Software defined networking approach for efficient information dissemination in vehicular networks

Kalupahana Liyanage, Kushan Sudheera

2020

Kalupahana Liyanage, K.S. (2020). Software defined networking approach for efficient information dissemination in vehicular networks. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/137888>

<https://doi.org/10.32657/10356/137888>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 17:47:07 SGT

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

**SOFTWARE DEFINED NETWORKING
APPROACH FOR EFFICIENT
INFORMATION DISSEMINATION IN
VEHICULAR NETWORKS**

**KALUPAHANA LIYANAGE KUSHAN SUDHEERA
SCHOOL OF ELECTRICAL & ELECTRONIC
ENGINEERING**

2020

**SOFTWARE DEFINED NETWORKING
APPROACH FOR EFFICIENT
INFORMATION DISSEMINATION IN
VEHICULAR NETWORKS**

**KALUPAHANA LIYANAGE KUSHAN
SUDHEERA**

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2020

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

29-JUL-2019

Date

KALUPAHANA LIYANAGE KUSHAN
SUDHEERA

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

29-JUL-2019

Date

ASSOC. PROF. MAODE MA

Author Attribution Statement

This thesis contains material from three papers published in peer-reviewed journals, five papers accepted and presented at conferences and one paper (journal) under review, in all of which I was the first and the corresponding author.

A preliminary version of the work in Chapter 3 is published as K. L. K. Sudheera, M. Ma, G. G. M. N. Ali and P. H. J. Chong, “Delay efficient software defined networking based architecture for vehicular networks,” 2016 IEEE International Conference on Communication Systems (ICCS), Shenzhen, 2016, pp. 1-6. doi: 10.1109/ICCS.2016.7833564.

The contributions of the co-authors are as follows:

- Prof. Peter Chong provided the initial project direction and edited the manuscript drafts.
- I reviewed the literature, came up with the new architecture, prepared the manuscript and performed simulations.
- Dr. Nawaz Ali participated in the initial brainstorming discussions, revised the manuscript and provided several technical directions and suggestions on simulation settings.
- A/Prof. Ma edited the manuscript and presented the paper at the ICCS conference.

A major part of Chapter 3 is published as Kushan Sudheera Kalupahana Liyanage, Maode Ma, Peter Han Joo Chong, “Controller placement optimization in hierarchical distributed software defined vehicular networks,” Computer Networks, Vol. 135, 2018, pp. 226-239, ISSN 1389-1286, <https://doi.org/10.1016/j.comnet.2018.02.022>.

The contributions of the co-authors are as follows:

- I reviewed the literature, conceived and analyzed the idea, developed the mathematical model, prepared the manuscript draft, developed the simulation platform and conducted the experiments.
- A/Prof. Ma reviewed the idea, guided me through the process and revised the manuscript.

- Prof. Peter Chong contributed with suggestions and edited the manuscript draft.

A part of Chapter 4 is published as K. L. K. Sudheera, M. Ma and P. H. J. Chong, “Link Dynamics Based Packet Routing Framework for Software Defined Vehicular Networks,” GLOBECOM 2017 - 2017 IEEE Global Communications Conference, Singapore, 2017, pp. 1-6. doi: 10.1109/GLOCOM.2017.8254597.

The contributions of the co-authors are as follows:

- I reviewed the literature, co-developed the idea and study with Prof. Peter Chong, formulated the mathematical model, prepared the manuscript draft, conducted the simulation experiments and presented the paper.
- A/Prof. Ma provided feedback, supervised the work and revised the manuscript.
- Prof. Peter Chong assisted in developing the idea and the study, and revised the manuscript.

Another part of the work in Chapter 4 is published as K. S. K. L., M. Ma and P. H. J. Chong, “Efficient Flow Instantiation via Source Routing in Software Defined Vehicular Networks,” 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall), Toronto, ON, 2017, pp. 1-5. doi: 10.1109/VTCFall.2017.8288201.

The contributions of the co-authors are as follows:

- I reviewed the literature, developed the idea, prepared the manuscript draft, conducted the simulation experiments and presented the paper.
- A/Prof. Ma guided me through the process and revised the manuscript.
- Prof. Peter Chong provided feedback and revised the manuscript.

A major part of Chapter 4 is published as K. L. K. Sudheera, M. Ma and P. H. J. Chong, “Link Stability Based Optimized Routing Framework for Software Defined Vehicular Networks,” in IEEE Transactions on Vehicular Technology, vol. 68, no. 3, pp. 2934-2945, March 2019. doi: 10.1109/TVT.2019.2895274.

The contributions of the co-authors are as follows:

- I reviewed the literature, developed the study, formulated the optimization problem, designed the solution mechanism, prepared the manuscript draft and response letters and conducted the simulation experiments.
- A/Prof. Ma provided valuable feedback, supervised the work, revised the manuscript and response letters and financially supported the publication.

- Prof. Peter Chong revised the manuscript and provided feedback on response letters.

A part of Chapter 5 is published as K. L. K. Sudheera, M. Ma and P. H. J. Chong, “Link Stability Based Hybrid Routing Protocol for Software Defined Vehicular Networks,” 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, 2018, pp. 1-6. doi: 10.1109/ICC.2018.8422399.

The contributions of the co-authors are as follows:

- I extended the idea from the previous GLOBECOM paper with the objective of resolving its practical issues, developed the mathematical model and the hybrid routing protocol, prepared the manuscript draft, conducted the simulation experiments and presented the paper.
- A/Prof. Ma provided feedback, supervised the work and revised the manuscript.
- Prof. Peter Chong contributed in developing the idea and the study, and revised the manuscript.

A major part of Chapter 5 is published as K. L. K. Sudheera, M. Ma, and P. H. J. Chong, “Connectivity Aware Tribrid Routing Framework For A Generalized Software Defined Vehicular Network,” Computer Networks, vol. 152, pp.167-177, 2019. doi: <https://doi.org/10.1016/j.comnet.2019.01.040>.

The contributions of the co-authors are as follows:

- I co-developed the study extending the ICC conference paper based on Prof. Ma’s suggestions, formulated the problem, prepared the manuscript draft and response letters and conducted the simulation experiments.
- A/Prof. Ma provided the initial study direction, supervised the work and revised the manuscript and response letters.
- Prof. Peter Chong revised the manuscript and response letters.

A part of Chapter 6 is published as K. L. Kushan Sudheera, M. Ma and P. H. J. Chong, “Cooperative Data Routing & Scheduling In Software Defined Vehicular Networks,” 2018 IEEE Vehicular Networking Conference (VNC), Taipei, Taiwan, 2018, pp. 1-8. doi: 10.1109/VNC.2018.8628383.

The contributions of the co-authors are as follows:

- I reviewed the literature, developed the idea, formulated the model, prepared the manuscript draft, conducted the simulation experiments and presented the paper.

- A/Prof. Ma supervised the work, provided valuable feedback and revised the manuscript.
- Prof. Peter Chong provided valuable comments and suggestions and revised the manuscript.

Another part of Chapter 6 is under review as K. L. K. Sudheera, M. Ma, and P. H. J. Chong, “Real-Time Cooperative Data Routing and Scheduling in Software Defined Vehicular Networks,” IEEE Transactions on Intelligent Transportation Systems, (Under Review).

The contributions of the co-authors are as follows:

- I reviewed the literature, developed the idea extending the conference paper (VNC), formulated the models and solution mechanisms, prepared the manuscript draft and conducted the simulation experiments.
- A/Prof. Ma supervised the work, provided valuable comments and suggestions and revised the manuscript.
- Prof. Peter Chong provided valuable feedback and revised the manuscript.

29-JUL-2019

Date

KALUPAHANA LIYANAGE KUSHAN
SUDHEERA

To My Loving Wife and Parents

Acknowledgement

I would like to sincerely thank my current supervisor, Prof. Maode Ma, for his aspiring guidance and continuous support during my PhD studies. He has been very patient and understanding in his supervision. I would also like to pay my gratitude to my former supervisor, Prof. Peter Chong, currently the head of the Department of Electrical and Electronic Engineering at Auckland University of Technology, New Zealand. I'm deeply honored by the fact that he selected me to be your PhD student even before the completion of my bachelor's degree. Even after transferring to AUT, he continued to supervise me as promised.

Both Professors were two major corner stones and support pillars in my PhD journey, and were always there for me even amidst their busy schedules. They regularly supported my thesis with illuminating views, invaluable constructive ideas and helpful comments. I must especially thank both of them for the academic freedom they provided, identifying and trusting my ability in conducting scientific research. It really helped me to have a less stressed life and in fact brought the best out of me.

I must also sincerely thank Dr. Nawaz Ali for his enormous support, valuable advice and constructive ideas, especially, in the first 2 years of my PhD study. A special thank goes to Dr. Bryan Ng as well, for the hands on experience he provided at the beginning of my PhD. It really helped to have a clear vision on the new research domain.

I had the privilege to work with a number of talented researchers locally and internationally. I was able to learn so much from them, and also had lots of eternal memories which I will cherish always. To this end, I would like to thank all of them for their sincere friendship. I would also like to gracefully mention the international collaborators for reaching out and also actively exchanging ideas on this new research domain, and re-enforcing my passion and belief on the thesis.

I should also acknowledge and thank the thesis advisory committee, qualifying examination panel and in advance the thesis examiners for their valuable comments and suggestions in improving the quality of the thesis. Recognition should also go to Nanyang Technological University (NTU), NTU Research Scholarship, and the Development of NTU/NXP-Intelligent Transport System Test-Bed Living Lab Fund (Project Reference No. S15-1105-RFLLF URBAN) for the financial support.

A massive thank should go to my lovely wife, Nadeesha, for being there for me, from all the way from undergrads, and encouraging me to pursue my passion and goals, even sacrificing her comforts. Also, I should thank my parents and siblings. Even though we live miles away, they wholeheartedly supported my dreams.

Finally, I would like to express my sincere gratitude towards each and every individual who has directly or indirectly helped me in reaching my goal.

Thank you.

Abstract

Vehicular Ad-hoc Networks (VANET) have emerged as a viable solution for the network related problems in Intelligent Transportation Systems (ITSs). However, with its current architecture, especially with the unavailability of the global network information, VANET has not been able to fulfill some of the necessities for efficient information dissemination and most critically, still lacks flexibility and programmability. Recently, Software Defined Networking (SDN) has been introduced into the vehicular domain in order to overcome the shortcomings in VANET. SDN is capable of handling the dynamic nature of networks with intelligent applications while lowering operational costs through simplified hardware and software. Yet, the architectural limitations of the presented Software Defined Vehicular Networks (SDVN), especially the centralization of control plane and its placement, have imposed doubts on its usability especially in the midst of stringent delay requirement of VANET applications.

Alternatively, in this thesis, we propose a novel Hierarchical Distributed controller architecture for SDVN (HD-SDVN) in which multiple controllers are hierarchically distributed and a layer of control is brought down to the Road Side Unit (RSU) level cutting down the southbound communication latency, while still leveraging the logically centralized pros such as collective global network view. Further, an optimized placement of controllers in several selected RSUs is proposed together with a proactive Flow Instantiation (FI) to further reduce the operational latency of SDVN. Implementation of HD-SDVN and the proposed proactive operational mode in NS3 exhibits latency measurements in tens of milliseconds which is satisfactory to most of the VANET applications.

The thesis further emphasizes the effect of software defined approach on efficient information dissemination in vehicular networks by scrutinizing the link stability in multi-hop data routing. As a middle ground for the shortest path approach and most stable path approach in the current literature, we propose an optimization

based data routing model for a connected vehicular network, scrutinizing multiple stable enough paths with less travel times that can deliver a given set of packets using SDVN. The problem formulation follows a minimum cost capacitated flow problem, embedded with an incremental packet allocation scheme to shift the computational complexity to a realizable level for real-time communication. Moreover, a source routing based FI scheme with effective route caching is presented to reduce the extent of communication with the control plane while still utilizing the intelligence of the controller.

We further acknowledge the possible limitations and uncertainties in network information which confine the improvement of the aforementioned proposal, through a combination of centralized and distributed routing techniques and making use of unicast, broadcast, and store, carry and forward concepts. The resulting tribrid routing framework similarly focuses on finding stable enough shortest routes that can deliver a given set of packets while satisfying the required Quality of Services (QoSs). In case of network uncertainties, the protocol incorporates broadcasting based distributed techniques along with unicast routing. In sparse network conditions, the model aims to deliver the packets in the optimal path(s) with least store and carry times within the QoS requirements.

