Energy Efficient and Co-operative Vehicle Scheduling Algorithms for Data Dissemination in Highway Vehicular Networks

Submitted in partial fulfillment of the requirements

for the award of the degree of

DOCTOR OF PHILOSOPHY

by

Satish Vemi Reddy

(Roll No. 717045)

Under the guidance of

Dr. Rashmi Ranjan Rout

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
TELANGANA - 506004, INDIA

OCTOBER 2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
TELANGANA - 506004, INDIA

THESIS APPROVAL FOR Ph.D.

This is to certify that the thesis entitled, Energy Efficient and Co-operative Vehicle Scheduling Algorithms for Data Dissemination in Highway Vehicular Networks, submitted by Mr. Satish Vemi Reddy [Roll No. 717045] is approved for the degree of DOCTOR OF PHILOSOPHY at National Institute of Technology Warangal.

Examiner

Research Supervisor
Dr. Rashmi Ranjan Rout
Dept. of Computer Science and Engg.
NIT Warangal
India

Chairman
Prof. P. Radha Krishna
Head, Dept. of Computer Science and Engg.
NIT Warangal
India

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
TELANGANA - 506004, INDIA

CERTIFICATE

This is to certify that the thesis entitled, Energy Efficient and Co-operative Vehicle Scheduling Algorithms for Data Dissemination in Highway Vehicular Networks, submitted in partial fulfillment of requirement for the award of degree of DOCTOR OF PHILOSOPHY to National Institute of Technology Warangal, is a bonafide research work done by Mr. Satish Vemi Reddy [Roll No. 717045] under my supervision. The contents of the thesis have not been submitted elsewhere for the award of any degree.

Research Supervisor
Dr. Rashmi Ranjan Rout
Associate Professor
Dept. of Computer Science and Engg.
NIT Warangal
India

Place: NIT Warangal
Date: October, 2021

DECLARATION

This is to certify that the work presented in the thesis entitled "*Energy Efficient and Co-operative Vehicle Scheduling Algorithms for Data Dissemination in Highway Vehicular Networks*" is a bonafide work done by me under the supervision of Dr. Rashmi Ranjan Rout and was not submitted elsewhere for the award of any degree.

I declare that this written submission represents my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/date/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Satish Vemi Reddy

(Roll No. 717045)

Date: 29-10-2021

ACKNOWLEDGMENTS

It is with great pleasure that I acknowledge my sincere thanks and deep sense of gratitude to my supervisor Dr. Rashmi Ranjan Rout, Department of Computer Science and Engineering for his valuable guidance throughout the course. His technical perception, profound knowledge, sincere effort in guiding a student and devotion to work have greatly charmed and constantly motivated me to work towards the goal. He always gave me ample time for discussions, reviewing my work and suggesting requisite corrections.

I extend my gratitude to all my Doctoral Scrutiny Committee members for their valuable suggestions and faculty members of Computer Science and Engineering Department. I wish to express my sincere thanks to Prof. D.V.L.N. Somayajulu, Prof. P. Radha Krishna, and Prof. L. Anjaneyulu for their insightful comments and suggestions during oral presentations.

I am also grateful to Dr. S. Ravichandra and Dr. Ch. Sudhakar, who has always been supportive and encouraging all through my tenure.

I am immensely thankful to Prof. P. Radha Krishna, Head of Dept. of CSE, for providing adequate facilities in the department to carry out the research work. I wish to express my sincere thanks to Prof. N.V. Ramana Rao, Director, NIT Warangal for his official support and encouragement.

I would also like to thank my friends Sudarshan, Sandipan Maiti, Sanjib, Sai krishna and Pavan for their valuable suggestions and for extending selfless cooperation. My gratitude to my wife Pallavi and my parents for their unconditional love, support and prayers for my success in achieving the goal. Lastly I am deeply indebted to the almighty for giving the opportunity to pursue this line.

Satish Vemi Reddy

Dedicated to

My Family

ABSTRACT

The cooperative vehicular networks enable wide verity of applications including road safety, real-time traffic management, location-aware advertisement, environment monitoring, remote region connectivity, etc. Vehicles communicate with other vehicles as well as Road Side Unit (RSU) deployed along road side in order to provide efficient data dissemination services to the vehicles outside the RSU coverage. The RSU exploits passing-by vehicles as store-carry-forwarders (relays) to serve the vehicles that are disconnected from RSU. On the other hand, the RSUs deployed in highway locations are energy-limited and they aim at reducing energy consumption during downlink communication to relay vehicles. Thus, improvement in data delivery is an important issue while reducing energy consumption of RSU, data delivery delay and response time in a highway vehicular network. Although energy harvesting technologies improve life time of RSU, the continuous arrival of task data to buffers with limited capacity leads to buffering delays at the RSU. Therefore, a dynamic power allocation mechanism is necessary to balance energy consumption and buffering delays under task deadline constraints. In addition, real-time scheduling of fog vehicles for the energy efficient offloading of tasks in RSU coverage is also challenging in order to reduce average response time of tasks.

This thesis focuses on energy efficient and cooperative vehicle scheduling algorithms for improving data dissemination services in the highway vehicular networks. The issues of energy efficiency and data delivery delay have been addressed while satisfying task deadline constraints. In this thesis, the proposed vehicle scheduling algorithms have achieved improvement in data delivery performance by reducing energy consumption of RSU, data delivery delay, buffering delay at RSU and response time of offloaded tasks. Unlike traditional wireless networks, the vehicular networks exist with various challenges including high mobility, rapid changes in network topology, limited RSU radio range and battery power. Four major problems have been addressed in this thesis, namely: a) a clustering based multi-relay scheduling algorithm is designed to minimize energy consumption of RSU by satisfying data delivery constraints. A polynomial time solution has been presented by modelling it as a minimum cost flow graph to improve energy efficiency and data delivery. b) an RSU assisted cooperative relay scheduling scheme is designed for bidirectional highway scenario. An Auction theory based polynomial time solution has been presented to achieve faster data delivery while minimizing energy consumption of RSU. c) a Lyapunov based optimization mechanism has been presented to analyse the trade-off between power consumption and buffering delays at the RSU. A max-weight greedy relay scheduling algorithm is proposed to improve the data delivery performance under task deadline constraints. d) a vehicular fog computing framework is designed to offload tasks to fog vehicles while minimizing energy consumption of RSU and response time of tasks. A real-time scheduling of fog vehicles has been presented by using the combination of fuzzy logic system and on-policy leaning model. Weights of fog vehicles observed from fuzzy logic act as input to learning model for faster convergence and improving long-term reward. Performance of the proposed scheduling algorithms are evaluated through simulations. It has been observed that the proposed algorithms have improved the overall performance of the Highway VANET in-terms of energy consumption, data delivery delay, buffering delay and response time. Furthermore, these problems have been extensively studied in this thesis, followed by discussion on future research directions.

Keywords: Highway vehicular networks, energy efficiency, minimum cost flow graph, clustering, end-to-end delay, Auction theory, buffering delay, Lyapunov optimization, task offloading, response time, Fuzzy logic, on-policy SARSA leaning, vehicular fog computing.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

DSRC	Dedicated Short Range Communication
DTA_DP	Distributed Task Allocation in Distribute Process
FCFS	First Come First Serve
FF	Fastest First
FRL	Fuzzy Reinforcement Learning
FRS	Forward Realy Scheduler
ILP	Integer Linear Programming
ITS	Intelligent Transport Systems
I2I	Infrastructure to Infrastructure
I2V	Infrastructure to Vehicle
LDPA	Lyapunov optimization based Dynamic Power Allocation
MCF	Minimum Cost Flow graph
NFS	Nearest Fastest Set
NNF	Nearest Neighbour Forwarder
MCF-NNF	MCF augmented with NNF
MRVS	Max-weight Relay Vehicle Scheduling
OBU	On Board Unit
RMS	Rate Monotonic Scheduling
RRS	RSU assisted Relay Scheduling
RSU	Road Side Unit
VFC	Vehicular Fog Computing
V2V	Vehicle to Vehicle

Chapter 1

Introduction

Recent developments in wireless communication technologies combined with Vehicular Ad hoc Network (VANET) enable safe and intelligent transportation systems which can provide road safety, infotainment and advertisement services[1]. In VANET, vehicles are equipped with On-Board-Units (OBUs) that are capable of communicating with other vehicles as well as nearby wireless access points called Road Side Units (RSUs). The RSU is a roadside infrastructure and it provides access to moving vehicles on the road. VANETs deployed in highway environment (i.e., highway vehicular networks [2]) observe that the vehicles maintain constant speed in highway road segment. Moreover, the vehicles remain in RSU coverage for a relatively short duration due to high mobility of the vehicles. According to Federal Communication Commission (FCC), the spectrum assigned for Dedicated Short Range Communication (DSRC) in 5.9GHz frequency band has been divided into one control channel (CCH) and six service channels (SCH). The channels CCH and SCH are used to transmit control messages and service messages, respectively. Moreover, the communication capabilities of VANET are classified into various modes including Infrastructure to Vehicle (I2V) and Vehicle to Vehicle (V2V) communications. These communication modes enable wide variety of applications such as safety (e.g., collision warning, real-time traffic, etc.) and non-safety (e.g., weather information, location-aware advertisement, etc.) services for the on-road vehicles. The RSU deployed along the highway provides data access to vehicles that are in RSU radio coverage via I2V communication. Due to high mobility, the vehicles may leave RSU without completely downloading the requested data.

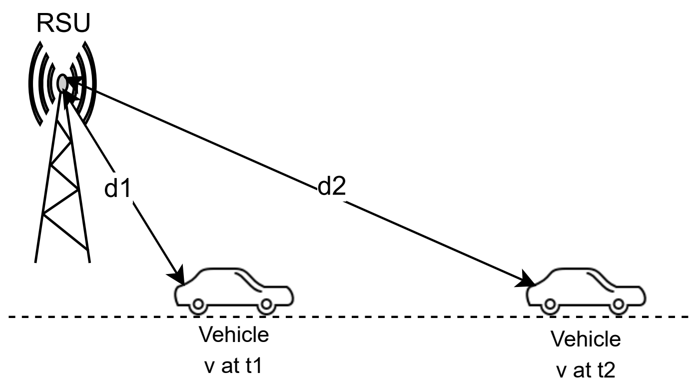


Figure 1.1: Downlink communication scenario

A vehicle leaves RSU region with unserved data requirement, is referred as *target* vehicle. When a target vehicle leaves RSU coverage, the RSU ensures data delivery to that vehicle with the aid of store-carry-forward vehicles (i.e. *relays*) and V2V forwarding.

In rural highway locations, providing direct wired electricity to RSU is difficult due to unavailability of power grid connections in that area. A viable alternative is to operate RSUs using sustainable energy sources such as wind power, solar power, etc. According to a deployment analysis presented by the U.S Department of Transportation [3], it has been estimated that over 40% of rural highway roadside infrastructure would be solar powered and nearly 63% of RSU costs would have to be spent on solar energy provisioning by 2050. Most often, the RSUs deployed in rural highways are equipped with rechargeable batteries with a support of energy harvesting technologies [4]. The energy provisioning cost of RSU significantly depends on its average energy consumption [5]. This is due to the strong dependency between energy consumption of RSU and RSU-to-vehicle distance[6]. For example, the downlink (RSU-to-vehicle) communication to a nearby vehicle consumes less RSU energy compared to a more distant vehicle in RSU coverage. As shown in Fig.1.1, vehicle v at time instance $t1$ is separated by a distance $d1$ from RSU, and the same vehicle is separated by a distance $d2$ from RSU at another time instance $t2$. Since $d1 < d2$, the RSU prefers downlink communication to vehicle v at $t1$ instead of $t2$. Although the greedy selection of nearby vehicles conserve energy, it may adversely impact system performance in terms of data delivery, end-to-end delay, response time, etc,. Therefore, optimal scheduling of vehicles in RSU coverage is an important issue to balance the trade-off between RSU energy consumption and other data dissemination service parameters of the system.

The major contributions in this thesis are as follows:

- **A clustering based energy efficient nearest neighbour forward approach for improving data dissemination in highway VANET:** This work presents an optimal relay scheduling algorithm to minimize the RSU energy consumption subject to satisfy residual data requirement of target vehicle. Furthermore, a clustering based Nearest Neighbor Forward (NNF) approach is proposed to identify the vehicles which are in the energy favorable locations (i.e., near to RSU) and multi-hop neighbors to the relay vehicles. Furthermore, combining the relay scheduling with NNF approach achieves reduction in the energy consumption of RSU and improvement in the data delivery to target vehicle.

- **An auction-based energy efficient cooperative relay scheduling for faster data retrieval in bidirectional highways:** This work presents an RSU assisted multiple relay scheduling algorithm to achieve faster data retrieval for the target vehicle using Auction theory principles. This work considers a bidirectional highway scenario where the relay vehicles are scheduled in both forward and opposite directions to minimize the RSU energy consumption and end-to-end delay to the target vehicle. The proposed algorithm utilizes the cooperation of a neighboring RSU for transferring the residual data of target vehicle.

- **A dynamic power allocation algorithm using Lyapunov optimization mechanism for data sharing between neighboring RSUs:** This work presents a delay-aware energy efficient dynamic relay scheduling algorithm to minimize the energy consumption of source RSU, buffering delay at source RSU, and maximize the average data delivery to the destination RSU in the network. The proposed Lyapunov based algorithm first decides the minimum power allocation for transmission of buffer content by observing the buffer back-log (buffered bits) sizes. This dynamic power allocation technique reduces energy consumption of RSU and ensures buffer stability. Further, the proposed algorithm schedules a set of relay vehicles to maximize the average data delivery to destination RSU.

- **A fuzzy reinforcement learning for energy efficient task offloading from RSU to mobile fog vehicles:** This work considers a Vehicular Fog Computing (VFC) paradigm where a stationary RSU supplements local computation and responsible for task allocation to fog vehicles. This work proposes an on-policy reinforcement learning based algorithm for energy efficient scheduling of fog vehicles to compute tasks within tolerable response latency. Further, a fuzzy logic based greedy heuristic is used to accelerate learning process and improve long term reward of the proposed algorithm for achieving minimization of energy consumption and response time.

The rest of this chapter is organized as follows. Motivation of this work and objectives are discussed in section 1.1. In section 1.2.1, the importance of clustering based approach for energy efficient scheduling of relay vehicles is discussed. In section 1.2.2, application of Auction theory principles for the relay scheduling in bidirectional highways has been presented. In section 1.2.3, requirements of dynamic power allocation approach for energy conservation is highlighted. Importance of greedy relay scheduling for maximizing data delivery is discussed. In section 1.2.4, the requirements of combining fuzzy heuristic and reinforcement learning for dynamic scheduling of fog vehicles is presented. Section 1.3 provides the details of experimental settings. Section 1.4 illustrates the organization of this thesis.

1.1 Motivation and objectives

Efficient data dissemination is crucial for achieving more reliable data services (e.g., large file download, sensor data transfer, etc.) by exploiting the synergetic effects between I2V and V2V communications [7, 8]. In highway locations, due to high installation cost of the vehicular infrastructure, the RSUs cannot provide seamless radio coverage and they leave uncovered area or outage area in between the neighboring RSUs [9]. Specifically, the RSUs may not complete the vehicles' requests inside the RSU coverage due to the limited I2V bandwidth, high mobility and high data demand of the vehicles [10]. Thus, the vehicles have unserved data requirement while entering into an uncovered area. Such vehicles are known as target vehicles. Nevertheless, the target vehicles can retrieve the unserved (residual) data by leveraging the services of passing by vehicles known as relay vehicles [11]. These relay vehicles follow store-carry-forward [12] mechanism to serve a target vehicle using V2V forwarding when they establish V2V links with the target vehicle. However, proper scheduling of relay vehicles is an important issue for improving data delivery services along with minimizing energy consumption.

As aforementioned, the deployment of RSUs creates uncovered areas in-between neighbouring RSUs due to their limited radio range and deployment cost. Specifically, some RSUs are placed in isolated rural highway locations without any connection to direct grid power or backbone network (which connects to other RSUs in the network). Therefore, a *source RSU* in an isolated location is equipped with rechargeable batteries and depends on energy harvesting technologies[13]. Moreover, the tasks generated by the applications running in the *source RSU* region need to be offloaded to a nearby *destination RSU* (which is equipped with high-end computation server and connects to direct grid power) via store-carry-forward vehicles or relays. However, it poses some challenges to design a good relay vehicle scheduling algorithm in such a dynamic scenario. First, future arrival of vehicles are completely unknown to *source RSU*, then the RSU needs to schedule the best possible relay vehicles available in its coverage region at current time instance. Second, rechargeable batteries equipped with RSU have limited storage capacity, consequently an efficient power allocation strategy is required for the effective utilization of stored energy. Third, the source RSU does not have control over the arrival of task data, and this may lead to continuous increase of buffer back-log size referred as buffer instability. Therefore, a dynamic power allocation technique is necessary to balance the trade-off between energy consumption and buffering delay, while improving data delivery to destination RSU.

To reduce the data transit delays between source and destination RSUs, a Vehicular Fog Computing (VFC) paradigm is realized to offload computation intensive tasks to mobile fog vehicles in RSU region. The efficient task offloading in VFC has challenges that need to addressed. High mobility, short connection time and heterogeneity of vehicles make difficult for smart devices to directly offload tasks to fog vehicles. On the other hand, selection of potential fog vehicles for task offloading is important since the vehicles with long staying period in RSU communication coverage may be busy in execution of other tasks while the vehicles with available resources may ready to leave communication coverage. Therefore, this work realizes that the minimization of energy consumption of RSU is equally important along with response time since the allocation of tasks to fog vehicles involves both communication cost of infrastructure and computation delay of fog vehicles.

The above mentioned challenges motivate the present work towards energy efficient and cooperative vehicle scheduling algorithms for data dissemination in highway vehicular networks. The major objectives of this dissertation is as follows.

1. Design of energy efficient relay scheduling algorithm using clustering mechanism to improve data delivery to target vehicle in highway VANET.

2. Design of energy efficient RSU assisted relay scheduling algorithm using auction theory principles for faster data retrieval to target vehicle in bidirectional highways.

3. Analysis of energy consumption and buffering delay at isolated RSU using Lyapunov optimization technique for dynamic power allocation under task deadline constraints.

4. Analysis of energy consumption and response time while offloading tasks to mobile fog vehicles by satisfying task deadlines in vehicular fog computing environment.

1.2 Overview of the contributions in this Thesis

This section presents overview of chapter-wise contributions discussed in this thesis work. Each sub section provides summary of corresponding chapter.

1.2.1 Clustering based energy efficient nearest neighbour forward approach

In this work, an energy efficient scheduling of multiple relay vehicles has been proposed to serve the residual requirement of a target vehicle moving in the uncovered area. The multi-relay scheduling problem is formulated as an optimization problem, and consequently presented a polynomial time solution by modeling it as Minimum Cost Flow (MCF) graph [14]. Furthermore, a clustering[15] based algorithm is proposed to identify the vehicles which are in the energy favorable locations (i.e., near to RSU) and multi-hop neighbors to the relay vehicles. Such vehicles are named as *Nearest Neighbor Forwarders (NNFs)*. Proper scheduling of downlink (RSU-to-NNF) communication further reduces the energy consumption of RSU and improves the data delivery to the target vehicle. Downlink channel time of RSU is divided into time slots of fixed duration. The proposed clustering based algorithm discovers vehicle clusters that are unaltered for a time slot duration. In a given time slot, the multi-hop relay vehicles are reachable from a NNF in the cluster. Then, the data forwarding from the NNF vehicle to its multi-hop relay vehicle follows *off-channel* V2V forwarding[16, 17].

The contributions of this work are as follows.

- Analyse the relationship between the energy consumption of RSU and the data delivery to the target vehicle and further determine a set of store-carry-forward relay vehicles which can establish a communication link with the target vehicle.
- Formulate the multi-relay scheduling as an optimization problem, and consequently present a polynomial time solution by modeling it as Minimum Cost Flow (MCF) graph.
- Design a clustering based Nearest Neighbor Forward (NNF) approach to identify the vehicles which are in the energy favorable locations (i.e., near to RSU) and multi-hop neighbors to the relay vehicles.

1.2.1.1 Minimum Cost Flow Graph (MCF) for relay scheduling

A Minimum Cost Flow (MCF) graph is presented to solve the optimization problem in polynomial time[14]. The graph $G = (N, E)$ is defined as set of N nodes and set of E arcs to connect the nodes. Suppose $s, r \in N$, then each arc $(s, r) \in E$ has an associated capacity $c_{s,r}$ and cost $\epsilon_{s,r}$, which denotes the maximum flow and the downlink communication cost per unit flow respectively. The associated capacity and energy of each arc has been labeled with an ordered pair $(c_{s,r}, \epsilon_{s,r})$. The minimum cost flow model ensures that minimum energy consumption of RSU while respecting the maximum flow from RSU to the target vehicle.

1.2.1.2 Nearest Neighbor Forward Approach

The proposed *Nearest Neighbor Forward* (NNF) approach selects the vehicles in energy favorable locations as the data forwarders to relay vehicles. *A nearest neighbor forwarder is a vehicle which is nearest to the RSU and multi-hop neighbor to a relay vehicle.* In a time slot, the NNF approach selects a nearest neighbor forwarder vehicle to each relay vehicle. The nearest neighbor forwarder of a relay r is defined as NNF(r). Then, it computes the energy cost of RSU to NNF(r) for all r. The energy consumption costs obtained from NNF becomes a new input to the scheduler (i.e., MCF). The MCF augmented with the NNF approach is referred as MCF-NNF. A schedule obtained from the MCF-NNF improves the energy consumption of RSU and data delivery to the target vehicle in the uncovered area.

Results from extensive simulations show that the proposed NNF approach augmented with MCF preform better when compared to scheduling algorithms Nearest Fastest Set (NFS)[6], MCF[18] and two more basic algorithms First Come First Serve (FCFS) and Fastest First (FF). The results also show the impact of vehicle arrival rate, vehicle transmission range and target vehicle speed on the power consumption of the RSU and data delivery to the target vehicle. The proposed MCF-NNF has significant improvement in energy consumption by 23% and 28% when compared to MCF and NFS respectively. The proposed approach shows clear dominance at higher vehicle arrival rates above 0.5 and its residual data completeness is 6%, 25% and 16% more when compared to FCFS, NFS and MCF respectively.

1.2.2 Auction-based RSU-assisted relay scheduling in bidirectional highways

In this work, a bidirectional highway scenario has been considered where the relay vehicles are scheduled in both forward and opposite directions to minimize the RSU energy consumption and end-to-end delay to the target vehicle. This service paradigm utilizes the cooperation of a neighboring RSU for transferring the unserved data of the target vehicle. Based on the received data, the neighboring RSU can schedule the relay vehicles driving in opposite direction by ensuring the energy consumption and end-to-end delay requirements. Minimizing end-to-end delay is also essential to improve the quality of service (QoS) requirements of the target vehicle when running infotainment applications such as online video transfer, online gaming, etc. Besides, the RSUs apply the principles of *Auction Theory*[19] to schedule the downlink communication for the suitable relay vehicles. Here, RSUs and relay vehicles act as sellers and bidders, respectively. The seller auctions the RSU channel time by subdividing into time slots of equal duration. Although the bidders have incomplete information about other bidders, they use only local information (e.g., speed, position, cooperative cache size, direction, etc.) and participate in Auctioning process. The seller or RSU select the bidders solely based on the bids received from the bidders or relay vehicles. Then, the RSU optimally assigns the relay vehicles to time slots. However, due to limited V2V bandwidth and half-duplex nature of OBUs [7], it is difficult to achieve simultaneous data transmission when multiple relay vehicles establish V2V links with the target vehicle. Therefore, the target vehicle is designated to perform ad hoc V2V scheduling in the uncovered area when multiple relay vehicles are present in its radio range. This will improve the data delivery ratio by making the best utilization of V2V communication bandwidth. The V2V forwarding cannot affect the power consumption of RSUs because the vehicles are assumed to have sufficient energy reserves.

Major contributions of this work are as follows.

- Determine the set of relay vehicles in both directions (forward and backward) in a bidirectional highway segment for faster data delivery to target vehicle. The relay scheduling problem is formulated as an Integer Linear Programming problem (ILP) and its NP hardness is proved.

- Propose a forward relay scheduler (FRS) based on *Auction Theory* to schedule the relay vehicles in target moving direction. In addition, proposed an Auction based RSU assisted relay scheduling (RRS) algorithm that uses cooperative sharing between neighboring RSUs and schedule the relay vehicles in both forward and backward directions for serving the target vehicle.

1.2.2.1 Auction theory for relay vehicle selection

This work presents an optimal assignment of relay vehicles to time slots by applying the concepts from *Auction Theory*[20]. This problem is modeled as an asymmetric assignment problem where the time slots are more than the relays. There exists non empty set of time slots T, relay vehicles V_r, and these are finite. Before selecting a suitable relay vehicle $V_i \in V_r$ at a time slot $T_j \in T$, the RSU estimates utility $\mathcal{U}_{ij}$ based on the Time-to-Contact (ΔC_{ij}) and the bit-rate (b_{ij}) in the downlink channel. These two parameters have direct impact on the end-to-end delay and the RSU power consumption, respectively. The mapping of time slot – relay vehicle pair is determined by the difference between the minimum utility and the second smallest utility.

The proposed Auction-based RRS algorithm utilizes the neighboring RSU cooperation and schedule the relay vehicles driving in both directions that can maximize the data delivery to the target vehicle. Extensive simulations show that the proposed RRS algorithm performs better compared to FCFS, GA and FRS in terms of average RSU energy consumption and end-to-end delay to the target vehicle. For the case of vehicle arrival rate 0.5 and target speed 20 m/s, improvement of RRS over FCFS, GA, and FRS is 60.17%, 22.27% and 15.69% in terms of average RSU energy consumption, respectively.

1.2.3 Lyapunov optimization mechanism for energy efficient data sharing between RSUs

This work presents a delay-aware energy efficient dynamic relay scheduling strategy to minimize the energy consumption of source RSU, buffering delay at source RSU and maximize the average data delivery to the destination RSU in the network. In highway locations, the deployment of RSUs create uncovered areas in-between neighbouring RSUs due to their limited radio range and deployment cost. Specifically, some RSUs are placed in isolated rural highway locations without any connection to direct grid power or backbone network (which connects to other RSUs in the network). Therefore, a *source RSU* in an isolated location is equipped with large batteries and depends on energy harvesting technologies[13]. Moreover, the tasks generated by the applications running in the *source*

RSU region need to be offloaded to a nearby *destination RSU* (which is equipped with high-end computation server and connects to direct grid power) via store-carry-forward vehicles or relays. Before selecting suitable relay vehicles, the proposed strategy first decides the minimum power allocation for transmission of buffer content by observing the buffer back-log sizes (buffer occupancy) in each time slot. This dynamic power allocation technique reduces energy consumption of RSU and ensures buffer stability. Second, depending on the amount of data to be transmitted to each vehicle via I2V communication, the proposed strategy schedules a set of relay vehicles to maximize the average data delivery to destination RSU. The selection criteria of relay vehicles are subjected to task deadlines constraints as well. The major contributions of this work is as follows.

- Present a dynamic relay scheduling strategy in a bidirectional highway scenario for data sharing in between the neighbouring RSUs. Specifically, such a system enables the RSUs to opportunistically exploit the store-carry-forward (relay) vehicles, which not only enhances the data delivery to destination RSU, but also realizes balancing the trade-off between buffer stability and energy consumption at the source RSU.

- Formulate two optimization problems namely, dynamic power allocation problem ($\mathcal{P}_1$) and a relay scheduling problem ($\mathcal{P}_2$). First, $\mathcal{P}_1$ minimizes the energy consumption of source RSU subject to satisfy the buffer stability and energy level in the rechargeable batteries. Second, $\mathcal{P}_2$ maximizes the data delivery to destination RSU subject to satisfy task deadlines.

- Propose a Lyapunov optimization based Dynamic Power Allocation (LDPA) algorithm (Section 5.2.1), which allocates minimum power required for the transmission of buffer content by observing the buffer back-log size and channel gain. Furthermore, a Max-weight Relay Vehicle Scheduling (MRVS) (Section 5.2.1) algorithm has been proposed to select the relay vehicles based on their speed, location and achievable data rates. In particular, it is observed that the combination of LDPA and MRVS improves the efficacy of the system in-terms of buffer stability, network life time and data delivery.

1.2.3.1 Lyapunov based dynamic power allocation

The proposed dynamic power allocation algorithm for determining the downlink transmission rate over each time slot τ, is designed by observing the buffer back-log size $B_i(\tau)$ and then deciding the power allocation $P_i(\tau)$. Rather than alone minimizing the drift $\Delta(B(\tau))$, the dynamic algorithm minimizes the bound on $\mathbb{E}\{P(\tau)|B(\tau)\} + \mathbb{V}\Delta(B(\tau))$ (i.e., *drift-plus-penalty*), where the constant $\mathbb{V} \geq 0$ is a parameter to control the trade-off exist between buffer stability and power allocation. Intuitively, large values of $\mathbb{V}$ emphasizes more on buffer stability but it consumes more power. Small values of $\mathbb{V}$ lead to less power consumption but there is possibility that the buffer becomes unstable.

1.2.3.2 Max-weight relay vehicle scheduling

A relay scheduling problem is formulated to select relay vehicles which satisfy task deadlines. As a solution, this work realizes the selection of relay vehicles with maximum achievable data rates (derived from dynamic power allocation technique), and transmits the buffer content from RSU to those scheduled relay vehicles via I2V communication. A simulation study has been conducted and demonstrated the performance of proposed algorithms interms of buffering and scheduling performance. It is observed that the proposed strategy provides significant improvement in terms of buffer stability, network life time and average data delivery in the system.

1.2.4 Fuzzy reinforcement learning for energy efficient task offloading from RSU to mobile fog vehicles

This work presents a latency-aware energy efficient scheduling of tasks to fog vehicles in Vehicular Fog Computing (VFC)[21]. The VFC extends fog computing to conventional vehicular networks, where the vehicles act as mobile fog nodes which support full utilization of computation resources. This computation model leverages latency-aware execution of applications and work load allocation among mobile fog nodes[22]. Most often, the RSUs deployed in rural highways are endowed with rechargeable batteries and depends on renewable energy sources[23]. The RSU provides third-party scheduling services for efficient allocation of tasks to potential fog vehicles. However, there is a requirement of energy efficient scheduling of fog vehicles in VFC for conserving stored energy and improving life time of the network.

In VFC, the stationary RSU not only supplements local computation but also responsible for task allocation to fog vehicles. As the number of vehicles in RSU coverage increases, it is difficult to find potential fog vehicles with exhaustive search techniques in real-time. Therefore, this work proposes a reinforcement learning based algorithm to identify potential fog vehicles in every time slot. Due to slower convergence of conventional learning algorithms caused by large action space and high-dimensionality, a fuzzy logic based greedy heuristic is used to accelerate learning process of the proposed algorithm. The major contributions of this work is described as follows.

- Present a Vehicular Fog Computing (VFC) framework for efficient offloading of tasks generated by real-time applications running in smart cities close to rural highways.

- Formulate an optimization problem as Integer Linear Programming Problem (ILP) which aims to minimise the communication and computation cost of RSU for efficient task allocation among fog vehicles while satisfying constraints on task deadline and resource availability.

- Propose a Fuzzy Reinforcement Learning (FRL) approach for energy efficient allocation of tasks to fog vehicles, where a Fuzzy logic based greedy heuristic is combined with an on-policy reinforcement learning (i.e., SARSA).

The FRL not only accelerate the learning process but also improves the selection of potential fog vehicles for reducing total energy consumption and average response time. This work presents a real-time scheduling of fog nodes by combining a greedy heuristic and reinforcement learning technique to improve long term reward and speedup learned outcome. Extensive set of experiments has been conducted and results show the proposed algorithm has better performance over other algorithms by 46.73% and 15.38% in terms of energy consumption and response time, respectively.

1.3 Experimental setup

In this thesis, as part of simulation settings, the neighbouring RSUs are separated by a distance 4000m and deployed in a bidirectional highway segment, where the communication range of each RSU is 1000m. The vehicles enter into source RSU region follow Poisson process with mean rate of arrival λ. The speeds of vehicles are assumed to be distributed uniformly in a range [12m/s, 28m/s] where the faster vehicles can overtake slower vehicles. In this simulation, the communication model uses parameters specified in [24] where the downlink bit rates vary from 3 to 27 Mb/s, maximum transmit power is 1W, bandwidth of channel is 10MHz and the noise at the relay vehicle is -174dBm/Hz. The number of task arrivals are considered as Poisson stream with a mean arrival of 1000 tasks in each time slot, and the length (in bits) of each task is chosen as 1024 bits. The simulation runs on a desktop system with 3.40GHz Intel core i7 CPU, 3.7 GiB of RAM, 64-bit ubuntu 16.04 LTS operating system and Python 3.6 for simulator development. The works in this thesis evaluate different scheduling algorithms using Monte Carlo simulations[25] over 100 time slots for 1000 iterations.

1.4 Organization of the Thesis

The main focus of this thesis is to design and analyze energy efficient and cooperative vehicle scheduling algorithms for data sharing in a dynamic highway vehicular environment. The proposed algorithms achieve energy efficiency and data delivery improvement by satisfying tolerable response delay requirement. The thesis has been organized into seven chapters.

Chapter 1: In this chapter, a brief introduction to vehicular ad hoc networks and downlink communication scenario in highway environment, and objectives of the thesis have been discussed. Moreover, it presents major contributions as an overview of the thesis.

Chapter 2: In this chapter, efficient scheduling algorithms based on minimum cost flow, auction theory and reinforcement learning have been surveyed. A survey on efficient data dissemination approaches is discussed. The challenges of energy conservation in

VANET have been presented.

Chapter 3: In this chapter, an energy efficient relay scheduling problem has been presented to serve the residual data requirement of a target vehicle. A polynomial time solution is proposed by modeling it as Minimum Cost Flow graph. A clustering based Nearest Neighbour Forward approach is proposed to further reduce energy consumption of RSU and improve data delivery to target vehicle.

Chapter 4: In this chapter, an energy efficient RSU-assisted relay scheduling algorithm is presented to achieve faster data retrieval for the target vehicle by minimizing end-to-end delay. Based on Auction theory principles, the proposed algorithm uses cooperative sharing between neighboring RSUs and schedule the relay vehicles in both forward and backward directions.

Chapter 5: In this chapter, a real-time scheduling of relay vehicles between neighboring RSUs with limited buffer capacity is discussed. A dynamic power allocation algorithm is presented to allocate minimum power required for the transmission of buffer content by observing the buffer back-log size. Furthermore, a Max-weight Relay Vehicle Scheduling algorithm has been proposed to schedule the relay vehicles for achieving maximum data delivery in the system.

Chapter 6: In this chapter, a vehicular fog computing scenario is considered for latency-aware energy efficient scheduling of tasks to mobile fog vehicles. A fuzzy reinforcement learning approach has been presented for efficient scheduling of fog vehicles. The proposed approach combines a Fuzzy logic based greedy heuristic with an on-policy reinforcement learning in order to accelerate the learning process and improve long term reward.

Chapter 7: This chapter concludes the contributions of this thesis work and discusses future scopes for extension of the work.

Chapter 2

Literature Survey

Vehicular Adhoc Network (VANET) is a sub class of Mobile Adhoc Networks (MANET) with the combination of wireless communication technologies and auto mobile industry[26]. With the emergence of 5G services and increasing number of vehicles equipped with communication devices, data dissemination via inter vehicular communication is more promising in the field of VANET research and development[27]. Different VANET service architectures enable wide variety of applications such as driver safety, collision warning, route scheduling, traffic monitoring, etc. Many VANET applications are categorized into two types. 1) safety applications 2) non-safety applications. Safety applications include sending of warning messages in order to avoid serious problems being confronted by vehicle users. Non safety applications include spreading of business advertisements and forwarding multi-media content. The content delivery to requested vehicles is challenging due to intermittent connections in the VANET[28]. However, combining the efforts of inter vehicular communication with infrastructure to vehicle communication can ensure delivery of requested content for the vehicle users.

VANET characteristics: the VANETs deployed in highway environment have its unique set of characteristics as discussed below.

- *High and predictable mobility:* Vehicles are the mobile nodes in VANET. The mobility of these vehicles are constrained by road network. In case of highway road network, the vehicles move with high mobility, but they tend to maintain constant speed over the highway road segment[29]. Therefore, the constrained and constant mobility leads to accurate prediction of vehicle's location in highways.

- *Power constraints:* Vehicles have adequate energy reserves as they are powered by vehicular engines. Besides, the stationary road side infrastructure deployed along the rural highways are deprived of direct power sources, and they depend on renewable energy sources such as solar power, wind power, etc,[30]. Moreover, large rechargeable batteries are used to store the energy conserved from energy harvesting technologies. However, effective utilization of the conserved energy is important to improve the operational efficiency and network life of the highway vehicular network.

- *Computation constraints:* Vehicles are equipped with computation resources such as processors, memory, sensors, wireless technologies and Global Positioning System (GPS). These resources strengthen the computational capability of vehicles that can support execution of tasks offloaded from other vehicles or road side infrastructure[31]. Since the road side infrastructure in rural scenarios has limited computation and energy reserves, it depends on the vehicle's resources for possible execution of computation intensive tasks.

- *Variable network density:* Vehicle density in the network depends on the number of vehicles, inter vehicular distance, speeds and direction. The vehicle density is high due to traffic jams in urban areas, and is low in case of highway or suburban locations.

- *Rapid changes in network topology:* High mobility of vehicles in highway environment leads to rapid changes in topology of vehicular network[32]. Specifically, life time of communication link (i.e. link time) between vehicles is affected by vehicle speeds, communication range and moving direction. Thus, increasing communication range of vehicles increases link time. Vehicles moving in opposite directions experience shorter link time when compared to vehicles moving in same direction. Moreover, high speed vehicles stay relatively shorter time in stationary infrastructure region when compared to slow vehicles. The rapid changes in network leads to disconnection of multi-hop paths before the path links are utilized.

- *Intermittent connections:* Due to high deployment cost of the stationary roadside infrastructure, it is difficult to provide seamless coverage for the entire highway road segment[33]. Therefore, it leads to uncovered areas or outage regions in-between equidistantly deployed stationary infrastructures. Vehicles in uncovered areas are completely disconnected from outside world. Thus, moving vehicles in highway road segment are intermittently connected to vehicular network.

2.1 Vehicular Adhoc Networks: preliminaries

The communication among vehicles, or between vehicles and road side infrastructure (Road Side Unit) has been achieved through Wireless Access for Vehicular Environment (WAVE) protocol stack[34]. Major components of VANET are On Board Unit (OBU), Application Unit (AU) and Road Side Unit (RSU). Typically, the RSUs host applications and act as content provider to the vehicles. The applications may be hosted by either RSU or OBU. The devices which host the applications are providers and the devices which use the application services are users.

On Board Unit (OBU):

Vehicles are equipped with On Board Units that are allowed to communicate with other vehicles or RSU. Recent advances in full duplex enhanced dual-radio OBUs enable the vehicles to send and receive messages concurrently over the same channel [35]. In dual-radio transceiver, one of the radio continuously tuned onto a service channel which is used to broadcast public safety messages, vehicle to vehicle collision avoidance messages, etc. Second radio switches between control channel and another service channel in regular intervals. The dual-radio transceivers ensure off-channel vehicle to vehicle forwarding in vehicular communications [36]. The main functions of OBU are adhoc routing, wireless radio access, message transfer and IP mobility [37].

Road Side Unit (RSU):

Road Side Unit (RSU) is a fixed vehicular infrastructure deployed along the road side that is equipped with wireless access technologies to provide content access to OBUs. The

RSUs connect to Internet and other RSUs through backbone network. The users registered with vehicle OBUs are allowed to access Internet via RSUs using wireless medium.

DSRC/WAVE protocol:

Dedicated Short Range Communication (DSRC)[34] based on IEEE 802.11p is a commonly used wireless communication protocol for vehicle to vehicle and infrastructure to vehicle communications. DSRC protocol is standardised by IEEE 1069 working group and is a part of Wireless Access for vehicular Environment (WAVE) protocol stack[34]. According to U.S department of Federal Communication Commission (FCC), the spectrum assigned for DSRC at 5.9GHz frequency band is a 75MHz licensed spectrum. As shown in Fig.2.1, the allocated 75MHz spectrum is divided into one control channel (CCH) and six service channels (SCH), where the capacity of each channel is 10MHz. These channels are numbered from 172 to 184. Here, channel 178 is a CCH, which is used to send control messages. The service channels of 172 and 184 are used to send messages related to safety critical applications such as collision warnings, driver safety messages, etc. Where as other SCHs are responsible for exchanging of non-safety messages such as video and audio content delivery. DSRC supports communication to moving vehicles with a maximum speed of 200Kmph, communication range 300m to 1000m, and data rates 3Mbps to 27 Mbps.

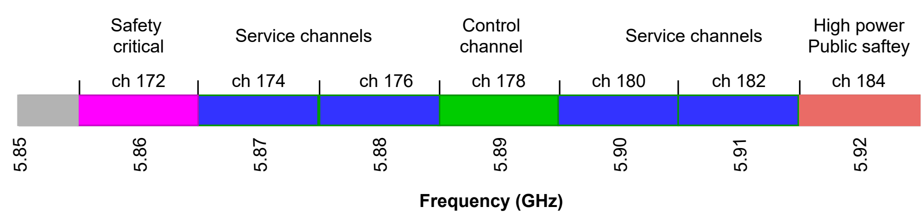


Figure 2.1: DSRC frequency spectrum

2.2 Cooperative communication in vehicular networks

Vehicular networks have been formed among vehicles, RSUs and pedestrians. The vehicular networks are deployed in urban, rural areas and highway environment. There exists three communication modes in vehicular networks; Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I) and Vehicle to Pedestrian (V2P)[38]. Cooperative communication in vehicular networks ensures efficient utilization of spectrum by exploiting the overhearing of broadcast signals transmitted from source node to destination node[39]. With cooperative communication the network can achieve higher spacial diversity[39], higher throughput[40] and lower delay[41].

Specifically, cooperative vehicular networks (CVN) exploit neighbouring vehicles as relay nodes or helper nodes to share the information with other vehicles that are in outside communication range. Vehicles communicate cooperatively using either direct one-hop links or with the help of RSU assistance. In cooperative communication, the relay nodes operate in various transmission modes that include compress-forward, decode-forward, and store-carry-forward. Fig 2.2 shows three different possibilities of cooperative vehicular networking. In case of direct transmission failure, the neighbouring vehicle is assisted to relay the transmission to destination as shown in Fig.2.2(a). Similarly, Fig.2.2(b) shows that a vehicle in RSU region acts as relay node to carry forward packets to destination which is out of RSU range. Besides, both RSU and vehicles are assisted to forward the transmitted packets to destination in between neighbouring RSUs as illustrated in Fig. 2.2(c).

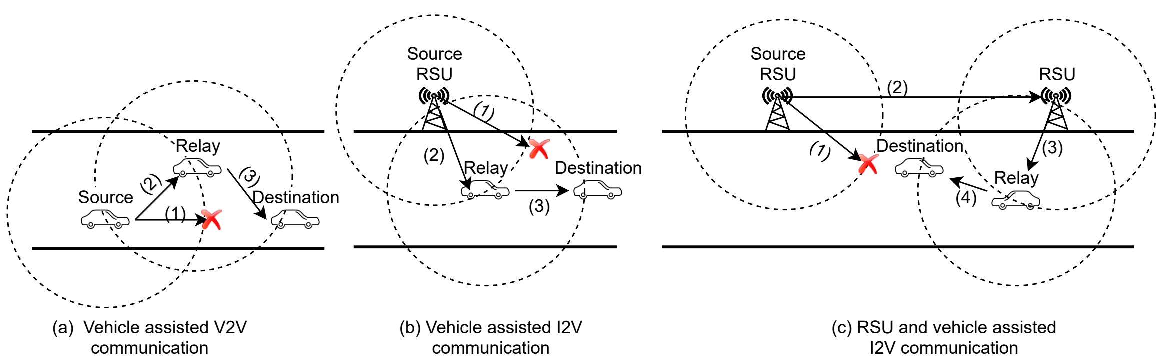


Figure 2.2: Cooperative vehicular networking

Cooperative vehicular networks enhance the spacial diversity at physical layer level by transmitting messages through two or more communication channels. Conventional Multi-In-Multi-Out (MIMO) systems ensure spacial diversity with multiple antennas[42]. But, achieving spacial diversity by employing multiple antennas incurs high installation cost. However, similar benefits as MIMO can be achieved by leveraging cooperative communication among vehicle nodes, referred as *cooperative diversity*[43]. The performance of coded transmission in MIMO downlink channels for cooperative relaying among vehicles is analysed and presented in [44]. In [44], the transmission from source to destination is divided into two stages: broadcast and relaying. Each stage is again divided in to two levels. In the first level of broadcast stage, pre-coded blocks are transmitted from two separate channels. In the second level, another set of pre-coded blocks are transmitted over same channels. In relaying stage, the received signal is transmitted by strengthening it to destination. Significant cooperative diversity gain has been observed from coded transmission in MIMO downlink channels. Another work in [45] has presented a energy efficient cooperative relying schemes in V2V and I2V scenarios. These cooperative relaying schemes combine the synergistic effects between multi-hop relaying and Cooperative MIMO techniques. Based on the transmission distance, the optimal selection of antenna configuration has been discussed.

