2010.

[90] B. Aslam, F. Amjad, and C. C. Zou, "Optimal roadside units placement in urban areas for vehicular networks," *Proceedings of the IEEE Symposium on Computers and Communications (ISCC)*, pp. 000 423–000 429, 2012.

[91] J. Lee and C. Kim, “A roadside unit placement scheme for vehicular telematics networks,” *AST/UCMA/ISA/ACN*, pp. 196–202, 2010.

[92] F. Zou, J. Zhong, W. Wu, D.-Z. Du, and J. Lee, “Energy-efficient roadside unit scheduling for maintaining connectivity in vehicle ad-hoc network,” *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication*, p. 64, 2011.

[93] S. Mehar, S. M. Senouci, A. Kies, and M. M. Zoulikha, “An optimized roadside units (rsu) placement for delay-sensitive applications in vehicular networks,” *Proceedings of the 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 121–127, 2015.

[94] L. Xue, Y. Yang, and D. Dong, “Roadside infrastructure planning scheme for the urban vehicular networks,” *Transportation Research Procedia*, vol. 25, pp. 1380–1396, 2017.

[95] C. M. Silva, A. L. Aquino, and W. Meira Jr, “Deployment of roadside units based on partial mobility information,” *Computer Communications*, vol. 60, pp. 28–39, 2015.

[96] A. Abdrabou and W. Zhuang, “Probabilistic delay control and road side unit placement for vehicular ad hoc networks with disrupted connectivity,” *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 129–139, 2011.

[97] A. Jalooli, M. Song, and X. Xu, “Delay efficient disconnected rsu placement algorithm for vanet safety applications,” *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2017.

[98] Y. Wang, J. Zheng, and N. Mitton, "Delivery delay analysis for roadside unit deployment in vehicular ad hoc networks with intermittent connectivity," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 8591–8602, 2016.

[99] Y. Sun, X. Lin, R. Lu, X. Shen, and J. Su, "Roadside units deployment for efficient short-time certificate updating in vanets," *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 1–5, 2010.

[100] Y. Liu, J. Ma, J. Niu, Y. Zhang, and W. Wang, "Roadside units deployment for content downloading in vehicular networks," *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 6365–6370, 2013.

[101] S.-W. Wang and M.-Y. Chang, "Roadside units allocation algorithms for certificate update in vanet environments," *Proceedings of the 17th Asia-Pacific Conference on Communications (APCC)*, pp. 472–477, 2011.

[102] A. B. Reis, S. Sargento, and O. K. Tonguz, "Parked cars are excellent roadside units," *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[103] O. K. Tonguz and W. Viriyasitavat, "Cars as roadside units: a self-organizing network solution," *IEEE Communications Magazine*, vol. 51, no. 12, pp. 112–120, 2013.

[104] H. Zheng and Y. Deng, "Evaluation method based on fuzzy relations between dempster–shafer belief structure," *International Journal of Intelligent Systems*, vol. 33, no. 7, pp. 1343–1363, 2018.

[105] Y. Wang, V. Menkovski, I. W.-H. Ho, and M. Pechenizkiy, “Vanet meets deep learning: the effect of packet loss on the object detection performance,” *Proceedings of the IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pp. 1–5, 2019.

[106] I. W.-H. Ho, S. C.-K. Chau, E. R. Magsino, and K. Jia, “Efficient 3d road map data exchange for intelligent vehicles in vehicular fog networks,” Technical Report, The Hong Kong Polytechnic University, 2019.

[107] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, “Towards 3d point cloud based object maps for household environments,” *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008.

[108] J. Kammerl, N. Blodow, R. B. Rusu, S. Gedikli, M. Beetz, and E. Steinbach, “Real-time compression of point cloud streams,” *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 778–785, 2012.

[109] R. Schnabel and R. Klein, “Octree-based point-cloud compression,” *Spbg*, vol. 6, pp. 111–120, 2006.

[110] Y. Huang, J. Peng, C.-C. J. Kuo, and M. Gopi, “A generic scheme for progressive point cloud coding,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 2, pp. 440–453, 2008.

[111] M. Mitzenmacher and E. Upfal, *Probability and computing.* Cambridge University Press, 2005.

[112] Y. Li, D. Jin, Z. Wang, L. Zeng, and S. Chen, “Exponential and power law distribution of contact duration in urban vehicular ad hoc networks,” *IEEE Signal Processing Letters*, vol. 20, no. 1, pp. 110–113, 2013.

[113] X. Liu, J. Biagioni, J. Eriksson, Y. Wang, G. Forman, and Y. Zhu, "Mining large-scale, sparse gps traces for map inference: comparison of approaches," *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 669–677, 2012.

[114] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "CRAWDAD dataset epfl/mobility (v. 2009-02-24)," Downloaded from https://crawdad.org/epfl/mobility/20090224, 2009.