We finally present a primordial study on cooperative scheduling of diverse vehicular communication channels for a better dissemination of data in multi-hop routing scenarios. The system utilizes all possible channel modes in Dedicated Short Range Communication (DSRC) and RSU backbone interface cooperatively to maximize the delivery of data packets under delay tolerance and link connectivity regions. The model makes use of the global network view available at the control plane of SDVN and avoids possible packet collisions, multiple simultaneous transmissions, routing loops by means of constraints while scrutinizing the wireless nature of links. The primary problem is formulated as an Integer Linear Programming (ILP) problem and it further thrives on the minimization of service delay based on the traffic type as well. Moreover, the proposed model is easily extended for cooperative data communication under the RSU domain application scenario. Furthermore, we present two computational improvement strategies based on incremental optimization and Maximum Independent Sets (MIS) for the application scenarios, to shift the computational complexity to a realizable level for real-time communication. The effectiveness of the proposed framework is comparatively evaluated via multiple network instances with realistic mobility.

The proposed SDVN system model and information dissemination schemes high-

light the improvement that can be brought upon vehicular networks via SDN, which is the core objective of this thesis. This study further exhibits, unlike in VANET, how intelligent applications can be easily incorporated to exploit diverse features/attributes of vehicular networks for an enhanced performance.

Table of Contents

Acknowledgement	i
Abstract	iii
List of Figures	xiii
List of Tables	xv
List of Algorithms	xvii
List of Abbreviations	xix
1 Introduction	1
1.1 Background	1
1.1.1 Vehicular Ad-hoc Network (VANET)	1
1.1.2 Software Defined Network (SDN)	4
1.1.3 Software Defined Vehicular Network (SDVN)	5
1.2 Motivation	6
1.3 Objectives and Contributions	10
1.3.1 Architecture and Operation	10
1.3.2 Routing under Ideal Network Conditions	10
1.3.3 Routing under Uncertain Network Conditions	11
1.3.4 Routing and Scheduling	12
1.4 Outline of the Thesis	12
2 Literature Review	15
2.1 SDVN Architecture and Operation	15
2.1.1 Flexibility and Programmability	16
2.1.2 Cost Efficient Communication	18
2.1.3 Scalability	19

2.1.4	Global Network View	20
2.1.5	Latency	22
2.1.6	Characteristics of SDVN	24
2.1.7	Motivation from wired SDN	25
2.2	Routing	26
2.2.1	Conventional VANET Routing	26
2.2.2	SDVN Routing Protocols	27
2.3	Scheduling	33
3	Hierarchical Distributed SDVN	39
3.1	Introduction	39
3.2	Proposed Architecture and Operation	40
3.2.1	Hierarchical and Distributed Controller Architecture	40
3.2.2	SDVN Operation	44
3.3	Controller Placement Problem	47
3.3.1	Factors Influencing in Controller Placement	47
3.3.2	Delay Analysis	50
3.4	Proposed Controller Placement Model	56
3.4.1	Notations	56
3.4.2	Objective Function	59
3.4.3	Constraints	60
3.4.4	Solution Scheme	63
3.5	Results and Discussion	64
3.5.1	Simulation Model	65
3.5.2	Result Analysis	68
3.6	Summary	72
4	Routing in SDVN under Ideal Conditions	75
4.1	Introduction	75
4.2	Link Stability based Packet Routing Algorithm	76
4.2.1	Problem Statement	76
4.2.2	Problem Formulation	79
4.2.3	Solution Approaches	86
4.3	Source Routing based FI Scheme	93
4.3.1	Route Valid Time Estimation	93
4.3.2	Source Routing	98
4.3.3	OpenFlow Realization	98

4.3.4	Path Encoding	99
4.4	Performance Evaluation	100
4.4.1	Agility and Accuracy	100
4.4.2	Simulation Scenarios	101
4.5	Summary	105
5	Routing in SDVN under Uncertain Conditions	107
5.1	Introduction	107
5.2	Problem Statement	108
5.3	Problem Formulation	110
5.3.1	Notations	110
5.3.2	Centralized Unicast Routing	111
5.3.3	Distributed Broadcast Routing	115
5.3.4	Store, Carry and Forward (SCF) routing	118
5.4	Performance Evaluation	122
5.4.1	Simulation Configuration	122
5.4.2	Result Analysis	124
5.5	Summary	127
6	Cooperative Data Routing and Scheduling in SDVN	129
6.1	Introduction	129
6.2	Motivation and Preliminaries	130
6.2.1	Problem Motivation	130
6.2.2	Preliminaries	131
6.3	Proposed Routing and Scheduling Framework	133
6.3.1	Network Model	133
6.3.2	Objective	135
6.3.3	Constraints	136
6.4	Implementation Details and Solution Mechanisms	141
6.4.1	Data Dissemination under an RSU Domain	141
6.4.2	Solution Mechanisms	142
6.5	Performance Evaluation	147
6.5.1	Simulation Configuration	148
6.5.2	Results Analysis	150
6.6	Summary	153
7	Conclusion and Future Work	155
7.1	Conclusion	155

7.2 Future Work 157

Author’s Publications 161

Bibliography 163

List of Figures

1.1	A general VANET instance.	2
1.2	Internal SDN architecture.	4
1.3	Existing Internet based SDVN architecture.	6
2.1	Full SDN control vs Partial SDN control in SDVN.	17
2.2	SDN based VANET with fog computing.	23
2.3	SDN based Geo-broadcasting in vehicular networks.	31
2.4	Data dissemination under a single RSU coverage.	34
2.5	Content distribution in SDVN.	35
3.1	Proposed hierarchical distributed SDVN architecture.	42
3.2	Distributed SDN controller architecture with the hybrid data ownership model.	43
3.3	SDVN operation with fallback mechanism.	46
3.4	Analysis of controller selection based on location status.	49
3.5	Several IFS relationships [1].	52
3.6	Positioning of normal RSUs with respect to the controlling RSU.	58
3.7	Simulation environment – NTU campus map.	65
3.8	Controller placement results for the worst delay constraint.	66
3.9	Controller placement results for the average delay constraint.	66
3.10	Latency results under the worst case delay constraint.	69
3.11	Latency results under the average case delay constraint.	69
3.12	Performance comparison.	70
4.1	An instance of a vehicular network.	77
4.2	Graphical illustration of case 1 scenario.	84
4.3	Graphical illustration of case 2 scenario.	84
4.4	Graphical illustration of case 3 scenario.	85
4.5	Illustration of a collision situation under a packet gap of $2 * t_{hop}^{wl}$	85

4.6	Node exploration in Dijkstra algorithms (circles indicate the nodes explored in finding the shortest path).	91
4.7	Illustration of vehicle interconnection and velocity profiles.	93
4.8	Selection of candidate vehicles.	95
4.9	Projection of vehicle velocity.	96
4.10	Proposed FI operation.	98
4.11	Packet format of <i>FLOW MOD*</i> message.	100
4.12	Effectiveness of different solution approaches.	100
4.13	PDR comparison.	103
4.14	Latency comparison.	103
4.15	Experimental comparison of routing attributes.	104
4.16	Theoretical comparison of FI attributes.	104
5.1	Disturbance in network information.	109
5.2	Multiple wireless links in a selected path.	114
5.3	Inclusion of broadcasting in a route.	116
5.4	Destination in a black hole.	117
5.5	A general instance of two vehicular nodes in the network.	119
5.6	A path with updated travel times for SCF.	120
5.7	Inclusion of next hop information in Flow Mod packet (*please note that Tx Method only require 2 bits and the characters are for the explanation purposes only).	122
5.8	Simulation network (NTU, Singapore campus map).	123
5.9	PDR improvement with respect to applied treatments.	125
5.10	Effectiveness of the SCF scheme.	125
5.11	Performance comparison.	126
6.1	Example vehicular network instances.	130
6.2	Conflicts at packet reception.	137
6.3	Looping situations.	139
6.4	A vehicular network instance (Shaded rectangular boxes and empty circles next to nodes indicate the data items that the corresponding nodes acquired and required, respectively).	146
6.5	Generated adjacency graph based on the network instance in Fig. 6.4.	146
6.6	Simulated grid type vehicular network.	149
6.7	Performance comparison of different attributes in routing schemes (scenario 1).	150

6.8	Performance comparison of different attributes in data dissemination schemes (scenario 2).	151
6.9	Comparison of service delay based on traffic type.	152
6.10	Comparison of solution time and accuracy of results (<i>Proposed</i> refers to the models with solution approaches and <i>Global</i> refers to the global optimal solution achieved via Gurobi).	152
6.11	PDR with realistic mobility profiles.	153

List of Tables

3.1	Default EDCA parameter values in IEEE 802.11p [1].	53
3.2	Theoretical parameter values used in the delay analysis.	56
3.3	Notations used in the ILP model.	59
3.4	Simulation parameters of HD-SDVN.	68
3.5	A summary of comparison of SDVN systems with respect to key features.	72
4.1	A summary of notations	81
4.2	Simulation parameters of link stability based routing framework. . .	102
5.1	A summary of notations	111
5.2	Simulation parameters of connectivity aware tribrid routing frame- work.	124
6.1	A summary of notations.	134
6.2	Simulation parameters of routing and scheduling study.	149

List of Algorithms

1	Incremental packet allocation scheme	90
2	Computation of path (route) valid time	97
3	Selection of the agent node	117
4	SCF routing	121
5	Incremental optimization based solution approach at an intermedi- ate SP	144

List of Abbreviations

5G	5 th Generation
AC	Access Class
ACAR	Adaptive Connectivity Aware Routing
AIFS	Arbitration Inter-Frame Space
AODV	Ad hoc On-Demand Distance Vector
AoI	Age of Information
API	Application Programming Interface
ATFM	Air Traffic Flow Management
BS	Base Stations
BSM	Basic Safety Message
CCDF	Complementary Cumulative Distribution Function
CCH	Control Channel
CDD	Cooperative Data Dissemination
CDRS	Cooperative Data Routing and Scheduling
CRP	Centralized Routing Protocol
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CTS	Clear-To-Send

DMR	Deadline Missed Ratio
DOM	Data Ownership Model
DSR	Dynamic Source Routing
DSRC	Dedicated Short Range Communication
EDCA	Enhanced Distributed Channel Access
ESR	Effective Service Ratio
EU	European Union
FCFS	First Come First Serve
FI	Flow Instantiation
FLOW MOD	Flow Modification
GPSR	Greedy Perimeter Stateless Routing
HD-SDVN	Hierarchical Distributed Software Defined Vehicular Network
I2I	Infrastructure-to-Infrastructure
I2V	Infrastructure-to-Vehicle
IFS	Inter-Frame Space
ILP	Integer Linear Programming
IQP	Integer Quadratic Programming
ISP	Internet Service Provider
ITS	Intelligent Transport System
LIDAR	Light Detection and Ranging
LSP	Link Stability based Protocol
LTE	Long-Term Evolution
MAC	Media Access Control
MANET	Mobile Ad-hoc Network
MEC	Mobile Edge Computing

MIS	Maximum Independent Set
MOT	Minimum Optimistic Time
MRF	Most Request First
MWIS	Maximum Weighted Independent Set
NTU	Nanyang Technological University
OBU	On Board Unit
OLSR	Optimized Link State Routing Protocol
PDF	Probability Density Functions
PDR	Packet Delivery Ratio
QoE	Quality of Experience
QoS	Quality of Service
REST	Representational State Transfer
ROMSGP	Receive On Most Stable Group Path
RREP	Route Reply
RREQ	Route Request
RSU	Road Side Unit
RSUC	Road-Side-Unit Controller
RTS	Request-To-Send
SCF	Store, Carry and Forward
SCH	Service Channel
SD	Service Delay
SDN	Software Defined Network
SDNR	Service Delay of Non-Real-time traffic
SDR	Service Delay of Real-time traffic
SDVN	Software Defined Vehicular Network

SP	Scheduling Period
SPA	Shortest Path Algorithm/Approach
SPOF	Single Point Of Failure
TDMA	Time-Division Multiple Access
TTL	Time To Live
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VANET	Vehicular Ad-hoc Network
WAVE	Wireless Access in Vehicular Environments
WDC	Worst Delay Constraint

Chapter 1

Introduction

Extensive transportation expectations and advances in modern technology have led to rapid growth in the usage of vehicles in today's world [2]. With this rapid increase in the number of vehicles together with many auxiliary factors such as limited space in road infrastructure and bad driving habits, the transportation industry has been facing various problems such as an increase in traffic accidents, long traffic congestion, damages to public property and human lives, inability in finding parking spots, etc. [3]. In order to overcome these issues and to make the transportation systems more efficient, a new paradigm emerged nearly two decades ago named as Vehicular Ad-hoc Network (VANET).