Various cooperative communication strategies for vehicular networks have been discussed in the literature [46, 47, 48]. In [46], the performance of throughput optimization is analysed by exploiting the combined efforts of V2V and V2I communications with the effects of node mobility. The proposed strategy ensures data dissemination to vehicle of interest (VoI) over V2I communication when VoI is in RSU coverage. When VoI leaves RSU coverage, the vehicles are assisted to relay data towards VoI in order to maximize throughput of the system. In [47], a bidirectional scenario has been considered to analyse symbol error rate and diversity during V2V communication through vehicle assisted and RSU assisted communications. The cooperative scheme presents an optimal power allocation strategy to achieve maximize diversity. In [48], data rate and vehicle communication range are considered as parameters to investigate the performance of the system in terms of ability to disseminate information. Effect of hidden nodes and channel busy conditions are used as feedback quantifiers for the measurement of information dissemination.

In this thesis, cooperative communication among vehicles via I2V and V2V has been considered in Chapters 3 and 5. Cooperation in between neighbouring RSUs via I2I has been used in Chapter 4. The cooperativeness among vehicles and RSUs has been investigated in these chapters 3, 4 and 5 to provide improvement in data dissemination services of vehicular network.

2.3 Applications of cooperative vehicular networks

Various applications of cooperative vehicular networks have been developed as part of safety and non-safety services of vehicles [49, 50]. Use cases of different V2I, V2V and V2X communications are discussed in this section.

V2I communication use cases:

A cloud based vehicle navigation service called SAINT (Self Adaptive Interactive Navigation Tool) has been developed to attain optimal navigation paths in road networks [49]. The vehicles report their travel experience to cloud center for the assessment of traffic congestion, real-time trajectories as a guidance to other vehicles. The cloud center maintains road traffic statistics, vehicle mobility information, congested locations and trajectories of vehicles. With this information SAINT uses a mathematical model to estimate real-time congestion in road networks. Moreover, SAINT estimates how much each vehicle contributes to road traffic in future travel of same vehicle. Furthermore, an emergency navigation service called SAINT+ (Self Adaptive Interactive Navigation Tool plus) has been developed to optimize the delivery delay of emergency services in an efficient navigation routes nearby accident locations [51]. SAINT+ uses the features of SAINT and it provides fast emergency vehicle services using virtual path reservation strategy. An accident area evacuation and protection scheme has been presented based on adjusted congestion contribution matrix. These real-time trajectories can help platooning of vehicles with fuel efficiency in road network[52].

An energy efficient speed recommendation system called SignalGuru[50] has been developed based on vehicular cloud services. The mobile phones are mounted on vehicles to capture pictures of traffic congestion in signalized intersections. The SignalGuru analyses gathered pictures and informs new moving speeds to vehicles for energy efficiency when they arrives to signal location (i.e., intersection). In this system, communication between vehicle and cloud happens through RSU. Autonomous vehicles need to cross the intersection without waiting for the signal with the help of Mobile Edge Computing (MEC) Server[53]. An MEC server receives mobility information of vehicles and schedules the vehicles for crossing the intersection. This signal free intersection can improve the throughput of the system and energy efficiency of vehicles.

V2V communication use cases:

V2V communication supports vehicle safety services in autonomous driving environment [54]. The safety services include context-aware navigation, adaptive cruise control and truck platooning. The V2V communication among vehicles is performed via DSRC. Context-Awareness Safety Driving (CASD) [54] is a safety driving service for human driving, autonomous driving and hybrid driving (i.e., both human and autonomous driving). In CASD, the vehicles exchange safety information (with other vehicles) that helps vehicle maneuvers in dangerous road locations. Vehicles employing CASD system use V2V communication via DSRC for real time collision avoidance in highway or urban road segment. An adaptive cruise control [55] extends the cruise control coordinates of vehicles to adjacent vehicles so that all vehicles in the highway road segment can maintain safe inter-vehicle distance via V2V communication. In case of any abrupt changes in vehicle speeds, the vehicle informs other vehicles so that vehicles in-front and behind adjust their speeds according to emergency situation. The notifications of such messages follow in progressive fashion and allow the vehicles to adjust their direction and speed accordingly. Platooning [56] is a series of trucks moving in highway with equal inter-vehicle distance sufficient to avoid collision. First vehicle acts as a leader and informs the changes in speed, direction other vehicle in the platoon. This platooning service reduces labour cost of drivers and reduces fuel consumption cost of vehicles. A cooperative automated driving (CAD) [57] system allows the vehicles to coordinate each other in order to adjust vehicle maneuvers with the help of collective participation mechanism and sharing of sensing information.

V2X communication use cases:

The V2X communication includes combination of V2I or I2V, V2V and V2P (vehicle to pedestrian) communications. A Safety-Aware Navigation Application (SANA)[58] is an example for pedestrian protection service. Mobile phones or smartphones with pedestrians shows alert messages received from vehicles on road via DSRC. But, mobile phones do not support DSRC. So, mobile phone to vehicle interaction happens through RSU via V2I2P (Vehicle to Infrastructure to Pedestrian). The RSUs are equipped with computation to support edge computing[59], where the vehicles and mobile phones can interact with RSU by scheduling their communication in energy efficient manner.

2.4 Energy efficiency in highway vehicular networks

Studies related to energy efficiency in vehicular infrastructure are gaining more attention in recent years. Energy consumption aspects of vehicular networks in the presence of cellular infrastructure are addressed in [60]. The authors in [61] have presented a joint placement and sleep scheduling of RSUs in order to reduce both the deployment cost and energy consumption cost of VANET. They have formulated a joint optimization and scheduling strategy when the RSUs are connected to both grid power and solar power as an alternative. Zhang *et al.* [62] presented an offline scheduling of switching on/off RSUs to maintain the connectivity of the vehicles while minimizing the energy consumption cost of multiple RSUs.

In offline scheduling, the RSU has prior knowledge of arrival instances of all the vehicles and their velocities[63]. A Nearest Fastest Set (NFS)[6] scheduler has been proposed for the offline scenario. It is a greedy scheduling algorithm which schedules a faster vehicle among the set of vehicles nearest to RSU. In [64], both offline and online scheduling algorithms for energy harvesting in RSUs have been described. Moreover, the authors have presented the scheduling of downlink communication from battery powered RSUs to vehicles in order to maximize the number of serving vehicles. They have focused more on energy harvesting RSUs in VANET.

In downlink communication, the data transmission can be considered as either constant or variable[23]. The RSU radio adopts power control technique when downlink transmission is constant bit-rate at a given time instant. For constant bit-rate case, offline and online downlink scheduling problems are discussed in [18]. Therein, an offline scheduling of downlink communication has been presented in order to minimize the total communication cost while satisfying the vehicle requests in the RSU coverage. Furthermore, three greedy online algorithms have been proposed for energy efficient scheduling of vehicles inside the RSU coverage. Azimifar *et al.* [16] have proposed an energy efficient scheduling for variable bit-rate case. In addition, the authors have used V2V data forwarding while serving the vehicle requests in the RSU coverage. In [65], an energy efficient ON-OFF scheduling of RSUs has been proposed and evaluated NP completeness of the problem by formulating it as an integer programming. Three low complexity online algorithms are presented to optimize the energy consumption in multi-RSU sparse vehicular networks. However, these works have not realized the relay vehicle scheduling in the intermittently deployed RSUs.

Atallah *et al.* [66] have presented an optimization of RSUs downlink communication towards vehicles by realizing the artificial intelligence at each RSU in order to exploit optimal scheduling policy. This scheduling approach maximizes the number of vehicle requests during the battery discharge period. Furthermore, an online scenario is considered and the problem is formulated as Markov decision process using reinforcement learning (RL) technique such as Q-Learning[66]. The results obtained from this formulation have been compared with three heuristic scheduling algorithms. In [9], an energy efficient scheduling of vehicles with multiple RSUs has been proposed in order to satisfy the maximum number of vehicle requests and to reduce the total energy cost of RSUs. The authors have presented an integer linear programming formulation for the problem and polynomial time approximation algorithms.

A multi-hop V2V forwarding with the aid of density based clustering approach is discussed in Chapter 3. Minimizing energy consumption of RSUs has been addressed in this thesis while satisfying constraints on data delivery (in Chapter 3), end-to-end delay (in Chapter 4), buffering delay (in Chapter 5) and response time (in Chapter 6) in the network.

2.5 Data dissemination approaches in vehicular networks

Many studies have focused on the efficient dissemination of data via store-carry-forward vehicles in between neighbouring RSUs that are deployed in a highway environment[67, 68, 6]. The authors in [69] have studied the problem of scheduling the store-carry-forward vehicles in between fixed road-side source and destination stations. The objective of source station is to transmit the packets to passing-by vehicles and this minimizes the queuing delay and transit delay in the rural communication network. In [33], the authors have proposed a mathematical model to analyze the delivery delay of the randomly generated road information in between two neighboring RSUs. The model considers vehicle speed, density and distance between RSUs.

In [70], a probabilistic bundle release scheme has been proposed to deliver the data from source RSU to an isolated destination RSU. In this scheme, the source opportunistically selects a passing-by vehicle to carry and forward the data bundle to destination. The main focus of this scheme is to minimize the bundle transit delay in between the intermittently connected source and destination. In [71], the authors have presented a cooperative store-carry-forward method to decrease the outage time in the uncovered area. They have considered a bidirectional highway scenario to deliver the data to a target vehicle located in the uncovered area by selecting store-carry-forward (relay) vehicles. In [12], the authors have considered the selection of multiple relay vehicles for faster retrieval of the requested data by a target vehicle. In [7], the authors have assumed a bidirectional highway environment and proposed a clustering based vehicle-to-vehicle data data sharing approach by exploiting the vehicles driving in opposite direction. Similarly, the authors in [10] have discussed the data sharing strategy by exploring the synergy between the centralized infrastructure-to-vehicle scheduling in the RSU region and adhoc vehicle-to-vehicle scheduling in the uncovered area.

In [72], a clustering based multi-hop vehicle to vehicle forwarding scheme has been proposed to minimize the energy consumption of the RSU. The authors in [16] have addressed the RSU energy consumption issues by considering off-channel and in-channel V2V forwarding for data dissemination in the uncovered area. For a bidirectional highway environment, the authors in [73] have proposed a RSU-assisted cooperative relay scheduling using auction theory principles for energy efficient data delivery in the uncovered area. Moreover, a recent work [74] has addressed the same problem by considering the full-duplex communication capabilities for the relay vehicles. In [75], an optimal stoppage strategy for scheduling the passing-by vehicles in source RSU is investigated. Similarly, the authors in [76] have addressed the minimization of the transit delay and energy consumption of an RSU while scheduling the passing-by vehicles in between neighbouring RSUs. Here, the source RSU stops a passing-by vehicle and transmit the accumulated data to the vehicle if queuing delay is above specific delay bound. Otherwise, the RSU skips the vehicle and waits for next vehicle arrival. The authors have assumed that both the transmission bit-rate from RSU to vehicle and the power allocation for the RSU are constant. In view of variable bit-rate case discussed in [23], the above assumption is not practical in a highly mobile environment where data rates are subjected to path loss and the vehicle to RSU distance changes as a function of time.

In this thesis, improvement of data dissemination services to target vehicle in uncovered area is presented in Chapter 3 and Chapter 4. Maximizing data sharing between neighbouring RSUs is addressed in Chapter 5.

2.6 Clustering in vehicular adhoc networks

According to [77], the clustering schemes in VANET have been categorized into three major strategies as shown in Fig. 2.3. These schemes are 1) intelligent-based strategies 2) mobility based strategies 3) multi-hop based strategies.

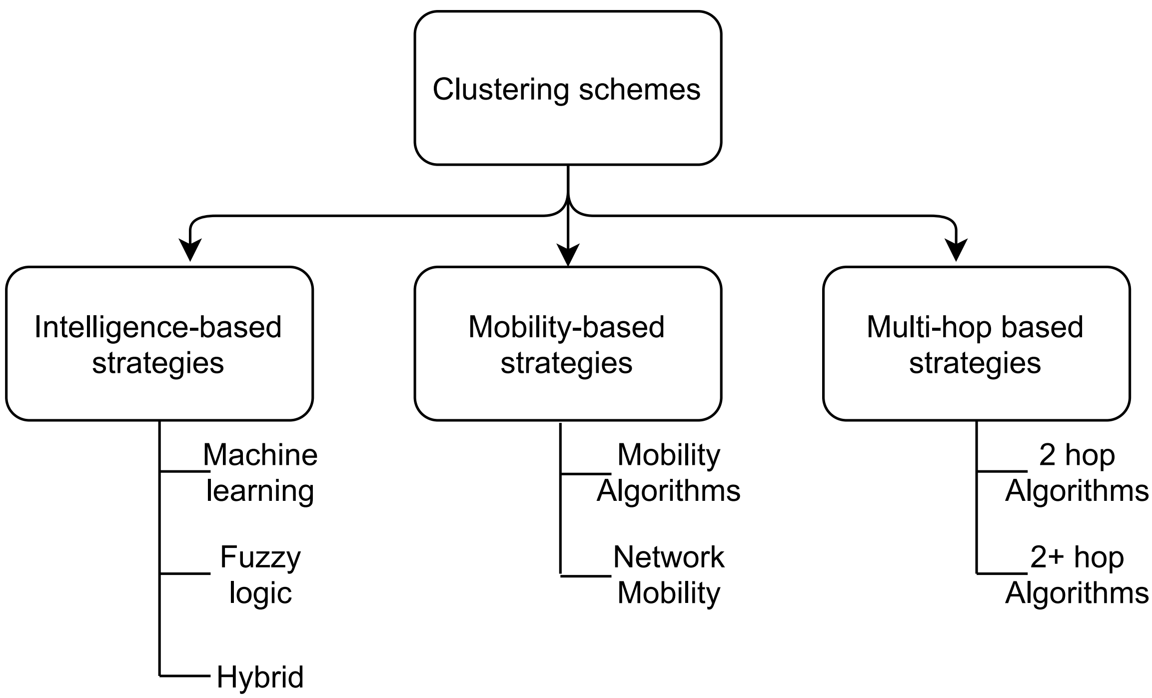


Figure 2.3: Taxonomy of clustering schemes in VANET

Intelligent-based clustering further categorized into machine learning algorithms, fuzzy logic algorithms and hybrid algorithms (combination of machine learning and fuzzy logic algorithms). In machine learning algorithms, K-means algorithm is one of the popular algorithm to divide vehicles into K number of clusters[78]. Initial centroid and vehicle coordinates are given as input. Based on the euclidean distances among the vehicles, a new centroid is computed and the centroid vehicle is elected it as cluster head. Other vehicles in the cluster act as cluster members which may leave or rejoin the cluster. Changes in the cluster members need to re-compute cluster for the election of new cluster head. Besides, nature inspired algorithms[79] are used to divide the vehicles into clusters based on their speed, direction, location and transmission range. On the other hand, fuzzy logic system[80] is used to identify clusters based on the inputs, speed and distance. Furthermore, an adaptive learning mechanism has been presented to identify more stable clusters by predicting future speeds of vehicles[81]. Hybrid algorithms combines the features of fuzzy logic with reinforcement learning (e.g., Q-learning) to improve stability and reliability of clusters[82]. On the other hand, a DBSCAN (Density Based Spacial Clustering of Applications with Noise) [83] is density based clustering technique which is used to identify arbitrary number of clusters.

Mobility based algorithms are classified into vehicle mobility and network mobility algorithms. In vehicle mobility algorithms, a Dynamic Clustering Algorithm (DCA) [84]has been presented to improve cluster stability in dynamic highway environment. This considers the relative speed of vehicles as metric along with average duration of cluster head and number of changes in the cluster. A mobility prediction-based clustering (MPBC) [85] method has been presented to predict the relative speeds of vehicles by exchanging Hello packets using Doppler effect. In network mobility algorithms, clustering strategies are introduced to reduce hand-off delays and packet losses in high speed VANET[86]. Multi-hop based clustering strategies include 2 hop and 2+ hop algorithms [87]. Multi-hop clustering is important to reduce the number of clusters and improve accessibility of cluster head in the cluster. In 2 hop clustering, cluster head is reachable from cluster member in two hops. In 2+ hop clustering, vehicles disseminate their speed, position and direction in N-hop neighbourhood.

Chapter 3 uses density based clustering mechanism for identifying nearest neighbour forwarder vehicles in the RSU coverage. The downlink communication to these nearest neighbour forwarders minimizes energy consumption of RSU and improves data delivery to target vehicle. A modified DBSCAN algorithm (Refer Chapter 3) is presented to cluster the vehicles in highway road segment.

2.7 Auction theory principles for vehicular networks

Auction theory is a powerful tool that works based on game theory principles where both sellers and buyers actively participate in auction process to determine price[88]. There are four basic types of auctions widely used in the literature.

- *Ascending-bid auction:* the seller successively raises price until only one bidder remains and wins the object for final price.
- *Descending-bid auction:* it is opposite to ascending-bid auction. The seller successively decreases price until any one of the bidder accepts the final price.
- *First-price sealed bid auction:* each buyer submits single bid for an object without having knowledge of others bids. The object is allotted to a buyer who makes highest bid.
- *Second-price sealed bid auction:* Similar to first-price bid auction, but the object is allotted to a buyer who makes highest bid for second highest bid price.

In [89], a multi-objective auction based caching scheme has been presented to allocate storage resources of content providers (CPs) over multiple overlapping RSUs. Since storage capacity of RSUs and OBUs are limited, the CPs bid for optimal allocation of RSU caching storage in order to improve data dissemination in vehicular network. On the other hand, RSUs can benefit from payments of CPs. A market matching algorithm is used to solve this multi-objective auction problem for RSU caching storage allocation. In [90], a reverse auction based computation offloading scheme has been presented for cloud-enabled vehicular environment. This scheme enables optimal offloading of computation intensive applications via opportunistic V2V channels. Buyers provide incentives to sellers for leasing their resources in order to execute offloading applications. This could allow the diversity of buyers to choose preferences over different sellers based on their computation capabilities, contact duration and transmission rates. In [91], an auction based graph allocation problem is presented to assign components (buyers) to virtual machines (sellers) while maximizing utility of buyer and satisfying concerns of sellers based on execution time and commission cost. A structure preserve matching algorithm is adopted to solve the problem with low computational complexity. An Agent assisted Smart Auction based Parking system has been discussed in [92] for optimal assignment of parking slots to autonomous vehicles. Vehicles act as bidders and parking facility provider acts as seller. A fair recurrent Vickrey-Clarke-Groves (VCG) auction mechanism is used to address truthful bidding for bidders while maximizing the utility value of parking facility providers.

Chapter 4 adopts Auction theory principles for the optimal assignment of relay vehicles to time slots, where RSU acts as a seller and relay vehicles act as bidders. The relay vehicles submit bids for time slots without having knowledge of bids submitted by other relay vehicles. The RSU allocates time slots to relay vehicles based on highest bidding increment and this has been discussed in Chapter 4.

2.8 Dynamic power allocation in vehicular networks

Dynamic power allocation is important to control the data transmission rates in order to maximize ergodic capacities of links between source and destination nodes. Lyapunov drift-plus-penalty optimization mechanism is a powerful technique which can be applied to dynamic queuing networks and other stochastic systems[93]. In [94], a dynamic power allocation scheme is presented to maximize the ergodic capacity of downlink transmission rates to relay vehicles from base station. The relay vehicles are exploited to store-carry-forward the requested data to a target vehicle in outage area. The proposed scheme[94] optimizes the power allocation to maximize downlink capacities between base station and relay vehicles subject to satisfy the up-link capacities when relay vehicles present in target vehicle range. The Lyapunov drift-plus-penalty theorem [95] has been included to determine the trade-off between amount of power allocation (penalty) and change in queue lengths (drift). A trade-off parameter V is used to balance the penalty and drift in the queuing system. This technique minimizes derived bound on the change in queue lengths along with amount of power allocation. Therefore, this optimization mechanism not only ensure queue stability but also limits the allocated power.

Minimizing latency is an important issue during transmission of packets from source to destination. But packets may belong to different priority classes of high and low priority. Satisfying the latency requirements along with the sufficient spectrum and power allocation in the system is presented in [96]. This will maximize the ergodic capacities of vehicle to network links while guaranteeing the latency requirements of transmitted packets. In [97], a multi-level water filling algorithm has been applied for dynamic power allocation in the up-link channels from vehicle transmitters to vehicle receivers. Different prices are assigned by vehicle receivers before vehicle transmitters do power allocation. This optimal pricing strategy is designed for optimization of amount of allocated power. In [98], a sub-channel matching scheme and power allocation scheme is presented by introducing alternating direction method of multipliers in order to enhance the non-orthogonal multiple access system performance.

To address the trade-off between energy consumption and buffering delays, a Lyapunov based dynamic power allocation algorithm has been presented in Chapter 5. The proposed algorithm allocates minimum power required for transmission of buffer content subject to buffer stability.

2.9 Fuzzy logic for vehicular networks

Fuzzy logic system is used to make decision on inferred values from imprecise and non-numerical information[99] . The term *fuzzy* refers to the things which are not clear or vague. To determine this situation into partially observable values, the fuzzy logic system is expressed in four stages: 1) Fuzzification 2) Fuzzy membership functions 3) Fuzzy rule base 4) Defuzzification. A fuzzy logic inference system is illustrated in Fig. 2.4

These four stages are represented as,

- *Fuzzification:* this stage converts input values (i.e., crisp value) into linguistic variables. For example, the input values are temperature, distance, height, etc.

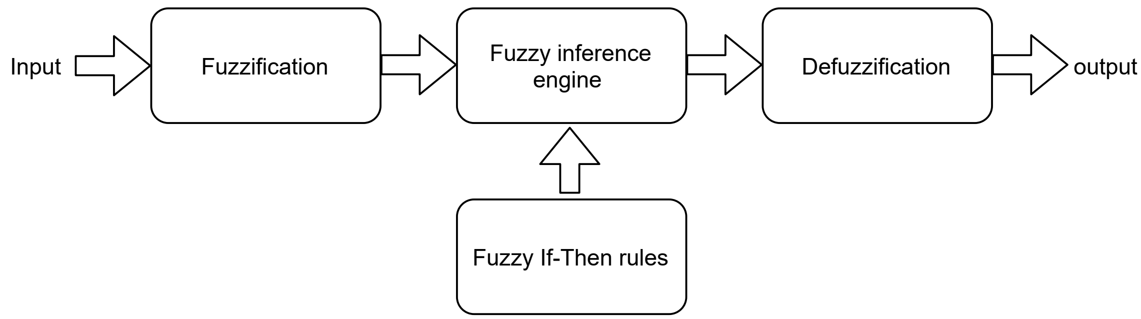


Figure 2.4: Fuzzy logic inference system

- *Fuzzy membership functions:* the stage is used to represent the degree of membership of input values with respect to linguistic variables. The membership functions are part of fuzzy inference engine.

- *Fuzzy rule base:* the fuzzy rules are set of IF-Then rules used to determine the decision making on linguistic information expressed in membership functions.

- *Defuzzification:* this stage is to convert decision on linguistic information into crisp value. The crisp values obtained from this stage act as output of the fuzzy logic system.

Finding location of neighbouring vehicles is important for efficient data dissemination among vehicles. In [99], a fuzzy logic based decision support system has been implemented for routing of emergency vehicle in least congested path. The fuzzy system decides the congestion on road location by taking inputs from sensor data such as vehicle speeds, noise on road, temperature and vehicle emissions. In [100], a fuzzy logic based intelligent vehicle localization algorithm has been presented to calculate vehicle's weights. The fuzzy logic system takes inter vehicle distance and moving direction of vehicles as input parameters. These vehicle weights act as indicator for efficient communication to neighbouring vehicles in such a way that the closer vehicles get highest weights and distant vehicles have lowest weights. In [101], a fuzzy logic intelligence mechanism for broadcasting of messages among vehicles has been presented. The vehicles transmit beacon messages to neighbouring vehicles and each vehicle identifies its multi-hop neighbours based on the input metrics such as mobility, connectivity and coverage factors. Then, the vehicles can send warning messages to their multi-hop neighbours observed using fuzzy logic system.

Intelligent data forwarding is crucial for improving the reliability of data services in VANET. In [102], a fuzzy logic based multi-criteria data forwarding algorithm has been presented to identify the best next hop node for data forwarding. The selection of next hop node depends on vehicle speeds, inter vehicular distance and link expiration time. This algorithm is used for dynamic selection of next node and evaluation of node stability during transmission. In [103], a fuzzy logic based data dissemination strategy has been presented by selecting next forwarding node in the vehicle's transmission range. This strategy has considered the inputs for fuzzy logic system as distance, vehicle mobility values and signal strength. The selected primary forwarding node defines secondary forwarding node in the propagation of forwarding message. In case of lost packet in the primary node, the secondary node takes it to advance the propagation of packet to next hop node.

In this thesis, a fuzzy logic system has been presented in Chapter 6 for computation of vehicles' weights. The weights of the vehicles are derived from set of vehicle parameters which act as metric to indicate the importance of that vehicle in scheduling process.

2.10 Fog computing for vehicular networks

Vehicular Fog Computing (VFC) extends the fog computing paradigm to a traditional VANET for efficient offloading of delay-sensitive tasks to computation enriched vehicles and this aims to improve response latency, energy consumption, throughput, etc [104, 105, 106]. However, high mobility, short connection time and diversity of resources available to vehicles lead to various challenges in VFC environment. In [104], a vehicular fog computing architecture is presented and a fog vehicle assisted traffic control system is discussed as an use case to control traffic lights in the intersection. To deal with an ever increasing demand of computation intensive applications, the RSUs that are deployed along roadside provide computation services with the aid of mobile vehicles in the RSU coverage[105]. In [106], the authors have studied a fog computing architecture for the detection of vehicular congestion which ensures an optimized communication and local processing of collected sensory data from vehicular clients. In [107], the authors have proposed task allocation of multi-user to multiple fog devices in an IoT architecture as a joint optimization problem which aims to minimize transmit power of IoT devices under delay constraints.

Various works [108, 109, 110, 111] have addressed the problems in VFC for its advancement. In [108], the authors have presented three layered vehicular fog computing architecture where tasks generated by IoT devices are offloaded to vehicular fog nodes. They have proposed a greedy task scheduling heuristic to lower the total response latency of time sensitive tasks when assigned to vehicular fog nodes with sufficient computation resources. The authors in [109] have presented an auction based approach that can assign vehicles to parking slots and exploit the fog computing services of these parked vehicles for the execution of delay sensitive tasks. In [110], the authors have presented a two stage intensive mechanism for allocation of vehicle resources and assignment of tasks to fog vehicles. The resource allocation and task assignment have been modelled as contract theory and two side matching game, respectively. A vehicular fog computing framework called vFog has been presented in [111], where the computation intensive tasks generated by user vehicles are offloaded to fog vehicles via V2V communication without roadside infrastructure support. The vFog has vehicle assisted mechanism that ensures multi-hop relaying of tasks between user vehicles and fog vehicles.

In this thesis, the Chapter 6 has presented a fog computing framework in vehicular network for the energy efficient allocation of tasks to fog vehicles in RSU communication range.

2.11 Learning algorithms in vehicular networks

The authors of [112] have presented an incentive based mechanism with dynamic pricing of vehicles to share their idle resources for task offloading in vehicular fog computing environment. To solve this priority-aware task offloading problem, the authors have proposed a soft actor-critic based deep reinforcement learning algorithm for the sake of maximising latency-aware utility and long-term reward. In [113], a resource allocation problem which aims to minimize service latency has been presented in the context of parked vehicles and slow moving vehicles for VFC. Challenges posed by high-dimensional search space has been addressed by integrating recurrent neural networks (RNN) with deep neural network

(DNN) and then combined with a heuristic algorithm for accelerating slow convergence of reinforcement learning algorithms.

A Vehicular Fog Computing System architecture has been presented in [114] for efficient offloading of computation intensive tasks to fog servers deployed in mobile public service vehicles. When a requester vehicle come up with offloading request, the computation scheduler (CS) decides whether allocate to fog vehicles or remote cloud. This resource allocation problem is modelled as semi Markov decision process and solved using value iteration algorithm. A ISVM-Q task scheduling algorithm[115] has been presented by combining the efforts of improved support vector machine(ISVM) and Q-learning, where ISVM as a value function approximate for the improvement of Q-leaning performance. A deep reinforcement learning algorithm based on SARSA (i.e., State-Action-Reward-State-Action) has been discussed in [116] for resource management in edge server and optimal allocation of tasks to edge servers while minimizing system cost and computation delay.

In this thesis, an on-policy learning algorithm (i.e., SARSA) has been presented (in Chapter 6) to learn the scheduling pattern of vehicles in vehicular fog computing environment. The proposed approach achieves faster convergence when compared to Q-learning and improves long term reward with the help of fuzzy logic system.

2.12 Summary

In this chapter, different types of cooperative communication modes in vehicular networks are discussed. Various applications of cooperative communications have been presented. A survey on energy efficient issues of road side infrastructure in highway vehicular networks has been presented. A discussion on clustering mechanisms for efficient data dissemination approaches used in vehicular networks has been included. Importance of Auction theory principles for optimal assignment problems is highlighted. Moreover, dynamic power allocation mechanism using minimization of Lyapunov drift-plus-penalty to determine trade-off between power allocation and buffer stability is discussed. Use of fuzzy logic and learning models in the context of vehicular fog computing environment have been presented. Furthermore, different works discussed in this thesis have been compared with other existing works in the literature. In next chapter, a clustering based mechanism combined with minimum cost flow graph is presented to address the energy efficient relay scheduling problem while improving data delivery to a target vehicle.

Chapter 3

Clustering based Energy Efficient Multi-Relay Scheduling in Highway Vehicular Infrastructure

In highway scenarios, the vehicles move with constant speed for long durations and they remain in RSU coverage for relatively short durations. In addition, RSUs may not provide seamless access to vehicles due to high deployment cost of the infrastructure. This leads to an uncovered area in between neighboring RSUs. A vehicle with large data requirement may leave the RSU without completely downloading all the requested data and enters into the uncovered area. Such a vehicle is referred to as *target vehicle*[71] and remaining requested data is denoted as *residual data*. Nevertheless, the RSU can transfer residual data to the target vehicle by exploiting other vehicles as store-carry-forward relays[12], which are referred to as *relay vehicles*. Proper selection of relay vehicles is an important issue in case the RSUs are energy-limited and powered by alternative energy sources. Therefore, it is important to schedule the downlink communication in order to reduce the RSU power consumption. It is worth mention that the energy efficiency of the vehicles is not an issue, since the vehicles are powered by dedicated vehicular engines.

In this chapter, an energy efficient scheduling of relay vehicles has been presented to improve data dissemination services to a target vehicle in an uncovered area. The scheduling problem is formulated as an optimization problem and a polynomial time solution is presented by modelling it as Minimum Cost Flow (MCF) graph [14]. A scheduling solution obtained from MCF provides optimal assignment of relay vehicles to time slots. Downlink channel time of RSU is divided into time slots of fixed duration. Furthermore, a clustering[15] based algorithm is proposed to identify the vehicles which are in the energy favorable locations (i.e., near to RSU) and multi-hop neighbors to the relay vehicles. Such vehicles are named as *Nearest Neighbor Forwarders (NNFs)*. Optimal scheduling of downlink communication from RSU to NNFs (instead of actual relay vehicles) further reduces the energy consumption of RSU and improves the data delivery to the target vehicle. In a given cluster, the relay vehicles are reachable from a NNF in a multi-hop distance. The data forwarding from the NNF vehicle to its multi-hop relay vehicle follows *off-channel* V2V forwarding[16, 17]. Here, the full-duplex enhanced dual-radio OBUs[36] equipped with vehicles ensure off-channel data forwarding from the NNF vehicle to its relay vehicle.

The major contributions of this chapter are as follows.

- Analyse relationship between the energy consumption of RSU and the data delivery to the target vehicle. And, determine a set of store-carry-forward relay vehicles which can establish a communication link with the target vehicle.

- Formulate multi-relay scheduling as an optimization problem, and consequently present a polynomial time solution by modeling it as Minimum Cost Flow (MCF) graph.

- A clustering based Nearest Neighbor Forward (NNF) approach is proposed to identify the vehicles which are in the energy favorable locations (i.e., near to RSU) and multi-hop neighbors to the relay vehicles. Combining the relay scheduling with the NNF approach achieves significant reduction in the power consumption of the RSU and improvement in the data delivery to the target vehicle.

Results from extensive simulations show that the proposed NNF approach augmented with MCF preform better when compared to scheduling algorithms Nearest Fastest Set (NFS)[6], MCF[18] and two more basic algorithms First Come First Serve (FCFS) and Fastest First (FF). The results also show the impact of vehicle arrival rate, vehicle transmission range and target vehicle speed on the power consumption of the RSU and data delivery to the target vehicle.

The remainder of this chapter is organized as follows. Section 3.1 presents the system dynamics along with the assumptions and derivation of effective communication time of vehicles. Section 3.2 formulates an optimization problem for energy efficient multiple relay scheduling along with a polynomial time solution. Section 3.3 proposes a *Nearest Neighbor Forward* approach to further reduce the RSU power consumption. Section 3.4 demonstrates a simulation scenario and analyzes the obtained results. Finally, section 3.5 summarizes the chapter.

3.1 System Model and Dynamics

This section introduces a system model for intermittently connected energy-constrained RSUs and the dynamics of the energy consumption of RSU versus data delivery to the target vehicle. Moreover, using trivial relative motion theory[71], a set of store-carry-forward relay vehicles have been derived along with their effective communication time to the target vehicle.

3.1.1 System model

In this work, an energy-constrained RSUs are considered to be deployed equidistantly in a highway road segment of interest, as shown in Fig.3.1. The distance between two neighboring RSUs in a road segment is considered as D and the radio coverage of each RSU is R. In this highway settings, RSUs can not provide seamless coverage (i.e., $D >> 2R$) due to significant deployment overhead. As a result, the system leaves an uncovered area $U = D - 2R$ between two neighboring RSUs[71]. Moreover, the RSUs are considered to be sustainable on renewable energy sources such as solar and wind power. The RSU consists of a single radio transceiver and can communicate with a single vehicle for a given time period, referred to as a *time slot*. The RSU uses transmit power control on downlink communication so that the constant bit rate can be achieved in each time slot, regardless of the vehicle's location within the RSU coverage[9]. The RSU obtains the estimate of downlink transmission energy cost by considering the distance-dependent exponential radio path loss model [117, 118]. In this model, a standard distance dependent exponential

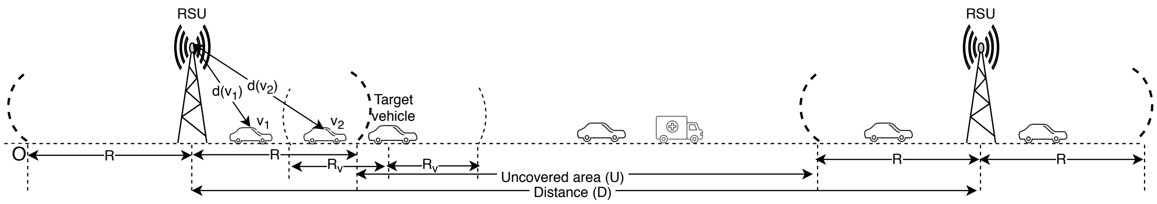


Figure 3.1: Energy-constrained Road Side Units in intermittently connected vehicular networks

path loss model[119] is used. Assuming constant bit-rate (B bits/slot) in each time slot, the relationship between distance (RSU-to-vehicle) and transmission power of RSU when communicating with a vehicle v at a time slot t is given by

$$B = \gamma \, P_{v,t} \, (D_{v,t})^{-\alpha} \tag{3.1}$$

$$P_{v,t} = \frac{B}{\gamma} \, (D_{v,t})^{\alpha} \tag{3.2}$$

where $P_{v,t}$ is the transmission power of RSU, α is a path loss exponent, γ is a scaling coefficient, and $D_{v,t}$ is the distance between the RSU and vehicle v at time t[119].

3.1.2 Assumptions

The following are the key assumptions in this model.

- The downlink (RSU-to-vehicle) communication of RSU serves the vehicles moving in one direction. The direction of vehicle flow assumed from left to right.
- The arrival of vehicles at the reference location O as shown in Fig.3.1, follows Poisson process [120] with the intensity λ in units of vehicles per second.
- RSU aware the velocities of all vehicles and their arrival instances at a location O, so that the RSU can accurately determine the positions of moving vehicles at a given instance [71].
- The velocities of vehicles are uniformly distributed within $[v_{min}, v_{max}]$ and remains constant in the interested highway road segment[121].

- All the vehicles are equipped with full-duplex dual-radio OBUs and they have same transmission range R_v. Downlink channel time of RSU is time-slotted, and all the time slots have equal duration δs.

3.1.3 Energy cost *Vs* Data delivery

In the highway scenario as shown in Fig.3.1, the large data requirement of a *target vehicle* may not be satisfied during its transit time inside the RSU coverage. However, the residual requirement of the target vehicle can be forwarded through a passing by vehicle known as *relay vehicle* which can establish a communication link with the target vehicle in the uncovered area.

From Eq.(3.2), it can be observed that the power consumption required by a vehicle near to RSU is significantly lower when compared to a more distant vehicle within the RSU coverage [9]. Therefore, the RSU prefers to communicate with a nearby vehicle where the RSU consumes less power. But, this may not always serve the target vehicle in the uncovered area. For example, as shown in Fig.3.1, suppose at a given time slot, two vehicles v_1, v_2 are moving with the speed of the target vehicle and only v_2 is within the radio range of the target vehicle. Although the RSU to vehicle distance $d(v_1) < d(v_2)$, the RSU does not prefer a nearby vehicle v_1 as a relay vehicle, because it can not establish a communication link with the target vehicle. So, it is important to select the vehicles as relays which can reduce the RSU power consumption and serves the target vehicle.

3.1.4 Effective communication time

In this section, the effective communication time of each vehicle has been estimated. It is defined as an amount of time a vehicle can act as a store-carry-forward relay to the target vehicle. Here, another scenario is considered where the vehicles are moving at different speeds and some of them can establish a communication link with the target vehicle in the uncovered area.

It depends on two factors:

i) *Dwell time* (ΔD_h): the amount of time a vehicle h spends in the RSU coverage when the target leaves RSU coverage.

ii) *Link time* (ΔL_h): the amount of time a vehicle h spends within the transmission range of the target vehicle in the uncovered area.

The minimum of ΔD_h and ΔL_h (i.e., $min(\Delta D_h, \Delta L_h)$) is the effective communication time of a vehicle $h \in V$. The effective communication time is the actual reduced outage time by a relay vehicle from the total outage time U/v_0. Then, the total number of time slots T is defined as

$$T = \lfloor (U/v_0)/\delta s \rfloor \tag{3.3}$$

A set of relay vehicles $V_r \subseteq V$ is defined in Eq.(3.4), where V is a set all vehicles in the system excluding the target vehicle.

$$V_r = \{h \in V \mid min(\Delta D_h, \Delta L_h) \geq \delta s\} \tag{3.4}$$

Effective communication time of each relay vehicle $r \in V_r$ is divided into number of time slots $\lfloor min(\Delta D_r, \Delta L_r)/\delta s \rfloor$, which defines the maximum number of times a relay vehicle can be selected for downlink communication.

3.2 Energy Efficient Multi-Relay Scheduling

This section goal is to minimize the downlink energy usage of RSU subjected to satisfy the residual data requirement of the target vehicle by selecting multiple relay vehicles. An *Integer Linear Programming* (ILP) is formulated for energy efficient relay selection and evaluated its NP-hardness. Moreover, this section provides a polynomial time solution of the problem by modeling it as a *Minimum Cost Flow* graph.

3.2.1 ILP formulation for scheduling relay vehicles

The RSU schedules downlink communication when it has a prior knowledge of all the vehicles. This can be achieved by defining a binary decision variable as follows.

$$x_{s,r} = \begin{cases} 1, & \text{if RSU selects a vehicle } r \text{ at a time slot } s \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

The inputs and outputs of the scheduler is given as follows.

INPUT: Given an input set of relay vehicles V_r. For each vehicle $r \in V_r$, using basic relative motion theory, the RSU derives *dwell time* ΔD_r and *link time* ΔL_r. The target vehicle pass through the uncovered area in T time slots, where each time slot $s \in T$. The downlink energy cost of RSU to a vehicle r at a time slot s has been denoted as $\epsilon_{s,r}$. Therefore, input to the scheduler is given as a set I, as shown in Eq.(3.6). And, also given the residual data requirement (H) of the target vehicle and time slot duration δs.

$$I = \{(\epsilon_{s,r}, \Delta D_r, \Delta L_r)\} \qquad \forall r \in V_r, \forall s \in T \tag{3.6}$$

OUTPUT: Given an input I, the objective of the scheduler is to find a schedule such that the residual requirement of target vehicle is satisfied in the uncovered area and total RSU downlink energy cost is minimized.

Therefore, lower bound on RSU total energy consumption can be computed using the following ILP.

$$\underset{x_{s,r}}{\text{minimize}} \sum_{r \in V_r} \sum_{s \in T} \epsilon_{s,r}\ x_{s,r} \qquad (OPT)$$

$$\text{subject to} \sum_{r \in V_r} \sum_{s \in T} x_{s,r} \geq \frac{H}{B} \tag{3.7}$$

$$\delta s \sum_{s \in T} x_{s,r} \leq min(\Delta D_r, \Delta L_r) \quad \forall r \in V_r \tag{3.8}$$

$$\sum_{r \in V_r} x_{s,r} \leq 1 \quad \forall s \in T \tag{3.9}$$

$$x_{s,r} \in \{0, 1\} \quad \forall r \in V_r, \forall s \in T \tag{3.10}$$

The objective function of OPT denotes the total downlink energy consumed by the RSU. Constraint (3.7) ensures the fulfillment of data requirement of the target vehicle in uncovered area. Constraint (3.8) ensures the total RSU to r communication time does not exceed the effective communication time of r with the target vehicle. Constraint (3.9) and (3.10) restricts the downlink communication to a single relay vehicle at each time slot.

3.2.2 NP-hardness proof

In this section, the formulated ILP optimization problem (OPT) for scheduling relay vehicles is shown as NP-hard, which is described in the following theorem.

Theorem 3.2.1. *The OPT optimization problem is NP-hard.*

Proof. This theorem is proved by reducing from well known NP-hard problem Seminar Assignment Problem (SAP)[122], which is a special case of Generalized Assignment Problem. Let a SAP instance has a set of students I and a set of seminars B. Each seminar $j \in B$ has a capacity of allowable number of students at most C_j. For a seminar j, assignment of a student $i \in I$ has a profit (satisfaction) p_{ij}. The objective is to assign students to the subset of seminars such that the number of students in each seminar j does not exceed C_j and the total profit(satisfaction) is maximized.

The reduction from an instance of SAP to an instance of OPT is described as follows: (1) The seminars are mapped (one-to-one) to relay vehicles. (2) The students are mapped

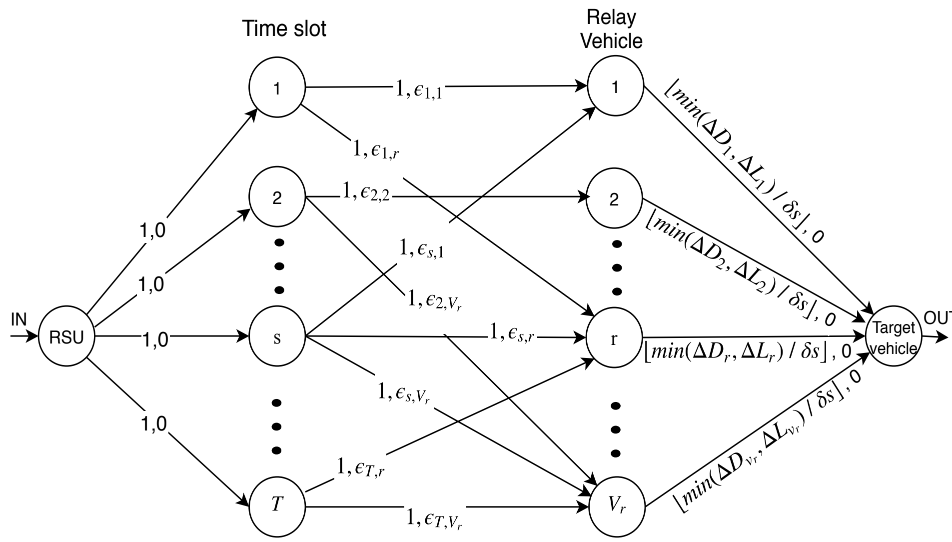


Figure 3.2: Minimum energy flow graph denoted as G. Input and Output links IN and OUT carry a flow of $min(T, \sum_{r \in V_r} \lfloor min(\Delta D_r, \Delta L_r)/\delta s \rfloor)$.

to time slots. (3) The capacity of j^{th} seminar is mapped to effective communication time of r^{th} relay vehicle, which is given as $min(\Delta D_r, \Delta L_r)$. (4) The profit of i^{th} student with respect to j^{th} seminar is negated for mapping to the energy cost of r^{th} relay vehicle at s^{th} time slot. Consider that it has sufficient number of time slots to carry residual data (H). It is clear that the transformation can be done in polynomial time. As the instance of SAP is directly mapped to an instance of OPT, can easily claim the following. The SAP instance has an assignment *if and only if* an instance of the problem has an assignment. This can establish the NP-hardness of the optimization problem. □

3.2.3 Minimum Cost Flow Graph

In this section, a Minimum Cost Flow (MCF) graph model has been presented to solve the optimization problem in polynomial time[14]. As shown in Fig.3.2, where $G = (N, E)$ is defined as set of N nodes and set of E arcs to connect the nodes. Suppose $s, r \in N$, then each arc $(s, r) \in E$ has an associated capacity $c_{s,r}$ and cost $\epsilon_{s,r}$, which denotes the maximum flow and the downlink communication cost per unit flow respectively. Each arc with associated capacity and energy cost has been labeled with an ordered pair $(c_{s,r}, \epsilon_{s,r})$.