[115] D. Zhang, J. Zhao, F. Zhang, and T. He, "Urbancps: a cyber-physical system based on multi-source big infrastructure data for heterogeneous model integration," *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*, pp. 238–247, 2015.

[116] L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici, and A. Rabuffi, "CRAWDAD dataset roma/taxi (v. 2014-07-17)," Downloaded from https://crawdad.org/roma/taxi/20140717, July 2014.

[117] X. Huang, Y. Yin, S. Lim, G. Wang, B. Hu, J. Varadarajan, S. Zheng, A. Bulusu, and R. Zimmermann, "Grab-posisi: An extensive real-life gps trajectory dataset in southeast asia," *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Prediction of Human Mobility*, pp. 1–10, 2019.

[118] B. Donovan and D. B. Work, "Empirically quantifying city-scale transportation system resilience to extreme events," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 333–346, 2017.

[119] G. N. Ali, M. A. Rahman, S. Samantha, Y. Gao, P. H. Chong, and Y. Guan, "Efficient coding based heterogeneous data access in vehicular networks," *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 1–6, 2017.

[120] T. V. Mathew and K. K. Rao, "Fundamental relations of traffic flow," *Introduction to Transportation Engineering*, vol. 1, pp. 1–8, 2006.

[121] G. Pandey, J. R. McBride, and R. M. Eustice, "Ford campus vision and lidar data set," *The International Journal of Robotics Research*, vol. 30, no. 13, pp. 1543–1552, 2011.

[122] R. C. Das, P. P. Purohit, T. Alam, and M. Chowdhury, "Location based atm locator system using openstreetmap," *Proceedings of the 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, pp. 1–6, 2014.

[123] J. Barrachina, P. Garrido, M. Fogue, F. J. Martinez, J.-C. Cano, C. T. Calafate, and P. Manzoni, "Road side unit deployment: A density-based approach," *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 3, pp. 30–39, 2013.

[124] M. Xu, J. Wu, M. Liu, Y. Xiao, H. Wang, and D. Hu, "Discovery of critical nodes in road networks through mining from vehicle trajectories," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2018.

[125] M. Papageorgiou, I. Papamichail, A. Messmer, and Y. Wang, "Traffic simulation with metanet," *Fundamentals of traffic simulation*, pp. 399–430, 2010.

[126] Z. Hou, J.-X. Xu, and J. Yan, "An iterative learning approach for density control of freeway traffic flow via ramp metering," *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 1, pp. 71–97, 2008.

[127] H. J. Qiu, I. W.-H. Ho, K. T. Chi, and Y. Xie, "A methodology for studying 802.11 p vanet broadcasting performance with practical vehi-

cle distribution," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4756–4769, 2015.

[128] I. W.-H. Ho, K. K. Leung, and J. W. Polak, "Connectivity dynamics for vehicular ad-hoc networks in signalized road systems," *Proceedings of the 21st International Teletraffic Congress*, pp. 1–8, 2009.

[129] "Fuel-efficient driving techniques," *URL https://www.nrcan.gc.ca/energy-efficiency/energy-efficiency-transportation/personal-vehicles/fuel-efficient-driving-techniques/21038*, 2020.

[130] Q. Zhao, Y. Shi, Q. Liu, and P. Fränti, "A grid-growing clustering algorithm for geo-spatial data," *Pattern Recognition Letters*, vol. 53, pp. 77–84, 2015.

[131] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, and O. Tonguz, "Routing in sparse vehicular ad hoc wireless networks," *IEEE journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1538–1556, 2007.

[132] P.-C. Lin, "Optimal roadside unit deployment in vehicle-to-infrastructure communications," *Proceedings of the 12th International Conference on ITS Telecommunications*, pp. 796–800, 2012.

[133] E. R. Magsino and I. W.-H. Ho, "Roadside unit allocation for fog-based information sharing in vehicular networks," *Proceedings of the 1st ACM International Workshop on Smart Cities and Fog Computing*, pp. 7–12, 2018.

[134] V. Latora and M. Marchiori, "Efficient behavior of small-world networks," *Physical review letters*, vol. 87, no. 19, p. 198701, 2001.

[135] L. B. Jiang and S. C. Liew, "Proportional fairness in wireless lans and ad hoc networks," *IEEE Wireless Communications and Networking Conference, 2005*, vol. 3, pp. 1551–1556, 2005.

[136] R. K. Jain, D.-M. W. Chiu, W. R. Hawe *et al.*, "A quantitative measure of fairness and discrimination," *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA*, 1984.

[137] I. W.-H. Ho and S. C. Liew, "Impact of power control on performance of ieee 802.11 wireless networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 11, pp. 1245–1258, 2007.