1.1 Background

1.1.1 Vehicular Ad-hoc Network (VANET)

The primary objective of VANET is to build a proper coordination between vehicles through a distributed communication mechanism to reduce potential accidents and provide rich traffic information to passengers [4]. The main entities in a VANET are vehicle nodes and Road Side Units (RSUs), and they are further bridged by Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Infrastructure-to-Vehicle (I2V), and Infrastructure-to-Infrastructure (I2I) communications. Vehicle nodes are generally equipped with GPS devices and Dedicated Short Range Communication (DSRC) radios [5,6]. Recently, there have been many studies on integrating multiple wireless technologies such as 5th Generation (5G), Long-Term Evolution (LTE), WiMax, Wi-Fi, ZigBee, etc. in the On Board Unit

(OBU) of vehicles [4, 7, 8]. VANET employs a periodical broadcasting of Basic Safety Messages (BSMs - also referred to as hello messages) which is primarily used for the safety operations such as collision avoidance. VANET further holds the luxury of unique stationary RSUs in the system, whose purpose is to disseminate important information such as road conditions, traffic light status, etc. to passing vehicles and also to act as a relay station [9]. The base stations connected with RSUs will provide access to the Internet. A general VANET architecture is shown in Fig. 1.1.

Figure 1.1: A general VANET instance.

VANET can be regarded as a subset of Mobile Ad-hoc Network (MANET). But, in contrast to MANET, VANET exhibits several unique characteristics such as predictable mobility patterns, frequent disconnections, high topology changing rates, no power constraints, and stringent delay requirements. VANET has been often much larger than MANET in terms of network size and consists of thousands of nodes.

VANET was first introduced to reduce the amount of traffic accidents and increase road safety [4]. Since then, it has been getting increasing attention both from the industry and academia due to the unique challenges and opportunities it presents. Now, it covers a wide range of applications, including safety message

dissemination, traffic information distribution, collision avoidance, traffic guidance, parking assistance, entertainment, etc. With the increase in extensive social networking expectations and seamless Internet connectivity requirement, VANET is even expanding over conventional boundaries, addressing applications such as on-the-go Internet and mobile clouds [10, 11]. To discuss a few applications in detail; in blind spot detection, periodic broadcasting of beacon messages is used to identify vehicles, which are out of the line of sight, whereas, collision detection can be achieved through sensor data (Light Detection and Ranging - LIDAR) and similarly, through beacon messages.

VANET applications and services can be divided into three main categories based on their latency requirement. For eminent safety applications such as collision avoidance, blind-spot warning, and lane change assistant, the upper bound on latency requirement is around 100 ms [12]. These applications generally involve one-hop transmissions and supported by sensor data. The second type is cooperative traffic efficiency applications, in which, the upper bound latency is around 200 ms [12]. These applications are intended to provide additional information to enhance traffic management. They may include dissemination of traffic information, traffic guidance, electronic toll collection, parking spot finding, etc. The third type is infotainment and the applications under this category can tolerate a delay up to 500 ms [12]. They mainly address passenger comfort, Internet services, instant messaging, etc.

Even though the research on VANET has been going on for nearly two decades, it is still facing a lot of challenges such as inefficiencies in information dissemination and resource utilization, security and privacy issues, unbalanced flow traffic, unreliable connectivity, inability to cope with dynamic topology changes, which collectively result in low Packet Delivery Ratios (PDRs), high end to end delays and packet overheads, lack of support for Quality of Service (QoS) provisioning, etc. [4, 13].

Some of these problems are very difficult to solve with current VANET architectures. Most critically, current distributed VANET architectures do not provide a sense of programmability and very resistive against innovation [7]. Once the protocols are implemented, given the scale of vehicular networks, it is practically impossible to change the behavior of vehicular entities. A comprehensive reconfiguration in each entity is required in case of deploying new services and protocols, prompting serious doubts on the flexibility, manageability, and expandability of existing VANET.

1.1.2 Software Defined Network (SDN)

In order to address these problems and enable flexibility and programmability, researchers have recently focused on incorporating Software Defined Networking (SDN)¹ into VANET [7, 14]. SDN has been receiving an increasing attention in the networking community and it was brought forward, especially to provide programmability to the existing networks and encourage innovation [15]. Initial research on SDN was more focused on wired networks, especially on data centers. But now, there has been an increasing attention on using SDN in wireless and ad-hoc domains [16, 17]. Moreover, SDN is capable of handling the dynamic nature of networks with intelligent applications while lowering operational costs through simplified hardware and software.

Figure 1.2: Internal SDN architecture.

The main distinguished feature of SDN is the separation of the control plane from the data plane. All the control functionalities are taken away from the network devices and hosted in a logically centralized control plane [18]. A typical SDN architecture is shown in Fig. 1.2. The centralized control plane handles all the logic and the data plane, which is composed of network devices, forwards the packets based on the flow rules supplied by the controller. As shown in Fig. 1.2, on top of the control plane, the users can have their own applications and change the network behavior according to their requirements. This enables programmability in SDN

¹The term SDN is used to abbreviate both Software Defined Networking and Software Defined Network

which is lacking in the conventional networks. The interface between the control plane and the application layer is called as Northbound interface and generally utilizes the Representational State Transfer (REST) Application Programming Interface (API). The control plane may consist of multiple controllers cascaded or distributed. The distributed controllers communicate with each other via the East-West interface using RESTful or any similar mechanism [19–21].

If an SDN enabled network device does not possess information on handling an incoming packet, the packet will be encapsulated and sent to the controller as a PACKET IN message through a secure channel. This communication between data plane and control plane is executed through the southbound interface and the standard protocol is OpenFlow [22]. The controller will then analyze the PACKET IN message, and supply flow rules to the SDN switch via a Flow Modification (FLOW MOD) message. Overall, SDN positively disrupts what is used to be a closed, proprietary and flat network paradigm and thus, it is expected to have rapid innovations in the network industry with the openness SDN brings into the network [23].

1.1.3 Software Defined Vehicular Network (SDVN)

Similar to SDN, SDVNs decouple the control algorithms from the hardware, allowing modularity and programmability in vehicular networks. The main elements in an SDVN are the control and data planes, assuming the applications are integrated with control, as shown in Fig. 1.3. The SDN controller possesses all the logical functionalities and takes decisions, including routing, scheduling, access control, etc. on behalf of the data plane. The data plane consists of vehicles/OBUs (these two terms are interchangeably used throughout this thesis) and RSUs, and executes actions specified by the flow rules which are supplied by the control plane [18]. Generally, a data plane node (OBU or RSU) consists of two wireless interfaces, LTE and DSRC.

In SDVN, in addition to the periodical broadcasting of BSMs, the OBUs at vehicles have to send beacon/status messages to the controller periodically. These messages typically contain information related to node positions, velocities, directions and neighbors and they are used to build topology graphs. At the same time, the frequency of status update should be high enough to constantly have fresh network state information and to get accurate decisions from the controller.

Figure 1.3: Existing Internet based SDVN architecture.

1.2 Motivation

As highlighted in the background, VANET lacks flexibility and programmability and generally a closed network environment. This is problematic, especially at the implementation stage, as there is no open interface to deploy new services and protocols. Given the size of VANET, it is not possible to reconfigure all the devices. This is where the potential of SDVN is unfolded, as it offloads all or most of the control software to a logically centralized control plane and opens-up the network, allowing operators to test their solutions even after the deployment, as well as improving the management of network resources, applications and users [7].

However, the integration of SDN into the vehicular domain is conceptualized in different ways by different researchers, and yet, to find its ideal layout and division of functionalities preserving the unique characteristics of vehicular networks.

From an architectural point of view, distinctive placements of SDN controllers and ways of communications can enable various new features such as the global overview for intelligent decision making. But some of the features are achieved at the expense of several essential characteristics of VANET. Most of the existing SDVN architectures control the network via a single, centralized controller hosted on the Internet, as shown in Fig. 1.3. This type of configuration enables an environment to make better decisions and provides a gateway for application deployment. However, as the controller is usually far away from the vehicles, it

tends to produce a significant operational latency [24]. Therefore, for VANET applications with low latency requirements [12], such a configuration may not be viable. Moreover, as VANETs consist of thousands of mobile nodes, controlling them via a single controller is not advisable. This imposes a lot of stress on the controller, represents a Single Point Of Failure (SPOF) and furthermore, raises questions on the practicality and scalability of the system.

Furthermore, the operational procedure of SDVN also takes a toll on the system as it requires a periodical collection of network information from vehicles and RSUs in addition to the conventional broadcasting of BSMs. In terms of an operational perspective, it generates a considerable packet overhead and communication cost.

The above issues raise serious questions on the applicability of existing SDVN systems. Thus, the initial part of this thesis attempts to address them through a flexible and programmable SDVN system model, while preserving stringent requirements of vehicular networks.

SDVN not only tears up the closed distributed network structure enabling innovation but also brings out a new perspective through the availability of global network view, which can be utilized to exploit diverse attributes of vehicular networks for enhanced performance. Thus, the second part of the thesis investigates one such critical attribute, link stability, under ideal network conditions.

The links in vehicular networks are highly vulnerable in their existence and not stable for a considerable time due to the highly mobile nature of vehicles, in comparison to other ad-hoc networks [25]. This dynamicity in links is one of the fundamental concerning factors for information dissemination in VANET. It causes frequent topology changes, and as repeatedly emphasized in [25, 26], imposes serious difficulties in successfully delivering packets. This is crucial, especially in multi-hop data transmission applications such as file sharing (entertainment and information), traffic information dissemination, mobile vehicular cloud services, public transport scheduling, and congestion avoidance, which require transmissions of sequences of packets over multiple hops. As the time frame in the multi-hop transmission is fairly large, packets frequently find it hard to reach the destinations before the route times out.

Distributed VANET routing protocols struggle to cope with this dilemma as they find it difficult to accurately estimate the route validity due to the lack of global network information [27, 28]. Hence, the discovered routes may not stably exist until the packets get through, likely resulting in packet drops.

However, the introduction of SDN enables global network information at the control plane, which can be exploited for a better route selection by considering multiple factors simultaneously such as distance, valid time, reliability, etc.

In SDVN, routing consists of two parts, the routing algorithm and the Flow Instantiation (FI) operation. As the routing information is being generated at the controller, the way this information is delivered to the respective nodes is as equally important as the efficiency of the routing algorithm. One of the recent focuses in SDN is on minimizing the extent of communication between the control and data planes, as it disrupts the continuous packet flow [29–31]. In SDVN, this is more critical as the controllers are generally placed far away from the vehicles [7, 29].

Thus, one major motivation of the thesis is to develop an improved routing scheme that can resolve the aforementioned issues by using the novel perspective (bird's eye view) provided by SDVN and at the same time, design an FI operation that can balance the extent of communication with the control plane.

The link stability/connectivity can be accurately scrutinized only if a global image of the underlying network is logged in the logically centralized control plane. Yet, at the deployment, this is not always realized, as the control plane may not always possess updated and complete network information which is required to accurately find stable paths. Lack of cellular and infrastructure support (RSUs) in certain road areas (rural) can generate an unreliable southbound interface. The control plane may not possess network information related to these blind areas. Also, fast-changing road networks such as urban, makes the current network information obsolete very quickly. Therefore, the above conditions may develop difficulties in finding routes between given source-destination pairs and accurately estimating the stability of paths. These impracticalities in the network information should also be addressed in the routing framework to enhance its applicability.

Moreover, unlike other networks, vehicular networks can be dense or sparse depending on the location, time, etc. The same area could be dense at peak-hours and sparse at off-hours. However, current VANET routing protocols generally target a specific network condition.

To this end, another aspiration of this work is to make the routing protocol independent of the network conditions and articulate the generality through intelligent routing via SDVN.