The minimum cost flow model ensures that minimum energy consumption of RSU while respecting the maximum flow from RSU to the target vehicle. As shown in Fig.3.2, the first column of nodes represent the total set of time slots (T) available to RSU, starting from when the target leaves the RSU coverage until it enters into next RSU coverage. The second column represents the set of relay vehicles (V_r), where each $r \in V_r$ satisfies the constraint $min(\Delta D_r, \Delta L_r) \geq \delta s$.

In Fig.3.2, an arc capacity of 1 from RSU to time slot s restrict the RSU to assign the time slot more than once. The capacity of 1 in the arc from time slot to relay vehicle represent that a time slot can be assigned to only one relay vehicle. An arc from a time slot node s to a relay vehicle node r is exists only if the vehicle r is available inside the RSU coverage during the time slot s. The capacity of arc from relay vehicle to target vehicle is derived as $\lfloor min(\Delta D_r, \Delta L_r)/\delta s \rfloor$, which denotes the maximum number of time slots that can be assigned to a relay vehicle r. The cost of an arc from RSU to time slot is zero as there is a dummy flow in the arc. Similarly, the cost of the arc from relay vehicle to target vehicle set to zero. It is already mentioned that the vehicles have unlimited energy resources and V2V data forwarding does not impact the energy cost of the RSU. A unit flow cost of the arc from a time slot s to a relay vehicle r is represented as $\epsilon_{s,r}$ if there exists an arc from s to r.

The optimization problem is reformulated by modifying the residual data requirement constraint in Eq.(3.7) as an integral data as shown in Eq.(3.11). And, the effective communication time constraint in Eq.(3.8) to an integral number of time slots for each relay vehicle is given in Eq.(3.12). This becomes apparent because the minimum cost flow graph model uses Integrality Property Theorem[14], which ensures the provided capacities and input flows of MCF are integers. Moreover, the decision variable $x_{s,r} = 1$, when RSU assigns time slot s to relay vehicle r; the variable $x_{s,r} = 0$, when the time slot s leaves unassigned for all $r \in V_r$. As a result, minimum cost flow is an integer minimum energy flow in this flow graph.

$$\sum_{r \in V_r} \sum_{s \in T} x_{s,r} = \left\lceil \frac{H}{B} \right\rceil \tag{3.11}$$

$$\sum_{s \in T} x_{s,r} = \left\lfloor min(\Delta D_r, \Delta L_r)/\delta s \right\rfloor \quad \forall r \in V_r \tag{3.12}$$

Finding the minimum cost flow of G provides the minimum energy consumption of RSU in order to schedule relay vehicles for carrying residual data requirement of the target vehicle. A standard minimum cost flow algorithm gives a schedule in polynomial time in terms of T and V_r.

3.3 Nearest Neighbor Forward Approach

This section presents a *Nearest Neighbor Forward* (NNF) approach, which selects the vehicles in energy favorable locations as the data forwarders to known relay vehicles V_r. *A nearest neighbor forwarder is a vehicle which is nearest to the RSU and multi-hop neighbor to a relay vehicle.* In a time slot $s \in T$, the NNF approach selects a nearest neighbor forwarder vehicle to each relay vehicle $r \in V_r$. The nearest neighbor forwarder of a relay r is defined as NNF(r). Then, the energy cost of RSU to NNF(r) is computed for all r, denoted as $p_{s,r}$. This procedure is repeated for all time slots and formulate an input set (I_{new}) as shown in Eq.(3.13), which becomes a new input to the scheduler (i.e., MCF). The MCF augmented with the NNF approach is referred as MCF-NNF. A schedule obtained from the MCF-NNF improves the energy consumption of RSU and data delivery to the target vehicle in the uncovered area. Fig.3.3 shows the flowchart of proposed approach.

$$I_{new} = \{(p_{s,r}, \Delta D_r, \Delta L_r)\} \qquad \forall r \in V_r, \forall s \in T \tag{3.13}$$

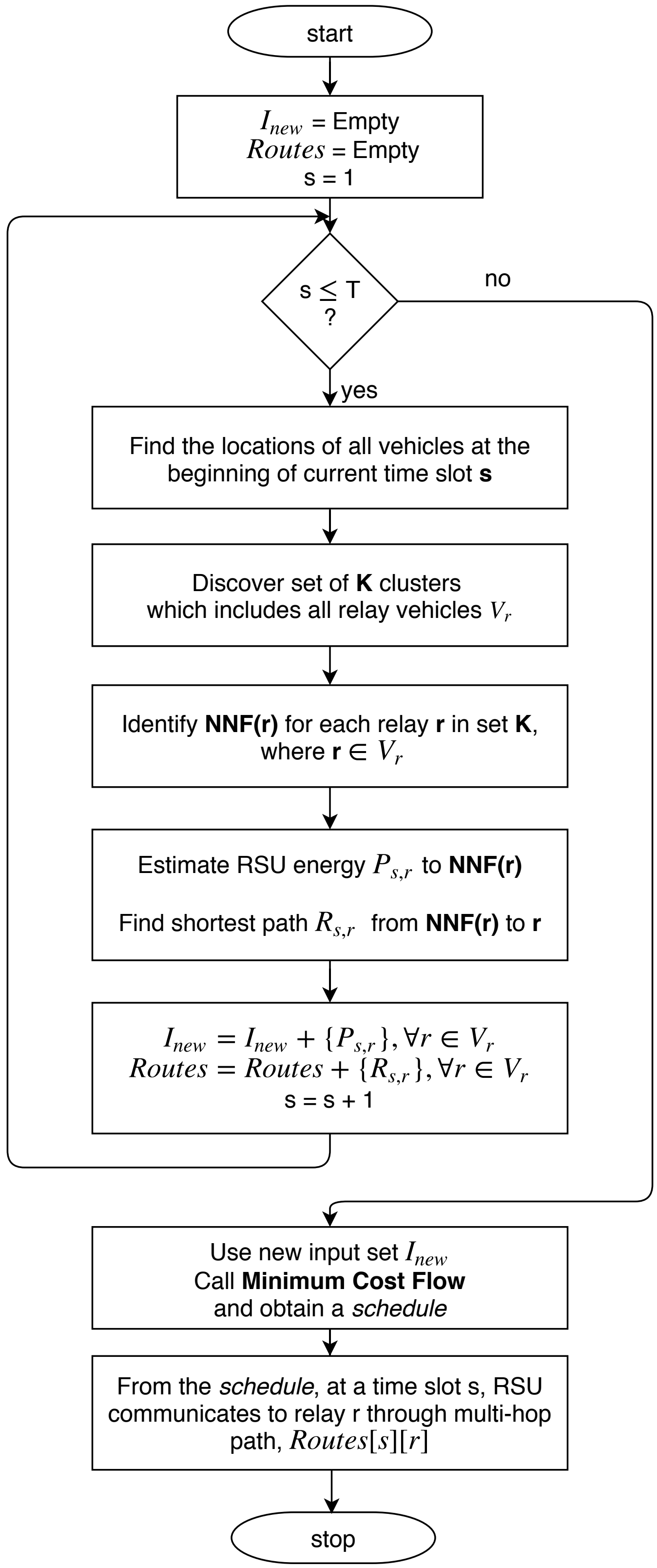


Figure 3.3: Flowchart of Nearest Neighbor Forward (NNF) approach

3.3.1 Two phase approach

The NNF approach has two phases A) clustering phase B) route discovery phase. In clustering phase, a DBSCAN (Density Based Spacial Clustering of Applications with Noise)[15] based algorithm discovers the stable clusters. A stable cluster contains a group of vehicles which are in the transmission range R_v of any other vehicle in the group and remains in the same group for the duration at least δs. For a given set of all vehicles V and relay vehicles V_r in 2-D space, the clustering phase identifies a set of clusters K wherein all the relay vehicles are included. In a cluster $C \in K$, a vehicle v_k that is nearest to RSU and in multi-hop distance to all the relay vehicles in the cluster is selected as the nearest neighbor forwarder. Then, the v_k acts as NNF(r), $\forall r \in C$. Furthermore, in a time slot s, for all $r \in V_r$, a new energy cost $p_{s,r}$ is evaluated from the Eq.(3.2), which is the energy cost of RSU to NNF(r).

The route discovery phase computes the shortest route from the vehicle NNF(r) to the relay vehicle r, where both NNF(r) and r belongs to the same cluster C. Suppose, at a given time slot, if MCF scheduler selects r as a relay, then RSU communicates to NNF(r) instead of r. Leveraging the full-duplex capabilities of dual-radio OBUs[35], the off-channel V2V forwarding accomplishes data delivery in shortest route from NNF(r) to r in the same time slot.

3.3.2 Energy cost and data delivery

At a given time slot, the set of all vehicles V are assumed to be 2-D co-ordinates in which the RSU positioned at the origin (0,0). The Fig.3.4 shows the downlink communication from the RSU to a vehicle only at the right side of the RSU. The proposed approach is applicable to the left side of the RSU also. In this example, only one relay vehicle for selection is considered. At a time slot s_1, as shown in case(i)(a), the RSU communicates to the relay r, since the relay vehicle is available inside the RSU coverage. In case (i)(b), using NNF approach, the RSU communicates to a vehicle v_k (i.e., NNF(r)), which is nearest to RSU and belongs to a cluster C, where also $r \in C$. It clearly shows the reduction in RSU energy consumption. At time slot s_2, in case (ii)(a), since no relay vehicle exists inside

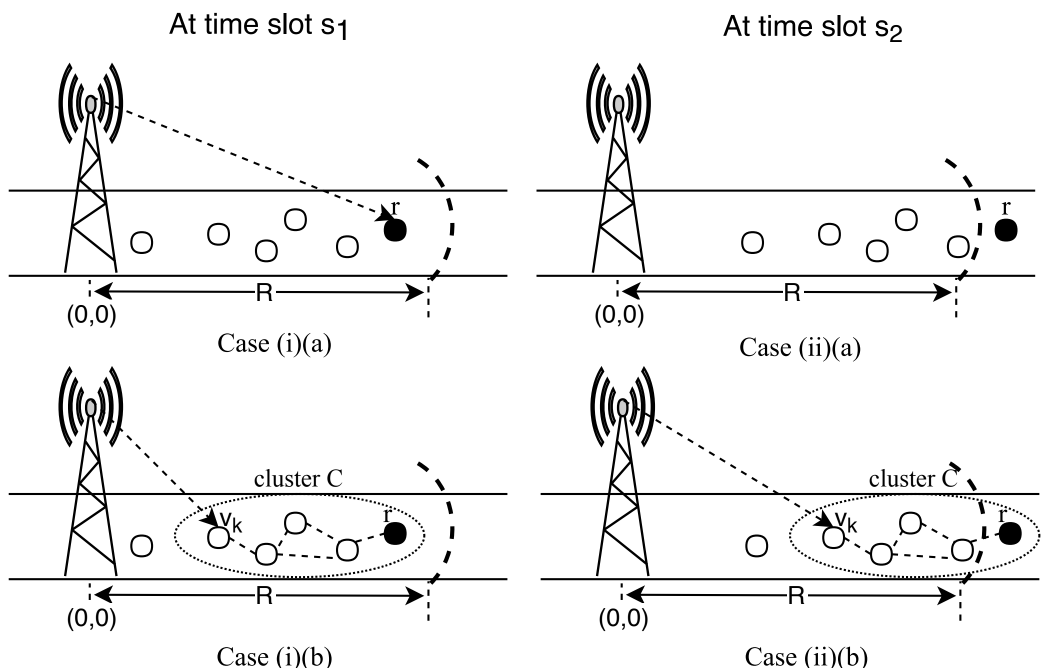


Figure 3.4: Nearest Neighbor Forward (NNF) approach. At time slot s_1, case(i)(a): RSU communicates to relay r. Case(i)(b): RSU communicates to nearest neighbor forwarder v_k, where r, v_k belongs to same cluster C. At time slot s_2, case(ii)(a): RSU cannot communicate to relay Case(ii)(b): RSU communicates to v_k, this is similar to case(i)(b).

the RSU coverage, RSU does not communicate to any vehicle. In case(ii)(b), using NNF approach, the RSU communicates to v_k, if the v_k is available inside the RSU coverage. This improves the data delivery ratio to the target vehicle.

Suppose more than one relay vehicle exists inside the cluster C, then the $v_k \in C$ acts as a nearest neighbor forwarder to all the relay vehicles in the same cluster. The reachability of a relay vehicle from the nearest neighbor forwarder is given in lemma.3.3.1. In worst case, if each discovered cluster has only one vehicle which is a relay vehicle, then $NNF(r) = r$. In such case there is no impact of NNF approach on RSU energy consumption and data delivery to the target vehicle. However, the performance of NNF approach depends on three factors such as vehicles transmission range (R_v), vehicles arrival rate (λ) and target vehicle speed (m/s). The details of performance evaluation is described in section 3.4.

3.3.3 Clustering phase

DBSCAN algorithm is one of the density based clustering algorithms[15, 123]. The density-based notion of DBSCAN generates arbitrary number of clusters $C = \{C_1, C_2, ..., C_k\}$ in arbitrary shapes, where $k \in \mathbb{N}$. Given a set of points X in 2-dimensional space, the algorithm takes two parameters, Eps is the radius of a point $p \in X$ and $MinPts$ is the minimum number of neighbors in the radius. This algorithm clusters the points which has at least $MinPts$ number of points in their Eps neighborhood. However, each cluster C_k has a set of core points and boarder points. A core point $p \in C_k$ has at least $MinPts$ points in its Eps neighborhood. The Eps neighborhood of p is defined as $N_{Eps}(p) = \{q \in X \mid dist(p, q) \leq Eps\}$ and the point p becomes a core point, if $|N_{Eps}(p)| \geq MinPts$. If a point is in Eps neighborhood of a core point, but it is not a core point then it is regarded as a boarder point. Moreover, the points which do not belong to any of the clusters are identified as noise or outliers, $noise = \{p \in X \mid p \notin C\}$. For better visualization of cluster shapes the DBSCAN uses euclidean distance function in 2-D space, i.e., $dist(p, q) = \sqrt{\sum_{i=1}^{d=2} (p_i - q_i)^2}$.

Modified DBSCAN

The proposed clustering based algorithm (Algorithm 3.2) generates stable clusters that are unaltered for the duration δs. The algorithm takes two input parameters, vehicle radio range R_v and time slot duration δs. The $Stable$ neighborhood of a vehicle v is defined as $N_{Stable}(v) = \{u \in V \mid dist(v, u) \leq R_v \,, CTime(v, u) \geq \delta s\}$. The $CTime(v, u)$ is the contact duration of neighbor vehicles v, u in a cluster, and it must be at least δs. The vehicle v becomes a core vehicle, if $|N_{Stable}(v)| \geq 1$. In order to increase the connectedness of a cluster, the algorithm takes minimum number of vehicles in the $Stable$ neighborhood as one (i.e., including v).

Since RSU has complete knowledge of the vehicles' positions and their velocities, it is easy to derive the contact time (i.e., $CTime(v, u)$) of any pair of neighbor vehicles $v, u \in V$. In a time slot, assume the vehicles v, u are in transmission range of each other, i.e., $dist(v, u) \leq R_v$. Let the location coordinates of v, u are (v_x, v_y),(u_x, u_y) and their velocities are v_s, u_s respectively. If $v_x > u_x$, then the vehicle v is ahead of u and $CTime(v, u)$

is derived as shown in Eq.(3.14). Similarly, $CTime(v, u)$ can be evaluated even if v is behind u.

$$CTime(v, u) = 1_{\{v_s > u_s\}} . \frac{R_v - dist(v, u)}{|v_s - u_s|} + 1_{\{v_s \leq u_s\}} . \frac{R_v + dist(v, u)}{|v_s - u_s|} \tag{3.14}$$

Definition 3.3.1. *($Stable$ neighborhood of a vehicle v). The stable neighborhood of a vehicle v is defined by a following condition:*
$N_{Stable}(v) = \{u \in V \mid dist(v, u) \leq R_v \,, CTime(v, u) \geq \delta s\}$

Definition 3.3.2. *(directly density reachable). A vehicle v directly density reachable from a vehicle u if*

1. $v \in N_{Stable}(u)$
2. $|N_{Stable}(v)| \geq 1$ *(core vehicle condition)*

Definition 3.3.3. *(density reachable). A vehicle v density reachable from a vehicle u if there is a chain of vehicles $v_1, ..., v_m$, $v_1 = u$, $v_m = v$ such that v_{i+1} density reachable from v_i.*

Definition 3.3.4. *(density connected). A vehicle v is density connected to a vehicle u if there is a vehicle w such that, both v,u directly reachable from w.*

Definition 3.3.5. *(cluster). For given set of vehicles V, a set of relay vehicles $V_r \subseteq V$. A cluster C is non-empty subset of V, $\emptyset \neq C \subseteq V$, satisfying the following conditions:*

1. $V_r \cap C \neq \emptyset$
2. $\forall v, u \in V :$ *if $v \in C$ and u is density-reachable from v then $u \in C$. (maximality)*
3. $\forall v, u \in C :$ *v is density-connected to u. (connectivity)*

Definition 3.3.6. *(noise). Let $C_1, ..., C_k$ be the clusters of V wrt. R_v and δs, then noise is defined as the set of vehicles in V which do not belong to any cluster $C_i, 1 \leq i \leq k$,i.e., $noise = \{n \in V | \forall i : n \notin C_i\}$*

Lemma 3.3.1. *Let C_k be a cluster wrt. R_v and δs, where $\{C_k \cap V_r\} \neq \emptyset$. Let v be any vehicle in C_k with $|N_{Stable}(v)| \geq 1$. Then C_k equals to the set $X = \{x \mid x$ is density-reachable from v wrt. R_v and δs $\}$*

As shown in Fig.3.4 and the lemma 3.3.1, a relay vehicle r is density-reachable from a vehicle v_k when $r, v_k \in$ cluster C. Hence, v_k can be a multi-hop neighbor to all relay vehicles that belongs to cluster C.

3.3.4 Route discovery phase

In route discovery phase, the NNF approach finds the shortest route $R_{s,r}$ from the NNF(r) to its corresponding relay vehicle r in a cluster. The $R_{s,r}$ is a multi-hop path denoted as NNF(r) $\mapsto$ r, in a time slot s. From the set of vehicles in a cluster, an adjacency graph can be formulated[124]. Given the adjacency graph, taking NNF(r) as a source and r as a destination, a polynomial time algorithm (e.g., Dijkstra's) finds a shortest route $R_{s,r}$. As the relative velocity of vehicles is low in highway locations, it is reasonable to assume that the mlti-hop path in a cluster is valid for the time slot duration.

3.3.5 Scheduling and data forwarding

Iterating the two phases (clustering and route discovery) over all the time slots, a schedule is obtained by MCF and shortest routes from NNF(r) to relay r, $\forall r \in V_r$. From the obtained schedule, if a relay r is scheduled in a time-slot s, then RSU communicates to NNF(r) instead of r and then data follows off-channel V2V forwarding in NNF(r) $\mapsto$ r.

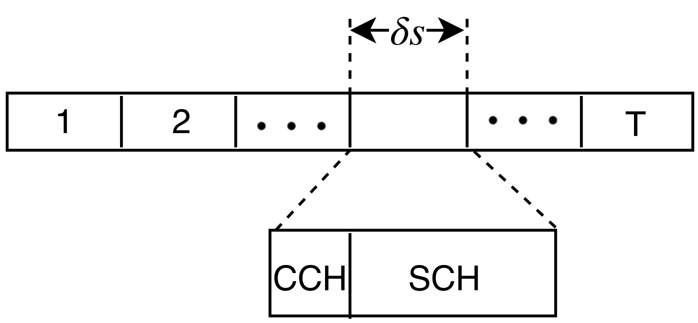


Figure 3.5: RSU communication channel time

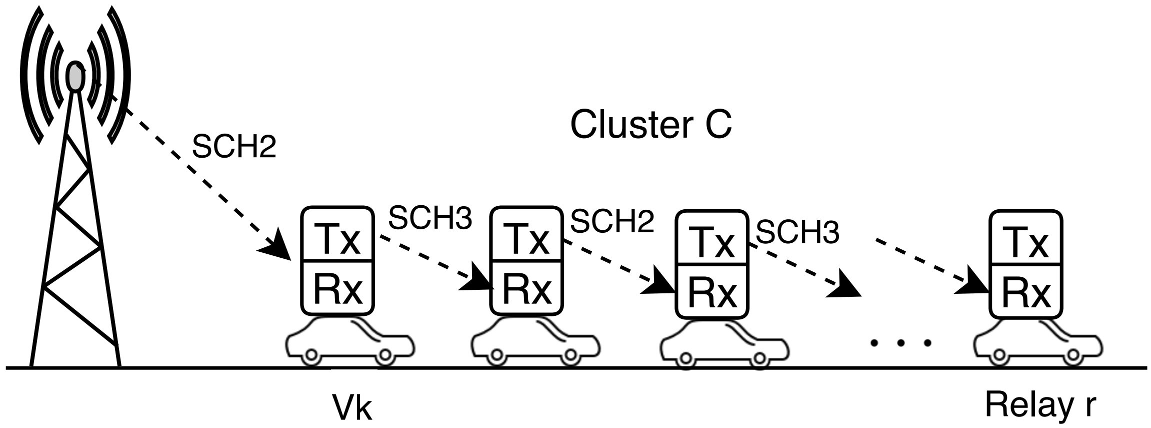


Figure 3.6: Data forwarding in service channel (SCH) at a given time slot

3.3.5.1 Full-duplex dual-radio OBUs

Leveraging the full-duplex (FD) capabilities of OBUs, the vehicles can simultaneously transmit/receive over the same channel[35]. The self-interference cancellation techniques[125] ensure the simultaneous transmission/reception in vehicular networks. A node either a vehicle or RSU, advertises a set of service parameters (e.g., SCH frequency, network information) in CCH interval, is referred as *provider*. Any other node interested in the service tuned onto advertised SCH, is referred as *user*[36]. In this scenario, the RSU (provider) advertises a set of services over a control channel (CCH). Upon overhearing the service advertisement on CCH, the interested vehicle (user) equipped with FD OBUs tuned onto advertised SCH frequency and exchange the data with the provider or acts as a relay to forward the data to another interested user. The full-duplex OBUs allows asymmetric links between the nodes (V2I or V2V), where one node targets second node, in turn second node targets another node. According to the WAVE standard [126], the service advertisement on CCH has been referred as WAVE Service Advertisement (WSA). The service providers broadcast WSAs to reach more users in the network.

3.3.5.2 Data forwarding

To ensure the data reachability from NNF(r) to r within a given time slot as shown in Fig.3.5, the RSU (provider) broadcasts WSA message to NNF(r) over the CCH channel. A WSA message contains multi-hop route NNF(r) $\mapsto$ r, SCH2, SCH3, network information, etc. Initially, the vehicle NNF(r) re-broadcasts the WSA message over the same CCH channel. The FD capabilities of OBUs allows the vehicles to transmit/receive WSA message during CCH interval, until it reach the relay r. Only vehicles in NNF(r) $\mapsto r$ rebroadcast the WSA message. Finally, the relay r stops broadcasting the WSA message. Meanwhile, the transceivers of vehicles in the multi-hop route NNF(r) $\mapsto r$ tuned to service channels SCH2 and SCH3 alternatively as shown in Fig.3.6. To this end, dual-radio of OBUs accomplishes the off-channel V2V data forwarding from NNF(r) to r in the same time slot.

Algorithm 3.1 MCF-NNF: Energy efficient RSU scheduling

// $V =$ set of all vehicles
// $T =$ set of all time slots, from Eq.(3.3)
// $V_r =$ set of relay vehicles, $V_r \subseteq V$, from Eq.(3.4)
// $R_v =$ Radio coverage of $v \in V$
// $\delta s =$ Time slot duration
initialize $newCost[s_1, ..., s_n; v_1, ..., v_m] = NULL$, $Route[s_1, ..., s_n; v_1, ..., v_m] = NULL$, $Schedule[s_1, ..., s_n; v_1, ..., v_m] = FALSE$
// $s_i \in T, v_j \in V_r; i : 1, .., n; j : 1, .., m; n = |T|, m = |V_r|$
for *all* $s \in T$ **do**
  $V' = \{$ Co-ordinates of all $v \in V$ wrt. RSU, at the beginning of current time slot $s\}$
  $(Cost, Path)$ =Call NNF-DBSCAN$(V', V_r, R_v, \delta s)$
  for *all* $r \in V_r$ **do**
    $newCost[s][r] = Cost[r]$, $Route[s][r] = Path[r]$
  end for
end for
// Call Minimum Cost Flow scheduling from section.3.2.3
$Schedule[T][V_r]$ = Call MinCostFlow($newCost[T][V_r]$)
for *all* $s \in T$ **do**
  for *all* $r \in V_r$ **do**
    if $Schedule[s][r] == TRUE$ **then**
      In time slot s, RSU communicates to r through the multi-hop path $Route[s][r]$
    end if
  end for
end for

Algorithm 3.2 NNF-DBSCAN $(V', V_r, R_v, \delta s)$

// V' is the set of coordinates of all vehicles in 2-D space wrt. RSU at origin (0,0)
// V_r is the set of all relays
initialize $Cost[v_1 : v_m] = NULL$, $Path[v_1 : v_m] = NULL$ // where, $v_i \in V_r$; $i : 1, ..., m$, $m = |V_r|$
assign$[\forall v \in V'] = UNCLASSIFIED$, cluster_id =1
for *all* $x \in V_r$ **do**
  if *assign[x] == UNCLASSIFIED* **then**
    /* ***Clustering Phase*** */
    initialize set $C = \{x\}$, set $Relays = \{x\}$, queue $seeds$ as empty, $seeds.enqueue(Neighborhood(V', x, R_v, \delta s))$, $assign[x + seeds] = cluster\_id$, $nearD = dist(x, RSU)$, $nearV = x$
    while $seeds \neq \emptyset$ **do**
      $y = seeds.dequeue()$, $C.append(y)$
      if $y \in V_r$ **then**
        $Relays.append(y)$
      end if
      $d = dist(y, RSU)$
      if $d < nearD$ **then**
        /* $nearV$ is a NNF vehicle at distance $nearD$ wrt. RSU */
        $nearD = d$, $nearV = y$
      end if
      $neighbors = Neighborhood\ (V', y, R_v, \delta s)$
      for *all* $w \in neighbors$ **do**
        if *assign[w] == UNCLASSIFIED* **then**
          $seeds.enqueue(w)$ $assign[w] = cluster\_id$
        end if
      end for
    end while
    /* ***End of Clustering Phase*** */
    /* ***Route Discovery Phase*** */
    $\mathcal{G}$ =Call Construct-Adjacency-Graph (C)
    for *all* $r \in Relays$ **do**
      // Call Dijkstra′s polynomial time algorithm
      $Path[r]$ =Find-Shortest-Path $(\mathcal{G}, nearV, r)$, $Cost[r] = \mathcal{E}$ /* From Eq. (3.2), $\mathcal{E}$ is the communication cost of RSU to the vehicle $nearV$ at a distance $nearD$. */
    end for
    /* ***End of Route Discovery Phase*** */
    $cluster\_id = cluster\_id + 1$
  end if
  /* Remaining vehicles are UNCLASSIFIED or NOISE */
end for
return $(Cost[v_1 : v_m], Path[v_1 : v_m])$

Algorithm 3.3 $Neighborhood\ (V', v, R_v, \delta s)$

$N_{Stable} = \{u \in V' \mid dist(v,u) \leq R_v, CTime(v,u) \geq \delta s\}$ **return** N_{Stable}

3.3.6 Algorithm for offline energy efficient RSU scheduling

The algorithm 3.1 presents an energy efficient relay selection (MCF-NNF) method, for a given set of all vehicles V, subset of realy vehicles V_r and number of time slots T. At the beginning of each time slot s, it finds a set V', which contains the 2D coordinates of all $v \in V$ with respect to the RSU positioned at the origin (0,0). It invokes the procedure NNF-DBSCAN given in the algorithm 3.2. The NNF-DBSCAN uses Nearest Neighbor Forward approach described in the section 3.3. The algorithm 3.2 identifies the Nearest Neighbor Forwarders corresponding to all $r \in V_r$. As aforementioned, a nearest neighbor forwarder of r can be defined as NNF(r). The NNF-DBSCAN computes the energy consumption cost of RSU to NNF(r) and finds the path from NNF(r) to r, $\forall r \in V_r$. The computed costs and paths have been updated in the $newCost$ and $Route$ respectively. The algorithm 3.1 repeats the steps 2∼5 for all $s \in T$.

The algorithm 3.2 presents a procedure NNF-DBSCAN as described in the section 3.3, given V', V_r, R_v and δs. The algorithm starts with an arbitrary relay vehicle $r \in V_r$. If r is *UNCLASSIFIED*, then the procedure retrieves all the vehicles in the set V' that are density-reachable from r. In clustering phase, If r satisfy the core vehicle condition, then the procedure yields a cluster C based on R_v and δs. It expands the cluster by finding a set of seed vehicles (i.e., $seeds$) from the neighborhood function defined in algorithm 3.3. If a seed vehicle is *UNCLASSIFIED*, then assign it to the current cluster. For each seed vehicle, it further finds the seed vehicles and includes in the set $seeds$. It repeats until the $seeds$ becomes empty. By the end of clustering phase, the procedure discovers a cluster $C \subseteq V'$ and a sub set of relay vehicles (i.e., $Relays$) which are equal to $C \cap V_r$. Moreover, the procedure identifies a NNF vehicle $nearV \in C$, which is nearest to RSU at a distance $nearD$.

The route discovery phase, from line 17∼19, finds shortest paths from vehicle $nearV$ to all relays that belongs to C. First, it constructs an undirected graph $\mathcal{G} = \{N, E\}$ where N is a set of nodes (vehicles) in the cluster C and E is a set of undirected edges between nodes.

The undirected graph $\mathcal{G}$ is defined as $\mathcal{G} = \{(v,u) \in E \mid dist(v,u) \leq R_v, \forall v,u \in N\}$. For each $r \in Relays$, the shortest path from $nearV$ will be determined using a polynomial time algorithm (e.g., Dijkstra's), taking hop count as a distance metric. Similarly, at line 36, it computes the energy cost of RSU to $nearV$. It repeats the steps 3∼22, until all the the relays classified to clusters. The remaining vehicles which does not belong to any cluster are denoted as noise. The computed $Cost$ and $Path$ returns to algorithm 3.1.

Finally, in algorithm 3.1, a polynomial time scheduler (e.g., Minimum Cost Flow graph) computes the schedule using the cost matrix $newCost$. If the ***Minimum Cost Flow*** schedules a relay vehicle r in a time slot s, then the $Schedule[s][r]$ is TRUE. Otherwise, it is FALSE. From the obtained schedule, at a time slot s the RSU communicates to NNF(r) instead of r. The data communication from NNF(r) to r follows the multi-hop route $Route[s][r]$, when RSU communicates to NNF(r).

3.3.7 Complexity analysis

The worst case time complexity of DBSCAN algorithm has been claimed as $O(n^2)$ [127], where n is the number of points in 2D space. The basic DBSCAN [15] starts with an arbitrary point and expands the clusters for every unclassified point in a set of points. The NNF-DBSCAN algorithm uses the original DBSCAN in clustering phase. In the clustering phase, for every unclassified arbitrary relay $r \in V_r$, the algorithm discovers a cluster C, where single *Stable* neighborhood query takes time $O(n)$. Furthermore, in route discovery phase, the adjacency graph can be represented in $O(n^2)$ time. The binary heap implementation of Dijkstra's shortest path algorithm needs $O(n\,log\,n)$ time. Finding shortest path for all relays $r \in C$ requires $O(n^2\,log\,n)$ time. Overall, the NNF-DBSCAN can be performed in polynomial time $O(n^3\,log\,n)$.

3.4 Performance evaluation

In this section, the performance of proposed *Nearest Neighbor Forward* (NNF) approach is investigated. The simulation model is described in the section 3.4.1 and the results are presented by comparing with existing algorithms *First Come First Serve* (FCFS), *Fastest*

First (FF), *Nearest Fastest Set* (NFS) and *Minimum Cost Flow* (MCF) graph. Moreover, it is demonstrated that the impact of key parameters such as vehicles arrival rate (λ), vehicles transmission range (R_v) and target vehicle speed (m/s).

3.4.1 Simulation setup

The input vehicular trace given to the schedulers is taken from the highway environment which has a special characteristic of maintaining constant vehicle speeds for long durations[121]. Moreover, the energy-limited RSU is assumed to be located close to highway segment where the vehicles are passing by one direction. The highway segment may consist of several lanes wherein the vehicles may overtake each other without changing their speeds. The RSU serves the target vehicle using direct downlink communication when it is available inside the RSU coverage. When the target vehicle leaves the coverage range, the RSU schedule the relay vehicles in order to deliver residual data to the target vehicle. The simulation time starts when the target vehicle leaves the RSU coverage and continues until it enters the next RSU coverage. The downlink communication of RSU follows the pre-computed schedule for all the time slots. Total number of time slots to be scheduled depends on the length of uncovered area and target vehicle speed.

In this scenario, arrival of vehicles are modeled as Poisson process[120]. Vehicle speeds follow uniform distribution in the range $[v_{min}, v_{max}]$, with $v_{min} = 15m/s$, $v_{max} = 25m/s$, the average speed is $20m/s$ and standard deviation 5. Energy cost estimates of RSU are readily available in this type of scenario[117, 118]. The associated energy costs are based on the vehicle location in the RSU coverage. It is assumed that the energy costs are derived from distance dependent path loss model using path-loss exponent $\alpha = 3$. Based on the constant bit-rate assumption, the RSU estimates the total energy consumption cost of the downlink communication of the offline schedule. The simulation has been conducted using a discrete event simulator, and obtained results are averaged over 1000 trials. The details of additional parameters are given in the Table 3.1.

Table 3.1: Simulation Parameters

Parameter	Value
RSU radio coverage (R)	1000m
Uncovered area (U)	10000m
Vehicle speeds $[v_{min}, v_{max}]$	15m/s $\sim$ 25m/s
Vehicle arrivals (λ)	Poisson process
Downlink bit-rate (B)	Constant
Time slot duration (δs)	1 sec
Scaling co-efficient (γ)	1
Path loss exponent (α)	3

3.4.2 Results and discussions

Various energy efficient offline schedulers have been proposed in the literature [18][16], including Nearest Fastest Set (NFS) scheduler and Minimum Cost Flow (MCF) graph scheduler. The NFS scheduler is a greedy algorithm, which selects a fastest vehicle among the set of vehicles nearest to RSU. The MCF scheduler is presented in [18], is a polynomial time algorithm, which minimizes the total RSU energy while serving the data requirement of *individual vehicles*. A polynomial time solution to the problem is presented in section 3.2.3, where the MCF scheduler is used to minimize the total RSU energy consumption while serving the residual data requirement of the *target vehicle* through store-carry-forward relay selection.

The proposed MCF-NNF approach is compared with the NFS, MCF and two more basic algorithms FCFS and FF. The FCFS selects a relay vehicle with least arrival time, where as FF schedules fastest relay vehicle in each time slot. Performance of the MCF-NNF is demonstrated in terms of *RSU Energy consumption per time slot* and *Residual Data Completeness*. The RSU Energy consumption per time slot is defined as the average energy consumption of RSU in order to deliver data (B bits) to the target vehicle. The *Residual Data Completeness* is defined as the percentage of data delivered to the target vehicle i.e., percentage of time slots successfully assigned to relay vehicles by the scheduler.

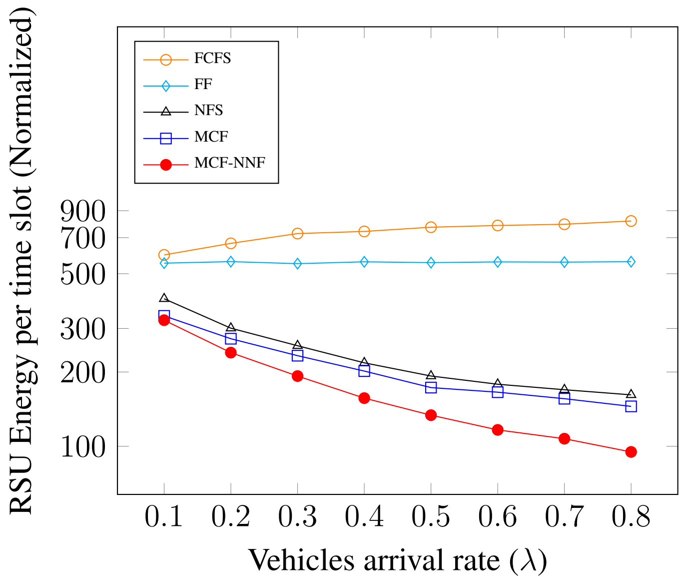


Figure 3.7: RSU energy per time slot versus Vehicles arrival rate (λ). Vehicles transmission range R_v=100m and target vehicle speed = 20m/s

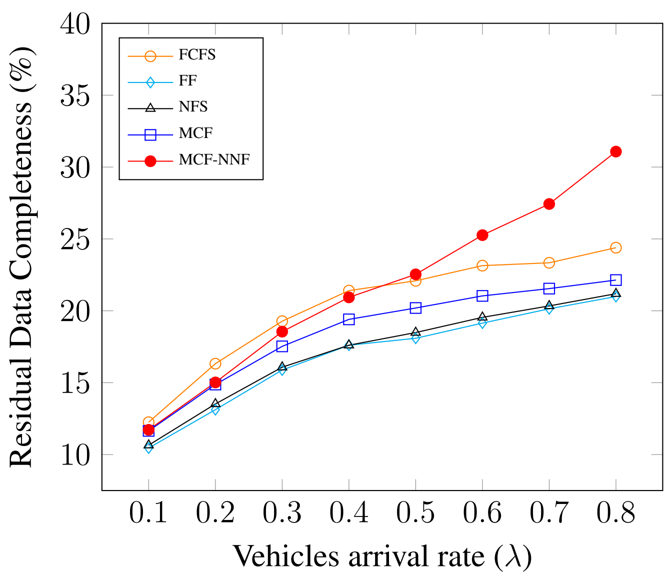


Figure 3.8: Residual data completeness versus Vehicles arrival rate (λ). Vehicles transmission range R_v=100m and target vehicle speed = 20m/s

3.4.2.1 Impact of vehicle arrival rate (λ)

Number of vehicles arrival per second into the highway segment is denoted as λ. The arrival rate of vehicles implies the number of vehicles present in the highway segment. Therefore, the number of possible relay vehicles for the target vehicle also increases. In

Fig.3.7, it is observed that as the vehicle arrival rate increases the possibility of selecting best relay vehicle increases. Therefore, the scheduler has a great chance of assigning the time slots to the best subset of relay vehicles with less energy consumption. However, FCFS results total energy consumption in the orders of magnitude much higher than that of other algorithms shown in graphs. This is because FCFS tends to serve the relay vehicles near to outer edge of the RSU. It raises energy consumption at higher λ values, because the serving relay vehicles are more close to the outer edge of the RSU. When compared to FCFS the other algorithms perform better and saves RSU energy. It is worth mention that the amount RSU energy consumption is normalized to RSU coverage range[16], as the downlink energy cost depends on the distance from RSU to vehicle.

The algorithms (NFS, MCF, MCF-NNF) result lower energy costs in case of higher vehicle arrival rate. Here, the MCF has complete knowledge of contention in all the time slots and therefore, it can schedule the vehicles with better performance and almost 5% of energy saving is recorded when compared to the NFS algorithm. In addition, the proposed MCF-NNF has significant improvement in energy consumption by 23% and 28% when compared to MCF and NFS respectively. When the number of vehicles are less at $\lambda = 0.1$, the proposed approach behaves similar to MCF, due to the fact that discovered clusters are formed by the limited number of multi-hop neighbors.

Fig.3.8 shows the number of time slots assigned to relay vehicles increases with the vehicle arrival rate. This is due to the increasing number of vehicles which can establish a communication link with the target vehicle in the uncovered area. Here, the performance of the FCFS is better because it only considers the assignment of time slots to relay vehicles regardless of their location in the RSU region. It is observed that FCFS outperforms all algorithms at lower vehicle arrival rates below 0.5, but it completely ignores the RSU energy efficiency. However, the proposed approach shows clear dominance at higher vehicle arrival rates above 0.5 and its residual data completeness is 6%, 25% and 16% more when compared to FCFS, NFS and MCF respectively.

The average energy consumption of FF is constant in all graphs because its performance is independent of vehicle arrival rate, vehicle transmission range and target vehicle speed. The residual data completeness of FF is same as NFS and the behavior is almost similar to

NFS because of its greedy nature.

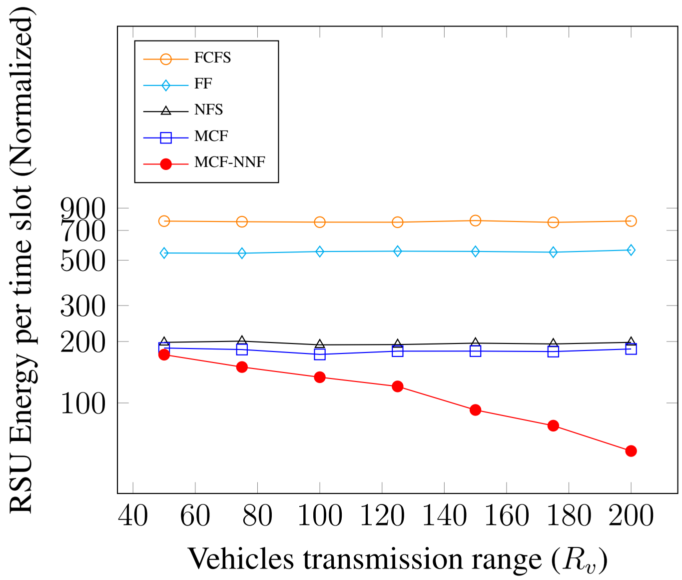


Figure 3.9: RSU energy per time slot versus Vehicles transmission range (R_v). Vehicles arrival rate λ=0.5 and target vehicle speed = 20m/s

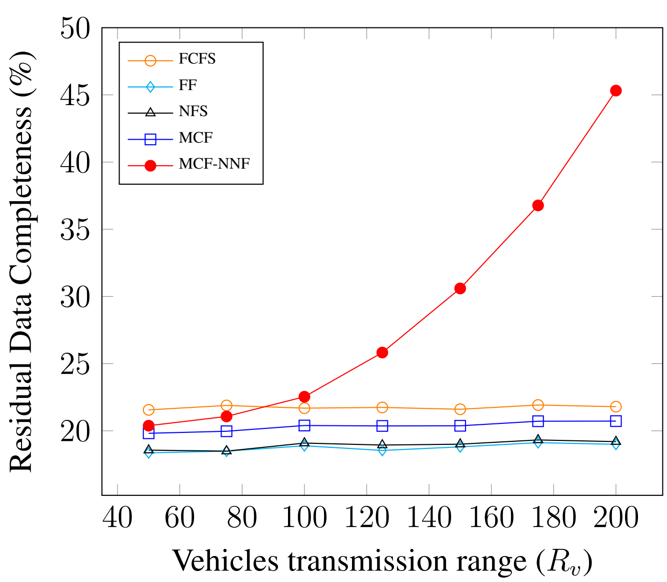


Figure 3.10: Residual data completeness versus Vehicles transmission range (R_v). Vehicles arrival rate λ=0.5 and target vehicle speed = 20m/s

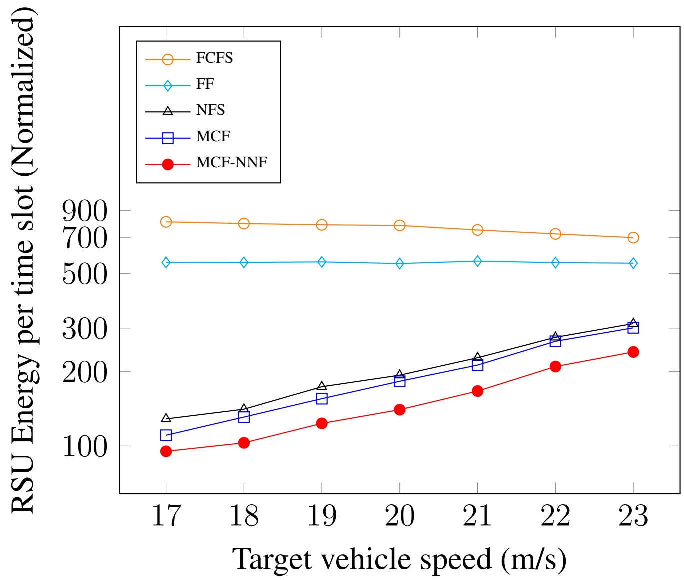


Figure 3.11: RSU energy per time slot versus target vehicle speed. Vehicles transmission range R_v=100m and vehicles arrival rate λ=0.5

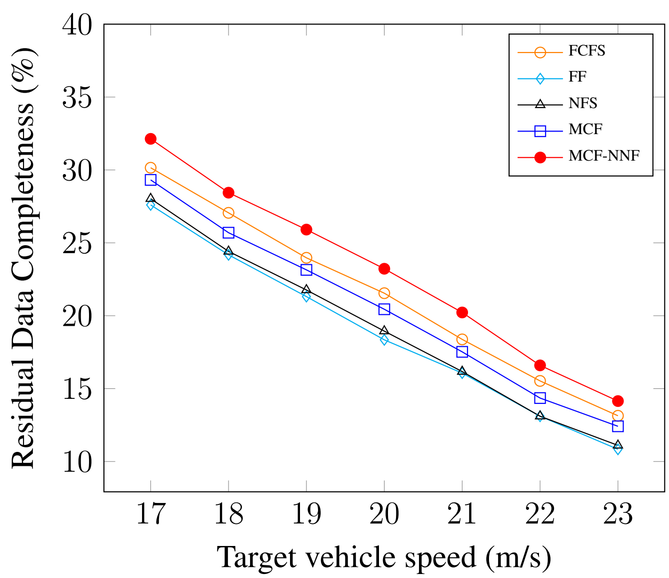


Figure 3.12: Residual data completeness versus target vehicle speed. Vehicles transmission range R_v=100m and vehicles arrival rate λ=0.5

3.4.2.2 Impact of vehicle transmission range (R_v)

As aforementioned, the RSU energy consumption and residual data completeness greatly depend on the number of vehicles present in the highway segment. Fig.3.9 shows the RSU energy per time slot versus vehicle transmission range. The proposed NNF approach with MCF has profound impact on reducing RSU energy consumption when compared to other algorithms. As the transmission range of vehicles increases, the number of vehicles in each cluster also increases. Therefore, it is feasible to select a vehicle in an energy favorable location as a nearest neighbor forwarder. Then, RSU can forward the data to a relay vehicle even if the relay vehicle is outside the RSU coverage and its nearest neighbor forwarder is inside the RSU coverage. Taking constant values of vehicle arrival rate 0.5 and the target vehicle speed 20m/s, it is observed that the performance of FCFS, NFS and MCF is not varying with increasing R_v. This due to the number of vehicles in the highway segment and number of relay vehicles that can reach the target vehicle are constant. When the vehicle transmission range is low, the NNF approach has no impact on energy saving and its performance is similar to other algorithms except FCFS and FF. The percentage of energy savings of MCF-NNF is observed nearly 71% and 57% when compared to NFS and MCF respectively.