With the recent boost in vehicular network-related applications and services,

VANET undergoes an exchange of enormous sets of packets between different entities. This encompasses transmissions of stacks of packets over single and multiple hops depending on the layout of the network. As links in vehicular networks are vulnerable in their existence, it is difficult to deliver all the packets in between source and destination pairs in a limited time. This is further challenged by the diverse deadlines of the packets. For instance, real-time traffic has a lower latency requirement and non-real-time traffic has a higher latency requirement.

The above scenarios together create contention for the access of channels. However, the channel access granting mechanism deployed by IEEE 802.11p, ‘Enhanced Distributed Channel Access’(EDCA) ignores these situations and grants channel access based on the timing of the Request-To-Send (RTS) / Clear-To-Send (CTS) packets [32]. Simply, it can be approximated to a First Come First Serve (FCFS) process, as first to request will get the channel access first. Still, due to the negligence of the global picture, which involves packet deadlines, link lifetime, etc. in granting channel access, it can cause some data packets to miss their delay tolerance bounds and some links to cease connectivity before delivering the packets. Furthermore, the packets could be queued up in nodes increasing the latency and possibility of packet collisions as well. Therefore, a better scheduling scheme is needed in general packet transmission scrutinizing the above issues and different channel types. In fact, the availability of diverse vehicular channels can be used for a better dissemination of data, as the differential characteristics of these channels generally allow multiple transmissions even inside the same transmission domain.

Moreover, routing and scheduling are separately addressed in VANET. Routing generally appeals to the selection of path(s) between the given source and destination pair, and the exact timings of the packets are not addressed. Scheduling generally refers to the sequential stacking of packets in queues for a better distribution among client vehicles. Hence, there is a need for a cooperative routing and scheduling framework exploiting the aforementioned parameters such as differential channel modes.

Another important aspect that should be considered in modeling the information dissemination problems and their respective solution approaches is the computational complexity. Especially, in vehicular networks, the solution time should be in the order of milliseconds for most of the applications to have a realistic realization. At the same time, the accuracy of the solutions should also be pushed closer to the global optimality as much as possible, and thus, this forms a commonly known trade-off between complexity and accuracy.

1.3 Objectives and Contributions

This thesis primarily focuses on four main aspects of SDVN which helps to enhance the information dissemination in vehicular networks.

1.3.1 Architecture and Operation

As depicted in the motivation, the fundamental objective in the integration of SDN and VANET is to enable flexibility and programmability while preserving the inherent characteristics of vehicular network. However, this is yet to be achieved in some forms, especially in terms of latency, in the existing literature. Thus, in the initial part of the thesis, we introduce a novel Hierarchical Distributed SDVN (HD-SDVN) architecture, in which the control plane is split into two layers with one layer brought down to the RSU level to comply with the stringent latency requirements of vehicular network applications and services, and at the same time enables the positives of SDN in vehicular networks, especially flexibility and programmability. The proposed SDVN architecture is scalable and the load is fairly distributed among multiple SDN controllers, removing SPOF. The controllers at the RSU level are placed with the objective of minimizing the communication delay and mathematically inspired by the p-median facility location problem. Concurrently, the controllers are able to collectively generate a localized global network view in order to take intelligent decisions.

The proposing operation adopts a proactive mode, and thus, less disruptive to the flow. The communication between data plane nodes and the control plane during the status collection and data transmission is kept minimized while extracting the necessary flow rules from the controller. This helps to reduce the communication cost (if cellular communication is used) and the latency in the operation significantly. Essentially, the focus is to keep the data flow in the data plane as much as possible without extending to the control plane.

1.3.2 Routing under Ideal Network Conditions

The primary objective of this part is to showcase the enhanced effect of the software defined approach on information dissemination in vehicular networks via an improved routing framework. Thus, we introduce an optimization based novel packet routing protocol for SDVN assuming a completely known deterministic network,

and considering multiple concerning factors, link stability² and path length. The aim is to select multiple shortest paths, which are collectively stable enough to deliver a given number of packets, while minimizing the extent of communication with the controller. The overall problem is modeled following a minimum cost capacitated flow problem, and linearized to an Integer Linear Programming (ILP) problem. Further, an incremental packet allocation scheme is introduced as a computational improvement strategy with the help of Bi-directional and K-shortest path algorithms [33,34].

As previously mentioned, an SDVN routing framework consists of a routing algorithm and also an FI scheme. For the FI scheme, we propose to adopt some of the unique features associated with conventional routing in VANET, and present a source routing based FI scheme with an effective route caching mechanism, which can efficiently deliver flow rules to the respective nodes with a minimum communication with the control plane.

1.3.3 Routing under Uncertain Network Conditions

We further consider the practical challenges in SDVN routing, which includes black holes in the network information, and different network conditions. As highlighted under the motivation, the former can impose challenges in finding stable paths and needs to be addressed to enhance the applicability of the routing protocol. For that, we incorporate broadcasting, a distributed routing technique in combination with a unicast protocol. The latter highlights the dire need to address specific network conditions in routing to establish a generality. Thus, we articulate a general vehicular network in routing, and additionally propose an improved scheme to find optimal Store, Carry and Forward (SCF) routes with least store time for sparse conditions. Referring to the above scenarios, we propose a tribrid routing protocol which consists of unicast, broadcast and SCF techniques for a general vehicular network. The focus is again on maximizing the delivery of packets while satisfying given latency based QoSs. The arising disturbances are treated with broadcast and SCF routing on a case by case basis.

²From link stability, we refer to the connectivity of links and also the associated dynamicity of links.

1.3.4 Routing and Scheduling

Availability of the global network view not only allows an opportunity to find optimal routes, but also optimal timings of packets that can enhance the overall efficiency of information dissemination within a vehicular network. Thus, we introduce a cooperative data routing and scheduling scheme with the help of SDVN for general multi-hop data transmission scrutinizing the different channel modes, packet deadlines, and link dwell times. In fact, the different channel modes (V2V, V2I, I2V, I2I) available in VANET are generally scrutinized in the Media Access Control (MAC) layer and avoided in the network layer applications such as routing and scheduling. However, in this study, we aim to show that when cooperatively scheduled, they can be assets, especially in terms of simultaneous transmissions. Later, we modify the proposed primary data routing and scheduling model to be applicable in the conventional data dissemination under RSU domains.

The problem is formulated as an ILP problem with the objective of maximizing the number of served/delivered data while minimizing their service delays. The time is considered in discrete steps, with the basic time unit as a Scheduling Period (SP). The link and data dwell times³ are implemented as constraints. Moreover, all the possible transmission conflicts such as packet collisions, multiple transmissions, looping conditions, etc. are constrained in the model. Further, we prioritize the real time traffic data over non real time data.

In order to reduce the associated computational complexity of the primary multi-hop routing and scheduling problem, we prioritize the packets based on their packet dwell times and traffic type, and introduce them incrementally into the problem, adhering to an incremental optimization approach. For the extended data dissemination under an RSU domain problem, we propose a Maximum Independent Set (MIS) based solution approach.

1.4 Outline of the Thesis

The rest of the thesis is structured as follows.

Chapter 2: Literature Review

In this chapter, we survey and review the existing related studies. Moreover, we expect to differentiate state-of-the-art studies in respective domains, and identify

³Link dwell time refers to the remaining life time of the link, whereas, data dwell time refers to the remaining deadline of the data packet.

the prevailing research gaps in them.

Chapter 3: Hierarchical Distributed SDVN

This chapter addresses the inefficiencies in existing SDVN system models and violations of intrinsic characteristics of vehicular networks through a novel HD-SDVN controller architecture and a proactive FI scheme. It further addresses the issue of the optimal placement of the SDN controllers and presents a novel controller placement model for the HD-SDVN system based on the delay parameters and significance of the RSU locations.

Chapter 4: Routing in SDVN under Ideal Conditions

In this chapter, assuming the availability of a complete global network image, we present an optimized routing framework for SDVN which collaboratively consists of a link stability based packet routing protocol and a source routing based FI scheme. This chapter also presents an incremental optimization based computational improvement strategy to have a lower solution time.

Chapter 5: Routing in SDVN under Uncertain Conditions

In this chapter, we drop the assumption of complete global network view, and address the disturbances in the network information and diversity in network conditions through a tribrid routing protocol, which consists of unicast, broadcast and SCF techniques. To this end, the proposed routing protocol entertains positives of both centralized and distributed routing techniques and shares the same objectives of the model in the chapter 4.

Chapter 6: Cooperative Data Routing and Scheduling in SDVN

In this chapter, we continue to expand the above routing framework by incorporating the timing dimension, and introduce a cooperative data routing and scheduling scheme which scrutinizes the different channel modes, packet deadlines, and link dwell times. The proposed model is also extended for the data dissemination under an RSU domain application scenario. Similar to above chapter, the proposed models are also associated with computational improvement strategies. For the former an incremental optimization approach and for the latter, an MIS based solution approach are proposed.

Chapter 7: Conclusion and Future works

We conclude the thesis in this chapter by summarizing the key contributions and drawing possible future directives.

Chapter 2

Literature Review

In this chapter, we make an extensive review of the existing literature relevant to SDVN and its information dissemination schemes. This includes studies on SDVN architectures and operations, SDN systems, FI schemes, conventional VANET routing protocols, recent SDVN routing protocols, cooperative scheduling schemes, etc. Through the survey, we expect to identify the state-of-the-art practices and possible improvement that can be made to have a better performance, in each domain.

The first paper on SDVN came up at the end of 2014 [7], and it was co-authored by a legendary figure in the networking domain, late Prof. Mario Gerla. SDVN is now becoming a hot research topic with the roll-out of the 5G road map [8] and evidently, with the extent of publications continues to surface in this area [35–39].

2.1 SDVN Architecture and Operation

The initial research focus of SDVN was on developing an effective SDVN architecture and operation which enables the essential features of SDN in vehicular networks [40]. Current distributed VANET architecture is highly resistive against innovation and thus, deployment of new services is really difficult. As mentioned previously, SDN simplifies the network management and thus, new services can be easily deployed and tested in vehicular networks.

The main entities in any SDN are the controllers and data plane nodes. Different SDVN architectures have placed the SDN controller in different positions and used various wireless communication technologies to connect the nodes and con-

troller. Based on the placement of the SDN controller and usage of communication technologies, they have been able to achieve different features.

2.1.1 Flexibility and Programmability

It is commonly known that the main stand out features of SDN are the flexibility and programmability [18]. Here, programmability refers to the capability to change and/or accept a new set of instructions in order to alter the functional behavior of a system, whereas, flexibility is the ability of a system to adapt and/or support unforeseen and/or new requirements/features (ex: new applications, protocols).

In existing systems, both hardware and software resides in the network nodes. So testing of new services and protocols require a comprehensive manual reconfiguration in each device. In networks like VANET which is composed of thousands of vehicle nodes, it will be a nightmare. But with the SDN, the control functions are taken away from nodes and located separately giving access to developers to test new services with ease, enabling flexibility and programmability. We can see that each and every paper that has presented SDVN architectures, has achieved these features fully [7, 8, 14] or partially [41, 42].

In most of the existing SDVN systems, the control functionality of both vehicles and RSU nodes are taken away, allowing a full SDN control over the network. But, several studies have only enabled these features in RSUs. We will first take a look at how the full SDN control is utilized in the proposed systems and then move on to the partial SDN control.

2.1.1.1 Full SDN Control

In an SDVN system, in addition to conventional VANET components, there is a main SDN controller residing on the Internet as shown in Fig. 2.1a. Vehicle nodes, RSUs, and Base Stations (BS), are all abstracted as SDN switches. In complete SDN control, the SDN controller provides the centralized intelligence and controls the entire network [7, 8, 14].

We can identify three operational modes in this type of architectures.

- **Central Control Mode:** Here, the connectivity with the controller is assumed to be stable. So the controller will supply exact flow rules and the nodes will forward the traffic based on the actions resides in the flow rules.

- **Distributed Control Mode:** This mode is triggered when the connectivity with the controller is lost. In this mode, [7] suggests the nodes to fall back to conventional routing protocols such as Greedy Perimeter Stateless Routing (GPSR) [43]. But in [14], the authors use trajectory prediction to estimate the possible positions when they are disconnected. Even though the disconnected vehicles' positions can be predicted using trajectory prediction algorithms, it is still unclear how the flow rules are supplied to those nodes from the controller. This creates an interesting research topic. But if the controller can successfully predict the positions of the disconnected nodes, it can be useful in packet forwarding, considering the disconnected nodes can act as intermediate switches.
- **Hybrid Control Mode:** In this mode, the controller supplies policy information rather than exact rules. So the nodes will change their behaviors based on the received policies.

These systems can be regarded as wireless SDN systems because the southbound communication is via a wireless channel. Usually the southbound communication utilizes either cellular or Wireless Access in Vehicular Environments (WAVE) wireless technologies, thus, it is more prone to data losses when compared with data center SDN.

Figure 2.1: Full SDN control vs Partial SDN control in SDVN.

2.1.1.2 Partial SDN Control

In partial SDN control, SDN has been applied only to the RSU backbone (OpenFlow enabled switches) as shown in Fig. 2.1b. Here, only wired SDN concepts

have been used. This is less prone to errors in the southbound due to its reliable wired connection. But, as the centralized controller only has control over the RSU backbone, it cannot control the entire network directly. The focus is only on controlling traffic among RSUs not between vehicles.

In this architecture, vehicles and RSUs communicate via IEEE 802.11 wireless technologies. RSUs are connected with OpenFlow switches through the Carrier Sense Multiple Access (CSMA) protocol. RSUs help vehicles to forward data and at the same time keep track of information on connected switches and vehicles. OpenFlow switches provide the interconnection between RSUs and form the backbone of the vehicular network. The global controller gathers topology and status information about the complete network and based on them, it supplies flow rules to the OpenFlow switches to manage the traffic flow.

Under this category, P. Duan et al. [41] and Yu-Chun et al. [42] have presented two similar SDVN architectures. Architecture wise, the main difference in these two studies is that [41] only proposes to control the OpenFlow switches, whereas [42] proposes to control both OpenFlow switches and OpenFlow RSUs.

In [41], when a vehicle requires a certain service, it first connects with a nearby RSU via broadcast. Upon reception of the data request, the RSU forwards it to the corresponding switch which it is connected to. Since switches collectively make the backbone, the polled switch deals with the packet accordingly or sends it to the controller based on the residing rules. In [42], the system uses a geo-information based constrained broadcasting approach to disseminate information. This will be discussed in detail in the routing subsection under the literature review.

2.1.2 Cost Efficient Communication

Most of the SDVN architectures propose to use multiple wireless interfaces in vehicle OBUs. Especially in full SDN control, vehicle nodes tend to use long range communication technologies in contacting the centralized controller on the Internet. Also, short range communication technologies are required for V2V and V2I communications.

I. Ku et al. [7] have proposed to use LTE or WiMax, long range communication technologies for control plane communication, i.e. between the SDN controller and data plane nodes (vehicles and RSUs). For data plane communication, they have proposed to use multiple high bandwidth Wi-Fi interfaces. Alternatively, H. Zongjian et al. [14] have proposed multiple different wireless interfaces such

as cellular, DSRC, and Wi-Fi for both control plane and data plane communications. However, the usage of multiple wireless interfaces in an OBU requires multiple bands in the frequency spectrum and thus, too idealistic. It would be more effective, if SDN can be adapted to existing vehicular network environment without using excessive resources such as additional expensive wireless interfaces and spectrum.

Z. He et al. [44,45] have extended these basic studies further with the objective of minimizing the communication cost. To achieve that, they propose to use multiple wireless interfaces as a combination. For an example, if a path exists between a source and a destination through multiple links with different wireless technologies, the controller can instruct nodes to use the path with the least transmission cost. The basic model assumes vehicles are equipped with Wi-Fi, Cellular and DSRC wireless devices as in [14]. Each of these communication mediums comes with a cost and it increases in the order of DSRC, Wi-Fi, and cellular. The usage of communication mediums is also bounded by the link availability. The problem is formulated as an optimization problem and the authors propose to use a greedy algorithm to solve the problem more efficiently than an exhaustive search.

2.1.3 Scalability

All of the above architectures are effective in terms of delivery of packets since in all the models, the control plane is centralized on the Internet as a single SDN controller and it has the complete global view which enables it to take better decisions. But this imposes a lot of stress on the controller, especially in networks like VANET because it consists of thousands of nodes. So it raises questions on the scalability of the system.

To overcome this problem, DeVANET [46] has proposed to use a distributed SDN controller architecture for SDVN. In [46], there exists separate controllers for each domain based on the geographical area in addition to a root controller in the system. However, all the controllers reside on the Internet. The presented architecture is much similar to Kandoo architecture in wired SDN [47]. Each controller has specific functionalities. Some of the domain controllers' functionalities are topology management, flow scheduling, mobility management, and connectivity management. The root controller is basically responsible for path computation, routing, and resource scheduling.

The objective of the aforementioned architecture is to uplift the scalability of

SDVN but, not to improve the performance. The packets generated at vehicles will go through RSUs using ad-hoc communication in contacting the controllers. Therefore, no long range communication links are required. However, in case of routing, the system might generate even a higher latency than centralized controller based SDVN because the requests have to go through many hops in the ad-hoc network and Internet Service Provider (ISP) network before reaching the root controller where the routing module resides. The resulting flow modification messages also have to undergo the same number of hops before reaching the vehicles. The results verify the stated problems with a delay profile of couple of tens of seconds, which is considerably higher, especially for safety applications.

2.1.4 Global Network View

Centralized control gives an edge to SDN systems over conventional distributed control systems in terms of the global network view. In conventional networks, nodes make decisions by themselves. But, they do not have a wide network view, rather they only know their neighbors. Similarly, in VANET, decisions such as routing are made based on broadcast mechanisms. So it cannot guarantee that the optimal path will be found.

But in SDVN, as depicted above, the centralized control plane gathers network information from the data plane nodes periodically via beacon messages. Therefore, the controller can build a better global image about the network. In fact, SDN provides a bird's eye view over the network to the controller. Therefore, the controller can, not only take accurate decisions, but also considers multiple factors such as link connectivity, trust, load, etc. in taking the decisions. All the single, centralized controlling SDVN architectures can easily achieve this property because all the beacon messages are delivered to one main data store.

However, distributed controllers find it difficult to generate a consistent global network view in each controller. For that, controllers need to have a synchronize network state. We can find several mechanisms related to this under the literature of data ownership models.

Data Ownership Models (DOMs): In the literature of distributed controllers in wired SDN, we can mainly find two DOMs depending on the location of the data store [48, 49]. The first one is single data DOM. Here, the data about the topology, switch status, etc. are stored in a single controller. Other controllers will access that data whenever needed or pass the routing decisions to that controller.

The second type is multiple DOM, where each controller will keep their separate data store inside them and share the information between controllers. Both DOMs have their own pros and cons. For an example, the single DOM represents a SPOF and but the system is very simple. On the other hand, the multiple DOM avoids SPOF and each controller can take their own decisions. But frequent information exchange between controllers can flood the east-west SDN interface, creating congestion in the control plane.

Age of Information (AoI): Similarly, the freshness of state information (age of information) also plays a crucial role in the accuracy of decisions and also helps to maintain a consistent global image over the network. State information usually has the highest value when it is fresh. For instance, real-time knowledge about the vehicular location and mobility will help immensely in estimating the residual lifetimes of the links.

This concept of AoI has recently emerged and now becoming a hot topic in a variety of fields including networking due to its applicability in a diverse set of applications [50]. AoI quantifies the freshness of the knowledge we have about the status of a remote system.

In SDVN, as pointed out above, it is desirable to have fresh network information at the control plane frequently. As this is accomplished through periodical beaconing, higher the beaconing frequency, fresher the information will be. However, when the beaconing frequency increases, the network gets congested, and thus, the space for data transmission depletes. A similar trade-off is identified in [51], where a source is transmitting packets containing status updates to a destination. Here, the utilization can be maximized by making the source send updates as fast as possible, similar to our case. However, it leads the destination to receive delayed statuses because messages are backlogged in the intermediate communication system. This delay can be minimized by reducing the rate of updates. Yet, it can also lead the destination to have outdated status information, and thus, an interesting trade-off emerges.

The same problem is addressed with respect to the broadcasting of BSMs in vehicular networks in [51]. Generally, the number of nodes in a broadcasting range is very high, and this can lead to congestion, undesirable levels of packet collisions and further to aged state information. The work in [51] has addressed this problem by balancing the optimal operating point that lies between the extremes of maximum throughput and minimum delay.

2.1.5 Latency

So far, all the presented SDVN architectures place the control plane on the Internet and thus, vehicles can use multiple methods to connect with it. However, there is a tradeoff between the communication cost and the delay, as some of the methods are high/low in terms of communication cost and low/high in terms of communication delay.

He Li, Mianxiong Dong and Kaoru Ota in [29] have proposed a method to balance this tradeoff between the cost and the latency of southbound communication by optimizing the usage of multiple wireless links. Their paper follows the full SDN control architecture with a single, centralized controller on the Internet. The vehicles can reach the controller either via cellular (5G) communication directly or using IEEE 802.11p (DSRC) technology via RSUs. The delay in reaching the controller via DSRC (ad-hoc) is very high because it needs to go through multiple hops to reach the controller. But the cost is very low because the vehicles do not have to pay for the packets transmitted through DSRC. In contrary, cellular links are expensive and each packet transmitted through cellular is charged at a fixed rate. But the delay is lower than DSRC. The paper proposes to use a hybrid method, where the low latency required packets are transmitted using cellular and others using ad-hoc. They further present a bandwidth rebating strategy to balance the cost and performance (latency) in communication between vehicles and controller. The bandwidth rebating problem is formulated as a two-stage Stackelberg game and the game equilibrium is being analyzed.

Moving further, Q. Zheng et al. [52] have presented a delay-optimal virtualized radio resource scheduling algorithm for SDVN via stochastic learning based on a distributed SDN controller architecture similar to DeVANET [46]. The purpose is to minimize the average delay of packets with optimal radio resource scheduling. The control plane allocates radio resources for the communication between road side infrastructures and vehicles considering the traffic parameters such as vehicle density in order to optimize the latency of queued packets. The delay optimization problem is formulated as an infinite-horizon average cost partially observed Markov decision process and then the equivalent Bellman equation is derived to solve the problem.

Even though the above studies propose mechanisms to optimize the latency the packets experience, still the placement of SDN controllers on the Internet, prevent it from lowering below a certain level. As mentioned above, irrespective of the

path packets take, they will have to go through the ISP network in reaching the controller. This generates a considerable amount of delay in control-data plane communication, prompting doubts on the usability of these architectures. In order to bring the latency to an acceptable level, especially for safety applications, the control plane should be fully or partially brought down from the Internet as much as close to the vehicles.

Figure 2.2: SDN based VANET with fog computing.

The authors of [24] have proposed to use the concept of Fog computing with SDN to bring a part of the control to the edge of the network. In Fog computing, services and data processing is done at the edge of the networks instead of sending to a central controller. This brings out numerous benefits such as low latency, location awareness, and improved QoS. The architecture is slightly different from the previous ones. As shown in Fig. 2.2, there is a centralized controller, which acts as the global intelligence. The paper introduces a new entity called SDN Road-Side-Unit Controller (RSUC). RSUC is also an SDN enabled device and controlled by the SDN controller. However, the SDN controller does not take full control over the network rather provides policy information to RSUC. Based on the received policy information, the RSUC decides the exact behavior depending on their local knowledge. Coordination between RSUCs is done by a Fog orchestration layer

located at the SDN controller. The RSUC forwards the data to the SDN controller and at the same time keeps a local storage. In case of an emergency service, RSUC serves the requests rather than sending to the controller, thus, reduces the latency by a significant margin. Yet, the above study has not been backup by any analysis or performance evaluation. Hence, further research is expected with comparative performance validations on essential networking metrics such as PDR and latency.

The same concern on the latency control is shared by [53], where they have highlighted the importance of an edge-up design (hierarchically distributed control with an edge layer), instead of a cloud-down design (Internet based SDVN). They have further reviewed the importance of latency with respect to existing studies, and also focused on the channel variations in terms of radio resource utilization.