Fig.3.10 clearly indicates the impact of increasing vehicle transmission range on residual data delivery to the target vehicle. This is due to the increasing cluster size and the number of multi-hop neighbors in each cluster. Increasing cluster size facilitates the data delivery to relay vehicles, even if the relays are outside of RSU coverage. It is observed that the MCF-NNF performance is more dominant from 100m transmission range. Although FCFS has edge over all algorithms in lower transmission ranges, it has poor performance in RSU energy saving. Except MCF-NNF, other algorithms results constant data delivery due to constant vehicle arrival rate and target vehicle speed. However, the MCF-NNF is exceptionally good at higher transmission ranges and its residual data completeness is 33%, 52% and 42% more when compared to FCFS, NFS and MCF respectively.

3.4.2.3 Impact of target vehicle speed (m/s)

The vehicle speeds are distributed uniformly in the range $[15m/s, 25m/s]$ with mean speed $20m/s$, as aforementioned. In the above analysis, the target vehicle speed has been considered as constant in the highway segment. Here, the performance of the proposed approach is demonstrated by varying the target vehicle speed. When the target speed is low, the number of vehicles that can reach the target in the uncovered area would be more. As the target speed increases, the number of possible relay vehicles decreases. In Fig.3.11, the algorithms NNF, MCF and MCF-NNF tends to increase the RSU energy consumption at higher target speeds. In contrast, FCFS tends to reduce energy consumption with increasing target speeds because the serving relay vehicles are not close to the outer edge of the RSU. However, the performance of MCF-NNF is better compared to FCFS, FF, NFS and MCF, this is due to fixed vehicle transmission range.

Fig. 3.12 indicates the decreased data delivery with increased target speed. The reasoning is much similar to the above analysis that the number of possible relay vehicles decreases with the increasing target speed. Therefore, lower residual data completeness is observed when the target speed is high. However, the overall performance of the proposed approach is better when compared to the other algorithms.

3.5 Summary

In this chapter, a set of store-carry-forward relay vehicles and their effective communication times with a target vehicle moving in an uncovered area are determined. An offline energy efficient scheduling of relay vehicles is formulated as an Integer Linear Programming and evaluated its NP-hardness by reducing from well-known Seminar Assignment Problem. A Minimum Cost Flow (MCF) scheduler is presented for assigning time slots to relay vehicles in polynomial time. Moreover, a clustering based Nearest Neighbor Forward (NNF) approach is introduced to identify vehicles which are in energy favorable locations and acts as data forwarders to the relay vehicles. This data forwarding follows off-channel V2V forwarding. Combining off-channel V2V forwarding with relay scheduling further reduces the energy cost of the RSU and improves the data delivery to the target vehicle. This is because of V2V forwarding does not incur extra cost on RSU energy usage, since vehicles have adequate energy reserves. The results from extensive simulations show that the NNF approach combined with MCF performs better when compared to offline algorithms NFS, MCF and two more algorithms FCFS and FF. The performance of the proposed scheduling shows a strong impact under the scenarios such as high vehicle arrival rate, high vehicle transmission range, low target vehicle speed.

In next chapter, an energy efficient multi-relay scheduling has been performed using cooperation between neighbouring RSUs in order achieve faster data delivery to the target vehicle in the uncovered area.

Chapter 4

Auction based Energy-Efficient Cooperative Relay Scheduling in Bidirectional Highway Scenarios

Efficient data delivery strategies are important for achieving more reliable data services (e.g., large file download, sensor data transfer, etc.) in highway vehicular networks [7, 8]. Proper scheduling of relay vehicles using cooperation between neighbouring RSUs in a bidirectional highway scenario is a useful strategy for improving data delivery services to the target vehicle. By exploiting cooperation among neighbouring RSUs, the data delivery can be improved for the target vehicle via relay scheduling in both forward and opposite directions. Although improvement of data delivery services is achievable using efficient relay scheduling, the amount RSU energy consumption and data delivery delay are also important performance measures to be considered for providing reliable data dissemination services in uncovered areas. Therefore, the objective of this chapter is to minimize the energy consumption of RSU while achieving faster data delivery to the target vehicle in bidirectional highway scenario.

In this chapter, a bidirectional highway scenario has been considered where the relay vehicles are scheduled in both forward and opposite directions to minimize the RSU energy consumption and end-to-end delay to the target vehicle. Cooperation of neighbouring RSU has been utilized for transmission of unserved data to the target vehicle in opposite (backward) direction of target vehicle. The neighboring RSU (which is next RSU in target moving direction) schedules the relay vehicles driving in opposite direction by ensuring the energy consumption and end-to-end delay requirements. Faster data delivery by minimizing end-to-end delay is also essential to improve the quality of service requirements when running infotainment applications such as online video transfer, online gaming, etc. In the proposed approach, the RSUs apply *Auction Theory*[19] principles to schedule the suitable relay vehicles in both forward and backward direction. Here, RSUs and relay vehicles act as sellers and bidders, respectively. The seller (RSU) auctions the time slots and receives bids from relay vehicles (Bidders). The bidders do not have bidding information of other bidders and therefore, the bidders only use their local information (e.g., speed, position, cooperative cache size, direction, etc.) and then submit bids to seller. The seller or RSU selects the bidders solely based on the bids received from the bidders or relay vehicles. Then, the RSU optimally assigns the relay vehicles to time slots. When target vehicle leaves RSU region, the target vehicle is designated to perform ad hoc V2V scheduling in the uncovered area when multiple relay vehicles are present in its radio range. This will improve the data delivery ratio by making the best utilization of V2V communication bandwidth. The V2V forwarding cannot affect the energy consumption of RSUs because the vehicles are assumed to have sufficient energy reserves.

Major contributions of this chapter are as follows.

- Determine the set of relay vehicles in both directions (forward and backward) in a bidirectional highway segment. The relay scheduling problem is formulated as an Integer Linear Programming problem (ILP) and its NP hardness is proved. To solve the problem, a greedy algorithm (GA) is presented by carefully scheduling the forward relay vehicles which causes minimum RSU energy consumption and end-to-end delay to target vehicle.

- Propose a forward relay scheduler (FRS) based on *Auction Theory* to schedule the relay vehicles in target moving direction. In addition, an Auction-based RSU assisted relay scheduling (RRS) algorithm is proposed to utilize the cooperative sharing between neighboring RSUs and schedule the relay vehicles in both forward and backward directions for serving the target vehicle. Besides, the target vehicle performs Vehicle to Vehicle (V2V) scheduling when multiple relays establish V2V links with the target vehicle.

Exhaustive set of simulations are performed to compare the proposed algorithm with other existing algorithms. Simulation results show the proposed Auction-based RSU assisted multiple relay scheduling algorithm outperforms other existing algorithms in terms of RSU energy consumption, end-to-end delay and residual data delivery for the vehicle arrival rate, target vehicle speed and cooperative cache size.

The remainder of this chapter is organized as follows. Section 4.1 describes the system architecture of energy-limited RSUs in bidirectional highway scenario. Further, it presents the relay selection in forward direction. Section 4.1.5 provides the ILP formulation of relay scheduling problem. Section 4.2 and 4.3 describes the proposed Auction-based forward relay scheduling and RSU assisted relay scheduling, respectively. Section 4.4 presents the performance comparison of the proposed algorithms in terms of average RSU energy consumption, average end-to-end delay and residual data delivery. Section 4.5 summarizes the chapter.

4.1 System model and problem formulation

The system considers energy-limited RSUs that are deployed along a bidirectional highway segment as shown in Fig.4.1. The RSUs are connected via backhaul wired links which can facilitate Infrastructure to Infrastructure (I2I) communication among the RSUs. Due to high installation overhead of the roadside infrastructure, the system leaves an uncovered area in between the neighboring RSUs as shown in Fig.4.1. Here, each RSU is enabled with an edge server in order to serve the vehicles on road. The vehicles are assumed to maintain constant speed in the highway segment and they remain in RSU coverage region for relatively less time. A vehicle in an RSU region can send requests to edge server in that RSU. The edge server consists a queuing buffer to store the requests received from the vehicles and it allows each request to wait for processing. After processing the request, the edge server sends response data to the requester vehicle. Meanwhile, the requester vehicle

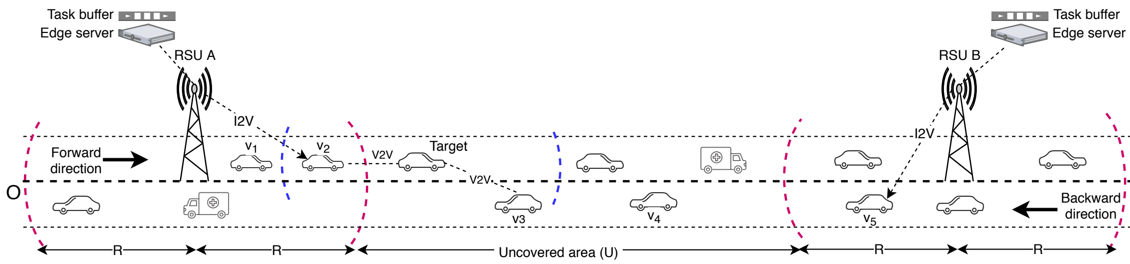


Figure 4.1: Energy-limited Road Side Units (RSUs) in bidirectional highway scenario

may leave the RSU region with residual data requirement and enters into an uncovered area. This requester vehicle is named as *target vehicle*. The target vehicle present in the uncovered area cannot get direct access to the edge servers. In this case, the passing by vehicles in the RSU region can act as store-carry-forward relays for data forwarding to the target vehicle. However, the arrival of vehicles follow Poisson process with a rate λ in both forward and backward directions. The vehicle speeds are assumed to be constant and are uniformly distributed in the range $[v_{min}, v_{max}]$. Furthermore, each vehicle v maintains a cooperative cache B_v to store the cached contents that are realized for the target vehicle.

Form Fig.4.1, if a target vehicle leaves RSU A before it receives a response data from the edge server, then the RSU A schedules Infrastructure to Vehicle (I2V) communication to relay vehicles (in forward direction) for serving the target vehicle in the uncovered area. For example, the relay vehicle v_2 is in forward direction. After that, if the target vehicle has any unserved data requirement, then RSU A sends that unserved data to next RSU (i.e., RSU B) through I2I communication. Then, the RSU B also schedule I2V communication to relay vehicles (in backward direction) for serving the target vehicle. For example, the relay vehicle v_3 is in backward direction. That means, the v_3 had been in RSU B coverage when the target vehicle leaves RSU A's coverage. These relay vehicles (in both directions) transfer their cached contents to the target vehicle using V2V forwarding, when they enter into target vehicle's radio range. The vehicles are assumed to be equipped with dual-radio OBUs[36] in which one radio tuned to a service channel (SCH) where the basic service messages (e.g, collision warning, safety messages, etc.,) are received. The second radio switches over control channel (CCH) and another service channel (SCH) for V2V data forwarding [128]. When a relay vehicle enters into the target vehicle radio range, they both tuned to a particular SCH for V2V forwarding in the uncovered area. Due to hardware limitations, the target vehicle cannot receive data from multiple relay vehicles simultaneously. Therefore, the target vehicle is designated to schedule the V2V forwarding from the in-range relay vehicles to improve the residual data delivery in the uncovered area. Note that the vehicles are not energy limited and they are assumed to have sufficient energy reserves.

The important notations used in this system model are summarized in Table 4.1.

4.1.1 Communication model

Variable bit rate transmission is assumed in downlink (I2V) channel to compensate the changes in channel path-loss, therefore RSU maintains fixed transmit power[23] during I2V communication. That means, a vehicle nearest to RSU receives more data compared to farthest vehicle when the RSU spends constant power regardless of the downlink (RSU-to-vehicle) distance. Besides, the transmission channel time is divided into equal durations, referred to as *time slots*. The RSU consists of a single radio transceiver that can communicate to only one vehicle v in a time slot t. Moreover, the wireless channel between RSU and the vehicles assumes flat fading [129]. Let $s_{v,t}$ be the small scale fading power gain of the downlink channel. $h_{v,t}$ is denoted as the channel gain from RSU to a vehicle v at a time slot t and it can be defined as follows,

$$h_{v,t} = s_{v,t} \cdot g_0 \cdot (\frac{d_0}{d_{v,t}})^{\alpha} \tag{4.1}$$

where g_0 is path-loss constant, d_0 is reference distance, $d_{v,t}$ is the downlink distance from the RSU to a vehicle v at a time slot t and α is path-loss exponent. The achieved bit-rate $b_{v,t}$ during downlink communication from RSU to vehicle v at time slot t is defined as follows[130],

$$b_{v,t} = B \cdot \log_2(1 + \frac{P.h_{v,t}}{B.N_0}) \tag{4.2}$$

where P is the fixed transmit power of the RSU, B is the bandwidth allotted to downlink channel, and N_0 is the Additive White Gaussian Noise at the vehicle.

Table 4.1: Important notations in system model

Notation	Description
U	Uncovered area or outage area.
V_0	Target vehicle in the outage area.
H	Residual data required by the target vehicle.
T	Set of time slots.
δt	Duration of a time slot t, where $t \in T$.
V_r	Set of forward relay vehicles.
V_r'	Set of backward relay vehicles.
R	Radio range of an RSU.
R_v	Radio range of a vehicle v.
B_v	Cooperative cache of a vehicle v.
$\Delta C_{v,t}$	Time to contact V_0 if v is selected at t.
$b_{v,t}$	Downlink bit-rate when v is selected at t.

4.1.2 RSU energy consumption

The RSU energy consumption depends on the downlink distance from the RSU to a vehicle in the RSU coverage[64]. The downlink communication to farthest vehicles consumes more RSU power when compared to nearby vehicles[18]. For example, from Fig.4.1, using variable bit-rate case discussed in section 4.1.1, if the RSU A transmits k bits to nearby vehicle v_1 in one time slot t_1, then the same RSU can transmit less than k bits to a farthest vehicle v_2 in another time slot t_2. To receive k bits, the farthest vehicle need to spend more than one time slot in downlink communication, consequently it increases communication time, as well as the RSU power consumption. Similarly, nearby vehicle requires less communication time and it decreases the RSU power consumption.

As shown in Fig.4.1, when a target vehicle leaves the RSU A coverage, a passing by vehicle v_2 is in the radio range of the target vehicle and v_1 is not in the target vehicle's radio range. Therefore, v_1 may take some time to contact (i.e., *Time-to-Contact*) the target vehicle. The downlink communication to nearby vehicle v_1 decreases RSU energy consumption, but it increases the time to contact the target vehicle, thereby overall end-to-end delay to the target vehicle increases. The end-to-end delay is defined as the difference between the start time of sending request to the edge server and end time of the total response data received at the target vehicle. Note that the end-to-end delay is directly proportional to *Time-to-Contact* the target vehicle (refer section 4.1.3). In a similar way, the downlink communication to a farthest vehicle v_2 increases RSU energy consumption, but it decreases the time required to contact the target vehicle, thereby overall end-to-end delay decreases. Therefore, the objective of this chapter is to minimize the total *RSU time spent in down-link communication* (proportional to RSU power consumption) and the *Time-to-Contact* (proportional to end-to-end delay) while improving data delivery to the target vehicle by scheduling the suitable relay vehicles.

4.1.3 End-to-end delay

Let a target vehicle V_0 with a velocity v_0 enters to one end of the RSU A coverage (to the reference point O shown in Fig.4.1) at time t_0. After some time Δt_1, the V_0 sends a request to the edge server placed with the RSU A. The end-to-end delay of the request includes computation delay (Δt_2) in the edge server and waiting time (Δt_3) in the queuing buffer. Here, up-link communication delay is assumed as negligible. If F is the clock frequency of an edge server and Z is the workload of the request, then the computation delay is defined as

$$\Delta t_2 = Z/F \tag{4.3}$$

In this system, the edge server has been modeled as M/M/1 queuing server. The arrived requests at the queuing buffer wait for execution. When the requests are generated as a Poission stream at a rate $\mathcal{A}$ and the service rate of an edge server is μ, then the average waiting time is defined as[131],

$$\Delta t_3 = \frac{\mathcal{A}}{\mu(\mu - \mathcal{A})} \tag{4.4}$$

The response data is ready to be served using downlink communication from RSU A to V_0, at time t_r.

$$t_r = t_0 + \Delta t_1 + \Delta t_2 + \Delta t_3 \tag{4.5}$$

Meanwhile, the V_0 may leave the other end of the RSU A coverage at time $t_k = t_0 +$

$2R/v_0$. If $t_r \leq t_k$, then V_0 is served inside the RSU A coverage. Otherwise, V_0 enters into the uncovered area and then the data delivery to V_0 is delayed. Therefore, the RSU A initiates selection of suitable relay vehicles which can deliver the response data to the target vehicle. This work assumes that the response data is ready when the target leaves RSU region and the response data will be served by multiple relay vehicles to the target V_0.

4.1.4 Forward relay selection

In this section, a set of relay vehicles has been derived for store-carry-forward the response data to the target vehicle in the uncovered area. By applying trivial relative motion theory [71], the RSU A selects relay vehicles that are moving in the same direction as the target vehicle. This forward relay selection is based on the following factors.

- *Dwell time (ΔS_x)* : the amount of time a vehicle V_x spent in the RSU A coverage.
- *Link time (ΔL_x)* : the amount of time a vehicle V_x is in the radio range of target vehicle.
- *Time-to-Contact (ΔC_x)* : the amount of time it takes a vehicle V_x to reach the target vehicle.

Let a vehicle V_x arrives (to the reference point O in Fig.4.1) with a velocity v_x at time t_x. As aforementioned, the target V_0 arrives to O at time t_0 with velocity v_0 and leaves RSU A region at time $t_k = t_0 + 2R/v_0$. However, at time t_0, the distance between vehicle V_x and the reference point O is

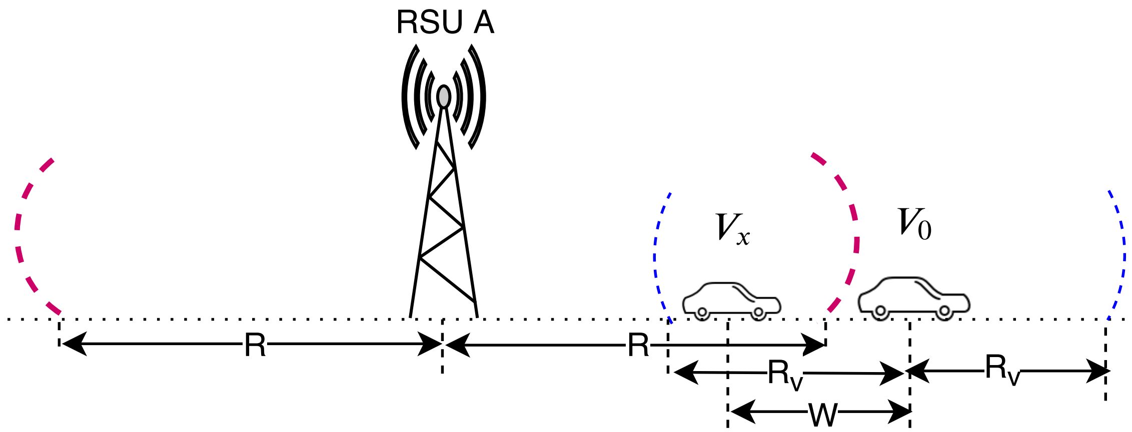
$$d_x(t_0) = (t_0 - t_x)v_x \tag{4.6}$$

When V_0 leaves RSU A at time t_k, the distance between V_x and O is

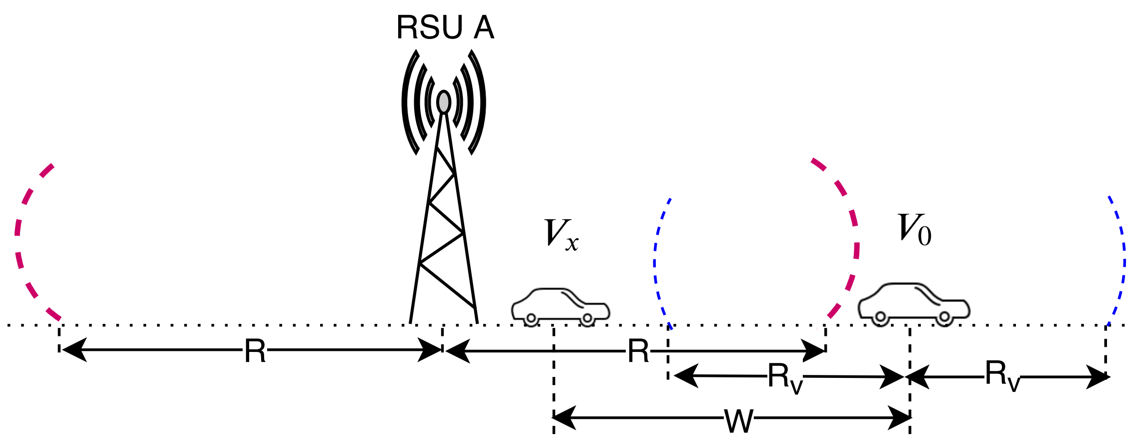
$$d_x(t_k) = d_x(t_0) + (2R/v_0)v_x \tag{4.7}$$

Then, the distance between V_0 and V_x is,

$$W = |2R - d_x(t_k)| \tag{4.8}$$



(a) Case I: $W \leq R_v$



(b) Case II: $W > R_v$

Figure 4.2: Relative positions of a vehicle V_x and target V_0

- If $d_{ox}(t_k) > 2R$, then V_x is ahead of V_0. In this case, V_x can not act as a relay vehicle.
- If $d_{ox}(t_k) \leq 2R$, then V_x is behind V_0. Then, V_x may act as relay to the target vehicle.

Two different cases illustrate the relative positions of V_0 and V_x as shown in Fig.4.2. The *Sojourn time* (ΔS_x), *Link time* (ΔL_x) and *Time-to-Contact* (ΔC_x) are defined as follows,

$$\Delta S_x = min\Big\{2R/v_x, W/v_x\Big\} \tag{4.9}$$

  - Case I: Fig.4.2a shows $W \leq R_v$, here ΔC_x is zero and then

$$\Delta L_x = \begin{cases} \frac{R_v + W}{v_x - v_0}, & if \quad v_x > v_0 \\ \frac{R_v - W}{v_0 - v_x}, & if \quad v_x \leq v_0 \end{cases} \tag{4.10}$$

– Case II: Fig.4.2b shows $W > R_v$, then

$$\Delta C_x = \begin{cases} \frac{W - R_v}{v_x - v_0}, & if \quad v_x > v_0 \\ \infty, & if \quad v_x \leq v_0 \end{cases} \tag{4.11}$$

After ΔC_x time, both V_x and V_0 can establish communication link with each other.

$$\Delta L_x = \begin{cases} \frac{2R_v}{v_x - v_0}, & if \quad v_x > v_0 \\ 0, & if \quad v_x \leq v_0 \end{cases} \tag{4.12}$$

However, the maximum amount of time a vehicle V_x is selected for downlink communication as a relay vehicle is referred to as *effective communication time*(ΔE_x). It has been defined as the minimum of *Link time* (ΔL_x) and *Sojourn time* (ΔS_x), i.e., $\Delta E_x = min(\Delta L_x, \Delta S_x)$.

A set of forward relay vehicles V_r and the set of time slots in the system is defined as follows,

$$V_r = \{V_x \in V | \, min(\Delta L_x, \Delta S_x) \geq \delta t \wedge B_x > 0\} \tag{4.13}$$

$$T = \{1, 2, 3, ..., \tau\}, \tau = U/(v_0 \, \delta t) \tag{4.14}$$

where δt is the duration of a time slot, and V denotes the set of all vehicles in the RSU A region.

Note that, if RSU A selects V_x as relay vehicle at the beginning of a time slot $t \in T$, then the *Time-to-Contact* the target V_0 is $\Delta C_{x,t} = \Delta C_x - (t-1)\delta t$. But, the effective communication time of V_x remains unchanged with the time slot at which the relay vehicle is selected.

4.1.5 Problem formulation

As aforementioned, proper scheduling of relay vehicles affects RSU energy consumption and the end-to-end delay to the target vehicle. Therefore, the objective of this chapter is to minimize the total RSU time spent in downlink communication and the time to contact the target vehicle that is by scheduling the suitable relay in each time slot. The input function to this problem contains a set of time slots T, set of forward relay vehicles V_r, residual data H, cooperative cache size B_v, and effective communication time ΔE_v. Here, a binary decision variable is used to represent the scheduling decision for the relay vehicle v at the beginning of a time slot t.

$$x_{v,t} = \begin{cases} 1, & \text{If RSU A chooses a } v^{th} \text{ relay vehicle at } t^{th} \text{ time slot} \\ 0, & \text{Otherwise} \end{cases} \tag{4.15}$$

In addition, the problem is formulated as follows.

$$\underset{x_{v,t}}{\text{minimize}} \sum_{v \in V_r} \sum_{t \in T} x_{v,t} \left(\Delta C_{v,t} / b_{v,t} \right) \tag{$\mathcal{P}$}$$

$$\text{subject to} \sum_{v \in V_r} \sum_{t \in T} b_{v,t} \, x_{v,t} \geq H \tag{4.16}$$

$$\sum_{t \in T} b_{v,t} \, x_{v,t} \leq B_v \quad \forall v \in V_r \tag{4.17}$$

$$\delta t \sum_{t \in T} x_{v,t} \leq \Delta E_v \quad \forall v \in V_r \tag{4.18}$$

$$\sum_{v \in V_r} x_{v,t} \leq 1 \quad \forall t \in T \tag{4.19}$$

$$x_{v,t} \in \{0, 1\} \quad \forall v \in V_r, \forall t \in T \tag{4.20}$$

The objective function of $\mathcal{P}$ represents a value that indicates the downlink time spent by RSU A and the average Time-to-Contact the target vehicle. Note that, the downlink communication time and Time-to-Contact are directly proportional to the RSU energy consumption and end-to-end delay to the target vehicle. Constraint (4.16) satisfies the residual data required by the target vehicle. Constraint (4.17) ensures the total data received from RSU A to a relay does not exceed cooperative cache size of the relay vehicle. Constraint (4.18) ensures the total downlink time used for a relay vehicle is to be less than its effective communication time. Constrains (4.19) and (4.20) allow only one relay vehicle for downlink communication in a time slot.

Theorem 4.1.1. *The proposed problem $\mathcal{P}$ is NP-hard.*

Proof. The proposed problem is proved as NP-hard by reducing it from Generalized Assignment Problem (GAP)[122] to the special case of the problem $\mathcal{P}$. Given an instance of GAP, it has a set of items I and knapsacks K. Each knapsack $j \in K$ has a capacity C_j. Assigning an item $i \in I$ to a knapsack $j \in K$ corresponds to profit p_{ij} and weight w_{ij}. The objective of GAP is to maximize the total profit by assigning each item to exactly one knapsack such that total weight of items does not exceed the knapsack capacity.

Consider a special case of the problem $\mathcal{P}$ that is without constraints (4.16) and (4.18). The reduction from GAP to the special case of $\mathcal{P}$ is as follows: (i) items are mapped to time slots. (ii) knapsacks are mapped to relay vehicles. (iii) knapsack capacity mapped to buffer size of relay vehicle. (iv) profit of each item mapped to negative sum of Time-to-Contact and time slot duration. (v) weight of each item mapped to bit-rate (from RSU to relay vehicle). It is clear that the special case of the problem can be reduced from GAP in polynomial time. Therefore, it concludes that the problem $\mathcal{P}$ is NP-hard. □

4.2 Forward relay scheduling algorithms

In this section, two polynomial time algorithms are presented for scheduling of suitable relay vehicles and this includes a greedy algorithm and Auction-based relay scheduling algorithm. These algorithms minimize both energy consumption of RSUs and end-to-end delay to the target vehicle in the uncovered area.

4.2.1 Greedy algorithm

This section describes a greedy algorithm (GA) to minimize the total RSU time spent in downlink communication and time required to contact the target vehicle. The details of the algorithm GA is as follows.

Step 1 Find a subset of relay vehicles which are near to RSU A and they must satisfy the constraint (4.18)

Step 2 Choose a relay vehicle v from the subset which has least Time-to-Contact and it satisfies the constraint (4.17)

Step 3 Schedule the relay vehicle v selected at Step 2.

Step 4 Update effective communication time (ΔE_v) and cooperative cache (B_v) of the relay vehicle v.

Step 5 Repeat Step 1 to Step 4, for all the time slots.

The algorithm 4.1 illustrates the above steps in-detail. Note that the calculation of effective communication time and Time-to-Contact are discussed in section 4.1.4. From algorithm 4.1, it runs in polynomial time with respect to set of time slots T and set of forward relay vehicles V_r.

4.2.2 Auction-based forward relay scheduling

This section presents an optimal assignment of relay vehicles to time slots by applying the concepts from *Auction Theory*[20]. Here, the RSU A acts as a seller and it auctions the time slots for downlink communication. The relay vehicles act as bidders and they participate in the auction. This problem is modeled as an asymmetric assignment problem where the time slots are more than the relays. It is observed that there exist non-empty set of time slots T, relay vehicles V_r, and these are finite. Before selecting a suitable relay vehicle $V_i \in V_r$ at a time slot $T_j \in T$, the RSU estimates utility $\mathcal{U}_{ij}$ based on the Time-to-Contact (ΔC_{ij}) and the bit-rate (b_{ij}) in the downlink channel. It is already mentioned that these two

Algorithm 4.1 GA: Greedy algorithm

Input: T, V_r
Output: Schedule X

for $t = 1$ to $|T|$ **do**
    for $v = 1$ to $|V_r|$ **do**
        $X[v][t] = 0$
    end for
end for
for $t = 1$ to $|T|$ **do**
    $d_{min} = R$
    for $r = 1$ to $|V_r|$ **do**
        $S = \varnothing$
        $d_{r,t}$ = Distance from RSU A to relay vehicle r
        if $d_{r,t} \leq d_{min}$ and $\Delta E_r \geq \delta t$ **then**
            $d_{min} = d_{r,t}$
            $S = S \cup \{r\}$
        end if
    end for
    $C_{min} = +\infty$
    for r in S **do**
        $\Delta C_{r,t}$ = Time to contact the target vehicle
        if $\Delta C_{r,t} < C_{min}$ and $B_r \geq b_{r,t}$ **then**
            $C_{min} = \Delta C_{r,t}$
            $v = r$
        end if
    end for
    $X[v][t] = 1$
    $\Delta E_v = \Delta E_v - \delta t$
    $B_v = B_v - b_{v,t}$
end for
return X

parameters have direct impact on the end-to-end delay and the RSU power consumption, respectively. The mapping of time slot – relay vehicle pair is determined by the difference between the minimum utility and the second smallest utility, known as *bidding increment*.

Let Q be a set of pairs (V_i, T_j), where a time slot $T_j \in T$ can be mapped to a relay vehicle $V_i \in V_r$. For each relay vehicle V_i, there are set of time slots $A(V_i)$ in which the V_i is present inside the RSU region and is expressed as

$$A(V_i) = \{T_j | (V_i, T_j) \in Q\} \tag{4.21}$$

Similarly, a time slot T_j can be mapped to a relay vehicle among the subset of relay vehicles $B(T_j)$ which are present inside the RSU region during that time slot and is denoted as

$$B(T_j) = \{V_i | (V_i, T_j) \in Q\} \tag{4.22}$$

4.2.2.1 Bidding phase

In this bidding phase, a bidder V_i (i.e. relay vehicle) bids for the subset of time slots $A(V_i)$. Each bidder estimates the bidding value based on the utility and it returns a bidding vector $B_T^i = \{B_1, B_2, ..., B_{|A(V_i)|}\}$. Note that the bid value for the remaining time slots in the set $T \setminus A(V_i)$ are taken as largest positive value, since the assignment problem has been modeled as a minimization problem. Here, each relay vehicle finds a time slot, and a relay vehicle is selected in each time slot which has minimum utility in the downlink communication.

$$T_{ji} \in arg \min_{T_j \in T} \{\mathcal{U}_{ij}\} \tag{4.23}$$

$$V_{ij} \in arg \min_{V_i \in V_r} \{\mathcal{U}_{ij}\} \tag{4.24}$$

where T_{ji} denotes a time slot T_j for V_i and it provides minimum utility by selecting that time slot. On the other hand, V_{ij} denotes a relay vehicle V_i for T_j and it provides minimum utility by selecting that relay vehicle. Then, the bidding increment is calculated as,

$$h_i = e_i - f_i, \forall V_i \in V_r \tag{4.25}$$

$$h_j = e_j - f_j, \forall T_j \in T \tag{4.26}$$

where f_i and f_j are the smallest utilities for the relay vehicle and time slot, respectively.

$$f_i = min_{T_j \in A(V_i)} \{\mathcal{U}_{ij}\} \tag{4.27}$$

$$f_j = min_{V_i \in B(T_j)}\{\mathcal{U}_{ij}\} \tag{4.28}$$

and e_i and e_j are the second smallest utilities for the relay vehicle and time slot, respectively

$$e_i = min_{T_j \in A(V_i), T_j \neq T_{ji}}\{\mathcal{U}_{ij}\} \tag{4.29}$$

$$e_j = min_{V_i \in B(T_j), V_i \neq V_{ij}}\{\mathcal{U}_{ij}\} \tag{4.30}$$

4.2.2.2 Allocation phase

The RSU A assigns a time slot T_j to a relay vehicle V_i, for which the bidding increment is maximum. The highest bidding increment is calculated as follows,

$$G_{ij} = max_{V_i \in V_r, T_j \in T}(h_i, h_j) \tag{4.31}$$

Then, the RSU excludes an assigned time slot T_j from the set T. Moreover, it excludes the relay vehicle V_i from the set V_r, if the vehicle's remaining buffer size is zero or the remaining effective communication time is less than the time slot duration (δt). Algorithm 4.2 illustrates the Auctioning process for the energy efficient relay scheduling. Next, the Auction procedure is explained with the following example.

Table 4.2: Biddings from relay vehicles

	T_1	T_2	T_3
V_a	7	6	8
V_b	9	4	7
V_c	6	10	2

Table 4.3: Bidding increment in Bidding phase

	T_1 (h_1)	T_2 (h_2)	T_3 (h_3)
V_a (h_a)	(1,1)	(1,2)	(1,5)
V_b (h_b)	(3,1)	(3,2)	(3,5)
V_c (h_c)	(4,1)	(4,2)	**(4,5)**

Algorithm 4.2 Forward Relay Scheduling (FRS)

Input: T : Set of time slots. V_r : Set of relay vehicles.
Output: X : Set of mapping pairs of time slots and relay vehicles.

Auction the time slot $T_j, \forall\, T_j \in T$.
Receive biddings from the relay vehicles $\forall\, V_i \in V_r$.
while $|T| \neq 0 \wedge |V_r| \neq 0$ **do**
  for $\forall V_i \in V_r$ **do**
    compute e_i from equation (4.29)
    compute f_i from equation (4.27)
    compute bidding increment $h_i = e_i - f_i$
  end for
  for $\forall T_j \in T$ **do**
    compute e_j from equation (4.30)
    compute f_j from equation (4.28)
    compute bidding increment $h_j = e_j - f_j$
  end for
  Find a time slot $t \in T$ and a relay vehicle $v \in V_r$ from equation (4.31)
  Map the time slot t to a relay vehicle v based on their highest bidding increment values $h_j^{(max)}$ and $h_i^{(max)}$, respectively
  $X = X \cup (v, t)$
  $\Delta E_v = \Delta E_v - \delta t$
  $B_v = B_v - b_{v,t}$
  Remove t from T, $T = T \setminus t$
  if $(B_v \leq 0) \vee (\Delta E_v \leq \delta t)$ **then**
    Remove v from V_r, $V_r = V_r \setminus v$
  end if
end while
return X

Example: Lets assume the RSU channel time is divided into three time slots T_1, T_2, and T_3. Three relay vehicles V_a, V_b, and V_c bid for those time slots as shown in table 4.2. In bidding phase, first it computes $h_i = e_i - f_i, \forall V_i \in V_r$ and $h_j = e_j - f_j, \forall T_j \in T$ from Eq.(4.25) and Eq.(4.26), respectively. Table 4.3 summarizes the computation. Here, for a relay vehicle V_c, h_c will be 4 (i.e., $h_c = 6 - 2 = 4$). Similarly, for a time slot T_3, h_3 will be 5 (i.e., $h_3 = 7 - 2 = 5$. After that, in allocation phase, the RSU maps the time slot T_3 to a relay vehicle V_c by using Eq.(4.31) as follows,

$G_{ij} = max_{V_i \in V_r, T_j \in T}((h_a, h_b, h_c)(h_1, h_2, h_3))$

$G_{ij} = max_{V_i \in V_r, T_j \in T}((1, 3, 4)(1, 2, 5))$

$G_{c3} = (h_c = 4, h_3 = 5)$

Then, the RSU A excludes the assigned time slot and the relay vehicle as aforementioned. After that, the RSU recomputes bidding phase, and it maps the time slots T_1, T_2 to relay vehicles V_b and V_a, respectively. At the end, mappings from the time slots to relay vehicles are as follows: (V_a, T_2), (V_b, T_1), and (V_c, T_3).

4.3 RSU-assisted relay scheduling

When the RSU A completes relay scheduling in forward direction, it initiates I2I communication to next RSU B which is present in the moving direction of the target vehicle. Then, the RSU B obtains a set of parameters $\{t_0, v_0, H'\}$ from RSU A via back-haul wired link. Here, t_0 is the arrival time of V_0 at the reference point O, v_0 is the velocity of V_0 and H' is the residual data after performing forward relay scheduling as represented in eq.4.32.

$$H' = H - \sum_{v \in V_r} \sum_{t \in T} b_{v,t}\, x_{v,t} \tag{4.32}$$

After that the RSU B initiates relay selection in backward (opposite to target) direction and schedule these relays in order to further minimize the energy consumption cost of the RSUs and end-to-end delay to the target vehicle. The RSU B also follows variable bit-rate transmission during the downlink communication to the relay vehicles.

4.3.1 Backward relay selection

Set of all vehicles in RSU B coverage region is denoted as V', when the target vehicle V_0 leaves RSU A at time $t_k = t_0 + 2R/v_0$. Suppose a vehicle $V_y \in V'$ with a velocity v_y enters into RSU B at a time t_y. Then, at time t_0, the distance between the reference point O and

V_y can be derived as,

$$d_y(t_0) = \begin{cases} U + 4R + (t_y - t_0)v_y, & if \quad t_y > t_0 \\ U + 4R - (t_0 - t_y)v_y, & if \quad t_0 \leq t_y \end{cases} \tag{4.33}$$

When V_0 leaves RSU A at time $t_k = t_0 + 2R/v_0$, the distance between O and V_y is,

$$d_y(t_k) = d_y(t_0) - (2R/v_0)v_y \tag{4.34}$$

and the distance between V_0 and V_y is,

$$W' = d_y(t_k) - 2R \tag{4.35}$$

The time (i.e., Time-to-Contact) it takes a vehicle V_y to establish a communication link with V_0 is,

$$\Delta C_y = \frac{W' - R_v}{v_0 + v_y} \tag{4.36}$$

Recall that the set of time slots is T and the remaining time to contact the target when the relay V_y is selected at time $t \in T$ is $\Delta C_{y,t} = \Delta C_y - (t-1)\delta t$.

The amount of time (i.e., link time) a vehicle V_y is in the radio range of V_0 is,

$$\Delta L_y = \begin{cases} \frac{2R_v}{v_0+v_y}, & if \quad \Delta C_{y,t} > 0 \\ \frac{W'+R_v}{v_0+v_y}, & if \quad \Delta C_{y,t} = 0 \end{cases} \tag{4.37}$$

The remaining time (i.e., sojourn time) a vehicle V_y can travel in the RSU B coverage region is derived as,

$$\Delta S_y = \frac{W' - U}{v_y} \tag{4.38}$$

Therefore, the set of relay vehicles V_r' in RSU B (i.e., backward direction) is defined as,

$$V_r' = \{V_y \in V' |\, min(\Delta L_y, \Delta S_y) \geq \delta t \wedge B_y > 0\} \tag{4.39}$$

4.3.2 Auction-based RSU-assisted relay scheduling

Upon identifying the set of relay vehicles V_r', the RSU B finds an energy efficient schedule for these vehicles in T time slots. By following the principles of Auction Theory[19], the RSU B assists RSU A to forward the residual data H' to the target vehicle while satisfying the constraints of energy consumption and end-to-end delay. Similar to the auction process described in section 4.2.2, the RSU B also acts as a seller and it auctions the time slots. The relay vehicles (in backward direction) act as bidders and participate in the auction process. Algorithm 4.3 provides auction based RSU assisted relay scheduling in a bidirectional highway by utilizing the cooperation between the RSUs. Although these relay vehicles (either in forward or backward direction) can only spend limited time in the target's coverage, the proper V2V forwarding can maximize data delivery for the target vehicle. However, due to hardware limitations of OBUs, the V2V forwarding from the relay vehicles cannot be performed simultaneously. In the case of simultaneous data transmissions in the same service channel may lead to data collisions at the target vehicle and it diminishes residual data delivery.

4.3.3 Vehicle to Vehicle (V2V) scheduling

As a solution, the target vehicle schedules V2V forwarding for its in-range relay vehicles in the uncovered area. As aforementioned, the vehicles are installed with dual-radio OBUs and one of its radio continuously tuned on to a service channel for receiving the basic service messages (BSMs) from their neighbor vehicles. Where as, the second radio switches between control channel and another service channel. In the uncovered area, all the relay vehicles act as service users and broadcast their BSMs over the service channel (e.g, SCH1), which contains vehicle identity, location, speed, direction and carrying data size. The target vehicle acts as a service provider and it broadcasts Wave Service Advertisement (WSA) message over control channel (CCH), which contains provider identity, a set of in-range vehicles (neighbors), set of offered services, service channel (e.g, SCH2) where the services are offered, user identity, etc. Based on the received BSMs in every time slot, the target vehicle updates neighbor set and computes link time, distance to its neighboring relay vehicles. Then, the target vehicle schedules a neighbor relay vehicle which has more carrying data and less link time. After that, the service provider (target vehicle) broadcasts the WSA containing the user (selected neighbor) identity and a service channel (e.g., SCH2) where the V2V forwarding can be performed. This V2V scheduling of relay vehicles in the uncovered area improves the overall performance of the system and maximizes data delivered to the target vehicle.