2.1.6 Characteristics of SDVN

Apart from the above papers on SDVN architectures, we find another research [54] conducted on the characteristics of SDVN. For the analysis, the authors have used an empirical data set from a project in China in which the information is collected from thousands of taxis in 2007 in the city of Shanghai. They have characterized the inter-contact time and contact duration in between intersections and vehicles, and according to their evaluation, at the long run, both aforementioned metrics follow an exponential decay. But when the time is very small, the Complementary Cumulative Distribution Function (CCDF) of aggregate inter-contact time follows a power-law distribution ($x^{-constant}$). As the time increases, it follows an exponential decay function (e^{-t}). So a combination model of power law and exponential cutoff can better describe the inter-contact time. These results benefit SDVNs, especially in terms of controller placement. It provides a great incentive for using RSUs as SDN controllers.

Moving from the theoretical aspects, [55] and [56] have highlighted the importance of a practical validation of the SDVN concepts through OpenFlow controllers and OpenFlow switches. The latter has further incorporated “Mininet-WiFi” [57] in validating the SDVN performance, which is an extension of mininet for the wireless domain. Yet, doubts exist on the support of the aforementioned platform for the vehicular domain, given the heterogeneity of vehicular networks with its unique DSRC radios, fast fading channels, high dynamicity, etc.

Even though there have been a few studies focusing on architectural challenges of SDVN as depicted above, they have not been able to achieve most of the expected

features in an ideal SDVN system. In brief, an ideal SDVN system should be able to achieve flexibility, programmability, and other inherent advantages of SDN such as the global network view, while preserving the unique characteristics of vehicular networks such as low latency, scalability, etc. To this end, we make an effort to bridge this gap in chapter 3. A design motivation is taken from wired SDN. A few studies on the controller placement in wired SDN is discussed in the next sub-section.

2.1.7 Motivation from wired SDN

One of the pioneering works that highlighted the importance of controller placement in SDN is found in [58]. It opens up the investigation with two fundamental problems: given a topology, how many controllers are needed, and where should they go? Their goal was not to find optimal minimum-latency placements at scale, and rather, to examine limitations in controller placement through known public WAN topologies. They have highlighted that the main concerning metric for controller placement as latency, and similarly, state distribution options to fault tolerance are also influential.

In [59], given a set of switches, a mathematical model is proposed to determine the optimal number, location, and type of controller(s) and also, the interconnections between all the network elements. The goal is to minimize the cost of the network while considering different constraints such as the capacity of the controllers, the latency of path setup. The model works well with small scale networks. Yet, find it hard to cope with large scale networks, and thus, not applicable for vehicular networks.

Targeting large scale SDN networks, [60] presents a framework that returns Pareto optimal controller placements with respect to different performance metrics. In the default configuration, it performs an exhaustive evaluation of all possible placements. While this is practically feasible for small sized networks, time and resource constraints in large scale networks require an alternative. For these scenarios, the study proposes a heuristic approach, which is less accurate, but faster in terms of computation time.

Even though a motivation can be taken from them, none of them perfectly fit for the vehicular network domain due to its unique characteristics such as stringent latency requirement, high mobility, frequent topological changes, scale of the network, etc. In our work, we consider these unique characteristics in designing of the

network architecture, placement of controllers, operational procedure, and even in routing and scheduling.

2.2 Routing

Routing is a well-recognized subject area in vehicular networks. A substantial amount of studies exists in this area, and they address a diverse set of parameters that affect routing such as load balancing, trust, security, link connectivity, overhead, etc. [61–64]. However, we are mainly concerned about the link stability in this thesis as depicted under the motivation.

2.2.1 Conventional VANET Routing

Conventional VANET routing protocols such as AODV [27], Dynamic Source Routing (DSR) [28], and Optimized Link State Routing Protocol (OLSR) [65], lack global network information and generally rely on the information exchange between vehicles in finding the routes. Therefore, despite being scalable, they struggle to maintain a competitive performance, especially in terms of packet reception. In addition to above, many other protocols, including GPSR [43] suffer the same fate. This is repeatedly emphasized in previous research such as [64, 66]. Among various reasons, inconsideration of link/path stability can be rooted as one of the major causes for the above performance deficiency.

More recent studies such as [64, 67, 68], have focused on minimizing the disruptions caused by the dynamicity of links. In [68], the authors have proposed an Adaptive Connectivity Aware Routing (ACAR) protocol that makes a better selection of routes based on the network transmission quality through a statistical model. The statistical model takes into account on-the-fly vehicle densities and traffic light periods and estimates the probability of network connectivity and data delivery ratio for transmitting packets. Moreover, in each road segment of the chosen route, the most efficient multi-hop path is selected that is likely to improve the delivery ratio and throughput. In [67], the authors have employed a stable directional forwarding mechanism. They group nodes based on the positions and only the nodes in the same group participate in the route discovery to reduce flooding of requests. Route selection is based on link duration and thus, it ensures a more stable selection of routes. Similarly, in [64], vehicles are grouped depending on their velocity vectors and routes are formed within the vehicles in the same group which again ensures a more stable selection of links as these links are

moving together with the vehicles. Also, they have used a constrained broadcast based route discovery process to prevent broadcast storms. Overall, the protocol triumphs on selecting the most stable path based on the *Receive On Most Stable Group Path* (ROMSGP) algorithm. Among the studies that consider link connectivity, ROMSGP can be identified as the state-of-the-art. In addition to above, there exist several other routing protocols which consider link connectivity such as [69], which can be stated as a preliminary version of [68] and [70], and primarily targets urban vehicular networks. The link stability is scrutinized in CAR [69] for a certain extent by each node distributedly acknowledging the links' connectivity. Furthermore, it actively caches successful routes that are able to transmit data and also predicts the node positions via the information extracted from beacon packets.

The studies that consider link stability generally focus on finding the most stable path which can lead to longer routes and higher delays. Similarly, in [71], the relay node is selected based on the longest lifetime and additionally, the best signal quality. However, the model works in a reactive manner, i.e. the relay node is selected after receiving all the signals, and analyzing the aforementioned parameters based on the received signal's characteristics. Further, the above protocols do not consider the future connectivity of links and thus, suffers heavily under sparse network conditions.

The uniqueness in the routing protocols that targets sparse networks lies in the storing of packets until new connections are made. Generally, these protocols forward packets to the first encountered node rather than waiting for a more optimal forwarding [72]. This creates possibilities of longer routes or route failures. In contrary, studies such as [73] utilize a more optimal selection of forwarding nodes based on the vector directions of vehicular position and velocity. However, the literature on store and forwarding, and in general routing, still lack a comprehensive routing protocol that scrutinizes a wider network perspective, which we aim to fulfill through the availability of global network view via SDVN.

2.2.2 SDVN Routing Protocols

SDN simplifies routing in vehicular networks. Routing with SDVN is about finding a path(s) from a source point to a destination point in a connected graph viewing from the top. In other words, it is similar to solving a maze. The important factors are how fast and accurate the solving algorithm and what are the features

considered in solving the maze. Two main approaches are commonly known for this, exhaustive searching of the path in the graph until the best path is found or running a proper routing algorithm on the graph as most of the authors have done below. The first approach takes a very long time and complexity will be in $O(n^{constant})$ depending on the number of nodes and the structure of the graph. The second approach is less complex and takes a lesser time. But it is hard to guarantee that the optimal path will be found.

Similar to finding routes, the way this route information is delivered to the data plane (flow instantiation - FI) is also important. SDN is mainly associated with two FI operations, reactive and proactive [74]. In reactive controlling, when a packet is initially routed from a source to a destination, the nodes along the path who miss matching flow entries will request the controller for route information. During each request, the controller will compute the complete path, but only supplies the next hop information to the node that requested the information. This causes each and every node that does not have route information to buffer the packet until it receives next hop information from the controller. So the flow is frequently disrupted, which in turn generate a considerable amount of delay and routing overhead¹. To worsen this problem, the links in these paths generally have limited lifetimes due to inherent high mobility [25], and thus, the supplied flow rules expire quickly creating needs for repeated consultations.

The proactive FI mode avoids the buffering time by actively forwarding required flow rules to all the nodes in the computed path and thus, less disruptive. Even though it helps to reduce the latency, the routing overhead continues to prevail, since almost the same amount of communication exists between the control and data planes. This can lead to an expensive communication cost as well, if a cellular link is used for this interface as in [7]. Further, the proactive FI when used with distributed controllers [75] has struggled to deliver flow rules to all the required nodes, as some vehicle nodes may fall out of local controllers' reach, downgrading the performance.

M. Zhu et al. [63] have introduced a routing scheme for Software Defined VANET based on a centralized controller architecture as in [14]. They assume that vehicles are equipped with at least two network interfaces, one for vehicle-to-vehicle and another for vehicle-to-controller communication. In their Centralized Routing Protocol (CRP), when a vehicle receives a data packet from another vehicle or

¹The routing overhead is defined as the total packet size utilized in delivery of routing information to the required nodes in the respective path(s).

from its inside OBU, first it checks for a matching flow entry in its flow table as in wired SDN. If a match is found, the packet will be handled based on the corresponding action. If not, a 'Route Query' message, including vehicle ID and destination IP will be sent to the controller.

Routing server application in the controller then takes full control of routing and uses the Dijkstra algorithm with the path metric as Minimum Optimistic Time (MOT) to find the shortest path in between the given source-destination pair. MOT symbolizes the time it takes for the two vehicles to meet assuming both vehicles move in the best direction to meet each other. The computed shortest path information is then sent to the client vehicle through a 'Route Reply' message. Based on the received Route Reply, the vehicle switch can generate a route entry in its flow table. The route entry will be valid until the time specified by the controller through the Route Reply message and after the time expires the entry will be deleted from the table.

Going through the routing process, we can identify few drawbacks and possible improvements. Evidently, the process symbolizes a reactive FI scheme, and as mentioned above, this type of controlling can induce a higher latency in multi-hop data transmission since at each hop in the initial run, the packet will be buffered until the controller supplies the flow rules.

A potential future research that can be carried out is on finding the optimal valid time of flow rules. This is very important for highly dynamic networks like VANET due to the involvement of high mobility which results in frequent topological changes. But this is not true for all the road environments. For an instance, in highways, the topology can persist for a considerable time. So, the valid time of flow rule should be carefully analyzed with respect to the environment.

In M. Zhu et al. [76], which is an extension of their work on [63], they have presented another routing scheme, in which the controller only involves partially in the routing process. The routing protocol is based on multiple-copy routing, and in contrast to the existing work such as Spray-and-Wait, Spray-and-Focus and Geo-Spray, the authors have proposed a new protocol named Spray-and-Prey. The protocol uses the global network information the SDN controller possesses to evaluate the optimal carriers.

In the spray phase, the source node creates multiple copies of the data and distribute them among a set of neighbor nodes. The set of neighbors is selected with the help of the controller based on a utility function value, which basically reflects

the number of hops other nodes will take to reach to the destination. As the copies of the data propagate in the network, the controller continues to monitor the process and eliminate copies whose utility values are significantly lower compared to others.

The prey process is about redistribution of the copies. For that, the protocol adopts the natural prey process of wolves hunting where they surround the target in a circle. So in the prey process, the protocol adds an angle term in the utility function so that the utility function not only depends on the distance to the destination, but also on the sum of angles formed with the destination. This process continues until one copy reaches the destination. After that the controller clears out other copies in the network to save buffer space and bandwidth.

This protocol has an edge over other protocols because of the global network knowledge the controller possesses. It helps in selecting better carriers and clearing out unwanted copies to save resources. Yet, the involvement of the controller at each step can lead to higher delays in the overall process.

Moving onto routing in partial SDN control, Yu-Chun, Chien and Suchandra in their paper, have presented a Geo-Broadcast approach for SDVN [42]. The architecture adapted in this study is described in the partial SDN control subsection (Fig. 2.3). The routing is only applied to the data flow through the RSU backbone.

According to the European Union (EU) Intelligent Transport System (ITS) standards [77], every warning message received by an RSU will be routed to the control center in ITS. Then, the control center redirects the corresponding message to every other RSU within an interested geographical area of that information. For an example, an accident warning message will only be applicable to RSUs to the back of the road from that accident point, not to the front end of the road as shown in Fig. 2.3. After the applicable RSUs receive the warning message, they periodically broadcast it to vehicles passing through its communication range. But this approach generates a huge overhead in the control center and also consumes a high network bandwidth.

In order to overcome these problems, the authors of [42] have presented a new Geo-Broadcasting mechanism using SDN. In the proposed routing scheme, the warning message received by an RSU is sent to the controller as a PACKET IN message as in conventional SDN. Then, the controller analyzes the packet and set up the path using the topological information. The interested destination

Figure 2.3: SDN based Geo-broadcasting in vehicular networks.

RSUs and optimum paths are found based on the geographical information and status of the events such as emergency vehicle request for make way, accident warning, etc. The controller then continues to install flow rules in corresponding RSUs and intermediate switches to setup the path. After that, all the incoming packets destined to those destinations will follow the already setup path without the intervention of the controller. So this will significantly reduce the overhead at the controller and bandwidth consumption. Unlike in the previous papers, here the controller actively installs flow rules along the path (proactive FI), in which the packets are destined to travel. Thus, the average delay in data transmission is also reduced.

Opposite to previous works, B. Dong et al. [78] have used a combination of distributed and centralized routing algorithms in a distributed SDVN environment to transmit packets between vehicles. In their system, RSUs which act as the local controllers, route the packets within road segments using centralized versions of the Bellman-Ford and Floyd shortest path algorithms. Within a local controller's span, the packets are transmitted following an optimized AODV scheme. Overall, the proposed routing scheme has shown promising improvements over conventional routing protocols, but with performance and architectural limitations. In terms of performance, the routing overhead is still at large owing to the broadcast and reactive nature of the operation. Architecture wise, the system is proposed only

for the city environment with one-way lanes, and thus, still to be tested in general vehicular environments.

One recent SDVN study that has considered stability in its routing protocol can be found in [79]. However, the algorithm refers to the stability as the maximum connectivity of the path which is simply calculated based on the Euclidean distance. The protocol does not consider the wireless channel access constraints or the mobility associated with links, and primarily focuses on cognitive routing, where idle segments of the spectrum are incorporated into vehicular communication. Similarly, in [80], the link lifetime is taken as a unit of measurement in estimating a V2V path to offload existing vehicular traffic in the cellular network to the vehicular network. This is aided by a Mobile Edge Computing (MEC) - SDN layer and a main SDN controller. In their system, the vehicles frequently report their context at the MEC server and based on reported status, the SDN controller estimates viable V2V paths between nodes that are currently communicating through the cellular network. Yet, similar to the previous studies, the path with the maximum lifetime is used to construct the V2V path.

Based on surveyed literature in both conventional VANET and SDVN routing, it is apparent that none of the studies globally scrutinize link stability. The distributed routing protocols that acknowledge link connectivity tend route packets in the strongly connected path with possible performance drawbacks in terms of latency, as highlighted above. SDVN routing protocols have rarely considered link connectivity, amid the availability of a wider network view.

Link connectivity plays a major role in link stability, as longer connected links are generally considered as more stable. Still, link stability differs from link connectivity through the associated mobility of links. It is found that [6], links which are associated with higher mobility are prone to higher error rates in data transmission due to Doppler spread. Thus, we aim to incorporate both link connectivity and link mobility, in general, link stability, in our proposed routing protocol.

Moreover, we aim to acknowledge practical limitations in network information as well, which none of the SDVN routing schemes has recognized. It is highly probable that the control plane may not possess all the required network information due to architectural limitations, and even if the information is collected, they may age quickly. These scenarios need to be anticipated in routing to have a practical realization.

2.3 Scheduling

In VANET, packets' timing dimension is usually considered only under the domain of data dissemination by a service RSU, which is also referred to as cooperative data scheduling in the existing literature. Here, as vehicles pass through the RSU within a limited amount of time, delivering all the required data before the vehicles go out of the RSU's transmission range is very challenging, and thus, the timings/order of packet transmissions is vital.

FCFS can be referred as the simplest study in this area, and it broadcasts the data items in the same order the RSU received the requests [81]. Most Request First (MRF) can be stated as a slightly more effective scheduling scheme as most requested data items are broadcasted first, and thus, increases the number of served items [81]. Several other studies prioritize the data items based on their service time, and deadline [81]. Further, some studies combine multiple features together in a hybrid scheduling scheme with the focus on the order of the request, size of the data, deadline of the request, and waiting time [81,82].

More comprehensive studies are found in [9,83,84], where the authors consider multi-item requests under multi-RSU domains. In multi-item queries, a single request demands for multiple data items, and such a request can only be satisfied upon delivery of all the data items. This leads to the query starvation problem, which is caused by the unbalance between less popular queries and high popular queries in the same request. This problem is handled by several studies in the literature, including [83,84]. In addition to the prioritizing of packets based on their demands, which is a common practice in this domain, several studies such as [84] have used network coding to broadcast multiple data items in a single transmission, to address the above issue.

In the multi-RSU scenario, un-served or partially served vehicles under one RSU are served under successive RSUs. As the road layout is fixed, and if an RSU knows the route of a corresponding vehicle, it can get an evaluation on the successive RSUs that the vehicle is going to interact with in future time steps. Using this information the corresponding set of RSUs can cooperatively serve the vehicle in their respective time slots. The above phenomena is utilized in [9] through an enhanced cooperative load balancing scheme.

However, all these scheduling schemes assume that the data items are only stored at RSUs, thus, only utilize the I2V channel. K. Liu et al. [85] have highlighted

the advantage of using the V2V channel in collaboration with I2V in cases where neighboring vehicles also hold the data items, and this study is identified as the state-of-the-art in this domain. In the proposed system, the RSU acts as the SDN controller and at the same time as a data disseminator. An example scenario of data dissemination is shown in Fig. 2.4.

Figure 2.4: Data dissemination under a single RSU coverage.

Vehicles under the RSU coverage update the RSU with details of its current neighboring vehicles and identifiers of already possessed and requested data. The RSU then assigns roles to vehicles to maximize data dissemination without any collisions. Some vehicles will get the responsibility to transmit data using V2V to several other vehicles. Another set of vehicles will be asked to switch into I2V mode to be ready to receive the data which will be broadcasted by the RSU. V2V channel and I2V channel use different bands in the frequency spectrum, and thus, collisions are avoided. The goal is to maximize the number of vehicles that receive the requested data while avoiding collisions. The selection of vehicles and their roles (scheduling) are made by converting the problem into a Maximum Weighted Independent Set (MWIS) problem and using a greedy method to approximately solve the MWIS. When estimating the solution, vehicle's dwell time (the time remaining for the vehicle to leave the RSU coverage) is also taken into consideration.

The above model has demonstrated improved performance over other conventional I2V protocols in terms of both service rate and service delay. Yet, the existing

literature does not highlight a scheduling scheme for general multi-hop data transmission, which may span over infrastructure support-less areas. In this paper, we aim to bridge this gap by introducing a routing and scheduling scheme for multi-hop scenarios via SDVN with consideration on link lifetime, packet deadline and different channel modes.

Apart from above scheduling schemes, the existing literature also highlights several other studies, which we consider as motivational studies for our work. Moreover, few papers have been published on efficient content distribution in SDVN that do not necessarily fall into scheduling, but serve a similar purpose.

Among them, Yi Cao, Jinhua Guo and Yue Wu have presented a novel idea via SDN to support large scale content distribution in vehicular networks [86]. It uses the SDN concept to support efficient data replication and caching. The main objectives of their approach are to balance the reliability and redundancy of content forwarding and to handle the frequent disconnection problem relevant to content distribution. The problem is based on the architecture shown in Fig. 2.5.

Figure 2.5: Content distribution in SDVN.

In the presented scheme, first the vehicles who are interested in the content will send requests to the content server which is located on the Internet. The content server will include the ids of those vehicles in its subscriber list, and then, the respective content will be forwarded to network edge switches once. SDN agents

in edge nodes will determine the number of update copies that should be sent to RSUs according to subscribers' locations. After the copies reach the RSUs, the data will be broadcasted among subscribers. Among the vehicles who received data, several vehicles will be selected as re-broadcasters, whose communication ranges do not overlap with each other. The re-broadcast can send data to those vehicles who did not receive them in the first broadcast period. Similarly, the content will be spread among the subscribers. If there are vehicles that are not under the coverage of an RSU or moves out of the coverage of RSUs with partial data, they will be served using the cellular link.

However, the authors have not performed proper simulation on their proposed composition. So the arguments they made are still inconclusive without evidence. Further research can be carried out on how to satisfy the QoS requirement on different categories of content. For an example, soft real time data need quick response while it can tolerate some loss of data. So including a prioritization scheme to the proposed system can yield better performances in terms of differential QoS.

Taking a different approach, E. Bozkaya and B. Canberk in their paper, have addressed the problem of flow and interference management in vehicular networks using SDN [87]. The architecture this paper follows is similar to [41], where only the RSUs are managed by the controller (partial SDN control).

The fundamental problem here is that when the coverage of RSU is high, it causes interference in the network. At the same time it helps to disseminate more data flows, but with degraded Quality of Experience (QoE). When the RSU coverage is low with a limited number of vehicles connected, RSUs can supply services more effectively. But the number of satisfied vehicles will be less. The current implementation of the RSU coverage is static, which is highly inefficient in terms of resource usage. In order to handle the aforementioned issues, they have introduced two modules in the controller to handle flow and power. When vehicles are connected with an RSU, the new flow management module in the SDN controller evaluates the QoE level of each vehicle and identifies unsatisfied vehicles whose QoE levels are below a threshold.

After identifying unsatisfied vehicles, the power management module in the controller attempts to connect these vehicles to new RSUs, who can provide better QoEs to these vehicles. For that the new RSUs need to increase their transmission powers to get enough coverage to connect with these unsatisfied vehicles. To achieve that, first the controller needs to figure out the signal levels of unsatisfied

vehicles and calculate the required new transmission powers of RSUs. The authors use Kriging spatial interpolation method with exponential weight factors to address this problem.

In [88], the authors have used SDN in scheduling time slots of Time-Division Multiple Access (TDMA) based MAC protocols considering the vehicular network status. In their system model, they propose a novel entity called an OpenFlow enabled RSU switch/controller, which, when act as a controller, manages the vehicles, and when act as a switch, is controlled by a centralized OpenFlow controller. The SDN controller cooperatively assigns the time slots to vehicle nodes and decides their transmission ranges based on the network dynamics such variations in vehicle density and thus, brings agility and responsiveness into the MAC layer.

Apart from the all above SDVN studies, recently there have been several research such as [89] that reviews the existing work on SDVN and at the same time raises interesting questions on, to which extent, a vehicular network should be software defined to construct a flexible and programmable vehicular network while respecting its the unique features.

Chapter 3

Hierarchical Distributed Software Defined Vehicular Networks

3.1 Introduction

The fundamental objective of this chapter is to develop a novel SDVN architecture and operation with an optimized placement of controllers to enable positives of SDN including flexibility and programmability while preserving stringent latency requirements of vehicular network applications and services [75, 90]. We have highlighted the lack of such SDVN system model in the literature in chapter 2 (section 2.1) with respect to existing studies.

When applying SDN into the vehicular domain, it is critical to consider the unique properties and requirements associated with VANETs. Many VANET applications such as safety (delay < 100 ms), and cooperative traffic efficiency (delay < 200 ms), are delay sensitive [12]. Most of the existing centralized SDVN architectures have provided flexibility and programmability to VANETs at the expense of latency as comparatively analyzed by Deng et al. [91]. The latency generated in the operation is mainly dictated by the placement of SDN controllers relative to the SDN nodes and the ways of communication. Thus, a higher consideration should be given to these factors in order to make the SDVN a useful reality.

As a solution to some of the above critical concerns, especially to reduce communication latency, we propose to bring a layer of control down, closer to vehicles. Thus, in this chapter, we present a delay efficient SDVN system model consisting of a hierarchical distributed controller architecture in which the top tier of con-

trollers are placed on the Internet and the bottom tier of controllers are placed in several selected RSUs closer to vehicles so that the latency induced by the system/operation becomes low. Moreover, extending the proposed HD-SDVN system model, we further present a novel controller placement model for the RSU level controllers based on the p-median facility location problem with the delay and the significance of the RSU location as factors. With the help of simulation, we show that our proposed SDVN architecture, proposed operation and controller placement model generates a lower latency compared to existing SDVN architectures and also conventional VANET. To the best of our knowledge, this is the first piece of research work to present a hierarchical and distributed controller architecture for SDVN and the first to address the issue of the optimization of the SDN controller placement in SDVN.