Algorithm 4.3 RSU-assisted Relay Scheduling (RRS)

Input: T: Set of time slots, V_r': Set of backward relay vehicles, V_r : Set of forward relay vehicles

Output: X: Set of mappings of time slots to relay vehicles

Perform forward relay scheduling using Algorithm 4.2
Compute remaining residual data H' from Eq.4.32
Send $\{t_0, v_0, H'\}$ to RSU B
RSU B auctions the time slots
RSU B receive biddings from the relay vehicles in backward direction
while $|T| \neq 0 \wedge |V_r'| \neq 0$ **do**
    if $H' == NULL$ **then**
        return X
    else
        Compute the bidding increment for each relay vehicle
        Compute the bidding increment for each time slot
        Find a suitable time slot t and relay vehicle v pair based on their highest bidding increment values
        Append the mapped pair (v, t) to set X
        Update effective communication time and cooperative cache size of the relay vehicle v
        Remove time slot t from set T
        Remove relay v from set V_r' either if its cooperative cache is full or effective communication time is less than the time slot duration
        Update residual data $H' = H' - b_{v,t}$
    end if
end while
return X

4.4 Simulation results

In this section, the performance of proposed scheduling algorithms are evaluated by leveraging the Monte Carlo simulations [25]. The input data of vehicular traces to the scheduling algorithms is considered from a bi-directional highway road segment. In the highway scenario, the vehicles maintain constant speed [9] and the speeds are uniformly distributed in a range [15m/s,25m/s]. The arrival of vehicles into the highway segment follows Poisson process with a rate λ vehicles per time slot in both directions (forward and backward). The road segment may have multiple lanes and the vehicles moving in same direction can overtake other vehicles without varying their speeds. For communication model, a distance dependent based path-loss model is used in the simulation with a path-loss exponent $\alpha = 3$. This model is obtained from [24] and it assumes the downlink channel bit rates vary based on the location of vehicle in the RSU region from 3 to 27 Mb/s. These bit rates for the downlink communication can be derived from the RSU to vehicle distance. In this simulation, the transmit power of RSU is set to 1W, channel bandwidth is 10MHz, Additive White Gaussian Noise at the vehicle is -174dB/Hz and reference distance $d_0 = 1m$. In addition, the radio range of vehicles, RSUs and the uncovered area between the RSUs are assumed 100m, 1Km and 5Km, respectively.

For evaluating the proposed algorithms, a naive algorithm called First Come First Serve (FCFS) has been considered for comparing with the proposed algorithms. In each time slot, the FCFS schedules a vehicle with least arrival time among the vehicles in the RSU region. Further, the proposed greedy algorithm (GA) is a variation of the known benchmark called Nearest Fastest Set (NFS) scheduler [6] and it includes additional parameters cooperative cache size and effective communication time of the relay vehicles. Besides, this section evaluates the performance of the proposed algorithms in terms of average RSU energy consumption, average end-to-end delay and residual data delivery for the vehicle arrival rate, speed of target vehicle and co-operative cache size.

4.4.1 Impact of vehicle arrival rate (λ)

Fig. 4.3 shows the performance comparison of relay scheduling algorithms RRS, FRS, GA and FCFS in terms of average energy consumption of RSU for different vehicle arrival rates. The average arrival rate λ indicates the number of vehicles enter into the bi-directional highway segment per time slot. From Fig. 4.3, the RRS scheduler outperforms

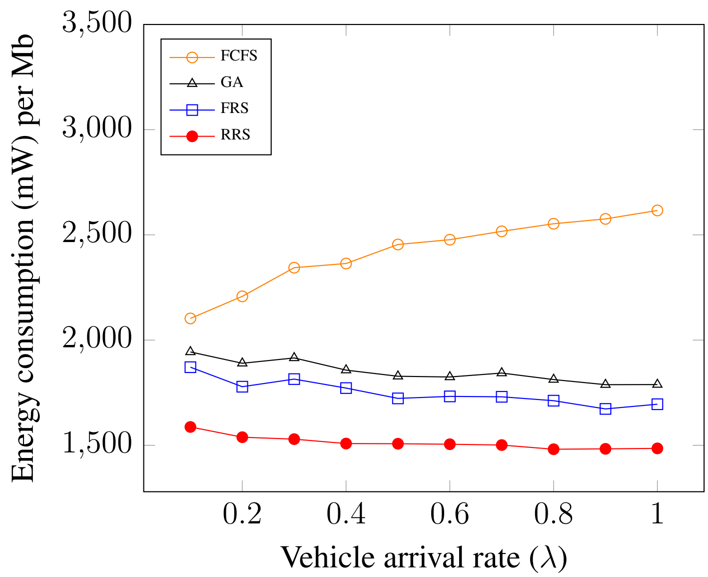


Figure 4.3: RSU energy consumption and Vehicle arrival rate

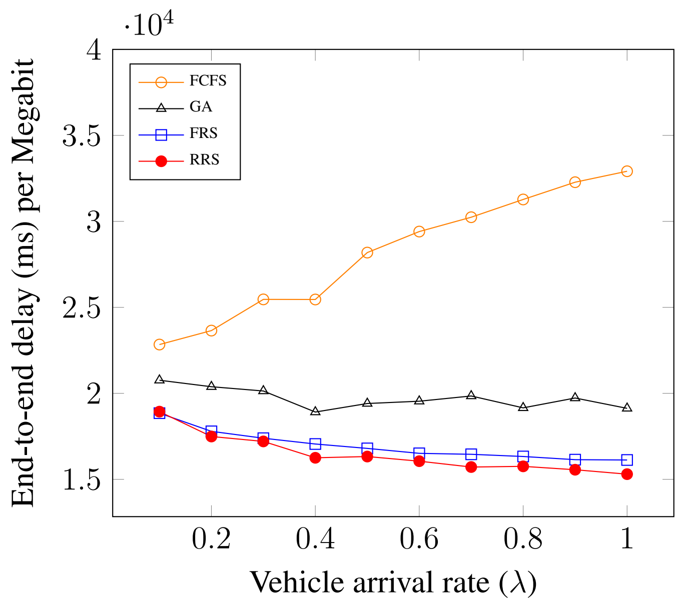


Figure 4.4: End-to-end delay and Vehicle arrival rate

other algorithms FRS, GA and FCFS. The average energy consumption of schedulers RRS, FRS, GA except FCFS decreases with the increase of vehicle arrival rate. This is because the RRS, FRS and GA selects suitable relay vehicles that are more close to RSU, as the number of vehicles in the RSU coverage increases. This leads to higher bit rate transmissions and then the RSU energy consumption per Megabit delivery to the target vehicle decreases. Besides, the FCFS tends to select the relay vehicles that are available more close

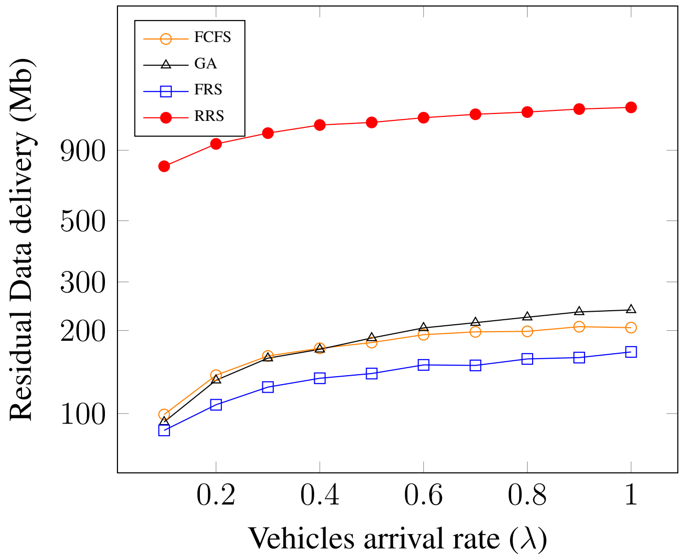


Figure 4.5: Residual data delivery and Vehicle arrival rate

to the end of RSU coverage, as the number of vehicles increases in the RSU coverage. This leads to lower bit rate transmissions and then the RSU energy consumption per Megabit increases. For example, for $\lambda = 0.5$ with constant target speed 20 m/s and unlimited cooperative cache size, the average RSU energy consumption by the schedulers RRS, FRS, GA and FCFS are 1.5×10^3 mW ,1.7×10^3 mW, 1.8×10^3 mW and 2.4×10^3 mW, respectively. In other words, the average energy consumption for the algorithms RRS, FRS and GA decreases by 60.17%, 38.35% and 30.93% when compared to FCFS algorithm.

Fig.4.4 shows the performance comparison of schedulers in terms of average end-to-end delay to the target vehicle for the vehicle arrival rate. As shown in Fig.4.4, the GA, FRS, RRS outperform FCFS, especially the RRS outperforms other schedulers. This is because GA, FRS and RRS schedule the relay vehicles with higher bit rate transmissions as the vehicle arrival rate increases. Therefore, the average end-to-end delay to the target vehicle for delivering one Megabit data decreases with arrival rate. However, the FCFS schedules the relay vehicles close to the outer edge of RSU coverage that leads to lower bit rate transmissions. Therefore, total amount of residual data delivered to the target vehicle decreases and then the average end-to-end delay increases with the vehicle arrival rate. In addition, the residual data delivery by the RRS is nearly 6.8 times the other algorithms as shown in Fig.4.5. Although the residual data delivery of FRS is less compared to other algorithms, the FRS achieves lower average end-to-end delay with the aid of proper auctioning process. The GA and FCFS deliver nearly same amount of residual data due to their greedy nature, but GA outperforms FCFS. This is because FCFS takes more number of downlink transmissions compared to GA. From Fig.4.4, it is observed that the average end-to-end delay by the RRS is low when the uncovered area in between the RSUs is not more than 5Km. Note that the increased uncovered area degrades the performance of the RRS algorithm. Therefore, this simulation assumes the uncovered area is constant at 5km. For $\lambda = 0.5$, without varying target vehicle speed and co-operative cache size, the average end-to-end delay to the target vehicle by the schedulers RRS, FRS, GA and FCFS are 1.63×10^4 ms, 1.67×10^4 ms, 1.94×10^4 ms and 2.81×10^4 ms, respectively. The performance improvement of RRS, FRS and GA over FCFS are 51.15%, 46.82% and 39.86%, respectively.

4.4.2 Impact of target vehicle speed (m/s)

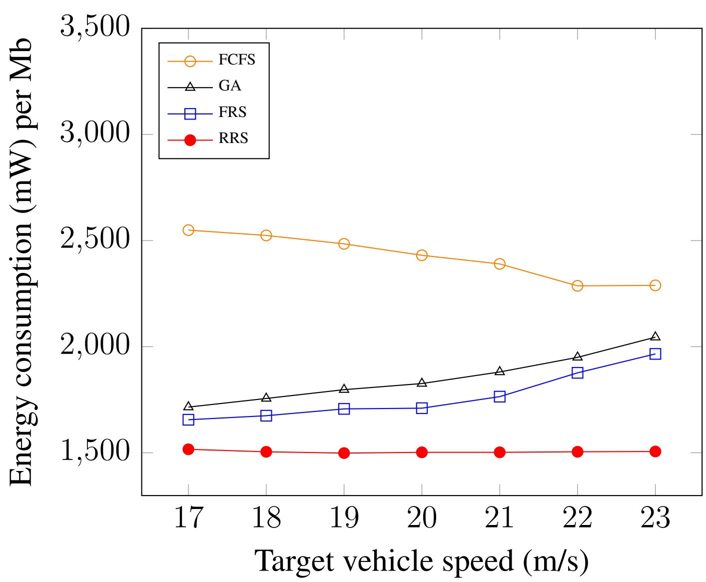


Figure 4.6: RSU energy consumption and target vehicle speed

Fig.4.6 shows the performance comparison among RRS, FRS, GA and FCFS in terms of average RSU energy consumption for the target vehicle speed. As shown in Fig.4.6, the RRS, FRS and GA outperform FCFS. Especially, FCFS reduces average energy con-

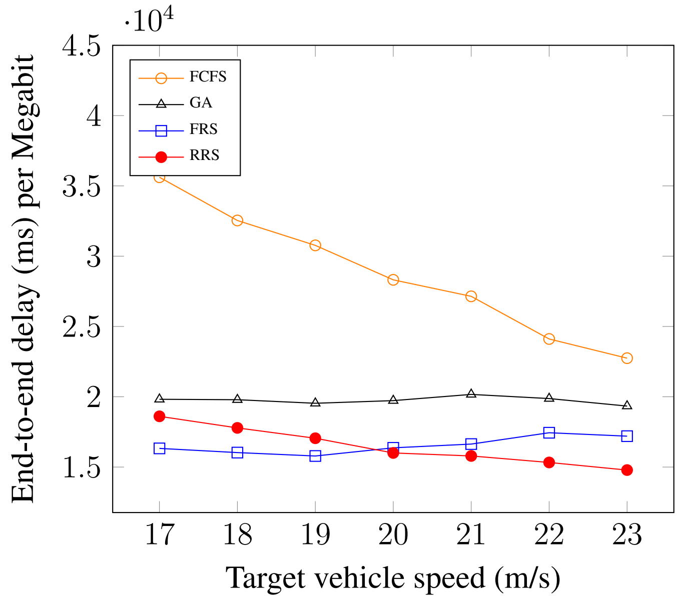


Figure 4.7: End-to-end delay and target vehicle speed

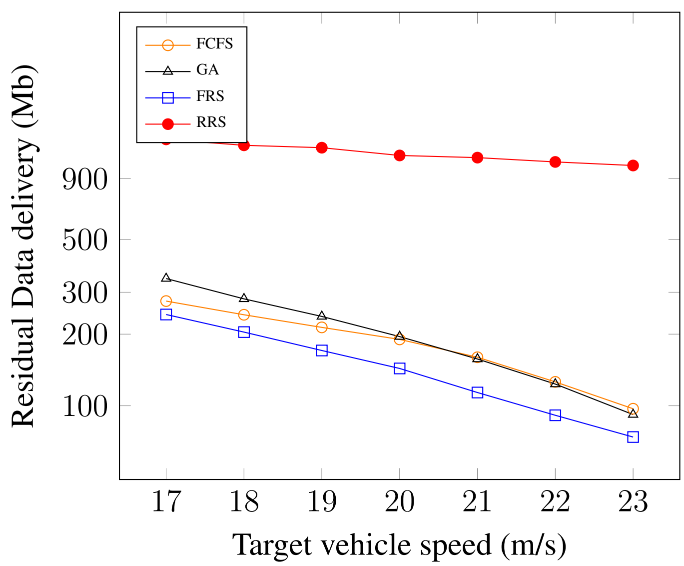


Figure 4.8: Residual data delivery and target vehicle speed

sumption and FRS, GA increases average energy consumption as the target vehicle speed increases. This is because, with increased target vehicle speed, the number of relay vehicles available in the RSU region decreases. As a result, the FCFS tends to select the relay vehicles away from (not close to) the end of RSU coverage, then RSU to relay distance decreases and it improves average energy consumption. In case of GA and FRS, these schedulers tend to select relay vehicles away from (not close to) the RSU with the decreased number of relay vehicles in the RSU coverage. In this comparison, the vehicle arrival rate is constant at $\lambda = 0.5$ with unlimited cooperative cache size. For example, when the target vehicle speed is 20 m/s, the average energy consumption per one Megabit delivery to the target vehicle by the schedulers RRS, FRS, GA and FCFS are 1.5×10^3 mW, 1.7×10^3 mW, 1.8×10^3 mW and 2.4×10^3 mW, respectively. Thus, for this case, the average energy consumption for the algorithms RRS, FRS and GA when compared to FCFS decreases by 60.93%, 37.24% and 30.71% , respectively.

Fig. 4.7 shows the performance comparison among the schedulers RRS, FRS, GA and FCFS in terms of average end-to-end delay for the target vehicle speed. As shown in Fig. 4.7, the RRS and FRS outperform GA and FCFS, especially RRS performs better compared to other schedulers. This is because the time required by relay vehicles (in backward direction) to reach the target vehicle decreases with the increasing target vehicle speed. In addition, residual data delivery by the RRS is nearly 6.3 times when compared to other algorithms as shown in Fig. 4.8. From Fig.4.7, it is observed that the FRS performs better when the target speed is lower than its average speed. This is due to the target vehicle moving slowly and then the time required by the relay vehicles (in forward direction) to reach the target decreases. For example, at target speed 23 m/s, the average end-to-end delay for the RRS, FRS, GA and FCFS are 1.47×10^4 ms, 1.71×10^4 ms, 1.93×10^4 ms and 2.27×10^4 ms, respectively. The average end-to-end delay from RRS decreases by 74.46%, 19.85% over FCFS and GA, respectively.

4.4.3 Impact of co-operative cache (B_v)

Fig.4.9 shows the performance comparison among the scheduling algorithms RRS, FRS GA and FCFS in terms of average RSU energy consumption for different co-operative cache sizes of the relay vehicles. In this case, the average energy consumption increases with the increased size of the cooperative cache. This is because the relay vehicles with larger cache can participate in more number of downlink communications while they move close to the end of RSU coverage. Therefore, the average amount of data transmission per time slot decreases, and then it increases the average RSU energy consumption per

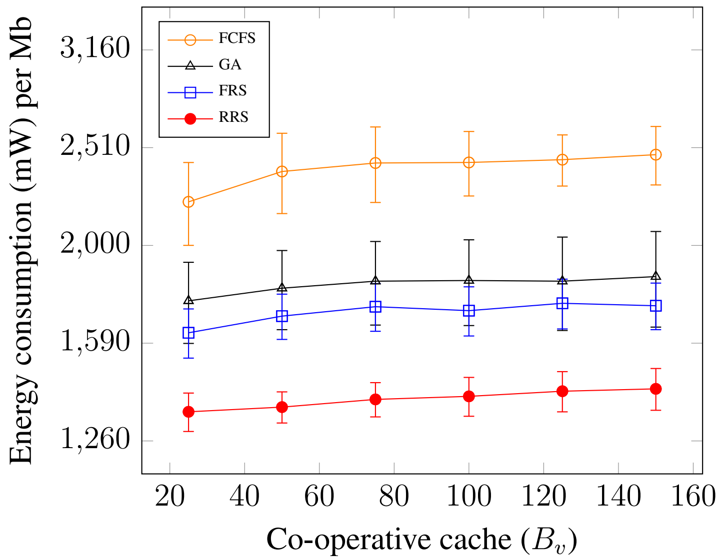


Figure 4.9: RSU energy consumption and Co-operative cache

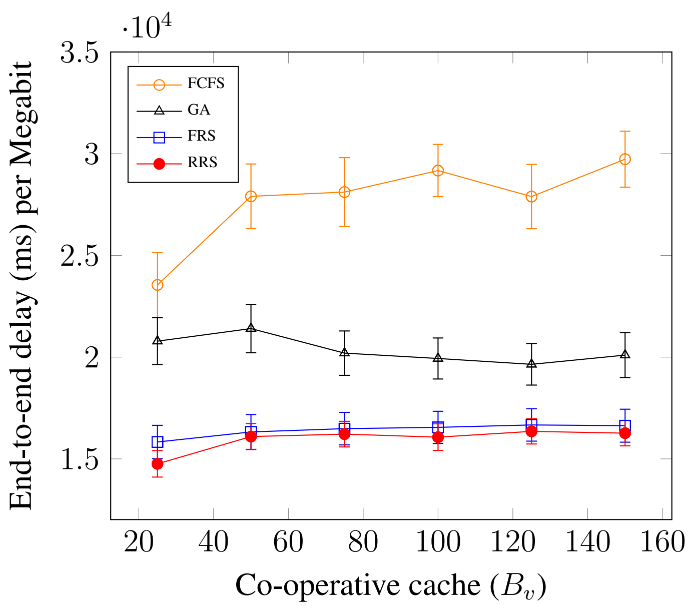


Figure 4.10: End-to-end delay and Co-operative cache

one megabit data delivery to the target vehicle. However, the proposed scheduler RRS outperforms other algorithms FRS, GA and FCFS in terms of average energy consumption. Here, the algorithms are compared by taking constant arrival rate 0.5 and target's speed 20 m/s. For example, when the cache size is 100Mb, the average energy consumption by the schedulers RRS, FRS, GA and FCFS are 1.39×10^3 mW, 1.71×10^3 mW, 1.83×10^3 mW and 2.42×10^3 mW, respectively. Meanwhile, the performance improvement for RRS over

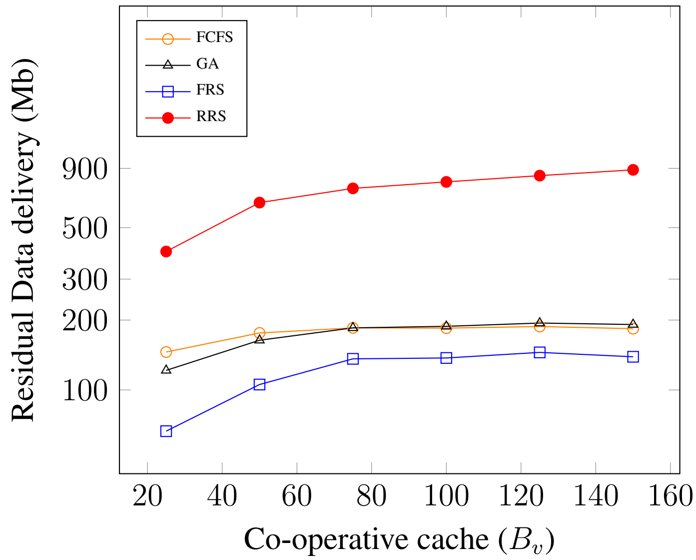


Figure 4.11: Residual data delivery and Co-operative cache

FRS, GA and FCFS are 72.08%, 30.91% and 22.63%, respectively.

Similarly, Fig.4.10 shows performance comparison of algorithms in terms of average end-to-end delay for different cache sizes of the relay vehicles. In this case also, the end-to-end delay per one megabit delivery will be much higher in FCFS when compared to other proposed algorithms. This is because that the FCFS selects relay vehicles that are more close to the end of RSU coverage. Therefore, the average amount of data transmission by FCFS per time slot is less compared to other algorithms, and then it increases the end-to-end delay per one mega bit data delivery. The performance improvement of RRS, FRS and GA over FCFS are 36.28%, 68.92% and 73.74%, respectively. In addition, Fig.4.9 and Fig.4.10 shows the standard deviation for multiple runs of the simulation. It is observed that the standard deviation for FCFS is more compared to other proposed algorithms, especially RRS shows more stable performance irrespective of co-operative cache sizes of the relay vehicles. However, the increased cache size of relay vehicles increases the residual data delivery to target vehicle as shown in Fig.4.11. When the cache sizes are in the range [25Mb,150Mb], it is observed that the proposed RRS scheduler delivers residual data nearly 4.5 times on average compared to other schedulers. In conclusion, although increased cache sizes perform lower in terms of average energy consumption and end-to-end delay, but it increases the data delivery for the target vehicle.

4.5 Summary

In this chapter, the energy efficient relay scheduling algorithms have been proposed for faster data retrieval to the target vehicle in the uncovered area. The proposed model selects the relay vehicles in a bidirectional highway by considering the minimization of RSU energy consumption and end-to-end delay to the target vehicle. The downlink (RSU to vehicle) channel uses variable bit-rate data transmission and the vehicle arrivals follow independent Poisson stream. The relay scheduling problem has been formulated and proved its NP hardness. By applying the concepts of *Auction Theory*, the RSU or seller optimally schedule the relay vehicles or bidders, based on the bids received from bidders. The bidders use local information such as location, speed, cooperative cache size, direction, time required to reach target vehicle, etc, and participate in Auctioning process. The proposed Auction-based RRS algorithm utilizes the neighboring RSU cooperation and schedules the relay vehicles driving in both directions that can maximize the data delivery to the target vehicle. Extensive simulations show that the proposed RRS algorithm performs better compared to FCFS, GA and FRS in terms of average RSU energy consumption and end-to-end delay to the target vehicle. For the case of vehicle arrival rate 0.5 and target speed 20 m/s, improvement of RRS over FCFS, GA, and FRS is 60.17%, 22.27% and 15.69% in terms of average RSU energy consumption, respectively.

In next chapter, to address the buffering delays at RSU, a trade-off between energy consumption and buffering delay is investigated while improving the data delivery between neighboring RSUs.

Chapter 5

Delay-aware Energy Efficient Dynamic Relay Scheduling in Isolated Vehicular Infrastructure

In highway locations, uncovered areas have been created due to deployment cost and limited radio range of RSUs. It has been observed that some of the RSUs deployed in isolated rural highway locations connect to neither grid power nor other RSUs. Therefore, such isolated RSUs (i.e., *source RSUs*) are equipped with rechargeable batteries and depend on energy harvesting technologies[13]. Moreover, the source RSU is responsible for gathering the tasks (stored into finite buffers) generated by the applications running in that source RSU region. The tasks waiting in buffer of source RSU need to be offloaded to a nearby *destination RSU* (which connects to high-end computation server and direct grid power) via store-carry-forward vehicles or relays. A mechanism is necessary to determine the minimum amount of power required to reduce the buffering delays at the RSU.

In this chapter, an energy-limited and isolated source RSU has been considered to store the task data (in finite buffers) before it forwards to destination RSU through relay vehicles via I2V communication. On the other hand, the vehicles are assumed to have different speeds, but each vehicle maintains constant speed in the highway segment[6]. However, it poses some challenges to design a good relay vehicle scheduling algorithm in such a dynamic scenario. First, the source RSU does not have control over the arrival of task data, and this may lead to continuous increase of buffer back-log size referred as buffer instability. Second, RSUs equipped with rechargeable batteries have limited storage capacity, consequently an efficient power allocation strategy is required for the effective utilization of stored energy in rechargeable batteries. Third, the future arrival of vehicles are completely unknown to *source RSU*, then the RSU needs to schedule the best possible relay vehicles available in its coverage region at current time slot. Fourth, due to half-duplex nature of On-Board-Units (OBUs)[132], it is a challenging issue for simultaneous V2V forwarding and receiving of data. Therefore, this work realizes only one hop RSU-to-relay (I2V) communication in the source RSU region. Fifth, although this kind of system tolerates the delivery delay to destination RSU, the tasks that are buffered in source RSU need to be computed within their deadlines.

To address the above challenges, a delay-aware energy efficient dynamic relay scheduling strategy is proposed to minimize the energy consumption of source RSU, buffering delay at source RSU and to maximize the average data delivery to the destination RSU in the network. Before selecting suitable relay vehicles, the proposed strategy first decides the minimum power allocation for transmission of buffer content by observing the buffer back-log sizes in each time slot. This dynamic power allocation technique conserves energy consumption of RSU and ensures buffer stability. Second, depending on the amount of data to be transmitted to each vehicle via I2V communication, the proposed strategy schedules a set of relay vehicles to maximize the average data delivery to destination RSU. The selection criteria of relay vehicles are subjected to task deadline constraints as well. The major contributions of this chapter is as follows.

- Present a system architecture in a bidirectional highway scenario for data sharing in between the neighbouring RSUs. Specifically, such a system enables the RSUs to opportunistically exploit the store-carry-forward (relay) vehicles, which is not only enhances the data delivery to destination RSU, but also realizes balancing the trade-off between buffer stability and energy consumption at the source RSU.

- Formulate two optimization problems namely, dynamic power allocation problem ($\mathcal{P}_1$) and a relay scheduling problem ($\mathcal{P}_2$). First, $\mathcal{P}_1$ objective is to minimize the energy consumption of source RSU subject to satisfy the buffer stability and energy level in the rechargeable batteries. Second, $\mathcal{P}_2$ objective is to maximize the data delivery to destination RSU subject to satisfy task deadlines. In particular, it aims to best exploring the data sharing services in source RSU via I2V communication and in destination RSU via V2I communication, respectively.

- Propose a Lyapunov optimization based Dynamic Power Allocation (LDPA) algorithm (Section 5.2.1), which allocates minimum power required for the transmission of buffer content by observing the buffer back-log size and channel gain. Furthermore, a Max-weight Relay Vehicle Scheduling (MRVS) algorithm (Section 5.2.1) has been proposed to select the relay vehicles based on their speed, location and achievable data rates. In particular, it is observed that the combination of LDPA and MRVS improves the efficacy of the system in-terms of buffer stability, network life time and data delivery.

The organization of remaining sections of this chapter is as follows. Section 5.1 presents system architecture and dynamics of buffering model, energy harvesting and consumption models. Section 5.2.1 and 5.2.2 describe the problem formulation and proposed algorithms for dynamic power allocation and relay scheduling problems, respectively. Section 5.3 presents experimental results in comparison to different parameters including vehicle arrival rate, vehicle speed and task arrival rate. Section 5.4 summarises the work in this chapter.

5.1 System Model and Dynamics

Fig.5.1 shows the deployment of *source RSU* and *destination RSU* separated by a distance $\mathbb{D}$ in highway locations. These RSUs are not connected via either direct wired back-haul links or wireless links. The distance between these neighboring RSUs is more than their communication region R, i.e., $\mathbb{D} >> 2R$. Generally, very few RSUs in rural highway locations are connected to wired grid power and such RSUs also equipped with edge servers with sufficient computation capabilities, referred to as *destination* RSUs. But, most of the

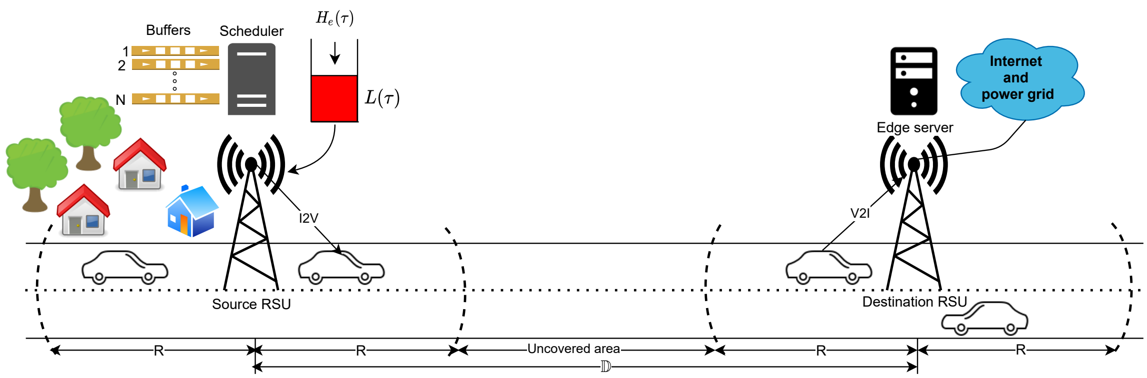


Figure 5.1: Intermittently deployed roadside units in bidirectional rural highways

remaining RSUs are isolated and they do not have direct wired power. Such an isolated RSU depends on renewable energy and acts as a *source* RSU to gather the task data generated by user applications running in that *source* RSU region. The gathered task data is buffered at source RSU before transmitted to destination RSU via store-carry-forward (relay) vehicles which move in the direction of destination RSU. A set of $\mathcal{N} = \{1, 2, ..., N\}$ user applications is assumed to run in the source RSU region. The tasks generated by i^{th} application have specific computation deadline d_i. The results computed at the destination RSU are expected to relay back to source RSU within the specified computation deadline. Sufficient computation capabilities of edge server realizes computing and buffering delays incurred at the destination RSU as negligible in this model.

The detailed description of notations mentioned in this chapter is given in Table 5.1.

5.1.1 Task data arrival and buffering model

The system time is divided into a discrete set of time slots $\tau \in \{0, 1, 2, ...\}$ where each time slot is of equal length δt. The arrival process of task data $A_i(\tau)$ from a user application i to the source RSU are distributed independently and identically in each time slot τ. The set of arrived task data is denoted as $A(\tau) = \{A_1(\tau), A_2(\tau), ..., A_i(\tau), ..., A_N(\tau)\}$. The arrived data $A_i(\tau)$ from application i is stored in a corresponding ith buffer which has a back-log size $B_i(\tau)$ at time slot τ. The set of buffer back-log sizes at the source RSU represents $\mathcal{B}(\tau) = \{B_1(\tau), B_2(\tau), ..., B_i(\tau), ..., B_N(\tau)\}$. The dynamics of ith time variant buffer associated with the source RSU in each time slot τ is represented as follows,

Table 5.1: Notations and Descriptions

Notation	Description
$\mathbb{D}$	Distance between source and destination RSUs
R	Radio coverage of RSUs
N	Number of user applications (or) buffers (or) channels
i	Index i refers to ith application or ith buffer or ith channel
j	Index refers to vehicle
d_i	Deadline of tasks generated by ith application
τ	Time slot of duration δt
$A_i(\tau)$	Task data (in bits) arrives to ith buffer in time slot τ
$B_i(\tau)$	Back-log size (in bits) of ith buffer in time slot τ
$B(\tau)$	Vector of N buffer back-log sizes
$C_i(\tau)$	Task data (in bits) transmitted from ith buffer at time slot τ
δt	Duration of time slot τ
$r_{ij}(\tau)$	Data rate over channel i to vehicle j at time τ
$\mathcal{G}_{ij}(\tau)$	Power gain of channel i when assigned to vehicle j at τ
W	Bandwidth of any channel
$D_j(\tau)$	Distance between source RSU and vehicle j at time τ
$V(\tau)$	Set of all vehicles in the source RSU region at time τ
$P_i(\tau)$	Power allocated to transmit task data over channel i at time τ
$P(\tau)$	Total power consumption of source RSU at time τ
$H_e(\tau)$	Energy harvested at time slot τ
$L(\tau)$	Energy level of battery at time τ
P^{tx}_{max}	Maximum transmit power allocates to each channel
$\mathbb{V}$	Control parameter for power allocation and buffer stability
$x_{ij}(\tau)$	Decision variable refers assignment of channel i to vehicle j

$$B_i(\tau + 1) = max\{B_i(\tau) - C_i(\tau), 0\} + A_i(\tau), \ \forall i, \forall \tau \tag{5.1}$$

where $B_i(\tau)$ is the buffer back-log size that counts the task data (in bits) stored in ith buffer at time slot τ, $C_i(\tau)$ is the departure process (observed from Eq. 5.5) that represents downlink transmission rate (in bits) from ith buffer in source RSU to a selected relay vehicle j at time slot τ, and $A_i(\tau)$ is the arrival process that denotes received task data (in bits) to ith buffer in source RSU at time slot τ. It can be observed that, $B_i(0) = 0$ denotes an initial condition of empty buffer at the first time slot $\tau = 0$. In addition, the average back-log size of the buffers is represented in equation (5.2), and consider that the buffers are stable when their average back-log size $\overline{B(\tau)}$ at time slot τ is bounded[133], and it is given as,

$$\overline{B(\tau)} = \lim_{T\to\infty} \frac{1}{T} \sum_{\tau=0}^{T-1} \sum_{i=1}^{N} \mathbb{E}\{B_i(\tau)\},\ \forall\tau \tag{5.2}$$

5.1.2 Communication and energy consumption model

Task data stored in each buffer are transmitted to a selected vehicle in the source RSU region. The data transmission follows FIFO (First-In-First-Out) order in the buffers. Moreover, it is assumed that the number of orthogonal channels available for data transmission is equal to number of buffers. Note that the index i is used to refer both channel i and its corresponding ith buffer. Data rate (i.e., $r_{ij}(\tau)$) over channel i to a vehicle j at a time slot τ depends on two factors: 1) the channel power gain $\mathcal{G}_{ij}(\tau)$ because it is a function of distance $D_j(\tau)$ between RSU to vehicle j 2) the transmit power $P_i(\tau)$ allocated to the channel i at time slot τ.

The source RSU calculates the distance to vehicles available in that RSU region. When a vehicle j arrives into an RSU region, the vehicle informs its arrival time t_j and velocity v_j to the RSU. It is assumed the velocity of vehicles is constant in the highway scenarios[12]. Therefore, the distance $D_j(\tau)$ from RSU to vehicle j in time slot τ is derived as,

$$D_j(\tau) = R - (\delta t\,\tau - t_j)\,v_j,\ \forall\tau \tag{5.3}$$

Furthermore, the set of vehicles present in the source RSU region in each time slot τ can be denoted as

$$V(\tau) = \{j \mid |D_j(\tau)| \leq R\ and\ t_j \leq \delta t\,\tau\}, \forall j \tag{5.4}$$

Moreover, the task data (in bits) transmitted over channel i to vehicle j at time slot τ is defined as,

$$\left.\begin{aligned} r_{ij}(\tau) &= W\log_2\left(1 + \tfrac{|\mathcal{G}_{ij}(\tau)|^2\ P_i(\tau)}{N_0}\right),\ \forall i, \forall j, \forall\tau \\ C_i(\tau) &\triangleq r_{ij}(\tau)\,\delta t,\ \forall i, \forall j, \forall\tau \end{aligned}\right\} \tag{5.5}$$

where W is the allocated bandwidth for each channel and N_0 is the background noise power. However, total power consumption of the source RSU at time slot τ is represented as τ as $P(\tau) = \sum_{i=1}^{N} P_i(\tau)$.

5.1.3 Battery capacity and energy harvesting model

The source RSU is equipped with energy harvesting devices to capture the energy from alternative sources e.g., solar, wind, etc. The $H_e(\tau)$ is the harvested energy captured and stored in the battery during a time slot τ. It is assumed that $H_e(\tau)$ values are i.i.d with a maximum value of $H_e^{max}(\tau)$ and is defined as[134],

$$0 < H_e(\tau) \leq H_e^{max}(\tau),\ \forall\tau \tag{5.6}$$

The energy harvested in all the previous time slots can be used for transmission of data in current time slot. Energy level of the battery in a time slot τ is denoted as $L(\tau)$ and the bounds on amount of battery discharge in each time slot is represented as,

$$L_{min}(\tau) \leq L(\tau) \leq L_{max}(\tau),\ \forall\tau \tag{5.7}$$

where $L_{min}(\tau)$ and $L_{max}(\tau)$ are the minimum and maximum energy that can be consumed respectively by the source RSU in a time slot τ.

As aforementioned, the total energy consumption for data transmission in a time slot τ is $P(\tau)$, but $P(\tau)$ is constrained by the energy level of battery as,

$$0 < P(\tau) \leq L(\tau),\ \forall\tau \tag{5.8}$$

Furthermore, the energy level in the battery in the next time slot can be computed as follows,

$$L(\tau+1) = min\{L(\tau) - P(\tau) + H_e(\tau), L_{max}(\tau)\},\ \forall\tau \tag{5.9}$$

where $L_{max}(\tau)$ is the maximum capacity of battery at time slot τ.

5.2 Problem Formulation and Energy Efficient Dynamic Scheduling of Relay Vehicles

This section presents a dynamic power allocation scheme for source RSU using Lyapunov optimization technique (refer section 5.2.1), and the assignment of relay vehicles to downlink channels while maximizing the task data delivery to destination RSU (refer section 5.2.2).

5.2.1 Decision making on power allocation

The problem $\mathcal{P}_1$ is formulated to minimize the running time averages of RSU power consumption while it seeks to stabilize the queuing buffers in each time slot τ. If the buffer is not stable, then the buffer back-log size can grow continuously when the downlink transmission rate is not more than the data arrival rate of the buffers. From Eq.5.5, the transmission rate is a function of power consumption of RSU. The higher transmission rate incurs more power consumption and the buffers are also stable. The lower transmission rate needs low power consumption that conserves the energy but the buffers are not stable. Therefore, It is required to optimize the RSU power consumption based on the buffer backlog size. Here, an indicator variable $x_{ij}(\tau)$ is considered to be equal to 1 when buffer i is chosen to transmit to vehicle j over channel i, otherwise $x_{ij}(\tau)$ is zero.

The formulation of problem $\mathcal{P}_1$ is as follows,

$$\mathcal{P}_1 : \underset{x_{ij}(\tau)}{\text{minimize}} \lim_{T\to\infty} \frac{1}{T} \sum_{\tau=0}^{T-1} \sum_{i=1}^{N} \mathbb{E}\{P_i(\tau)\} \tag{5.10}$$

$$\text{s.t. } (5.6), (5.7) \tag{5.11}$$

$$0 \leq P_i(\tau) \leq P_{max}^{tx}, \ \forall i, \forall \tau \tag{5.12}$$

$$\sum_{i=1}^{N} P_i(\tau) \leq L(\tau), \ \forall i, \forall \tau \tag{5.13}$$

$$\lim_{\tau\to\infty} \frac{\mathbb{E}\{|\overline{B(\tau)}|\}}{\tau} = 0, \ \forall \tau \tag{5.14}$$

The objective function in (5.10) represents the minimization of running time averages of power consumption of source RSU in each time slot τ. The constraint (5.11) denotes the limitation on the harvested energy and battery discharge capacity in each time slot τ. The constraints (5.12) represents the upper bound on transmit power (where P^{tx}_{max} is maximum transmit power). The constraint (5.13) gives that total power consumption of source RSU and it is not more than the energy level of battery in each time slot τ. The buffer stability constraint in (5.14) is considered by setting the buffering rate equals to zero.

A closed-form expression is derived for downlink transmission rate only by investigating the stability of the buffer. Let $B(\tau)$ be a vector of buffer back-logs. Then, a quadratic Lyapunov function has been defined as a scalar measure of buffer back-logs[93], given as

$$L(B(\tau)) \triangleq \frac{1}{2}B^T(\tau)B(\tau) = \frac{1}{2}\sum_{i=1}^{N} B_i(\tau)^2 \tag{5.15}$$

where $B^T(\tau)$ represents the transpose of $B(\tau)$. To consistently maintain lower congestion in the buffers, Eq. 5.1 has been used to compute bound on the difference in Lyapunov function from current to next time slot,

$$\begin{aligned} & L(B(\tau+1)) - L(B(\tau)) \\ & = \frac{1}{2}\sum_{i=1}^{N}[B_i(\tau+1)^2 - B_i(\tau)^2] \\ & = \frac{1}{2}\sum_{i=1}^{N}[(max\{B_i(\tau) - C_i(\tau), 0\} + A_i(\tau))^2 - B_i(\tau)^2] \end{aligned} \tag{5.16}$$

Since $(max\{B_i(\tau) - C_i(\tau), 0\} + A_i(\tau))^2 \leq (B_i(\tau) - C_i(\tau) + A_i(\tau))^2$ [135], then (5.16) can be represented as,

$$\begin{aligned} L(B(\tau+1)) - L(B(\tau)) \leq \sum_{i=1}^{N} \frac{[A_i(\tau)^2 + C_i(\tau)^2]}{2} \\ - \sum_{i=1}^{N} A_i(\tau)\, C_i(\tau) + \sum_{i=1}^{N} B_i(\tau)[A_i(\tau) - C_i(\tau)] \end{aligned} \tag{5.17}$$

Let $\Delta(B(\tau))$ be a conditional Lyapunov drift which can keep the buffers stable. The drift in the time slot τ is denoted as,

$$\Delta(B(\tau)) \triangleq \mathbb{E}\{L(B(\tau+1)) - L(B(\tau))|B(\tau)\} \tag{5.18}$$

According to (5.18), the drift in time slot τ can be expressed as,

$$\begin{aligned}\Delta(B(\tau)) \leq \mathbb{E}\Bigg\{\sum_{i=1}^{N}\Big\{\frac{[A_i(\tau)^2 + C_i(\tau)^2]}{2} - A_i(\tau)\,C_i(\tau)\Big\}\Big|B(\tau)\Bigg\}\\ +\mathbb{E}\Bigg\{\sum_{i=1}^{N}B_i(\tau)[A_i(\tau) - C_i(\tau)]\Big|B(\tau)\Bigg\}\end{aligned} \tag{5.19}$$

For all τ, the first term on the right hand side of (5.19) is finite because of the bound on the maximum value of $C_i(\tau)$. Let $\mathbb{Z}$ be a finite constant and $\mathbb{Z} > 0$ to express the above inequality as follows,

$$\Delta(B(\tau)) \leq \mathbb{Z} + \mathbb{E}\Bigg\{\sum_{i=1}^{N}B_i(\tau)[A_i(\tau) - C_i(\tau)]\Big|B(\tau)\Bigg\} \tag{5.20}$$

where

$$\mathbb{E}\Bigg\{\sum_{i=1}^{N}\Big\{\frac{[A_i(\tau)^2 + C_i(\tau)^2]}{2} - A_i(\tau)C_i(\tau)\Big\}\Big|B(\tau)\Bigg\} \leq \mathbb{Z}$$

The proposed dynamic algorithm for determining the downlink transmission rate over each time slot τ, is designed by observing the buffer back-log size $B_i(\tau)$ and then deciding the power allocation $P_i(\tau)$. Rather than minimizing only the drift $\Delta(B(\tau))$, the dynamic algorithm minimizes the bound on $\mathbb{E}\{P(\tau)|B(\tau)\} + \mathbb{V}\Delta(B(\tau))$ (i.e., *drift-plus-penalty*), where the constant $\mathbb{V} \geq 0$ is a parameter to control the trade-off existing between buffer stability and power allocation. Intuitively, large values of $\mathbb{V}$ emphasizes more on buffer stability but it consumes more power. Small values of $\mathbb{V}$ leads to less power consumption but there is possibility that the buffer become unstable.

Therefore, the proposed algorithm minimizes the bound on *drift-plus-penalty*, which can be expressed as follows,

$$\mathbb{E}\{P(\tau)|B(\tau)\} + \mathbb{V}\,\Delta(B(\tau)) \leq Z + \mathbb{E}\left\{\sum_{i=1}^{N} P_i(\tau)\middle| B(\tau)\right\} + \mathbb{V}\,\mathbb{E}\left\{\sum_{i=1}^{N} B_i(\tau)[A_i(\tau) - C_i(\tau)]\middle| B(\tau)\right\} \tag{5.21}$$

According to the concept of opportunistic minimization of expectation [93], the expression (5.21) can be minimized by minimizing the values inside the expectation. Therefore, the proposed algorithm accomplishes the expression (5.21) by minimizing,

$$\sum_{i=1}^{N} P_i(\tau) + \mathbb{V}\,\sum_{i=1}^{N} B_i(\tau)[A_i(\tau) - C_i(\tau)] \tag{5.22}$$

From (5.5), the downlink transmission rate $C_i(\tau)$ can be substituted in (5.22). Moreover, it is observed that the expression (5.22) is clearly separable for each channel i [136], and it is a function of power allocation $P_i(\tau)$ as follows,

$$\mathcal{F}(P_i(\tau)) \triangleq P_i(\tau) + \mathbb{V}\,B_i(\tau)A_i(\tau) - \mathbb{V}\,B_i(\tau)W\log_2\left(1 + \frac{|\mathcal{G}_{ij}(\tau)|^2\,P_i(\tau)}{N_0}\right) \tag{5.23}$$

where the product of buffer back-log size $B_i(\tau)$ and arrival rate $A_i(\tau)$ is constant since the arrival rate is independent of power allocation.

It is clear that the function $\mathcal{F}(P_i(\tau))$ depends on power allocation decision in each time slot τ. The minimum power required to allocate for each channel i can be obtained by setting the derivative of the objective function in (5.23) equal to 0 (i.e., $\frac{\partial}{\partial P_i(\tau)}\{\mathcal{F}(P_i(\tau))\} = 0$)[137] and constant $1/\ln 2 \approx 1.44$, as

$$P_i(\tau) \approx 1.44\,\mathbb{V}\,B_i(\tau)W - N_0/|\mathcal{G}_{ij}(\tau)|^2 \tag{5.24}$$

Re-substitute the derived minimum power allocation $P_i(\tau)$ into (5.5). Then, the possible downlink transmission rate $r_{ij}(\tau)$ which can be used as an input argument to the Max-

Weight downlink relay scheduling problem $\mathcal{P}_2$.

Algorithm 5.1 LDPA: Lyapunov based dynamic power allocation for wireless channels in a time slot

Input parameters:

- $\mathbb{V}$: parameter indicates power-delay trade-off
- W : bandwidth of each wireless channel
- N : number of wireless channels
- τ : current time slot

Dynamic power allocation:

```
Initialize i = 1, j = 1
Find a set of vehicles V(τ) in source RSU region at time slot t using eq. (5.4)
Choose trade-off value V to adjust the power P_i(τ)
while i ≤ N do
   while j ≤ |V(τ)| do
      Observe B_i(τ), G_ij(τ), N_0
      Make a decision on power allocation P_i(τ):   P_i(τ) ≈ 1.44 𝕍 B_i(τ)W − N_0/|G_ij(τ)|^2
      Substitute P_i(τ) in eq. (5.5) and obtain downlink transmission rate r_ij(τ)
      j ← j+1
   end while
   i ← i+1
end while
```

Based on the above discussion, this work proposes an energy efficient dynamic power allocation algorithm (Algorithm 5.1) to make a decision on power allocation for each channel i when it is assigned to a vehicle $j \in V(\tau)$. The algorithm can be described as follows: Given the input parameters $\mathbb{V}$, N and W, for each channel i, the source RSU (i) observes the corresponding ith buffer back-log size $B_i(\tau)$, channel gain $\mathcal{G}_{ij}(\tau)$ and noise N_0 (ii) computes the power allocation $P_i(\tau)$ from eq. (5.24) (iii) obtains the downlink transmission rate $r_{ij}(\tau)$. The Algorithm 5.1 runs in each time slot τ for allocating minimum transmission power while satisfying the constraint on buffer stability. Moreover, the computational complexity for determining the amount of power allocation in the algorithm is as $O(1)$.

5.2.2 Max-weight scheduling of relay vehicles

The problem $\mathcal{P}_2$ aims to maximize the downlink transmission of task data (in bits) to suitable relay vehicles which satisfy the constraints on task deadline and distance to RSU at a time slot τ. The formulation of problem $\mathcal{P}_2$ is as follows,

$$\mathcal{P}_2 : \underset{x_{ij}(\tau)}{\text{maximize}} \sum_{i=1}^{N} \sum_{j=1}^{V(t)} M_{ij}(\tau)\, x_{ij}(\tau) \tag{5.25}$$

$$\text{s.t.} \sum_{j=1}^{V(\tau)} x_{ij}(\tau) \leq 1, \forall i \tag{5.26}$$

$$\sum_{i=1}^{N} x_{ij}(\tau) \leq 1,\ \forall j \tag{5.27}$$

$$\frac{\mathbb{D} - R + D_j(\tau)}{v_j} + \frac{\mathbb{D} - 2R}{v_{avg}} \leq x_{ij}(\tau)\, d_i,\ \forall i, \forall j \tag{5.28}$$

$$D_j(\tau) \leq R,\ \forall j \tag{5.29}$$

$$x_{ij}(\tau) \leq 1,\ \forall i, \forall j \tag{5.30}$$

$$x_{ij}(\tau) \in \{0, 1\},\ \forall i, \forall j \tag{5.31}$$

where $x_{ij}(\tau)$ is a decision variable denotes the assignment of downlink channel i to a relay vehicle $j \in V(\tau)$ at a time slot τ if $x_{ij}(\tau)$ is set to 1. Otherwise, $x_{ij}(\tau)$ is set to 0.

The aim of objective function in (5.25) is to maximize the summation of weights of edges between the downlink channels and relay vehicles. The weight of each edge is defined as,

$$M_{ij}(\tau) \triangleq B_i(\tau)\, r_{ij}(\tau)\, \delta t,\ \forall i,\ \forall j \tag{5.32}$$

where $B_i(\tau)$ denotes ith buffer back-log size and $r_{ij}(\tau)$ denotes downlink transmission rate (in bits) when a channel i assigned to a relay vehicle $j \in V(\tau)$. Moreover, the constraints (5.26) and (5.27) ensure the one-to-one assignment of channels to relay vehicles. The constraint (5.28) ensures that the sum of travel time of relay vehicle j to reach the destination RSU and time required for the computation reply to reach the source RSU should not exceed the deadline d_i of tasks associated to ith buffer. It assumes the average speed of a vehicle (which carries computation reply) from destination RSU to source RSU as v_{avg}, and ignores the computation delay at the destination RSU. The constraint (5.29) restricts the selection of a vehicle as relay vehicle when it is outside the RSU region. Further, (5.30) and (5.31) represents the integer constraints.

To find the optimal scheduling decision for the problem $\mathcal{P}_2$, the algorithms are required to run in exponential computation time. For this problem, the scheduling decision of assigning the relay vehicles to down link channels can be viewed as a special case of 0/1 multiple knapsack problem[138], which is a well-known NP-hard problem. Therefore, a greedy heuristic based relay vehicle scheduling algorithm is proposed for obtaining the sub-optimal solution to the problem $\mathcal{P}_2$. The proposed greedy algorithms select the edges (between relay vehicles and downlink channels) with maximum weights in each time slot τ is described in Algorithm 5.2.

Algorithm 5.2 MRVS: Max-weight Relay Vehicle Scheduling algorithm

Input: Set of weights is $E = \{M_{ij}(\tau)\}$, where $i \in \{1, ..., N\}$, $j \in \{1, ..., \left|V(\tau)\right|\}$, $t \leftarrow$ current time slot
Output: Schedule matrix X

1: Sort elements of E such that $e_1 \geq e_2 \geq ... \geq e_{|E|}$, where $e_k \in E$
2: **for** $i = 1$ to N **do**
3:     **for** $j = 1$ to $\left|V(\tau)\right|$ **do**
4:         $X[i][j] = 0$; /* initialization*/
5:     **end for**
6: **end for**
7: **for** $k = 1$ to $|E|$ **do**
8:     $i, j \leftarrow e_k$ /* store associated indices of element e_k */
9:     **if** $\left(\text{Sum}[row_i(X)] > 0\right)$ or $\left(\text{Sum}[column_j(X)] > 0\right)$ **then**
10:         go to step 7;
11:     **else if** vehicle j satisfy eq.(28) and eq.(29) **then**
12:         $X[i][j] = 1$; /* Schedule the vehicle j as a relay*/
13:     **else**
14:         $X[i][j] = 0$;
15:     **end if**
16: **end for**
17: **return** X

In detail explanation for the Algorithm 5.2 is as follows: given the input set E having the elements as edge weights computed from the Eq. (5.32), the set E is sorted in the decreasing order of edge weights. Initially, a scheduling matrix X is considered and its elements are initialized to zero. From line 7, the algorithm iterates over each element of sorted set E and then applies the constraints for the selection of current element. Each element $e_k \in E$ has associated indices that represents ith channel and jth relay vehicle. The condition given in line 9 verifies whether both ith channel and jth vehicle are unassigned or not. Here, Sum$[row_i(X)]$ and Sum$[column_j(X)]$ denotes the summation of elements in ith row and summation of elements in jth column of X, respectively. Further, line 11-15 represents the assignment of jth vehicle to ith channel if the vehicle j satisfies the task deadline constraint and distance to RSU constraint shown in line 11. Finally, the proposed greedy algorithm produces matrix X to denote the near optimal assignment of channels to relay vehicles. The computational complexity of the proposed algorithm is as $O(N \log N)$.

5.3 Experimental Analysis

This section presents experimental analysis of proposed algorithms in terms of buffering performance (refer to 5.3.1) and scheduling performance (refer to 5.3.2).

For simulation settings, the neighbouring RSUs are deployed in a bidirectional highway segment and are separated by a distance 4000m. The communication range of each RSU is 1000m. The vehicles entering into source RSU region follow Poisson process with mean rate of arrival $\lambda = 1$. The speeds of vehicles are suppose to be distributed uniformly in a range [12m/s, 28m/s] such that the faster vehicles can overtake slower vehicles. In this simulation, the communication model uses parameters specified in [24] where the downlink bit rates vary from 3 to 27 Mb/s, maximum transmit power is 1W, bandwidth of channel is 10MHz and the noise at the relay vehicle is -174dBm/Hz. Furthermore, considering simple simulation criteria, the number of buffers associated with the source RSU is assumed as ten (i.e., $N = 10$). The number of task arrivals are considered as Poisson stream with a mean arrival of 1000 tasks in each time slot, and the length (in bits) of each task is chosen as 1024 bits. The computation deadlines of tasks are in the range 200 sec $\sim$ 300 sec. The simulation runs on a desktop system with 3.40GHz Intel core i7 CPU, 3.7 GiB of RAM, 64-bit ubuntu 16.04 LTS operating system with Python 2.7.12 for simulator development. Different scheduling algorithms are evaluated using Monte Carlo simulations over 100 time slots over 1000 iterations.

The performance of proposed LDPA is evaluated in terms of normalized buffering delay. The combined effect of LDPA and MRVS is analysed in terms of average buffer back-log size, normalized network life time and average data delivery.

5.3.1 Buffering performance

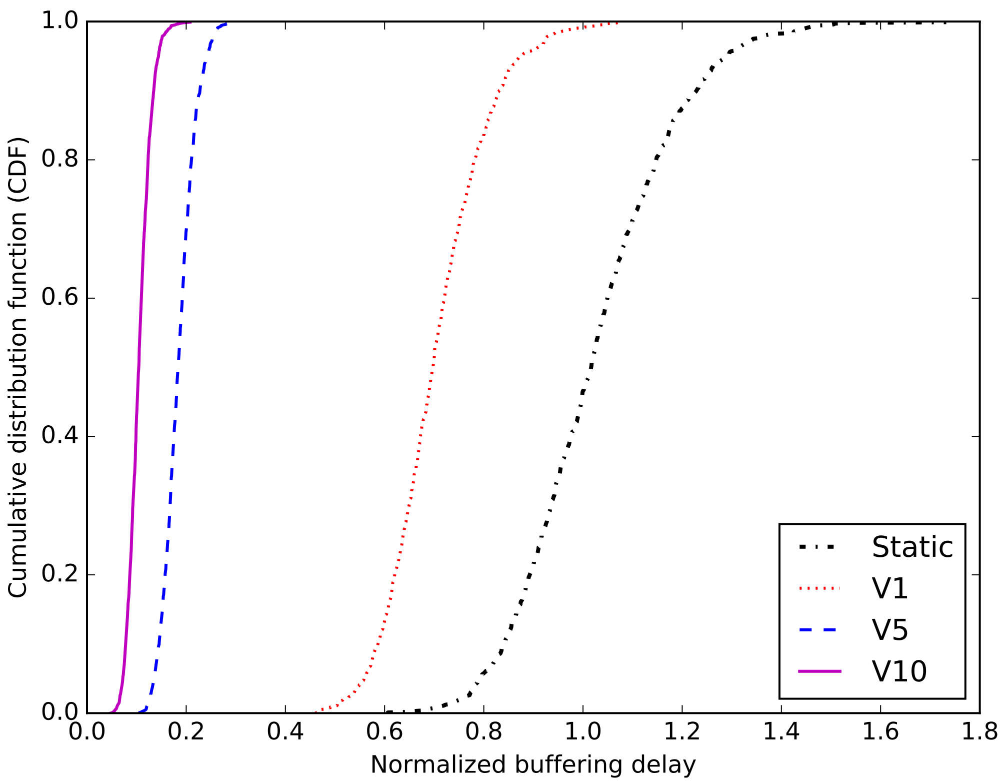


Figure 5.2: Buffering delay for control parameter $\mathbb{V}$

To evaluate the buffering performance in the proposed model, the stochastic buffering is simulated using Lyapunov based dynamic power allocation (LDPA) for different $\mathbb{V}$ values 1, 5 and 10 entitled as $V1$, $V5$ and $V10$ respectively. In addition, this LDPA based stochastic buffering is then compared with a *Static* power allocation method. This *Static* method allocates fixed transmit power which is an average of maximum and minimum transmit powers. Figure 5.2 shows simulation results of dynamic power allocation for different $\mathbb{V}$

values compared to static power allocation. In Fig. 5.2, normalized buffering delay (where maximum delay is set to 2) is shown on x-axis and cumulative distributive function of delay is given on y-axis. It is observed that the median values of normalized buffering delays for $V10$, $V5$, $V1$ and *Static* are 0.156, 0.212, 0.642 and 1.082, respectively. Moreover, dynamic power allocation for $V = 10$ gives minimum buffering delay with maximum buffer stability. However, the *Static* power allocation method does not perform well when compared to the dynamic power allocation.

5.3.2 Scheduling performance

This section presents the performance evaluation of the proposed Max-weight Relay Vehicle Scheduling (MRVS) algorithm when augmented with dynamic power allocation (LDPA) algorithm. The performance of proposed MRVS is evaluated in terms of following parameters.

- *Average buffer back-log size* denotes the amount of task data (in bits) buffered for transmission per time slot. With this evaluation parameter, it is possible to asses buffering delay and stability performance of the buffers.
- *Normalized network life time* denotes the normalized time duration (where maximum time is set to 48 hours) in which the source RSU available for the operations of scheduling and data transmission.
- *Average data delivery* denotes the average amount of task data (in bits) to be transmitted to destination RSU per time slot.

Furthermore, the proposed scheduling algorithm MRVS has been compared with four scheduling algorithms namely,

- *First Come First Serve (FCFS)* selects a relay vehicle which has lowest arrival time and present in the source RSU region at each time slot[139].
- *Fastest First (FF)* selects a vehicle as relay which has highest speed among the other vehicles in source RSU[23].

- *Rate Monotonic Scheduling (RMS)* selects a vehicle as relay which spends least time (or period) in the source RSU region[140].
- *Nearest Fastest Set (NFS)* selects a vehicle as relay which is fastest out of all vehicles nearest to source RSU[18].

Similar to *NFS*, the proposed *MRVS* algorithm also selects relay vehicles (with maximum downlink bit rates) nearest to RSU. In addition, the *MRVS* considers the back-log size of buffers for edge weight calculation (see Eq.5.32). Intuitively, the *MRVS* gives highest priority to transmit the content from buffer with maximum back-log size to a nearest relay vehicle. Therefore, the The *MRVS* ensures better buffer stability and data delivery with minimum energy consumption compared to *NFS*. The *MRVS* observes 1∼2% improvement over *NFS* when analysed in terms of buffer back-log size, network life time and data delivery for increased number of vehicles, speeds of vehicles and number of arrived tasks. Furthermore, *MRVS* is compared with basic well-known algorithms like *FCFS*, *FF* and *RMS*, in this section.

5.3.2.1 Effect of vehicle arrival rates

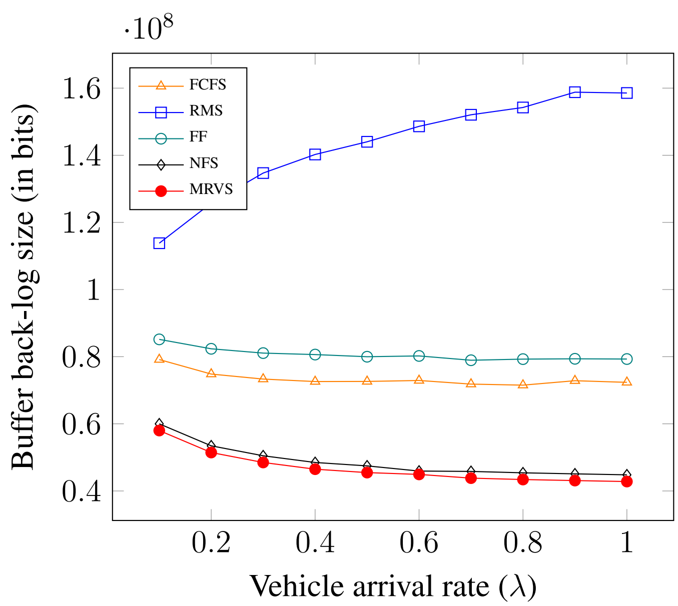


Figure 5.3: Vehicle arrival rate and Buffer back-log size when mean vehicle speed=20 m/s and task arrival rate=1000 are constant.

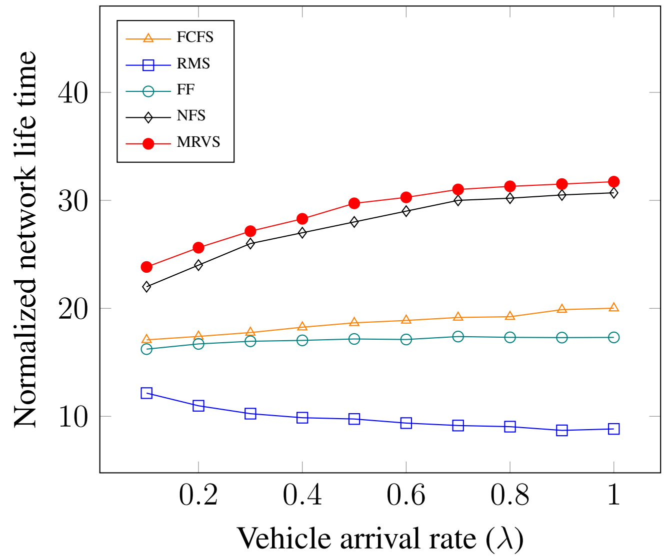


Figure 5.4: Vehicle arrival rate and normalized network life time when mean vehicle speed=20 m/s and task arrival rate=1000 are constant.

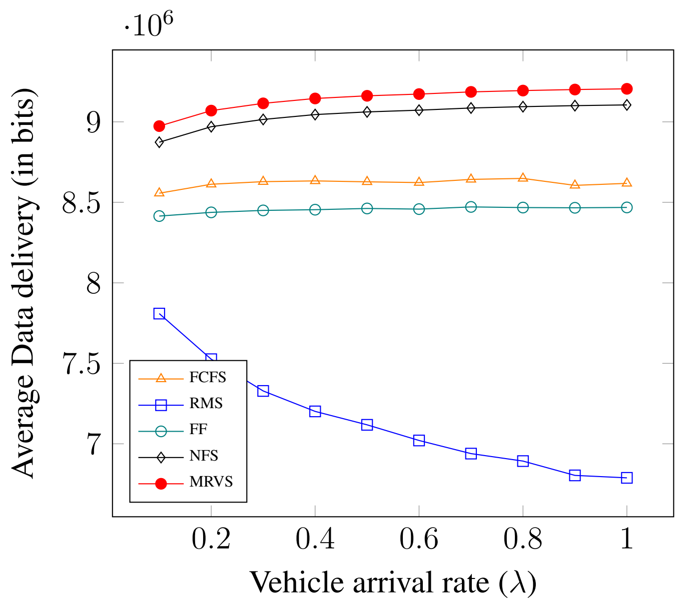


Figure 5.5: Vehicle arrival rate and data delivery when mean vehicle speed=20 m/s and task arrival rate=1000 are constant.

Figure 5.3 shows comparison results for the scheduling algorithms *MRVS*, *NFS*, *RMS*, *FF* and *FCFS* in terms of buffer back-log size for different vehicle arrival rates. The results are obtained by varying the mean vehicle arrival rate λ form 0.1 to 1, and keeping the mean vehicle speed and mean task arrival rate as constant at 20 m/s and 1000, respectively. Intuitively, the vehicle arrival rate in the source RSU region can affect the number of vehicles existing in that RSU region for any time instant. When number of vehicles are high, then the likelihood of availability of a relay vehicle near to RSU increases. Consequently, the RSU can transmit more task data to nearby relay vehicle, thereby average buffer back-log size decreases with the increasing vehicle arrival rates. In contrast, *RMS* selects farthest vehicles near to end of RSU coverage because such vehicles leave RSU quickly. With increased vehicle rates the selected farthest relay vehicle receives less task data. Therefore, it is clear that the buffer stability is high at higher vehicle arrival rates for except RMS. However, the proposed *MRVS* algorithm outperforms other four algorithms (*NFS*, *RMS*, *FF* and *FCFS*). Here, the average buffer back-log size for *MRVS* decreases by 75.9%, 59.71% and 2.16 times when compared to *FF*, *FCFS* and *RMS*, respectively.

Figure 5.4 shows the performance analysis of the algorithms in terms of normalized network life time for different vehicle arrival rates. Since, the average buffer back-log sizes are less at higher vehicle arrival rates, the amount of power allocation (using LDPA algorithm) of RSU for data transmission also less. Consequently, with the increased vehicle arrival rates the network life time increases for the algorithms *NFS*, *FF* and *FCFS*, and decreases for *RMS*. However the *MRVS* outperforms other four algorithms. This is because, the amount of average buffered task data for *MRVS* is less compared to other four algorithms. Therefore, from Fig 5.4, the network life time of *MRVS* increases by 37.93%, 58.62% and 68.96% when compared to *FCFS*, *FF* and *RMS* algorithms, respectively. Furthermore, Fig. 5.5 shows the performance analysis of these algorithms in terms of average data delivery per time slot. It is observed that the proposed algorithm outperforms other four algorithms. Moreover, it indicates that the average data delivery of *MRVS* is higher by 5.83%, 7.6% and 22.3% when compared to *FCFS*, *FF* and *RMS* algorithms, respectively.

5.3.2.2 Effect of vehicle speeds

Figure 5.6 shows the performance comparison of *MRVS*, *NFS*, *RMS*, *FF* and *FCFS* algorithms in terms of buffer back-log size for different mean vehicle speeds. The results are obtained by varying the mean speed of vehicles from 12 m/s to 28 m/s, and keeping the vehicle arrival rate and task arrival rates are constant at $\lambda = 1$ and 1000, respectively. Increasing the mean speed of vehicles clearly shows decrease in buffer back-log size for *NFS*,

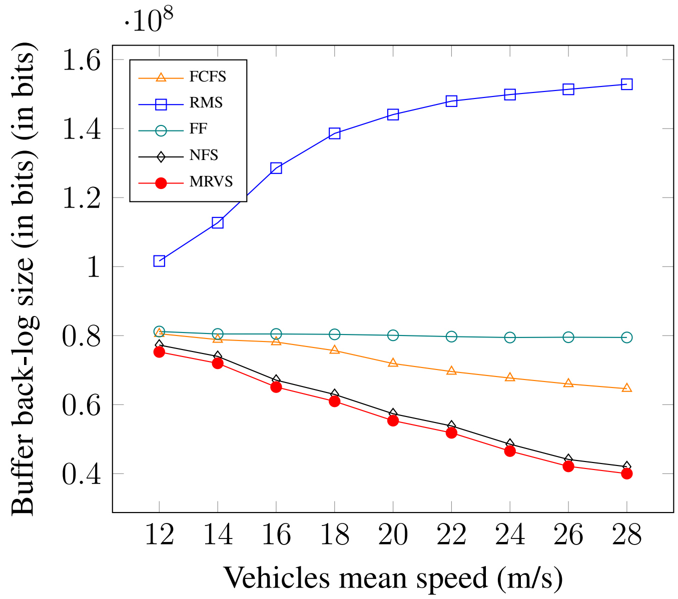


Figure 5.6: Vehicle speeds and buffer back-log size (in bits) when vehicle arrival rate λ=1 and task arrival rate=1000 are constant.

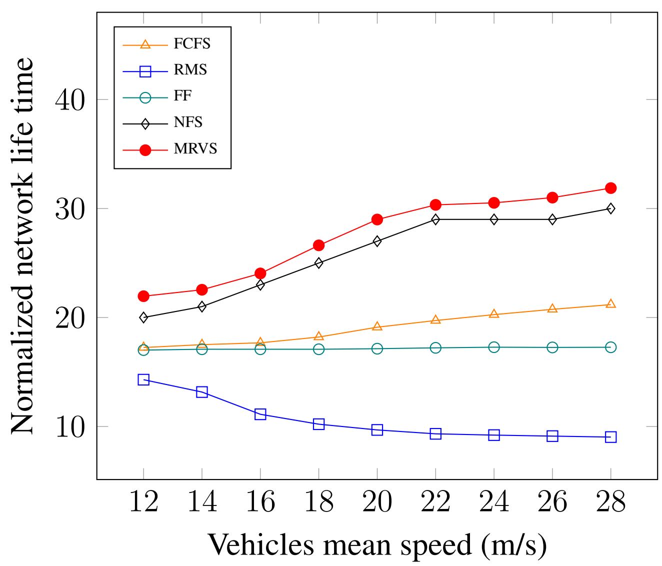


Figure 5.7: Vehicle speeds and normalized network life time when vehicle arrival rate λ=1 and task arrival rate=1000 are constant.

FF and *FCFS* algorithms, except for *RMS*. This is because, at higher vehicle speeds, more number of vehicles can reach the destination RSU by satisfying task deadline $d_i = 300sec$. Consequently, there exists more number of relay vehicles in the nearby source RSU. Therefore, source RSU transmits with higher data rates to nearby relay vehicles. Thereby the buffer back-log sizes reduces at higher vehicle speeds. In contrast, *RMS* selects farthest

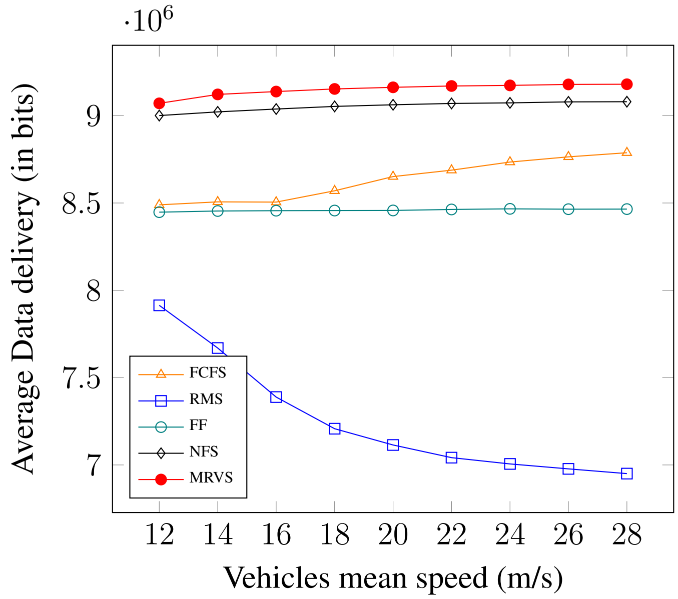


Figure 5.8: Vehicle speeds and data delivery when vehicle arrival rate λ=1 and task arrival rate=1000 are constant.

relay vehicle at the end of RSU coverage that leaves RSU first. It leads to increase of average buffer back-log size per time slot. However, the proposed *MRVS* outperforms other four algorithms. The average buffer back-log sizes of *MRVS* decreases by 29.88%, 44.67% and 1.6 times when compared with *FCFS*, *FF* and *RMS* respectively. Fig. 5.7 shows the relative comparison of algorithms in terms of normalized network life time for different mean vehicle speeds. Since the buffer back-log sizes are less at high vehicle speeds, the amount of energy required by source RSU to transmit task data also reduces tremendously. As a result, the network life time increases with the increasing vehicle speeds, except for *RMS*. The normalized network life time for *MRVS* raises by 32.14%, 39.28% and 67.85% when compared to *FCFS*, *FF* and *RMS*, respectively. Furthermore, Fig. 5.8 shows the comparison of relative performance of algorithms in terms of average data delivery to the destination RSU for varying speeds of vehicles. It is observed that the data delivery of *MRVS* is higher by 5.57%, 7.69% and 22.35% when compared to *FCFS*, *FF* and *RMS*, respectively.

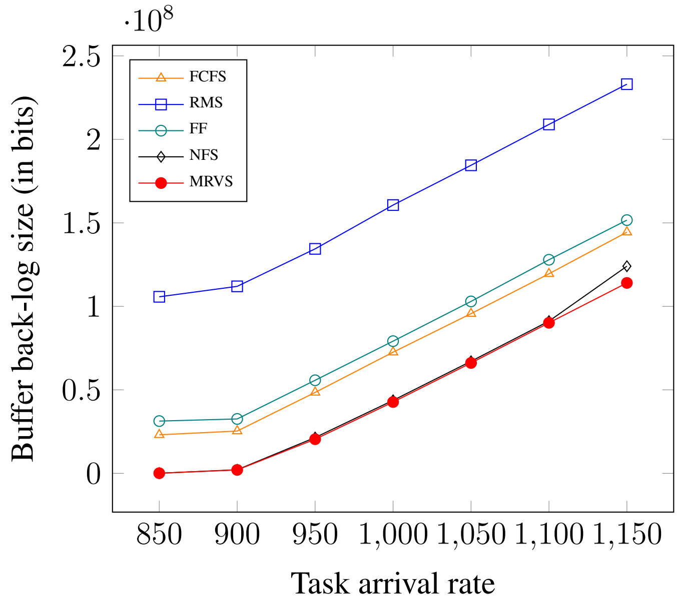


Figure 5.9: Task arrival rate and buffer back-log size when vehicle arrival rate λ=1 and mean vehicle speed=20 m/s are constant.

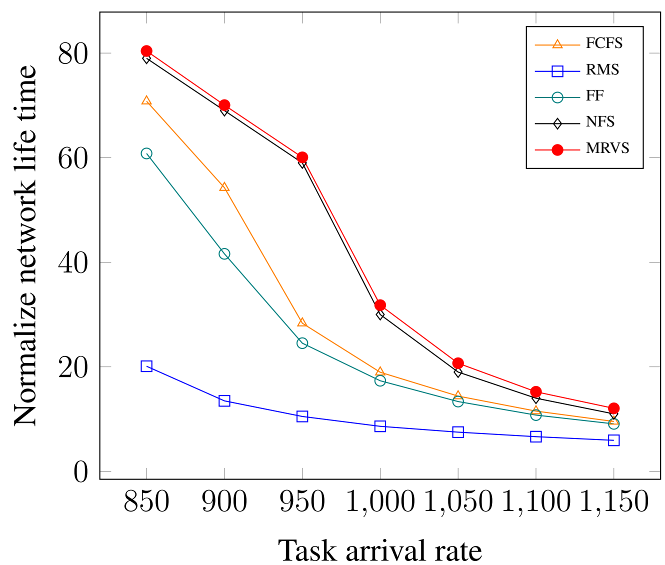


Figure 5.10: Task arrival rate and normalized network life time when vehicle arrival rate λ=1 and mean vehicle speed=20 m/s are constant.

5.3.2.3 Effect of task arrival rate

Figure 5.9 illustrates the performance comparison of *MRVS*, *NFS*, *RMS*, *FF* and *FCFS* algorithms in terms of buffer back-log size by varying task arrival rates. The results are obtained by simulation with mean task arrival rates from 850 to 1150, and keeping the vehicle

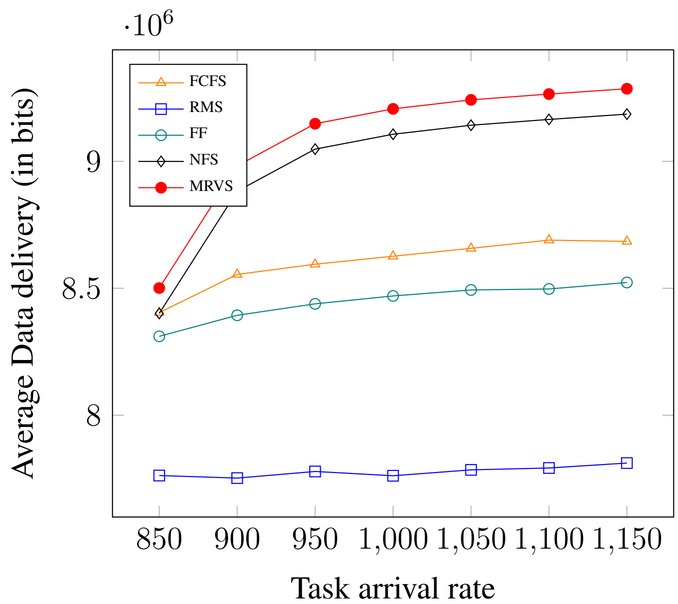


Figure 5.11: Task arrival rate and data delivery when vehicle arrival rate λ =1 and mean vehicle speed=20 m/s are constant.

arrival rate and mean vehicle speed are constant at $\lambda = 1$ and 20 m/s, respectively. From Fig. 5.9, it is clear that the increasing task arrival rates leads to increasing of average buffer back-log sizes. This is because, the power allocation algorithm (LDPA) is constrained by maximum transmit power (P^{tx}_{max}). Consequently, transmission rates can not grow beyond certain threshold, as a result, buffer back-log size increases at higher task arrival rates. However, the proposed *MRVS* outperforms other four algorithms since it chooses the relay vehicles with maximum possible transmission rate. The average buffer back-log size for *MRVS* reduces by 69.91%, 85.4% and 2.76 times when compared to *FCFS*, *FF* and *RMS*, respectively. Figure 5.10 illustrates the comparison of these algorithms in terms of network life time for different arrival rates of tasks. When task arrival rate is low, the accumulation of task data in the buffers also low. Then, the allocation of transmit power is less, and source RSU conserves more energy for its future operations. The network life time reduces with the increasing task arrival rates, because of growing buffer back-log sizes. However, the proposed *MRVS* outperforms the other algorithms, and shows nearly 41%, 45% and 74% improvement over *FCFS*, *FF* and *RMS*, respectively. Furthermore, Fig. 5.11 shows the data delivery performance of algorithms for different task arrival rates. It is observed that the data delivery increases with the increase of task arrival rates. This is because, as the buffer sizes grows with the task arrival rate, the data transmission rates also increase to keep buffers stable. Therefore, the data delivery is high at increasing task arrival rates. However, the proposed *MRVS* outperforms other algorithms, and its data delivery performance is higher by nearly 6%, 8% and 15% when compared to other algorithms *FCFS*, *FF* and *RMS* , respectively.

5.4 Summary

In this chapter, an online relay scheduling algorithm has been presented for energy efficient sharing of data in between neighbouring RSUs. Specifically, a dynamic power allocation problem is formulated to determine data rates of a transmission channel when assigned to a relay vehicle in the RSU region. Edge weights are calculated from data rates and backlog sizes of buffers. Further, a relay scheduling problem is formulated to select a set of relay vehicles (with maximum edge weights) which satisfy task deadlines. As a solution, this work realizes a Lyapunov optimization technique for minimization of RSU energy consumption subject to buffer stability, and transmission of buffer content from RSU to a set of scheduled relay vehicles via I2V communication. Finally, a simulation study is conducted to demonstrate the performance of proposed algorithms in-terms of buffering and scheduling performance. It is observed that the proposed strategy shows significant improvement in terms of buffer stability, network life time and average data delivery to destination RSU.

In next chapter, to reduce the data delivery delay between neighbouring RSUs, a vehicular fog computing framework is presented for the execution of tasks in fog vehicles within tolerable response times while minimizing the energy consumption of RSU.

Chapter 6

Fuzzy Reinforcement Learning for Energy Efficient Task Offloading in Vehicular Fog Computing

Vehicular Fog Computing (VFC)[21] extends fog computing to conventional vehicular networks, where the vehicles act as mobile fog nodes which supports full utilization of computation resources. Fig.6.1 illustrates the VFC environment in rural highways that are close to smart villages. The efficient task offloading in VFC has few challenges that need to be addressed. First, high mobility, short connection time and heterogeneity of vehicles make difficult for smart devices to directly offload tasks to fog vehicles. Although vehicles are equally functional in VFC, there is a need for centralize infrastructure (e.g., RSU) to perform scheduling and task allocation and use vehicles exclusively for task execution. Second, selection of potential fog vehicles for offloading is important since the vehicles with long staying period in communication coverage may be busy in execution of other tasks while the vehicles with available resources may ready to leave communication coverage. Third, few works in the literature [141, 142] aims to minimise response time. Optimization of response time has been considered as an indicator for performance evaluation in VFC. However, it is realized that the minimization of energy consumption of centralize infrastructure is equally important along with response time since the allocation of tasks to fog vehicles involve both communication cost of infrastructure and computation delay of fog

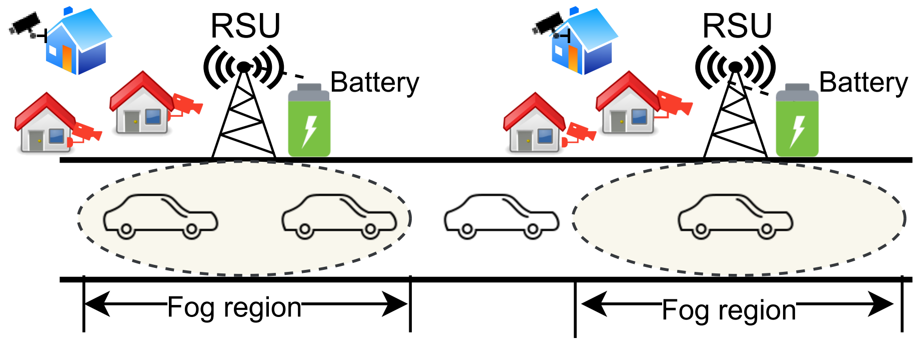


Figure 6.1: Vehicular Fog Computing (VFC) in rural highways near smart villages

vehicles. Fourth, computation overhead in determining potential fog vehicles may degrade quality of experience of smart devices and the overall performance of VFC. To achieve real-time selection of fog vehicles, it is required to follow some learned patterns caused by heterogeneity of vehicles in dynamic vehicular fog computing environment. Therefore, real-time allocation of tasks to fog vehicles improve user experience as well as minimize energy consumption of RSU within tolerable response latency.

To address the above mentioned issues, a latency-aware energy efficient scheduling of tasks to fog vehicles in VFC is presented. The major contributions of this chapter is described as follows.

- Present a Vehicular Fog Computing (VFC) framework for efficient offloading of tasks generated by real-time applications running in smart cities close to rural highways.

- Formulate an optimization problem as Integer Linear Programming Problem (ILP) which aims to minimise the communication and computation cost of RSU for efficient task allocation among fog vehicles while satisfying constraints on task deadline and resource availability.

- Propose a Fuzzy Reinforcement Learning (FRL) approach for energy efficient allocation of tasks to fog vehicles. A Fuzzy logic based greedy heuristic is combined with an on-policy reinforcement learning (i.e., SARSA[143]) that not only accelerate the learning process but also improves the selection of potential fog vehicles for decreasing total energy consumption and average response time.

- Extensive simulations are performed to evaluate the proposed scheduling approach along with other scheduling heuristics in terms of various performance metrics such as RSU energy consumption, total task service time, average response time and percentage of tasks processed locally at RSU.

The remaining sections of this chapter are organized as follows. Section 6.1 presents a frame work for vehicular for computing in the context of smart villages in rural highways. Section 6.2 illustrates system model and formulation of optimization problem. Section 6.3 presents a reinforcement leaning solution combined with a greedy heuristic. Section 6.4 describes experimental evaluation of scheduling algorithms. Section 6.5 summaries this chapter.

6.1 Vehicular Fog Computing (VFC) Framework

6.1.1 Preliminaries

Smart home devices: The smart home devices are the terminal devices (e.g., IoT devices, smart phones, smart cameras, PDAs etc.) with limited computation capabilities. Tasks generated by the smart home devices cannot be handled efficiently due to inherent limitations of these devices. In this case, it is necessary to offload the tasks to nearby nodes with adequate computation resources. Because of high mobility and short connection time between smart home devices and vehicles, it is unrealistic to offload tasks directly to computation enriched vehicles (i.e., fog vehicles). Therefore, the smart home devices tend to offload their computation-intensive tasks to nearby Road Side Units (RSU) where the RSU is responsible for the efficient allocation of tasks to fog vehicles at the edge of the network.

Tasks: Broadly speaking, the task refers to a basic unit of service requirement of an application running in a smart home device. Each task is defined by three parameters such as task data size, computation demand and tolerable response latency (i.e., deadline). A task cannot be subdivided into smaller sub tasks. When smart home devices offload computationally intensive tasks to third party (i.e., RSU) for execution, the tasks are buffered at RSU before allocating efficiently over the vehicular fog nodes.

Fog nodes: Two types of fog nodes are assumed in this framework.

- *Stationary fog nodes:* Road Side Unit is considered as the stationary fog node. The RSU is equipped with re-chargeable batteries powered by alternative sources such as wind, solar, etc. Moreover, the RSU is endowed with an edge server which incurs computation cost while executing tasks with short deadlines. To execute tasks without short dead lines, the RSU depends on computation resources of mobile vehicles, consequently the RSU incurs communication cost for task assignment to fog vehicles in the RSU region. Therefore, the RSU acts as an intermediary between the smart home devices and mobile vehicles for efficient computation of tasks within specific deadlines.

- *Mobile fog nodes:* Mobile vehicles are considered as the mobile fog nodes on road. These mobile vehicles have adequate computation capabilities for task processing, and possess on-board communication (i.e., Dedicated Short Range Communication (DSRC)) module in order to ensure communication to nearby RSU. The mobile vehicles have sufficient energy reserves to support computation requirements of the tasks generated by the smart home devices. For better representation, mobile fog nodes are termed as *fog vehicles* or *vehicles* in the rest of this chapter.

6.1.2 Process of task allocation

The task allocation process consists of following steps.

Identification of fog vehicles: It is assumed that the time is divided into time slots of equal length. At the beginning each time slot, the RSU broadcasts a probe message over DSRC to the vehicles in its service region and collects response from the interested vehicles. Based on the received responses, the RSU updates the candidate list of fog vehicles in each time slot.

Assignment of tasks to fog vehicles: In each time slot, the RSU receives tasks which are independent and identically distributed (i.i.d) with a fixed mean arrival rate. A scheduling algorithm runs in RSU to decide each task whether it is assigned to edge server or fog vehicle. This task assignment needs to satisfy the deadline and resource constraints of

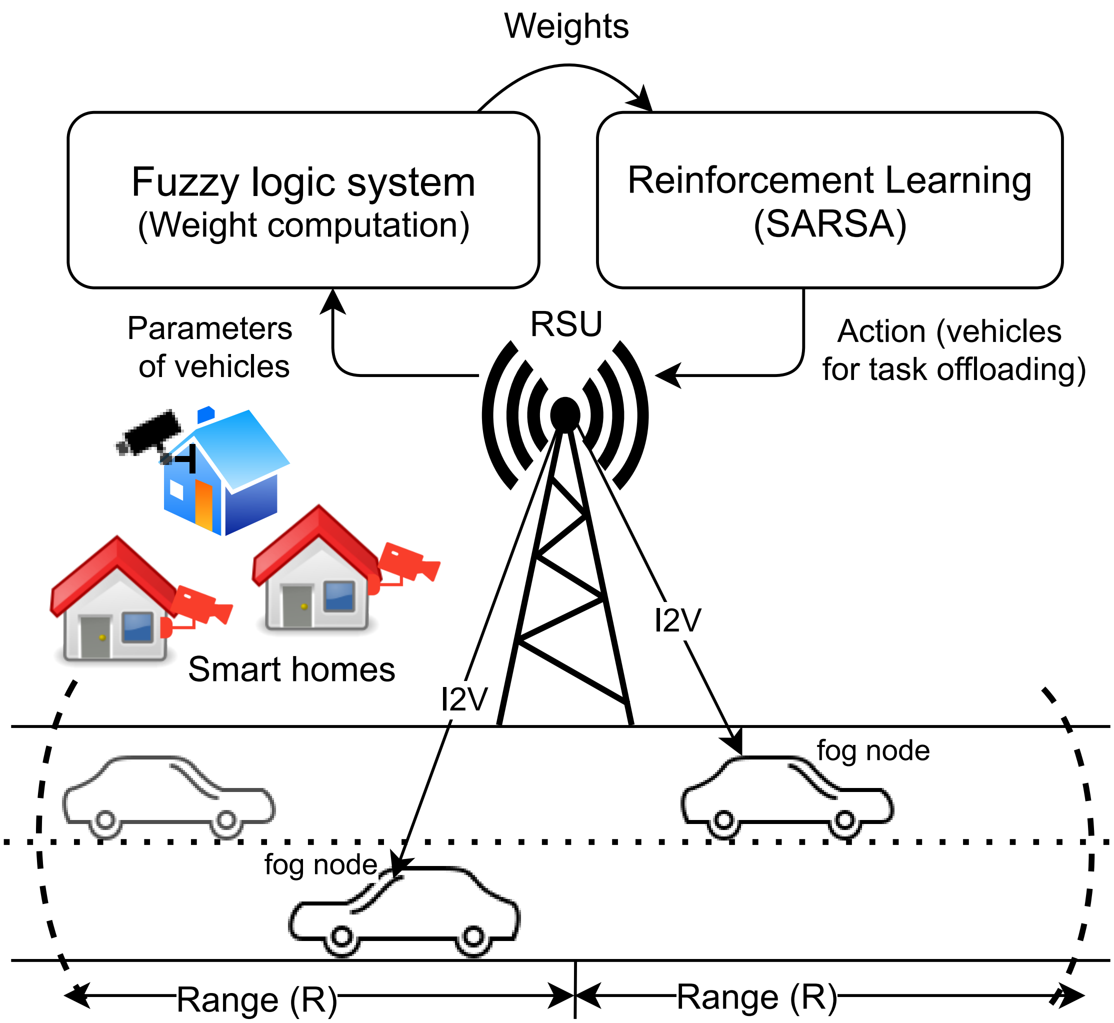


Figure 6.2: VFC architecture for offloading of tasks to fog vehicles

tasks and vehicles, respectively. Based on the arrival sequence of tasks that are awaiting in each time slot, the RSU assigns the tasks for execution with an objective to minimize the energy consumption cost of RSU. The incurred energy consumption cost of RSU includes communication cost when task assigned to fog vehicle, and computation cost when task assigned to edge server. The task allocation process is repeated in every time slot for the overall minimization of energy consumption and response time.

6.2 System model and Problem formulation

This section presents communication and computation model in section 6.2.1, and formulation of the problem as an Integer Linear Programming problem in section 6.2.2.

6.2.1 System model

The system considers an energy limited RSU endowed with an edge server which is deployed near highway road segment as shown in Fig.6.2. The RSU acts as a stationary fog node for the execution of tasks and the assignment of tasks to fog vehicles. The system time is divided into fixed length time slots of set $\mathcal{K}$. In each time slot $k \in \mathcal{K}$, the RSU observes a set of buffered tasks $\mathcal{I}$ for execution, and identifies a set of fog vehicles $\mathcal{J}$ within the RSU's communication range R. Each task $i \in \mathcal{I}$ is described as (H_i, C_i, τ_i). H_i represents the amount of ith task data (in bits). C_i represents the task requirement (in CPU cycles) for the execution of task i. τ_i denotes deadline (in seconds) of task i. The number of orthogonal downlink (from RSU to fog vehicle) channels with equal bandwidth is assumed as N.

In each time slot, two kinds of execution scenarios exist in this system model. Firstly, the RSU assigns the tasks to fog vehicles for execution subject to satisfy deadlines of tasks, vehicle staying period and availability of computation resources of fog vehicles. Secondly, if either buffering time of a task is more than its deadline or the computation resources of vehicles are not sufficient for task execution then that particular task can be processed locally at the edge server with RSU. Notations used in this chapter are described in Table 6.1.

6.2.1.1 Communication model

The vehicles with available computational resources periodically disseminates beacon messages. A beacon message of a vehicle j is described as $(v_j, \mu_j, S_j^{avail})$ where v_j is velocity, μ_j is processing rate in number of CPU cycles per second and S_j^{avail} is the available resources of vehicle j. It is assumed that the velocity of vehicles is constant inside RSU communication range as the vehicles tend to move without changing their speed for shorter distances in highway scenarios[23]. However, the RSU receives beacon messages of vehi-

Table 6.1: Abbreviations and Notations

Notation	**Description**
$k, \mathcal{K}$	time slot k in set of time slots $\mathcal{K}$
$i, \mathcal{I}$	task i in set of tasks $\mathcal{I}$
$j, \mathcal{J}$	vehicle j in set of fog vehicles $\mathcal{J}$
H_i	i^{th} task data (in bits)
C_i	i^{th} task requirement (in cycles)
τ_i	deadline of task i
N	number of vehicles in action space
R	radio coverage range of RSU
v_j	velocity of vehicle j
μ_j	processing rate of vehicle j
S_j^{avail}	resources available at vehicle j
D_j^k	distance from RSU to vehicle j at time k
$\mathcal{E}_{ijk}^{Comm}$	communication cost of RSU when task i is assigned to vehicle j at time k
$\mathcal{E}_i^{Proc}$	processing cost of RSU when task i executed locally
$\mathcal{E}(k)$	sum of communication and computation costs
x_i^k	indicator variable to denote task i arrived to RSU at time k
x_j^k	indicator variable to denote vehicle j available in RSU region at time k
x_{ij}^k	indicator variable to denote the assignment of tasks to fog vehicles at time k
M_j^k	processing rate of vehicle j at time slot k
T_j^k	dwell time of vehicle j at time slot k
I2V	infrastructure to vehicle communication
RSU	Road Side Unit

cles and updates candidate list of fog vehicles at the beginning of each time slot. The RSU observes arrival time of first beacon as arrival time instance A_j of vehicle j, and calculates the distance D_j^k between RSU to vehicle j at the beginning of each time slot k as follows,

$$D_j^k = R - (\delta t\, k - A_j)\, v_j \tag{6.1}$$

where δt is length of time slot, D_j^k value is negative when vehicle j is in arrival side of RSU and positive when vehicle j is in leaving side of RSU.

Furthermore, this system adopts a distance dependent path loss communication model[72] for the transmission of ith task data from RSU to a vehicle j. For a given bit rate (B) in each time slot k, the energy consumption of RSU is represented as,

$$\mathcal{E}_{ijk}^{Comm} = \frac{B}{\psi}(D_j^k)^{\beta},\ \forall j\ \forall k \tag{6.2}$$

where ψ represents scaling co-efficient and β denotes path loss constant.

6.2.1.2 Computation model

Let the computation capacity of edge server with RSU in-terms of number of CPU-cycles per second denoted as F_{rsu}. According to [144], the energy consumption for execution of one CPU-cycle is considered as $\kappa\, F_{rsu}^2$, where κ is a coefficient and depends on the switched capacitance of chip architecture. If RSU fails to offload tasks to fog vehicles either due to unavailability of fog vehicles or fog resources, then those tasks will be processed locally at RSU. For local processing, the RSU choose to execute each task i with minimum resource requirement (C_i) in order to minimize the energy consumption of RSU. Moreover, the RSU is required to process the task locally before deadline of that task exceeds. This local execution is possible only if the task satisfies deadline constraint $\frac{C_i}{F_{rsu}} \le \tau_i$, otherwise the RSU drops the task. However, the computation model and communication model works independently based on the offloading decision of each task. Suppose, a task i decides local processing in the RSU's edge server, then the energy consumption of RSU is,

$$\mathcal{E}_i^{Proc} = \kappa\, F_{rsu}^2\, C_i\ \forall i \tag{6.3}$$

In this case, the energy consumption cost of fog vehicles has been ignored in the model, as the vehicles do not depend on battery power.

6.2.2 Problem formulation

An indicator variable x_i^k is defined to denote whether a task i is received by RSU at time slot k, and another indicator variable x_j^k is defined to denote whether a vehicle j is present in the RSU radio coverage at time slot k.

$$x_i^k = \begin{cases} 1, & \text{If } i \in \mathcal{I} \text{ at time k} \\ 0, & \text{Otherwise} \end{cases} \qquad x_j^k = \begin{cases} 1, & \text{If } j \in \mathcal{J} \text{ at time k} \\ 0, & \text{Otherwise} \end{cases} \tag{6.4}$$

Moreover, an indicator variable x_{ij}^k is defined to denote whether a task i is assigned to a vehicle j at a time slot k. The assignment is possible only if the vehicle j is present in RSU region and task i arrives to RSU in the same time slot k. Therefore,

$$x_{ij}^k \leq min(x_i^k, x_j^k),\ \forall i,\ \forall j,\ \forall k \tag{6.5}$$

where x_{ij}^k is bounded by minimum of x_i^k and x_j^k. That means, x_{ij}^k is equal to 0 if either task i or vehicle j, or both are not present with RSU at time slot k. Further, x_{ij}^k is equal to 1 only if both task i and vehicle j are available at RSU, and the task i is assigned to vehicle j at time k.

Total energy consumption of RSU in a time slot k is defined as the sum of communication cost of RSU to vehicles and computation cost of tasks at edge server, and is given as

$$\mathcal{E}(k) = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} x_{ij}^k\ \mathcal{E}_{ijk}^{Comm} + \sum_{i \in \mathcal{I}} 1_{\{\sum_j x_{ij}^k = 0\}}\ \mathcal{E}_i^{Proc} \tag{6.6}$$

where $1_{\{Z\}}$ is a binary function which is equals to 1 if function Z is true, otherwise equals to 0.

Accordingly, the formulation of the proposed problem is described as follows,

$$\mathcal{P}_3 : \underset{x_{ij}^k}{\text{minimize}} \sum_{k \in \mathcal{K}} \mathcal{E}(k)$$

$$\text{s.t.} \sum_{j \in \mathcal{J}} \left(\frac{H_i}{B} + \frac{C_i}{\mu_j} \right) x_{ij}^k \leq \tau_i, \ \forall i, \forall k \tag{6.7}$$

$$\sum_{j \in \mathcal{J}} \left(\frac{R - D_j^k}{v_j} \right) x_{ij}^k \geq \tau_i, \ \forall i, \forall k \tag{6.8}$$

$$\sum_{i' \in \mathcal{I}, i' \neq i} C_{i'} + C_i \, x_{ij}^k \leq S_j^{avail}, \ \forall j, \forall k \tag{6.9}$$

$$\sum_{j \in \mathcal{J}} x_{ij}^k \leq 1, \ \forall i, \forall k \tag{6.10}$$

$$\sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} x_{ij}^k \leq N, \forall k \tag{6.11}$$

$$|D_j^k| \leq R, \ \forall j, \forall k \tag{6.12}$$

$$x_{ij}^k \leq min(x_i^k, x_j^k), \ \forall i, \ \forall j, \ \forall k \tag{6.13}$$

$$x_{ij}^k \in \{0, 1\}, \ \forall i, \forall j, \forall k \tag{6.14}$$

The objective function of problem $\mathcal{P}_3$ represents the minimization of total energy consumption (includes communication cost and processing cost) of RSU in all time slots. Constraint (6.7) ensures the sum of communication time and computation time of task i into vehicle j must not be more than the assigned task deadline τ_i. Constraint (6.8) ensures the staying period of vehicle j in RSU coverage should not be less than the assigned task deadline. Constraint (6.9) ensures total resource requirement of assigned tasks should not be more than the resources available with vehicle j. Constraint (6.10) denotes the assignment of each task to at most one vehicle. Constraint (6.11) ensures number of one-to-one assignments of tasks to vehicles is less than or equal to number of wireless channels. Constraint (6.12) ensures a fog vehicle j is within the RSU coverage R. Further, (6.13) and (6.14) are the integer constraints.

Suppose there exists a case that the arrived tasks in every time slot are without short deadlines, the RSU decides to allocate all tasks to fog vehicles. To obtain the optimal allocation, the RSU has to perform a naive assignment of N tasks over a set of $\mathcal{J}$ fog vehicles by using exhaustive search technique with a time complexity $O(|\mathcal{J}|^N)$. It is difficult to attain optimal solution when number of fog vehicles and tasks are large. Therefore, a fuzzy logic based heuristic is proposed in order to obtain a sub-optimal solution for the problem $\mathcal{P}_3$. Further, the obtained solution is enhanced using reinforcement learning approach as discussed in section 6.3.

6.3 Fuzzy Reinforcement Learning Approach

In this approach, Reinforcement Learning (RL)[145] agent learns an optimal scheduling policy for the assignment of tasks to fog vehicles. The RL agent observes set of actions in every system state. Then, the agent explores all possible actions greedily until it exploits the best set of actions in order to maximize the log term reward. In this problem, states and actions are represented as time slots and fog vehicles, respectively. In addition, the reward is considered as a function of service time and energy consumption of RSU. However, the RL agent can not identify vehicles directly as fixed set of actions in this dynamic scenario. Rather, the agent is trained to act on the weights of vehicles as actions. That means, an action is a set of sub set of vehicles and it is identified by vehicle weights but not the vehicle identities. Therefore, this section presents a fuzzy logic system for calculation of vehicle weights based on the parameters such as processing rate of vehicle, staying period (i.e. Dwell time) in RSU and distance to RSU. Then, the selection of maximum weighted vehicle set acts as current policy to the RL agent. Consequently, an *on-policy* version of reinforcement learning system converges towards the selection of best possible set of fog vehicles with maximum reward. Since the problem is a minimization problem, the reward is considered as the inverse function of service time and energy consumption cost of RSU in order to maximize the log term reward.

6.3.1 Calculation of vehicle weights using fuzzy logic system

As aforementioned, the RSU receives beacon messages disseminated by the vehicles and maintains candidate list of fog vehicles in each time slot. In addition, the RSU collects the following information from the vehicles residing in its coverage at the beginning of every time slot k.

1. η^k assumes fixed number of candidate fog vehicles in the RSU region. If number of in-range vehicles are less than η^k, then it is required to add dummy (zero weighted) vehicles to reach the number of vehicles equal to η^k. On the other hand, when number of in-range vehicles more than η^k, then system ignores some vehicles which have least computational capacity. Here, η^k is constant for all k.

2. $M^k = \{M_1^k, M_2^k, ..., M_j^k, ..., M_{\eta^k}^k\}$ is a vector of processing rates of vehicles, where $M_j^k = \mu_j$.

3. $T^k = \{T_1^k, T_2^k, ..., T_j^k, ..., T_{\eta^k}^k\}$ is a vector containing dwell times of vehicles in RSU coverage. The dwell time is defined as the staying time or remaining travel time of a vehicle inside the RSU. Dwell time of vehicle j is defined as $T_j^k = \frac{R - D_j^k}{v_j}$.

4. $D^k = \{|D_1^k|, |D_2^k|, ..., |D_j^k|, ..., |D_{\eta^k}^k|\}$ is a vector of distances between vehicles to RSU. D_j^k can be obtained from the Eq.6.1.

Further, the proposed vehicle weight calculation method uses Fuzzy logic system [146] which takes the input parameters processing rate, dwell time and distance to RSU. It is required to normalize these input values in the interval (0,1). Let $\widehat{M_j^k}$ is the normalized processing rate of a vehicle j in a time slot k, which is defined as,

$$0 \leq \widehat{M_j^k} = \frac{M_j^k - min_j\ M^k}{max_j\ M^k - min_j\ M^k} \leq 1 \tag{6.15}$$

Also, the normalized dwell time ($\widehat{T_j^k}$) of a vehicle j and distance ($\widehat{D_j^k}$) to RSU and they are defined as,

$$0 \leq \widehat{T_j^k} = \frac{T_j^k}{max_j\ T^k} \leq 1 \tag{6.16}$$

$$0 \leq \widehat{D_j^k} = \frac{|D_j^k|}{R} \leq 1 \tag{6.17}$$

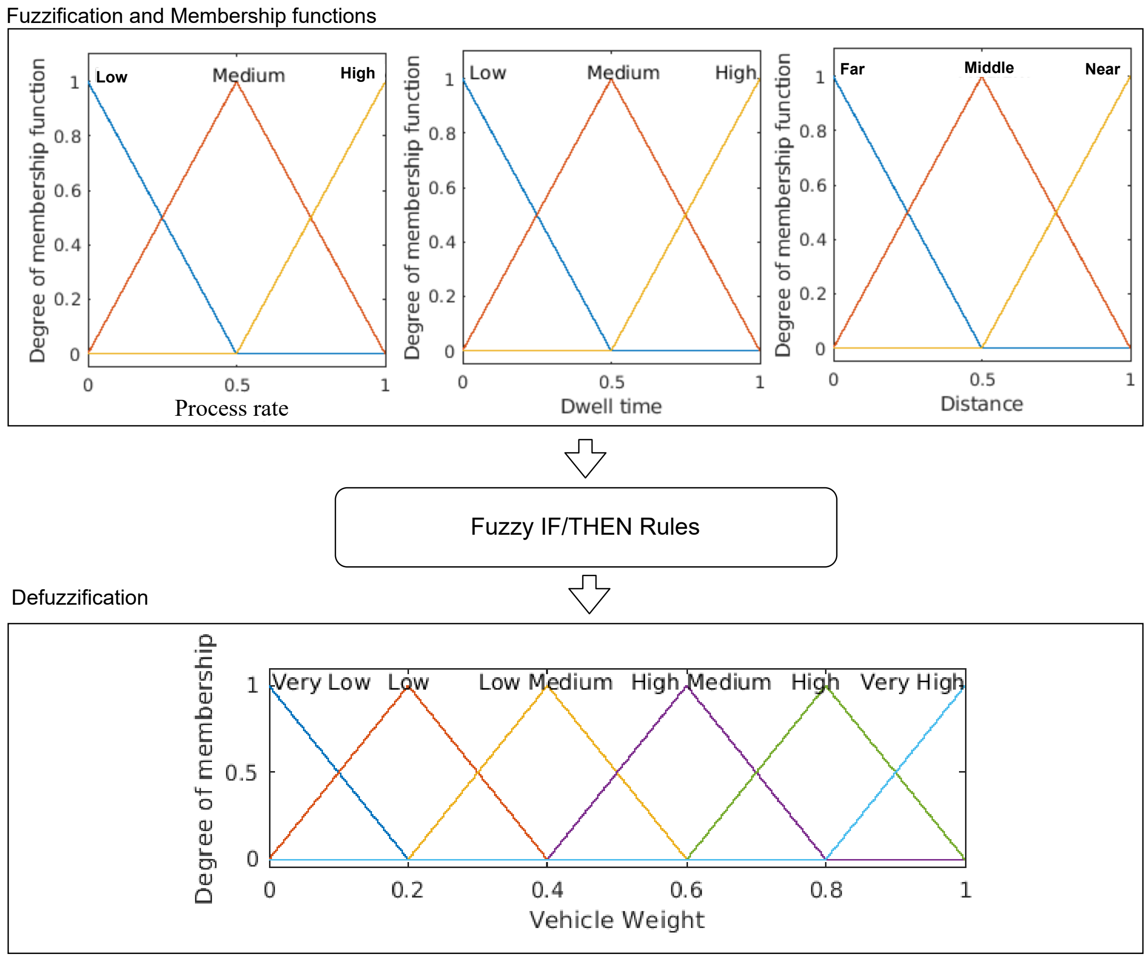


Figure 6.3: Fuzzy logic system for calculation of vehicle weights

The vehicle weight calculation using Fuzzy logic system has been shown in Fig. 6.3, which is Mamdani fuzzy inference system represented as of four stages: 1) fuzzification 2) membership functions 3) fuzzy rule base 4) defuzzification. In fuzzification, the three quantifiable input parameters (crisp values) are converted to linguistic variables represented as $\{Low, Medium, High\}$, $\{Low, Medium, High\}$, and $\{Far, Middle, Near\}$, respectively. The membership functions give the degree of membership for the input values. Further, the fuzzy rule base i.e., IF/THEN rules (in table 6.2) are defined to calculate the vehicle weight (linguistic variable). The linguistic variables for the vehicle's weight are represented as $\{Very\ Low, Low,\ Low\ Medium,\ High\ Medium, Very\ High\}$. Finally, the defuzzification uses centroid method in order to calculate the vehicles' weights (crisp value).

Table 6.2: Fuzzy IF/Then rule base

S.No	**Input Parameters**			**Output**
	Process rate	**Dwell time**	**Distance**	**Vehicle Weight**
1	High	High	Near	Very High
2	High	High	Middle	High
3	High	High	Far	Low Medium
4	High	Medium	Near	Very High
5	High	Medium	Middle	High Medium
6	High	Medium	Far	Low Medium
7	High	Low	Near	High
8	High	Low	Middle	Low Medium
9	High	Low	Far	Low
10	Medium	High	Near	High
11	Medium	High	Middle	High Medium
12	Medium	High	Far	Low Medium
13	Medium	Medium	Near	High
14	Medium	Medium	Middle	High Medium
15	Medium	Medium	Far	Low Medium
16	Medium	Low	Near	High
17	Medium	Low	Middle	High Medium
18	Medium	Low	Far	Low
19	Low	High	Near	High Medium
20	Low	High	Middle	Low Medium
21	Low	High	Far	Low
22	Low	Medium	Near	High Medium
23	Low	Medium	Middle	Low Medium
24	Low	Medium	Far	Low
25	Low	Low	Near	Low Medium
26	Low	Low	Middle	Low
27	Low	Low	Far	Very Low

6.3.2 Scheduling of fog vehicles using Reinforcement Learning

The Reinforcement Learning allows an agent to learn an optimal vehicle scheduling policy while interacting with its environment. The RL system can be modeled as a Markov Decision Process (MDP). In this problem, the MDP over finite horizon $\mathbb{N} = |\mathcal{K}|$ is composed of a set $(\mathcal{X}, \mathcal{A}, r^k, \phi, \gamma)$ defined as follows:

1. *State space:* $\mathcal{X}$ is a system state-space, where a state $p^k \in \mathcal{X}$ and a time step $k = \{1, 2, .., \mathbb{N}\}$.

2. *Action space:* $\mathcal{A} = \{a_1, a_2, ..., a_z\}$ as a set of actions at time step k. Here, each action denotes a set of fog vehicles over which tasks are being offloaded for execution. The number of fog vehicles in every action equals to number of wireless channels N. Therefore, combination of N fog vehicles is selected from η^k, and then $z = \binom{\eta^k}{N}$. Since η^k and N are constants, the number of actions are constant in every state. The actions in set $\mathcal{A}$ are identified uniquely over all time steps. More clearly, vehicles in action a_z at current time step k may be different from vehicles in same action a_z at next time step $k+1$, but a_z can be chosen by RL agent based on the current policy (i.e, max-weight policy) of the agent as given in algorithm 6.1. Notably, RL agent choose a_z if a subset of vehicles with highest weight is selected. These vehicle weights are derived from the fuzzy logic system which is discussed in section 6.3.1.

3. *Reward:* r^k is a single step immediate reward at time step k, which is obtained from the learning environment when an action a^k is taken in the current state p^k, denoted as $r^k(p^k, a^k)$. In our problem, the reward of an action (a subset of vehicles) is taken as an inverse function of energy consumption cost of RSU ($\mathcal{E}(k)$) and average service time (ST_k) of assigned tasks in current time step k. Therefore, RL agent strives to conserve RSU energy and quick execution of tasks. Then, it prefers to choose subset of vehicles (action) close to RSU which have high computation resources. This is because, vehicles nearby RSU consume less energy when compared to more distant vehicles in the RSU coverage. Therefore,

$$r^k(p^k, a^k) = \frac{1}{f(\mathcal{E}(k), Avg(ST_k))}, \forall k \tag{6.18}$$

where f(.) is a normalized average of energy consumption and service time $Avg(ST_k) = \{\frac{\sum_{i=1}^{N} C_i/\mu_j}{N}\}_{j \in a^k}$. The assignment of arrived tasks to fog vehicles in a^k is done on the basis of first in first out.

4. ϕ is the transition probability from current sate p^k to next state p^{k+1} provided that the action a^k is taken in the current state. Here, ϕ is chosen as equal to one, when RL agent moves to next time step $k+1$ from current time step k.

Algorithm 6.1 Max-weight policy of SARSA's Learning agent

Input: η^k vehicles and their weights
Output: Action set $\mathcal{A} = \{a_1, a_2, ..., a_z\}$ where $z = \binom{\eta^k}{N}$
1: Let V be a set of vehicles in RSU region, and $|V| = \eta^k$
2: Find power set $\mathbb{P}(V)$
3: Find a set $A' = \{a| \ a \in \mathbb{P}(V)$ and $|a| = N\}$
4: Find action a_z = Max(A') such that $w(a_1) \leq w(a_2) \leq ... \leq w(a_n) \leq ... \leq w(a_z)$, where $w(a_z)$ is the total weight of all vehicles in set a_z calculated from section 6.3.1.
5: **return** a_z

5. γ is a discount factor that can take value in a range (0,1).

At time step k, the aim of agent is to take best action a^k in current state p^k which maximizes its reward. Furthermore, when the agent follows a policy π by observing an immediate reward $r^k(p^k, a^k)$ and next state p^{k+1}, then the agent is able to adjust its policy π towards optimal policy. In MDP, the main goal of agent is to find an optimal policy $\pi^* : \mathcal{X} \to \mathcal{A}$ that will maximize overall reward of the system. A value function $V^\pi : \mathcal{X} \to \mathcal{R}$ gives the overall reward that is obtained by following agent's policy π when the agent starts from a state $p^k \in \mathcal{X}$. The value function V quantifies an expected value for the policy π throughout finite horizon $\mathbb{N}$, is the total sum of discounted rewards expressed as follows:

$$\mathcal{V}^\pi(p^k) = \mathbb{E}_\pi\Big[\sum_{k=0}^{\mathbb{N}} \gamma r^k(p^k, a^k) \mid p^0 = p^k\Big] \tag{6.19}$$

$$= \mathbb{E}_\pi\Big[r^k(p^k, a^k) + \gamma \mathcal{V}^\pi(p^{k+1}) \mid p^0 = p^k\Big] \tag{6.20}$$

Let Π be a set of all acceptable policies. Then, the optimal policy is given by

$$\pi^* = \underset{\pi \in \Pi}{\text{argmax}} \ \mathcal{V}^\pi(p^k) \tag{6.21}$$

Generally, RL has two popular learning techniques: 1) Q-learning 2) SARSA learning. The Q-learning is an off-policy algorithm where as SARSA learning follows on-policy approach. A Q-learning agent updates its policy based on the maximum reward (i.e., reinforcement signal) received from all possible actions, and that is independent of agent's

Algorithm 6.2 SARSA based Fuzzy Reinforcement Learning (FRL) algorithm

Initialize $Q(p^k, a^k)$ values arbitrarily
Initialize ϵ, α and γ
for every Episode **do**
  Initialize p^k as starting state
  Choose a^k from p^k using agent's policy derived from Algorithm. 6.1
  while p^k is not a terminal state **do**
    Take action a^k, observe reward r^k and next state p^{k+1}
    if random(0,1) $> \epsilon$ **then**
      Choose a_z from p^{k+1} using ϵ-greedy policy
    else
      Choose a_z from p^{k+1} using random policy
    end if
    Update $Q(p^k, a^k)$ value using Eq. 6.23
    Current state $p^k \leftarrow p^{k+1}$
    Current action $a^k \leftarrow a_z$
  end while
end for

policy. In contrast, SARSA learning updates the agent's policy directly from the actions taken, that is based on the applied policy. In this problem, the agent's policy is Max-weight policy given in Algorithm 6.1.

In Q-learning, an optimal Q-function $Q^*(p^k, a^k)$ is denoted as an approximation of $\mathcal{V}^\pi(p^k)$. Therefore, an optimal policy can be written as $\pi^* = \max_{a^k}\{Q^*(p^k, a^k)\}$. Here, the agent learns an optimal policy $\pi^* : \mathcal{X} \rightarrow \mathcal{A}$ and it starts by initializing estimates of Q-values $Q_k(p^k, a^k)$, where each state $p^k \in \mathcal{X}$ maps to its best action $a^k \in \mathcal{A}$. In the learning process, the agent first observes current state p^k and then takes an action a^k. Thereafter, it receives a reward $r^k(p^k, a^k)$ and a new state p^{k+1}, consequently it updates $Q_k(p^k, a^k)$ with respect to observed outcomes $r^k(p^k, a^k)$and p^{k+1}. The process is repeated at each time step until the agent learns optimal policy π^*. In particular, the estimated Q-values at each time step k are updated using learning rate α as follows,

$$\begin{aligned} Q_{k+1}(p^k, a^k) &= Q_k(p^k, a^k)+ \\ &\alpha \left[r_k(p^k, a^k) + \gamma \max_{a'} Q_k(p^{k+1}, a') - Q_k(p^k, a^k)\right] \end{aligned} \tag{6.22}$$

Furthermore, the SARSA is named as an acronym for State-Action-Reward-State-Action. SARSA learning is an online Reinforcement Learning algorithm. It has an advantage of selecting most optimal action in the new state p^{k+1} which lead to faster learning. The difference between Q-learning and SARSA learning is that the former can find optimal actions only after all the Q-values converge. Where as, the latter one can choose an optimal action in online fashion, without waiting to converge the algorithm. In particular, the estimated Q-values of SARSA at each time step k are updated using learning rate α as follows,

$$\begin{aligned} &Q_{k+1}(p^k, a^k) = Q_k(p^k, a^k) + \\ &\alpha \left[r_k(p^k, a^k) + \gamma \, Q_k(p^{k+1}, a_z) - Q_k(p^k, a^k) \right] \end{aligned} \tag{6.23}$$

where a_z is an action from Max-weight policy in Algorithm 6.1. The fuzzy based on-policy reinforcement learning algorithm is presented in Algorithm 6.2.

6.4 Experimental results and analysis

In this section, a set of simulation results is presented for different scheduling heuristics when compared to the proposed fuzzy reinforcement learning (FRL) scheduler. These algorithms use different scheduling strategies to select fog vehicles for assigning of tasks in real time. The performance of the proposed FRL approach is evaluated in terms of collected rewards when compared to Q-learning. Furthermore, performance of other scheduling algorithms along with FRL is evaluated using various parameters such as RSU energy consumption, total service time of tasks, percentage of tasks processed locally at RSU and average response time of tasks.

6.4.1 Experimental settings

Input vehicular traces for the highway environment have been considered from [121], where the vehicles tend to maintain constant speed in RSU coverage. The vehicle mobility model considered in this chapter is from [9, 18]. Due to stochastic arrival nature of vehicles, it is assumed the number of vehicle arrivals are Poisson independent process with

Table 6.3: Simulation parameters

Parameter	**Value**
Number of time slots	100
RSU radio range	1000m
Vehicle speeds	[15m/s, 25m/s]
Vehicle processing rate	[10, 100] cycles/sec
Task data size (H_i)	[60Mb, 100Mb]
Task requirement (C_i)	[400, 600] cycles/task
Total episodes	800
Learning rate (α)	0.05
Discount factor (γ)	0.95
Epsilon (ϵ)	0.1
Bit rate(B)	[3 Mb/s, 27 Mb/s]
Path loss constant (β)	3
Scaling co-efficient (ψ)	1

mean arrival rate λ. The speeds of the arrived vehicles in RSU coverage are distributed uniformly in a range $[v_{min}, v_{max}]$. In this kind of scenario, the estimates of RSU energy consumption costs are readily made in [117]. The associated energy costs can be derived from distance dependent path loss model with path loss constant $\beta = 3$. The downlink transmission energy costs are based on vehicle position. Moreover, due to high deployment cost, the RSUs are distributed sparsely in a highway segment with non-overlapping radio regions. The scheduling algorithms run in RSU to take decision on the selection of fog nodes. Each RSU maintains potential fog candidates present in its region when the RSU receives broadcast messages (contains vehicle identity, speed, processing rate and resources available) from fog vehicles. That means, same vehicle cannot be fog node for multiple RSUs at a time and the set of fog candidates belongs each RSU is distinct. Furthermore, the arrival of tasks follow Poisson process with a mean arrival rate which is more than λ. This is because, in a time slot usually the number of tasks to be assigned are more than the fog vehicles in RSU coverage. A set of parameters involved in this experimental settings is listed in a Table 6.3.

The simulations have been conducted in a desktop machine with a Central Processing Unit of 3.40GHz Intel core i7, Random Access Memory of 3.7 GiB and an Operating System of 64-bit Ubuntu 16.04 LTS. Python 3.5.6 programming language is used to develop the simulation environment. An intensive set of experiments has been performed for evaluating the proposed scheduling approach in comparison with other scheduling strategies under various parameters. Further, Monte Carlo simulations are performed for evaluation of each scheduling strategy and the comparison results are averaged over each time slot for 1000 simulation runs.

6.4.2 Scheduling approaches and performance metrics

The proposed a FRL approach uses an *on-policy* reinforcement learning algorithm called SARSA. The current policy of FRL agent is derived from maximum weighted action in each state. Therefore, the convergence of FRL is much faster as compared to commonly used *off-policy* Q Learning algorithm as shown in Fig.6.4. From Fig. 6.4, it is observed that the collected rewards of FRL in the initial episodes are much higher when compared to Q learning. Overall performance of proposed FRL is nearly 12% higher when compared to collected rewards of Q learning. The results of proposed FRL approach has been recorded at 200th episode rather than waiting for its full convergence after 600th episode. However, various scheduling heuristics have been adopted in this chapter to solve the efficient task assignment to fog vehicles. To investigate the performance of scheduling algorithms, the proposed FRL approach is compared with other four scheduling algorithms as follows,

- *FCFS:* First Come First Serve scheduling [139] is a naive algorithm which assigns the tasks in each time slot to a set of fog vehicles with least arrival time.
- *RMS:* Rate Monotonic Scheduling algorithm [140] that can select the set of vehicles which has least staying time (i.e., Dwell time) in the RSU coverage.
- *Fuzzy:* Greedy algorithm which is based on the selection of vehicles with maximum weights that are derived using Fuzzy logic system (Discussed in section 6.3.1).
- *DTA_DP:* Distributed Task Allocation in Distributed Processes [147], is a heuristic algorithm which can assign the tasks to fog vehicles with minimum completion time (i.e., service time).

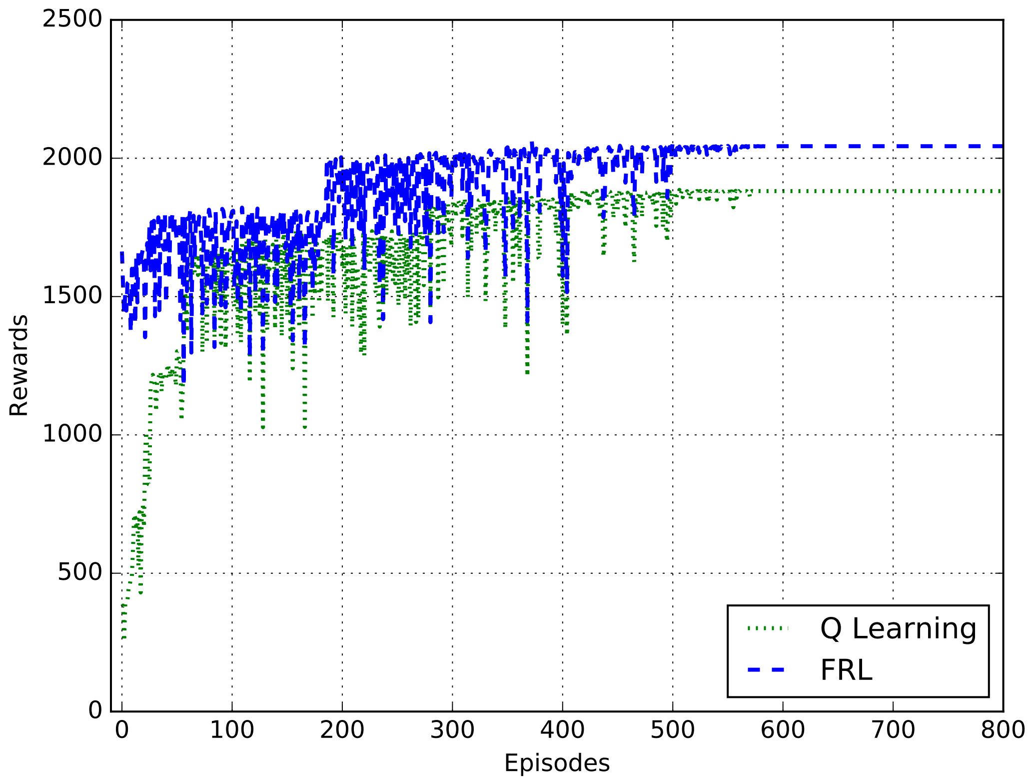


Figure 6.4: Convergence of FRL and Q Learning

Furthermore, the performance of these scheduling algorithms have been evaluated in-terms of following performance metrics. First, *RSU Energy consumption* is an important evaluation metric which shows effective utilization of battery power and improvement of network life time. Energy consumption estimates of scheduling algorithms are normalized (with respect to RSU radio coverage) and plotted on the graphs. Second, *Service time* is the completion time of task when it is assigned to a fog vehicle. Third, *Response time* is the round trip time which includes the communication time and processing time of task when it is assigned to a fog vehicle. The *Service time* and *Response time* are the two important evaluation metrics which affects the Quality of Service (QoS) of the network and the Quality of Experience (QoE) of IoT devices in smart homes, respectively. Fourth, *Local processing tasks* indicate the percentage of tasks that can not be processed by the fog vehicles and then processed by the RSU locally.

6.4.3 Results and analysis

In this section, the results obtained from simulations are analysed the proposed approach (FRL) when compared to DTA_DP, Fuzzy, RMS and FCFS. This evaluation has been performed by varying different parameters such as number of vehicles in action space, task requirement and task deadlines.

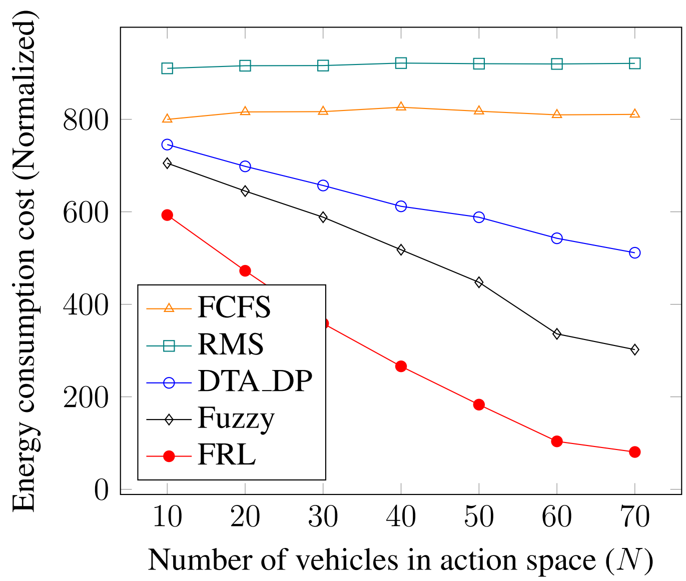


Figure 6.5: Number of vehicles and energy consumption

A set of experiments is performed in order to analyze the energy consumption of RSU for varying the number of vehicles in RSU coverage as shown in Fig. 6.5. As aforementioned, vehicle arrivals are assumed as Poisson process and the vehicles are distributed exponentially in the RSU coverage. The velocities of vehicles are uniformly distributed in a range [15m/s, 25m/s]. It is observed that the increasing number of vehicles in RSU coverage gives advantage to the proposed FRL approach, DTA_DP and *Fuzzy* for the selection of vehicles that are near to RSU. This is because, as the number of vehicles in the action space (N) increases then there is a possibility of vehicles located nearby RSU also increases. Therefore, the downlink communication from RSU to nearby fog vehicle reduces energy consumption of RSU and the energy costs decrease linearly with increased action space. However, the energy consumption cost for RMS and FCFS is observed as constant with varying action space. This is because, the RMS selects the fog vehicles with least staying period (dwell time) in RSU region. Such vehicles are located near to leaving edge of RSU even though the number of vehicles in the action space increases. Similarly, the FCFS tends to select a vehicle that arrives first which is almost at the leaving edge of RSU. Since the downlink communication cost from RSU to farthest vehicles is high, the RMS and FCFS do not perform well in terms of energy consumption. From Fig. 6.5, it can be observed that the proposed FRL approach reduces average energy cost by 48.27% and 51.61% when compared to *Fuzzy* and DTA_DP, respectively. Eventually, the FRL approach shows better performance in-terms of RSU energy consumption when the number of vehicles in the RSU coverage is high.

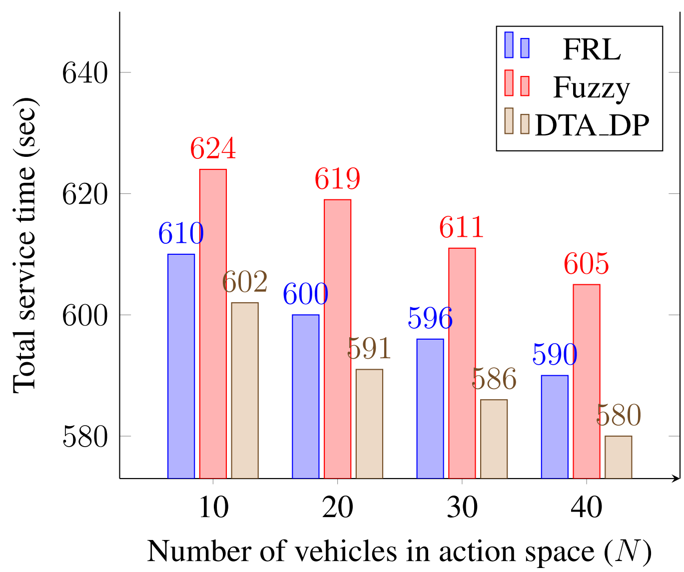


Figure 6.6: Number of vehicles and total service time

Another set of experiments has been conducted for evaluating the total service time of tasks when assigned to fog vehicles. The comparison of proposed FRL with other two scheduling algorithms are illustrated in Fig. 6.6, where x-coordinate represents the number of vehicles in RSU coverage and y-coordinate represents the total service of tasks (in seconds). As aforementioned, service time is the amount of time it takes to complete the tasks when assigned to a fog vehicle. In this evaluation, requirement of each task is considered as 500 cycles and the processing rate of fog vehicles are assumed to be distributed uniformly in the range of [10, 100] cycles/sec. When tasks are assigned to fog vehicles, the maximum and minimum service time of each task is 50 sec and 5 sec, respectively. It is observed that the total service time of all tasks executed on one fog vehicle (on average) for 100 time slots is at least 500 sec. From Figure 6.6, it is observed that the total service time for DTA_DP is less compared to the proposed FRL and *Fuzzy*. Although DTA_DP performs better in terms of service time due to the selection of fog vehicles with minimum completion time, the DTA_DP fails to achieve better results in terms of energy consumption as shown in Fig. 6.5. From Fig. 6.6, it has been shown that the largest gap between DTA_DP and the proposed FRL is less than 1.8%. Moreover, it is observed that the total service time of tasks is decreasing with the increased number of vehicles in action space. This is because, the scheduling algorithms have better choice of selection when more number of fog vehicles present in RSU coverage. However, the primary focus of this work is to minimize the energy consumption of RSU while assigning the tasks to fog vehicles subject to satisfy service time and deadline constrains. Since minimization of energy consumption has highest priority over other factors, the performance of proposed FRL has considerable gain over DTA_DP and *Fuzzy*.

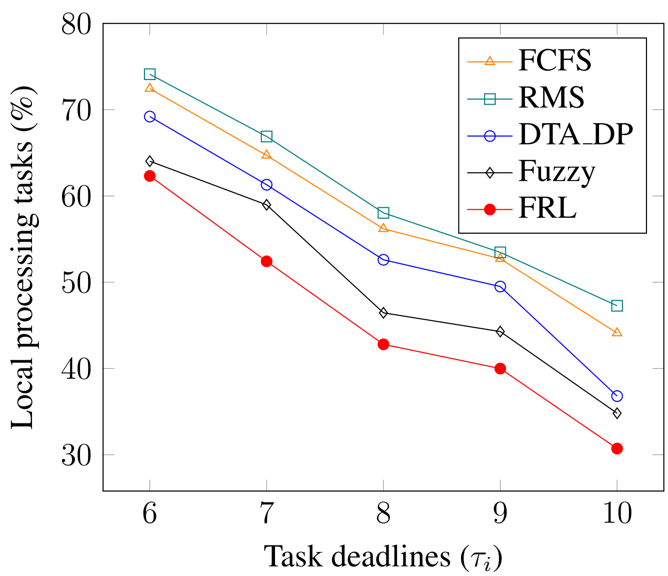


Figure 6.7: Task deadlines and local processing tasks

Number of local processing of tasks is another important evaluation metric to determine the computation load on RSU. Deadline of tasks affect the decision on offloading to fog vehicles or local execution at RSU. Intuitively, tasks with tight (or short) deadline need to be processed immediately when compared to the tasks with relaxed deadlines. If the tasks deadline constraints do not satisfy, these tasks cannot be offloaded to fog vehicles irrespective of availability of fog resources. Then, such tasks with immediate processing requirement have to be scheduled for local processing at RSU. In this simulation environment, the percentage of tasks that can be processed locally at RSU are evaluated for different task deadlines as shown in Fig. 6.7. It is observed that the percentage of tasks for local processing is higher when task deadlines are tight. As the task deadlines are relaxed, the number of tasks assigned to fog vehicles also increases and percentage of local processing at RSU reduces. Since the proposed FRL has an advantage of choosing fog vehicles that are near to RSU and with high processing rate, the number of tasks transmitted and processed at the fog vehicles are more for FRL when compared to other scheduling algorithms. From Fig. 6.7, it has been shown that the average number of local processing tasks reduces up to 8.33% and 19.86% when compared to *Fuzzy* and DTA_DP, respectively. Therefore, it is clear that the proposed FRL decreases processing load on RSU and the proposed scheduler outperforms other schedulers even in tight deadline scenarios.

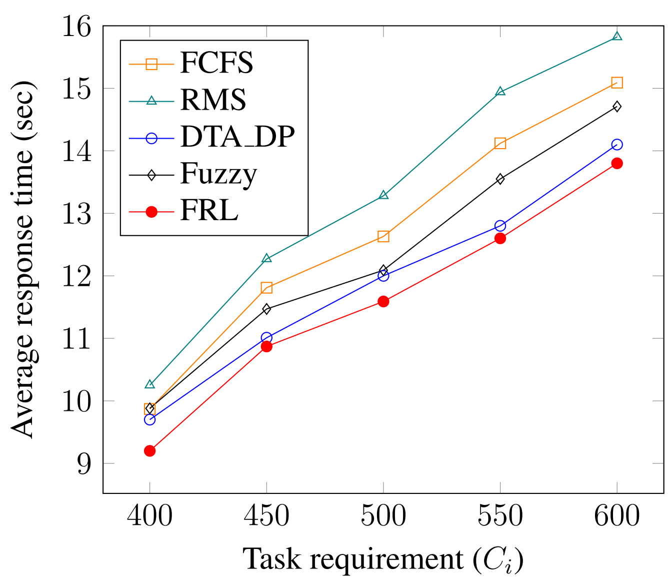


Figure 6.8: Task requirements and Average response time

To asses the quality of experience of the smart home applications in the RSU region, the response time is considered as an evaluation metric for comparison of proposed FRL approach with other scheduling algorithms. As aforementioned, the response time includes two way communication time and task execution time. As the computation result from fog nodes is small, the communication delay for the transmission of computed result is considered as negligible. However, response time depends on several factors such as distance to fog node, bit rate and processing rate of fog node. The comparison of proposed FRL with other scheduling heuristics is shown in Fig. 6.8, where x-coordinate represents the task requirement (in cycles) and y-coordinate represents the average response time (in seconds). From the Fig 6.8, it is observed that the average response time of tasks increases as the task requirement increases. For example, if task requirement increases from 400 cycles to 600 cycles and assume that the processing rate of selected fog node is 100 cycles/sec, then processing delay (as part of response time) increases from 4 sec to 6 sec. On the other hand, if the same task has data size 100Mb and achieved bit rate of selected fog node varies from 25Mb/s (nearby fog node) to 10Mb/s (farthest fog node), then communication delay can be observed from 4 sec to 10 sec. Consequently, total response time of a task increases from 8 sec to 16 sec. However, the proposed FRL approach has an advantage to select fog vehicles with high processing rate and nearby fog nodes. From Fig. 6.8, it is observed that the proposed FRL reduces average response time up to 4.34% and 8.33% when compared to DTA_DP and *Fuzzy*. Therefore, the FRL approach outperforms other scheduling algorithms in terms of average response time.

6.5 Summary

The development of autonomous and connected vehicles realize the utilization of vehicle resources for the computation of IoT applications in smart villages nearby rural highways. In this chapter, a vehicular fog computing framework is presented for energy efficient allocation of delay-sensitive tasks to mobile fog vehicles. This computation model supports execution of tasks near to edge of network by either energy constrained stationary fog node (RSU) or mobile fog nodes (vehicles). Minimization of RSU energy consumption is considered as performance metric subject to execution of tasks within tolerable response time. Further, a real-time scheduling of fog nodes has been performed by combining a greedy heuristic and on-policy reinforcement learning technique (i.e. SARSA) to improve long term reward and speedup learning process. Extensive set of experiments has been conducted and the obtained results show that the proposed algorithm has better performance over other algorithms up to 46.73% and 15.38% in terms of average energy consumption and response time, respectively.

Chapter 7

Conclusion and Future Scope

In this thesis, various challenges related to highway vehicular networks such as high mobility of vehicles, variable traffic density of vehicles, limited RSU radio range and bandwidth, limited battery storage and buffer capacity of RSU have been discussed. Specifically, high deployment cost of RSUs leads to uncovered areas in between neighbouring RSUs and the vehicles in uncovered areas are completely disconnected from the RSUs. Passing-by vehicles act as store-carry-forward relays in order to deliver data to the target vehicle in uncovered area. However, scheduling of downlink communication to relay vehicles is a challenging issue, which is addressed to minimize RSU energy consumption and improve data delivery in the system. On the other hand, due to limited buffer capacity at RSU, the data related to different tasks received at RSU experiences buffering delays. Therefore, a dynamic power allocation strategy to balance the trade-off between power allocation and buffering delays has been presented. In addition, optimal selection of relay vehicles is proposed to maximize the data delivery to neighbouring RSUs. Furthermore, a vehicular fog computing frame-work is considered for execution of tasks by offloading to mobile fog vehicles at the edge of the network. A fuzzy reinforcement learning based approach has been presented to schedule the mobile fog vehicles for energy efficient offloading of tasks while satisfying task deadlines. Overall contributions of this thesis is to investigate the RSU energy consumption, end-to-end delay, buffering delay and response time while achieving efficient data dissemination in highway vehicular network.

This thesis investigates the design and development of co-operative vehicle scheduling algorithms for data dissemination while minimizing energy consumption of RSUs deployed in rural highway environment. Different vehicle scheduling algorithms have been presented in this thesis to improve data sharing between neighbouring RSUs, reduce the end-to-end delay while delivering data to target vehicle and decrease the response time of tasks when offloaded to mobile fog vehicles. The proposed polynomial time scheduling algorithms achieve better performance in terms of energy consumption, end-to-end delay and response time. The proposed vehicle scheduling strategies have been implemented and evaluated using Mote Carlo Simulations. Extensive set of experiments has been conducted for multiple runs. Then, comparative analysis of the proposed algorithms has been presented based on their merits and capabilities.

7.1 Major contributions of this thesis

A clustering based energy efficient relay vehicle scheduling algorithm has been presented in Chapter 3. Firstly, this work investigates the relationship between energy consumption of RSU and amount of data delivery to target vehicle in the uncovered area. Secondly, a set of relay vehicles are derived to formulate a relay scheduling problem as an optimization problem. As a solution, a polynomial time algorithm has been presented using Minimum Cost Flow graph in order to schedule the relay vehicles with minimum energy consumption and maximum data flow. Furthermore, a density based clustering technique is adopted to identify Nearest Neighbour Forwarder (NNF) vehicles that are near to RSU and multi-hop distance to relay vehicles. The RSU forwards data to NNF instead of corresponding relay vehicle. Therefore, RSU can be able to minimize its energy consumption and improve data delivery performance even if the relay is outside RSU coverage. It is observed that the NNF approach combined with MCF performs better when compared to offline algorithms NFS, MCF and two more algorithms FCFS and FF.

To achieve faster data delivery to the target vehicle, an RSU assisted relay scheduling strategy has been presented in bidirectional highway scenarios. This work is discussed in Chapter 4. When a target vehicle enters into uncovered area, the amount of time a relay vehicle takes to establish a communication link with the target vehicle affects the average end-to-end delay in the system. The major objective is to minimize energy consumption of RSU and end-to-end delay to the target vehicle while scheduling the relay vehicles. The relay scheduling problem has been solved using Auction theory principles, where the RSU acts as a seller and relay vehicles act as bidders. The assignment of relays to time slots can be accomplished based on the bids received from relay vehicles. Therefore, Auction theory principles have been adopted to schedule relay vehicles in both forward and opposite directions from neighbouring RSUs. It is observed that the proposed RRS algorithm performs better compared to FCFS, GA and FRS in terms of average RSU energy consumption and end-to-end delay to the target vehicle.

Isolated deployment of RSUs in rural highway region poses limited battery storage and buffer storage capabilities. In Chapter 5, a Lyapunov optimization based dynamic power allocation technique has been presented to investigate the trade-off between energy consumption and buffer back-log sizes. Based on the derivation of achievable data rates from dynamic power allocation strategy, further a max-weight relay vehicle scheduling algorithm is proposed to maximize data delivery to a neighbouring destination RSU. Therefore, the proposed relay scheduling algorithm along with dynamic power allocation mechanism efficiently address the challenges related to the limited resources available at isolated RSU.

In order to reduce transit delays between neighbouring RSUs, a vehicular fog computing framework has been presented to offload computation intensive tasks to mobile fog vehicles within the RSU coverage. This work is discussed in Chapter 6. Since the RSUs are energy limited, they prefer to offload tasks to fog vehicles while satisfying task deadline constraints. The energy efficient task offloading has been formulated as an optimization problem. As a solution, a fuzzy reinforcement learning approach is proposed to schedule the fog vehicles for real-time assignment of tasks. It is observed that the proposed scheduling approach outperforms traditional Q-learning approach.

7.2 Future scope

Although the proposed RSU based centralized vehicle scheduling algorithms have shown better performance when compared to other existing scheduling heuristics in the literature, there are other contemporary methods and techniques that need to be investigated. A limitation of this research work is that the velocity of the vehicles in highway segment has been considered as constant. Maintaining constant velocity may be unrealistic in real time scenarios where the vehicles do not follow lane discipline and exceed the speed limit. This can be further investigated as a future research direction and methods can be proposed for realistic traffic scenarios.

Some of the extensions of this thesis as future research directions are listed here.

1. In this thesis, the vehicles are assumed to have sufficient energy reserves and effective utilization of stored energy at RSU during I2V communication has been addressed. In some practical scenarios, both vehicles and RSUs may have energy limitations. Therefore, there is a need to design and develop distributed vehicle scheduling algorithms that can address the energy consumption of vehicles during V2V communication in a dynamic highway vehicular environment. This is because electric vehicles are witnessed as current trend in recent times.

2. Secure communication in vehicular environment is an another important issue that need be addressed. Considering the nature of vehicular safety applications, without secure communication may lead to not only originates network vulnerabilities and attacks but also fatal accidents. Therefore, secure communication mechanisms are necessary to provide authentication, privacy, integrity and non-repudiation that can account for proper functioning of vehicular safety applications.

3. Unmanned Aerial Vehicles (UAVs) can be used as mobile RSUs to enhance the connectivity in the uncovered areas as shown in Fig. 7.1. Multi-hop communication discussed in this thesis has few challenges such as broken paths and throughput reduction. Using UAVs, the broken paths can be re-established and throughput can be improved. Furthermore, UAVs can be used to mitigate the impact of non-cooperative vehicles that affect the overall connectivity of vehicular network.

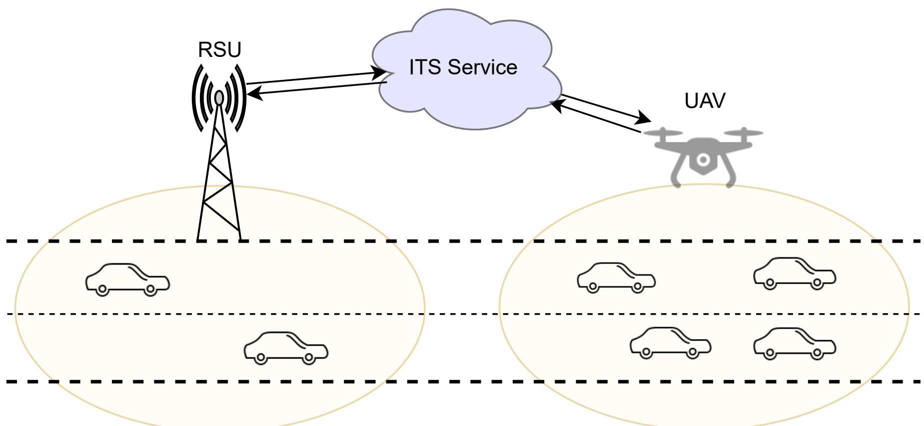


Figure 7.1: UAVs as mobile RSUs

Bibliography

[1] E. Ahmed and H. Gharavi. Cooperative vehicular networking: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 19(3):996–1014, March 2018.

[2] Changle Li, Yao Zhang, Tom H. Luan, and Yuchuan Fu. Building transmission backbone for highway vehicular networks: Framework and analysis. *IEEE Transactions on Vehicular Technology*, 67(9):8709–8722, 2018.

[3] S. Peirce and R.Mauri. *Vehicle-Infrastructure Integration (VII) initiative: Benefit-cost analysis: Pre-testing estimates*. Intell. Transp. Syst. Joint Program Office, Washington, DC, USA, mar 2007.

[4] Ruben Hidalgo-León, Javier Urquizo, Jose Macías, Diego Siguenza, Pritpal Singh, Jinsong Wu, and Guillermo Soriano. Energy harvesting technologies: Analysis of their potential for supplying power to sensors in buildings. In *2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM)*, pages 1–6, 2018.

[5] G. H. Badawy, A. A. Sayegh, and T. D. Todd. Energy provisioning in solar-powered wireless mesh networks. *IEEE Transactions on Vehicular Technology*, 59(8):3859–3871, Oct 2010.

[6] A. A. Hammad, G. H. Badawy, T. D. Todd, A. A. Sayegh, and D. Zhao. Traffic scheduling for energy sustainable vehicular infrastructure. In *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, pages 1–6, Dec 2010.

[7] J. Wang, K. Liu, K. Xiao, C. Chen, W. Wu, V. C. S. Lee, and S. H. Son. Dynamic clustering and cooperative scheduling for vehicle-to-vehicle communication in bidirectional road scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 19(6):1913–1924, 2018.

[8] B. Ko, K. Liu, and S. H. Son. Towards efficient data services in vehicular networks via cooperative infrastructure-to-vehicle and vehicle-to-vehicle communications. In *2016 Intl IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, pages 82–89, 2016.

[9] A. Khezrian, T. D. Todd, G. Karakostas, and M. Azimifar. Energy-efficient scheduling in green vehicular infrastructure with multiple roadside units. *IEEE Transactions on Vehicular Technology*, 64(5):1942–1957, May 2015.

[10] B. Ko, K. Liu, S. H. Son, and K. Park. Rsu-assisted adaptive scheduling for vehicle-to-vehicle data sharing in bidirectional road scenarios. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–13, 2019.

[11] Osama Rehman, Mohamed Ould-Khaoua, and Hadj Bourdoucen. An adaptive relay nodes selection scheme for multi-hop broadcast in vanets. *Computer Communications*, 87:76 – 90, 2016.

[12] S. H. Ahmed, D. Mu, and D. Kim. Improving bivious relay selection in vehicular delay tolerant networks. *IEEE Transactions on Intelligent Transportation Systems*, 19(3):987–995, March 2018.

[13] R. Atallah, M. Khabbaz, and C. Assi. Energy harvesting in vehicular networks: a contemporary survey. *IEEE Wireless Communications*, 23(2):70–77, April 2016.

[14] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[15] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press, 1996.

[16] M. Azimifar, T. D. Todd, A. Khezrian, and G. Karakostas. Vehicle-to-vehicle forwarding in green roadside infrastructure. *IEEE Transactions on Vehicular Technology*, 65(2):780–795, Feb 2016.

[17] Ba Cao Nguyen, Tran Manh Hoang, and Le The Dung. Performance analysis of vehicle-to-vehicle communication with full-duplex amplify-and-forward relay over double-rayleigh fading channels. *Vehicular Communications*, 19:100166, 2019.

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

[19] Dimitri P. Bertsekas. Auction algorithms for network flow problems: A tutorial introduction. *Computational Optimization and Applications*, 1(1):7–66, 1992.

[20] S. Misra, B. E. Wolfinger, M. P. Achuthananda, T. Chakraborty, S. N. Das, and S. Das. Auction-based optimal task offloading in mobile cloud computing. *IEEE Systems Journal*, 13(3):2978–2985, 2019.

[21] Xueshi Hou, Yong Li, Min Chen, Di Wu, Depeng Jin, and Sheng Chen. Vehicular fog computing: A viewpoint of vehicles as the infrastructures. *IEEE Transactions on Vehicular Technology*, 65(6):3860–3873, 2016.

[22] Md. Muzakkir Hussain and M.M. Sufyan Beg. Code-v: Multi-hop computation offloading in vehicular fog computing. *Future Generation Computer Systems*, 116:86–102, 2021.

[23] A. A. Hammad, T. D. Todd, and G. Karakostas. Variable-bit-rate transmission schedule generation in green vehicular roadside units. *IEEE Transactions on Vehicular Technology*, 65(3):1590–1604, March 2016.

[24] Yalan Wu, Jigang Wu, Long Chen, Jiaquan Yan, and Yuchong Luo. Efficient task scheduling for servers with dynamic states in vehicular edge computing. *Computer Communications*, 150:245 – 253, 2020.

[25] Reuven Y. Rubinstein. *Simulation and the Monte Carlo Method*. John Wiley & Sons, Inc., USA, 1st edition, 1981.

[26] Fabrício A. Silva, Azzedine Boukerche, Thais R. M. Braga Silva, Linnyer B. Ruiz, Eduardo Cerqueira, and Antonio A. F. Loureiro. Vehicular networks: A new challenge for content-delivery-based applications. *ACM Comput. Surv.*, 49(1), June 2016.

[27] Saif Al-Sultan, Moath M. Al-Doori, Ali H. Al-Bayatti, and Hussien Zedan. A comprehensive survey on vehicular ad hoc network. *Journal of Network and Computer Applications*, 37:380–392, 2014.

[28] Hakim Badis and Abderrezak Rachedi. Chapter 23 - modeling tools to evaluate the performance of wireless multi-hop networks. In Mohammad S. Obaidat, Petros Nicopolitidis, and Faouzi Zarai, editors, *Modeling and Simulation of Computer Networks and Systems*, pages 653–682. Morgan Kaufmann, Boston, 2015.

[29] Muhammad Rizwan Ghori, Kamal Z. Zamli, Nik Quosthoni, Muhammad Hisyam, and Mohamed Montaser. Vehicular ad-hoc network (vanet): Review. In *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, pages 1–6, 2018.

[30] Ronald Adrian, Selo Sulistyo, I Wayan Mustika, and Sahirul Alam. Roadside unit power saving using vehicle detection system in vehicular ad-hoc network. In *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, pages 198–202, 2020.

[31] Zizhen Deng, Zhen Cai, and Mangui Liang. A multi-hop vanets-assisted offloading strategy in vehicular mobile edge computing. *IEEE Access*, 8:53062–53071, 2020.

[32] Baihong Dong, Jian Deng, Weigang Wu, and Tianyu Meng. Topology analysis system for vehicular ad hoc network. In *2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, pages 299–302, 2016.

[33] 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*, 65(10):8591–8602, 2016.

[34] Stephan Eichler. Performance evaluation of the ieee 802.11p wave communication standard. In *2007 IEEE 66th Vehicular Technology Conference*, pages 2199–2203, 2007.

[35] C. Campolo, A. Molinaro, and A. O. Berthet. Full-duplex communications to improve platooning control in multi-channel vanets. In *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 936–941, May 2017.

[36] Claudia Campolo, Antonella Molinaro, Alexey Vinel, and Yan Zhang. Modeling and enhancing infotainment service access in vehicular networks with dual-radio devices. *Vehicular Communications*, 6:7 – 16, 2016.

[37] Qiong Yang, Lin Wang, Weiwei Xia, Yi Wu, and Lianfeng Shen. Development of on-board unit in vehicular ad-hoc network for highways. In *2014 International Conference on Connected Vehicles and Expo (ICCVE)*, pages 457–462, 2014.

[38] Yung-Lan Tseng. Lte-advanced enhancement for vehicular communication. *IEEE Wireless Communications*, 22(6):4–7, 2015.

[39] Hamid Gharavi and Bin Hu. Cooperative diversity routing and transmission for wireless sensor networks. *IET Wireless Sensor Systems*, 3(4):277–288, 2013.

[40] Yuanfeng Li, Yinglei Teng, Ya'nan Xiao, and Mei Song. Joint channel allocation and relay selection with multi-qos provisioning in cognitive cooperative networks. In *2013 5th IEEE International Conference on Broadband Network Multimedia Technology*, pages 191–195, 2013.

[41] Naveen Gupta and Vivek Ashok Bohara. An adaptive subcarrier sharing scheme for ofdm-based cooperative cognitive radios. *IEEE Transactions on Cognitive Communications and Networking*, 2(4):370–380, 2016.

[42] Cheng-Xiang Wang, Xuemin Hong, Xiaohu Ge, Xiang Cheng, Gong Zhang, and John Thompson. Cooperative mimo channel models: A survey. *IEEE Communications Magazine*, 48(2):80–87, 2010.

[43] Ezzeldin Hamed, Hariharan Rahul, Mohammed A. Abdelghany, and Dina Katabi. Real-time distributed mimo systems. In *Proceedings of the 2016 ACM SIGCOMM Conference*, SIGCOMM '16, page 412–425, New York, NY, USA, 2016. Association for Computing Machinery.

[44] Mohamed F. Feteiha and Hossam S. Hassanein. Cooperative vehicular ad-hoc transmission for lte-a mimo-downlink using amplify-and-forward relaying. In *2013 IEEE Global Communications Conference (GLOBECOM)*, pages 4232–4237, 2013.

[45] Tuan-Duc Nguyen, Olivier Berder, and Olivier Sentieys. Energy-efficient cooperative techniques for infrastructure-to-vehicle communications. *IEEE Transactions on Intelligent Transportation Systems*, 12(3):659–668, 2011.

[46] Jieqiong Chen, Ammar Zafar, Guoqiang Mao, and Changle Li. On the achievable throughput of cooperative vehicular networks. In *2016 IEEE International Conference on Communications (ICC)*, pages 1–7, 2016.

[47] Mehrdad Shirkhani, Zeinab Tirkan, and Abbas Taherpour. Performance analysis and optimization of two-way cooperative communications in inter-vehicular networks. In *2012 International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–6, 2012.

[48] Yaser P. Fallah, Ching-Ling Huang, Raja Sengupta, and Hariharan Krishnan. Analysis of information dissemination in vehicular ad-hoc networks with application to cooperative vehicle safety systems. *IEEE Transactions on Vehicular Technology*, 60(1):233–247, 2011.

[49] Jaehoon Jeong, Hohyeon Jeong, Eunseok Lee, Tae Oh, and David H. C. Du. Saint: Self-adaptive interactive navigation tool for cloud-based vehicular traffic optimization. *IEEE Transactions on Vehicular Technology*, 65(6):4053–4067, 2016.

[50] Emmanouil Koukoumidis, Li-Shiuan Peh, and Margaret Rose Martonosi. Signalguru: Leveraging mobile phones for collaborative traffic signal schedule advisory. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, page 127–140, New York, NY, USA, 2011. Association for Computing Machinery.

[51] Yiwen Shen, Jinho Lee, Hohyeon Jeong, Jaehoon Jeong, Eunseok Lee, and David H. C. Du. Saint+: Self-adaptive interactive navigation tool+ for emergency service delivery optimization. *IEEE Transactions on Intelligent Transportation Systems*, 19(4):1038–1053, 2018.

[52] Sebastian van de Hoef, Karl Henrik Johansson, and Dimos V. Dimarogonas. Fuel-efficient en route formation of truck platoons. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):102–112, 2018.

[53] Pavankumar Tallapragada and Jorge Cortés. Hierarchical-distributed optimized coordination of intersection traffic. *IEEE Transactions on Intelligent Transportation Systems*, 21(5):2100–2113, 2020.

[54] Yiwen Shen, Jaehoon Jeong, Tae Oh, and Sang Hyuk Son. Casd: A framework of context-awareness safety driving in vehicular networks. In *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pages 252–257, 2016.

[55] Chaojie Wang, Siyuan Gong, Anye Zhou, Tao Li, and Srinivas Peeta. Cooperative adaptive cruise control for connected autonomous vehicles by factoring

communication-related constraints. *Transportation Research Procedia*, 38:242–262, 2019. Journal of Transportation and Traffic Theory.

[56] Koichi Serizawa, Manabu Mikami, Kohei Moto, and Hitoshi Yoshino. Field trial activities on 5g nr v2v direct communication towards application to truck platooning. In *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pages 1–5, 2019.

[57] Ignacio Llatser, Thomas Michalke, Maxim Dolgov, Florian Wildschütte, and Hendrik Fuchs. Cooperative automated driving use cases for 5g v2x communication. In *2019 IEEE 2nd 5G World Forum (5GWF)*, pages 120–125, 2019.

[58] Taehwan Hwang, Jaehoon Paul Jeong, and Eunseok Lee. Sana: Safety-aware navigation app for pedestrian protection in vehicular networks. In *2014 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 947–953, 2014.

[59] Gorkem Kar, Shubham Jain, Marco Gruteser, Fan Bai, and Ramesh Govindan. Real-time traffic estimation at vehicular edge nodes. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, SEC '17, New York, NY, USA, 2017. Association for Computing Machinery.

[60] M. Patra, R. Thakur, and C. S. R. Murthy. Improving delay and energy efficiency of vehicular networks using mobile femto access points. *IEEE Transactions on Vehicular Technology*, 66(2):1496–1505, Feb 2017.

[61] M. Patra and C. S. R. Murthy. Performance evaluation of joint placement and sleep scheduling of grid-connected solar powered road side units in vehicular networks. *IEEE Transactions on Green Communications and Networking*, 2(4):1197–1209, Dec 2018.

[62] L. Zhang and Y. Wang. An offline roadside unit on-off scheduling algorithm for energy efficiency of ad hoc networks. *IEEE Access*, 6:59742–59751, 2018.

[63] W. S. Atoui, M. A. Salahuddin, W. Ajib, and M. Boukadoum. Scheduling energy harvesting roadside units in vehicular ad hoc networks. In *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, pages 1–5, Sept 2016.

[64] W. S. Atoui, W. Ajib, and M. Boukadoum. Offline and online scheduling algorithms for energy harvesting rsus in vanets. *IEEE Transactions on Vehicular Technology*, 67(7):6370–6382, July 2018.

[65] Yu Wang and Lei Zhang. Multiple rsus scheduling for energy efficiency in vehicular ad hoc networks. *Wireless Personal Communications*, Apr 2019.

[66] R. F. Atallah, C. M. Assi, and J. Y. Yu. A reinforcement learning technique for optimizing downlink scheduling in an energy-limited vehicular network. *IEEE Transactions on Vehicular Technology*, 66(6):4592–4601, June 2017.

[67] E. Baccelli, P. Jacquet, B. Mans, and G. Rodolakis. Highway vehicular delay tolerant networks: Information propagation speed properties. *IEEE Transactions on Information Theory*, 58(3):1743–1756, 2012.

[68] E. Altman, A. P. Azad, T. Baºar, and F. De Pellegrini. Optimal activation and transmission control in delay tolerant networks. In *2010 Proceedings IEEE INFOCOM*, pages 1–5, 2010.

[69] Venkatesh Ramaiyan, Eitan Altman, and Anurag Kumar. Delay optimal scheduling in a two-hop vehicular relay network. *Mobile Networks and Applications*, 15(1):97–111, Feb 2010.

[70] M. J. Khabbaz, W. F. Fawaz, and C. M. Assi. Modeling and delay analysis of intermittently connected roadside communication networks. *IEEE Transactions on Vehicular Technology*, 61(6):2698–2706, 2012.

[71] Y. Wang, Y. Liu, J. Zhang, H. Ye, and Z. Tan. Cooperative store carry forward scheme for intermittently connected vehicular networks. *IEEE Transactions on Vehicular Technology*, 66(1):777–784, Jan 2017.

[72] Satish Vemireddy and Rashmi Ranjan Rout. Clustering based energy efficient multi-relay scheduling in green vehicular infrastructure. *Vehicular Communications*, 25:100251, 2020.

[73] Satish Vemireddy and Rashmi Ranjan Rout. Auction based energy-efficient cooperative relay scheduling in bidirectional highway scenarios for vanet. *Wireless Personal Communications*, Mar 2021.

[74] A. A. Siddig, A. S. Ibrahim, and M. H. Ismail. Full-duplex store-carry-forward scheme for intermittently connected vehicular networks. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pages 1–6, 2020.

[75] L. Huang, H. Jiang, Z. Zhang, and Z. Yan. Optimal traffic scheduling between roadside units in vehicular delay-tolerant networks. *IEEE Transactions on Vehicular Technology*, 64(3):1079–1094, 2015.

[76] L. Huang, H. Jiang, Z. Zhang, Z. Yan, and H. Guo. Efficient data traffic forwarding for infrastructure-to-infrastructure communications in vanets. *IEEE Transactions on Intelligent Transportation Systems*, 19(3):839–853, 2018.

[77] Mohammad Mukhtaruzzaman and Mohammed Atiquzzaman. Clustering in vehicular ad hoc network: Algorithms and challenges. *Computers & Electrical Engineering*, 88:106851, 2020.

[78] Satendra Kumar Bansal, Anand Singh Bisen, and Rohit Gupta. A secure hashing technique for k-means based cluster approach in vanet. In *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, pages 2037–2041, 2016.

[79] Muhammad Fahad Khan, Farhan Aadil, Muazzam Maqsood, Syed Hashim Raza Bukhari, Maqbool Hussain, and Yunyoung Nam. Moth flame clustering algorithm for internet of vehicle (mfca-iov). *IEEE Access*, 7:11613–11629, 2019.

[80] Khalid Abdel Hafeez, Lian Zhao, Zaiyi Liao, and Bobby Ngok-Wah Ma. A fuzzy-logic-based cluster head selection algorithm in vanets. In *2012 IEEE International Conference on Communications (ICC)*, pages 203–207, 2012.

[81] Khalid Abdel Hafeez, Lian Zhao, Jon W. Mark, Xuemin Shen, and Zhisheng Niu. Distributed multichannel and mobility-aware cluster-based mac protocol for vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 62(8):3886–3902, 2013.

[82] Hamid Reza Arkian, Reza Ebrahimi Atani, Abolfazl Diyanat, and Atefe Pourkhalili. A cluster-based vehicular cloud architecture with learning-based resource management. *The Journal of Supercomputing*, 71(4):1401–1426, Apr 2015.

[83] Dingsheng Deng. Dbscan clustering algorithm based on density. In *2020 7th International Forum on Electrical Engineering and Automation (IFEEA)*, pages 949–953, 2020.

[84] Wei Fan, Yan Shi, Shanzhi Chen, and Longhao Zou. A mobility metrics based dynamic clustering algorithm for vanets. In *IET International Conference on Communication Technology and Application (ICCTA 2011)*, pages 752–756, 2011.

[85] Minming Ni, Zhangdui Zhong, and Dongmei Zhao. Mpbc: A mobility prediction-based clustering scheme for ad hoc networks. *IEEE Transactions on Vehicular Technology*, 60(9):4549–4559, 2011.

[86] Azzedine Boukerche, Zhenxia Zhang, and Xin Fei. Reducing handoff latency for nemo-based vehicular ad hoc networks. In *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, pages 1–5, 2011.

[87] Zhenxia Zhang, Azzedine Boukerche, and Richard Pazzi. A novel multi-hop clustering scheme for vehicular ad-hoc networks. MobiWac '11, page 19–26, New York, NY, USA, 2011. Association for Computing Machinery.

[88] Paul Klemperer. Auction theory: A guide to the literature. *Journal of Economic Surveys*, 13(3):227–286, 1999.

[89] Zhiwen Hu, Zijie Zheng, Tao Wang, Lingyang Song, and Xiaoming Li. Roadside unit caching: Auction-based storage allocation for multiple content providers. *IEEE Transactions on Wireless Communications*, 16(10):6321–6334, 2017.

[90] Minghui Liwang, Shijie Dai, Zhibin Gao, Yuliang Tang, and Huaiyu Dai. A truthful reverse-auction mechanism for computation offloading in cloud-enabled vehicular network. *IEEE Internet of Things Journal*, 6(3):4214–4227, 2019.

[91] Zhibin Gao, Minghui Liwang, Seyyedali Hosseinalipour, Huaiyu Dai, and Xianbin Wang. A truthful auction for graph job allocation in vehicular cloud-assisted networks. *IEEE Transactions on Mobile Computing*, pages 1–1, 2021.

[92] Syed R. Rizvi, Susan Zehra, Ravi Mukkamala, and Stephan Olariu. Asap: An agent-assisted smart auction-based parking system in internet of things. In *Proceedings of the 9th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*, DIVANet '19, page 1–8, New York, NY, USA, 2019. Association for Computing Machinery.

[93] M. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. 2010.

[94] Ali A. Siddig, Ahmed S. Ibrahim, and Mahmoud H. Ismail. An optimal power allocation and relay selection full-duplex store-carry-forward scheme for intermittently connected vehicular networks. *IEEE Access*, 8:163903–163916, 2020.

[95] Lorenzo Bracciale and Pierpaolo Loreti. Lyapunov drift-plus-penalty optimization for queues with finite capacity. *IEEE Communications Letters*, 24(11):2555–2558, 2020.

[96] Xiaoyan He, Chongtao Guo, and Bin Liao. Spectrum and power allocation for vehicular networks with diverse latency requirements. In *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–5, 2019.

[97] Huaqing Zhang, Xiao Tang, Reginald Banez, Pinyi Ren, Lingyang Song, and Zhu Han. An epec analysis for power allocation in lte-v networks. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pages 1–6, 2017.

[98] Yupei Liu, Haijun Zhang, Keping Long, Arumugam Nallanathan, and Victor C. M. Leung. Energy-efficient subchannel matching and power allocation in noma autonomous driving vehicular networks. *IEEE Wireless Communications*, 26(4):88–93, 2019.

[99] Rashmi Ranjan Rout, Satish Vemireddy, Sanjib Kumar Raul, and D.V.L.N. Somayajulu. Fuzzy logic-based emergency vehicle routing: An iot system development for smart city applications. *Computers & Electrical Engineering*, 88:106839, 2020.

[100] Lina Altoaimy and Imad Mahgoub. Fuzzy logic based localization for vehicular ad hoc networks. In *2014 IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems (CIVTS)*, pages 121–128, 2014.

[101] Elnaz Limouchi, Imad Mahgoub, and Ahmad Alwakeel. Fuzzy logic-based broadcast in vehicular ad hoc networks. In *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, pages 1–5, 2016.

[102] Rajesh Purkait and Sachin Tripathi. Fuzzy logic based multi-criteria intelligent forward routing in vanet. *Wireless Personal Communications*, 111(3):1871–1897, Apr 2020.

[103] Revati S. Jadhav, Manoj M. Dongre, and Geeta Devurkar. Fuzzy logic based data dissemination in vehicular ad hoc networks. In *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, volume 1, pages 479–483, 2017.

[104] Cheng Huang, Rongxing Lu, and Kim-Kwang Raymond Choo. Vehicular fog computing: Architecture, use case, and security and forensic challenges. *IEEE Communications Magazine*, 55(11):105–111, 2017.

[105] Zhaolong Ning, Jun Huang, and Xiaojie Wang. Vehicular fog computing: Enabling real-time traffic management for smart cities. *IEEE Wireless Communications*, 26(1):87–93, 2019.

[106] Arnav Thakur and Reza Malekian. Fog computing for detecting vehicular congestion, an internet of vehicles based approach: A review. *IEEE Intelligent Transportation Systems Magazine*, 11(2):8–16, 2019.

[107] Om-Kolsoom Shahryari, Hossein Pedram, Vahid Khajehvand, and Mehdi Dehghan TakhtFooladi. Energy-efficient and delay-guaranteed computation offloading for fog-based iot networks. *Computer Networks*, 182:107511, 2020.

[108] Chaogang Tang, Xianglin Wei, Chunsheng Zhu, Yi Wang, and Weijia Jia. Mobile vehicles as fog nodes for latency optimization in smart cities. *IEEE Transactions on Vehicular Technology*, 69(9):9364–9375, 2020.

[109] Yi Zhang, Chih-Yu Wang, and Hung-Yu Wei. Parking reservation auction for parked vehicle assistance in vehicular fog computing. *IEEE Transactions on Vehicular Technology*, 68(4):3126–3139, 2019.

[110] Zhenyu Zhou, Pengju Liu, Junhao Feng, Yan Zhang, Shahid Mumtaz, and Jonathan Rodriguez. Computation resource allocation and task assignment optimization in vehicular fog computing: A contract-matching approach. *IEEE Transactions on Vehicular Technology*, 68(4):3113–3125, 2019.

[111] Syed Sarmad Shah, Muhammad Ali, Asad Waqar Malik, Muazzam A. Khan, and Sri Devi Ravana. vfog: A vehicle-assisted computing framework for delay-sensitive applications in smart cities. *IEEE Access*, 7:34900–34909, 2019.

[112] Jinming Shi, Jun Du, Jingjing Wang, Jian Wang, and Jian Yuan. Priority-aware task offloading in vehicular fog computing based on deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, 69(12):16067–16081, 2020.

[113] Seung-Seob Lee and Sukyoung Lee. Resource allocation for vehicular fog computing using reinforcement learning combined with heuristic information. *IEEE Internet of Things Journal*, 7(10):10450–10464, 2020.

[114] Zhe Wang, Zhangdui Zhong, and Minming Ni. Application-aware offloading policy using smdp in vehicular fog computing systems. In *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6, 2018.

[115] Zhenchun Wei, Fei Liu, Yan Zhang, Juan Xu, Jianjun Ji, and Zengwei Lyu. A q-learning algorithm for task scheduling based on improved svm in wireless sensor networks. *Computer Networks*, 161:138–149, 2019.

[116] Taha Alfakih, Mohammad Mehedi Hassan, Abdu Gumaei, Claudio Savaglio, and Giancarlo Fortino. Task offloading and resource allocation for mobile edge computing by deep reinforcement learning based on sarsa. *IEEE Access*, 8:54074–54084, 2020.

[117] S. Y. Wang. The effects of wireless transmission range on path lifetime in vehicle-formed mobile ad hoc networks on highways. In *IEEE International Conference on Communications, 2005. ICC 2005. 2005*, volume 5, pages 3177–3181 Vol. 5, May 2005.

[118] C. Sommer, D. Eckhoff, R. German, and F. Dressler. A computationally inexpensive empirical model of ieee 802.11p radio shadowing in urban environments. In *2011 Eighth International Conference on Wireless On-Demand Network Systems and Services*, pages 84–90, Jan 2011.

[119] Zhigang Rong and Theodore S Rappaport. *Wireless communications: Principles and practice, solutions manual*. Prentice Hall, 1st ed. edition, 1996.

[120] K. Bilstrup, E. Uhlemann, E. G. Strom, and U. Bilstrup. Evaluation of the ieee 802.11p mac method for vehicle-to-vehicle communication. In *2008 IEEE 68th Vehicular Technology Conference*, pages 1–5, Sep. 2008.

[121] M. Khabazian and M. K. M. Ali. A performance modeling of connectivity in vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 57(4):2440–2450, July 2008.

[122] Sven O. Krumke and Clemens Thielen. The generalized assignment problem with minimum quantities. *European Journal of Operational Research*, 228(1):46 – 55, 2013.

[123] A. Bryant and K. Cios. Rnn-dbscan: A density-based clustering algorithm using reverse nearest neighbor density estimates. *IEEE Transactions on Knowledge and Data Engineering*, 30(6):1109–1121, June 2018.

[124] Sabri M. Hanshi, Tat-Chee Wan, Mohammad M. Kadhum, and Ali Abdulqader Bin-Salem. Review of geographic forwarding strategies for inter-vehicular communications from mobility and environment perspectives. *Vehicular Communications*, 14:64 – 79, 2018.

[125] C. Campolo, A. Molinaro, A. O. Berthet, and A. Vinel. Full-duplex radios for vehicular communications. *IEEE Communications Magazine*, 55(6):182–189, June 2017.

[126] WAVE. Ieee standard for wireless access in vehicular environments (wave) – multi-channel operation. *IEEE Std 1609.4-2016 (Revision of IEEE Std 1609.4-2010)*, pages 1–94, March 2016.

[127] Junhao Gan and Yufei Tao. Dbscan revisited: Mis-claim, un-fixability, and approximation. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, pages 519–530, New York, NY, USA, 2015. ACM.

[128] C. Campolo, A. Molinaro, and A. O. Berthet. Full-duplex communications to improve platooning control in multi-channel vanets. In *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 936–941, 2017.

[129] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief. Power-delay tradeoff in multi-user mobile-edge computing systems. In *2016 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, Dec 2016.

[130] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief. Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems. *IEEE Transactions on Wireless Communications*, 16(9):5994–6009, Sep. 2017.

[131] Donald Gross, John F. Shortle, James M. Thompson, and Carl M. Harris. *Fundamentals of Queueing Theory*. Wiley-Interscience, USA, 4th edition, 2008.

[132] X. Xie, F. Wang, K. Li, P. Zhang, and H. Wang. Improvement of multi-channel mac protocol for dense vanet with directional antennas. In *2009 IEEE Wireless Communications and Networking Conference*, pages 1–6, 2009.

[133] G. Zhang, W. Zhang, Y. Cao, D. Li, and L. Wang. Energy-delay tradeoff for dynamic offloading in mobile-edge computing system with energy harvesting devices. *IEEE Transactions on Industrial Informatics*, 14(10):4642–4655, 2018.

[134] Z. Chang, L. Liu, X. Guo, and Q. Sheng. Dynamic resource allocation and computation offloading for iot fog computing system. *IEEE Transactions on Industrial Informatics*, 17(5):3348–3357, 2021.

[135] S. Pan and Y. Chen. Energy-optimal scheduling of mobile cloud computing based on a modified lyapunov optimization method. *IEEE Transactions on Green Communications and Networking*, 3(1):227–235, 2019.

[136] J. Kim and W. Lee. Stochastic decision making for adaptive crowdsourcing in medical big-data platforms. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(11):1471–1476, 2015.

[137] J. Kim. Energy-efficient dynamic packet downloading for medical iot platforms. *IEEE Transactions on Industrial Informatics*, 11(6):1653–1659, 2015.

[138] Silvano Martello and Paolo Toth. Algorithm 632: A program for the 0–1 multiple knapsack problem. *ACM Trans. Math. Softw.*, 11(2):135–140, June 1985.

[139] W. Zhao and J. A. Stankovic. Performance analysis of fcfs and improved fcfs scheduling algorithms for dynamic real-time computer systems. In *[1989] Proceedings. Real-Time Systems Symposium*, pages 156–165, 1989.

[140] J. Lehoczky, L. Sha, and Y. Ding. The rate monotonic scheduling algorithm: exact characterization and average case behavior. In *[1989] Proceedings. Real-Time Systems Symposium*, pages 166–171, 1989.

[141] Chen Xu, Yahui Wang, Zhenyu Zhou, Bo Gu, Valerio Frascolla, and Shahid Mumtaz. A low-latency and massive-connectivity vehicular fog computing framework for 5g. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6, 2018.

[142] Qiong Wu, Hanxu Liu, Ruhai Wang, Pingyi Fan, Qiang Fan, and Zhengquan Li. Delay-sensitive task offloading in the 802.11p-based vehicular fog computing systems. *IEEE Internet of Things Journal*, 7(1):773–785, 2020.

[143] Mohd Azmin Samsuden, Norizan Mat Diah, and Nurazzah Abdul Rahman. A review paper on implementing reinforcement learning technique in optimising games performance. In *2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)*, pages 258–263, 2019.

[144] X. Chen. Decentralized computation offloading game for mobile cloud computing. *IEEE Transactions on Parallel and Distributed Systems*, 26(4):974–983, 2015.

[145] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[146] J. Park, Y. Sim, G. Lee, and D. Cho. A fuzzy logic based electric vehicle scheduling in smart charging network. In *2019 16th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pages 1–6, Jan 2019.

[147] Yi Zhou, Kai Liu, Xincao Xu, Songtao Guo, Zhou Wu, Victor Lee, and Sang Son. Distributed scheduling for time-critical tasks in a two-layer vehicular fog computing architecture. In *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, page 1–7. IEEE Press, 2020.

List of Publications

Publications from the thesis

Journal papers:

1. **Satish Vemireddy**, Rashmi Ranjan Rout, Clustering based energy efficient multi-relay scheduling in green vehicular infrastructure, *Vehicular Communications, Elsevier,* Volume 25, 2020.

2. **Satish Vemireddy**, Rashmi Ranjan Rout, Auction based Energy-Efficient Cooperative Relay Scheduling in Bidirectional Highway Scenarios for VANET, *Wireless Personal Communications, Springer*, 1-25, 2021.

3. **Satish Vemireddy**, Rashmi Ranjan Rout, Delay-aware Energy Efficient Dynamic Relay Scheduling in Isolated Infrastructure, *IEEE Transactions on Intelligent Transport Systems*. (Received revision).

4. **Satish Vemireddy**, Rashmi Ranjan Rout, Fuzzy Reinforcement Learning for Energy Efficient Task Offloading in Vehicular Fog Computing , *Computer Networks, Elsevier*, Volume 199, 2021, 108463, ISSN 1389-1286.