The rest of the chapter is organized as follows. Section 3.2 presents the proposed SDVN architecture and operation. In section 3.3 and 3.4, we present the background related to controller placement problem, and then, the controller placement model, respectively. Section 3.5 discusses the results and finally, section 3.6 summarizes the chapter.

3.2 Proposed Architecture and Operation

The motivation here is to develop an SDVN architecture and operational procedure which consists of the following requirements.

- Lower route setup (time taken to discover/setup the route from a source to a destination) and end-to-end transmission delay (time taken to transmit a packet from a source to a destination).
- Use wireless technologies dedicated for VANET such as DSRC - (No extra spectrum usage and no high communication cost).
- Use a proactive and less disruptive FI operation.
- Provide flexibility and programmability to the system.

3.2.1 Hierarchical and Distributed Controller Architecture

It is clear that the controllers should be placed as close as possible to the SDN data plane elements when applying SDN into VANET to reduce the latency experience by vehicles. In fact, the current trend in the research of SDN is to minimize the

extent of communication between the control and data planes as much as possible and also to bring the control plane closer to the data plane [92]. Given that the VANET has the advantage of unique fixed RSUs, we bring the control plane down from the Internet to the RSU level as in [36]. The RSU level would be ideal for placing the controller(s) because it reduces the time to reach the control plane and also allows to use the existing DSRC wireless links for the southbound interface instead of expensive cellular links, saving not only communication cost, but also extra usage of the spectrum. Research conducted on the characteristics of communication contacts between vehicles and intersection [54] also promotes the usage of RSUs as SDN controllers.

However, unlike a single controller on the Internet, a single controller in an RSU cannot cover the whole network without inducing a very large latency. In opposite, a controller in each RSU will not hold an enough network view or controlling area for individual controllers to be effective. In this case, the controllers will be most likely handling single hop communication requests, which OBUs can already process by themselves via beaconing of BSMs periodically. Further, it will induce a higher system cost as the number of controllers is too high [93]. Therefore, we propose to place a set of limited controllers at several selected RSUs giving them enough controlling areas in the network to be effective, especially in multi-hop data transmission scenarios. Yet, we should make sure that the time to reach the control plane for OBUs should not go too high. This path leads to distributed controllers at the RSU level and it is similar to some of the distributed controller architectures presented in wired SDN like DISCO [21].

However, if we completely take the control plane down to the RSU level, the application developers will lose a secure gateway to implement new applications, thus, the programmability. One important feature in SDN is that it allows operators of the network to change the network behavior through application. But in order to do that there should be a secure northbound interface. Since application deployment is done in the Internet IP network, it requires controllers to be on the same IP network to build a secure channel between them. Moreover, as several VANET applications such as congestion avoidance and traffic guidance require a certain regional network view which includes traffic statistics over that area, to function, it too appeals for a centralization of the control plane on the Internet with a wider network view.

This brings a contradiction in placing controllers because one aspect (latency) requires the control plane closer to SDN data plane elements and another aspect

(applications) requires controllers on the Internet. By considering both aspects, we deploy a two tier control plane in which the top tier is placed on the Internet and the bottom tier is placed at the RSU level as shown in Fig. 3.1. The top tier and bottom tier controllers are connected through broadband links. In the top tier, regional controllers are placed distributedly on the Internet and connected through the Internet backbone. The bottom tier accommodates local controllers in several selected RSUs and connected through the RSU backbone via Ethernet. A local controller RSU is in fact a combination of a conventional RSU and an SDN controller. The data plane consists of OBUs and other non-controlling RSUs. Each OBU is equipped with a DSRC interface and a 4G-LTE interface. Here, the LTE interface will only be used in cases, where the RSU coverage does not exist (more details in section 3.2.2.1). In this way, the communication cost can be compensated in comparison to other SDVN systems with only a LTE interface.

Figure 3.1: Proposed hierarchical distributed SDVN architecture.

There are several duties assigned to the local and regional controllers.

- **Local Controllers** - They handle the requests which only require a local knowledge such as multi-hop data transmission within few hops and lane change assistance, without the intervention of the regional controllers. Therefore, OBUs can get a quick response and do not have to use the expensive LTE links. The requests which need a broader network view will be passed to the top tier.
- **Regional Controllers** - These controllers handle the requests which require a

certain regional network information such as vehicular statistics, road conditions, and traffic statistics. These requests are either redirected by the local controllers, or received directly from the OBUs which are not under the RSU coverage. Upon estimating the responses for the incoming requests, the regional controllers will pass them to the OBUs either via the local controlling RSUs or directly via LTE depending on the type of the request such as real time, non-real time, and the reachability of the OBUs such as their existence in an RSU coverage. These controllers are further responsible for synchronizing the bottom tier controllers and their maintenance. In case of a local controller brake down or malfunction, the regional controller(s) should activate contingency plan to transfer the OBUs and the RSUs currently under the faulty controllers to other local controllers or take over the duties of the malfunctioned controller by itself.

As depicted in chapter 2, we can find two DOMs depending on the location of the data store [48] in the literature of distributed controllers. In our system, we propose to use a *hybrid* DOM by combining the positives of the existing DOMs as shown in Fig. 3.2. Here, the RSU level controllers will keep their own data storage as in a multiple DOM and frequently update the top tier controller with the data as in a single DOM. Thus, the bottom tier controllers can make their own decisions if they have the information and if not they can pass the request to the top tier controller. Therefore, the RSU level controllers do not have to exchange information between them, leading to a less congestion in the RSU backbone.

Figure 3.2: Distributed SDN controller architecture with the hybrid data ownership model.

The resulting SDVN architecture resembles Orion [94] and [95] in the wired SDN, while the physical placement of controllers, interaction between the control and data planes and division of functionalities are unique to the vehicular networks. The overall architecture not only reduces the latency, but also adds scalability to the system. As the system does not depend on a single controller, SPOF can also be avoided. Moreover, the proposed architecture helps to apply SDN into VANET, without disrupting the existing vehicular network significantly. Vehicles only require existing DSRC technology to connect with controlling RSUs. Only several selected RSUs in which the controllers are to be installed, need to have access to the Internet where the top tier of the control plane is located. The exact placements of RSU level controllers will be analyzed in section 3.3 and 3.4.

3.2.2 SDVN Operation

3.2.2.1 Acquiring Network Information

As mentioned in the background, SDVNs require an additional periodical collection of network information which is the basis for the control plane in taking accurate and intelligent decisions. Most of the existing SDVN systems use cellular links for the above logging of network information and it generates additional communication overheads and costs. In our system, we take this pressure off the OBUs and assign it to the RSUs. Basically, the RSUs collect network information such as vehicle locations, velocities, etc. by eavesdropping on the conventional BSM broadcasting and generate local topology maps based on the retrieved information, estimated distances of vehicles and the knowledge that they have on the local road network. This topological information is first sent to the nearest local controller where it combines all the information received from RSUs and generate aggregated local area maps. Finally, they will be sent to the regional controllers where they generate the regional network maps. It is a sequential process which does not impose any additional responsibilities on the OBUs. However, the OBUs that are out of an RSU coverage have to follow the existing status update mechanism and send beacon messages to the regional controllers via the LTE interface. Here, the network information exchange frequency is set identical to BSM frequency, as only upon arrival of new BSMs, the network topology changes.

3.2.2.2 Proposed Operation

In a general operation, OBUs can find out their nearest local controllers via the BSM/hello message broadcasted by the RSUs as these messages additionally advertise information about nearby local controllers. For an instance, the hello message received by the blue vehicle in the left hand side of Fig. 3.1 from the RSU which is connected via a blue dotted line, indicates that the blue vehicle can reach a local controller with that particular RSU as the next hop. Thus, conventional RSUs act as the agents for the local controllers.

When vehicles are under RSUs' coverage, the OBUs use Vehicle-to-RSU (V2I) and RSU-to-RSU (I2I) links to connect with local RSU controllers. The same links will also be used in the data communication and thus, this is called as **in-band** controlling [96]. As there is no separate communication medium between the control plane and the data plane, it brings out several advantages such as usability of existing OBUs with minor modification, use of inexpensive communication links and lower delay in operation. We further propose to allocate a Service Channel (SCH) with a priority higher than other conventional SCHs but lower than the Control Channel (CCH) out of six service channels for this communication, as immediate safety matters over data communication [97].

A distance based simple handover mechanism is used when vehicles move between RSUs which are under different controllers. When vehicles move out of the RSU coverage, the OBUs on them use the dedicated LTE channel to connect with the regional controllers. As this channel is dedicated to the control-data plane communication, it is called as **out-of-band** communication [96]. However, the costs associated with these links and the delays associated with the placement of regional controllers on the Internet, impose limitations on their usability.

In the data transmission phase, an OBU sends a PACKET IN message to the nearest local controller when it receives any incoming packet with an unknown destination. Then, the local RSU controller or the top level controller depending on the type of request and availability of information, computes the required path¹ from the source to the destination.

Unlike in *reactive FI*, where the response is only sent to the requested node, here, the controller sends all the corresponding FLOW MOD messages which carry the

¹In this chapter, we use the Dijkstra algorithm to compute the shortest path. However, in future chapters, we present more sophisticated and improved routing protocols scrutinizing multiple parameters.

flow rules, to each node² in the computed route, including the source node in parallel, thus, adopts a *proactive FI*. As mentioned under the motivation, it can avoid buffering at intermediate nodes hence, less disruptive to the flow and the delay could be significantly lower compared to the reactive FI. Finally, the nodes upon receiving FLOW MOD messages, update their routing tables based on the flow rules residing in FLOW MOD and route packets toward the destinations.

3.2.2.3 Fallback Mechanism

There could be situations, where both RSU coverage and cellular coverage do not exist. To handle these situations, the OBUs can be equipped with the GPSR protocol [43], to act as a fallback mechanism as suggested by I. Ku. et al. [7]. The GPSR protocol does not need a route setup procedure, rather, the OBUs will send the data to the next OBUs based on the geographical location information. When the data is being transmitted through multiple hops, it is possible to initiate the SDN operation if the data is received by an intermediate OBU that has the connectivity with either a regional or a local controller. The Fig. 3.3 illustrates such a scenario with the sequence numbers representing the steps in the process.

Figure 3.3: SDN operation with fallback mechanism.

Here, vehicle A (OBU inside the vehicle) wants to send a packet to vehicle D. But, vehicle A is not within the coverage of an RSU or a base station of the cellular system. So, it can use the GPSR protocol and send the data to the OBU at vehicle B via the V2V channel based on the geographical information (step 1). Now, vehicle B has a connection with an RSU which it can use to initiate the SDN operation. Thus, vehicle B sends a PACKET IN message to the local controller requesting for route information (step 2). The local controller then computes the entire path and sends flow mod messages to all the OBUs in parallel (step 3).

²The term 'nodes' refers to both OBUs and RSUs

Upon receiving the route information, the OBUs can update their routing tables and forward the messages towards D (step 4).

3.3 Controller Placement Problem

In this section, we discuss the background related to the controller placement problem in detail including factors affecting the controller placement and theoretical analysis on the delay parameters which is required to formulate the problem.

3.3.1 Factors Influencing in Controller Placement

When placing local controllers at several selected RSUs, it is important to place them at optimal positions so that the latency experience by OBUs will be minimized. The controller placement problem depends on many factors, which has been analyzed in depth in wired SDN [58]. The factors related to controller placement in the SDVN are different from those in the wired SDN due to the unique characteristics of the VANET. Here, we discuss several important factors which will be considered in the proposed model. We assume that the SDN controllers, including local and regional controllers, can handle all the receiving requests because the current commercial SDN controllers are capable of handling millions of client devices simultaneously [98]. However, as the number of clients (or requests) increase, performance degradation, especially in terms of responsiveness, can be expected.

3.3.1.1 Number of Controllers

One trivial way is to place controllers at each RSU so that the OBUs can directly contact the controllers. Since the controlling RSUs consist of two interfaces as DSRC and LTE, they are generally costly and complex. Obviously, placing the controllers at all RSUs increases the cost and the complexity of the total system. At the same time, one RSU might not possess the minimum required global knowledge on the topology which is required to take accurate routing decisions. Therefore, most of the PACKET IN messages might have to be diverted into the top tier controllers which again induce a higher latency. Also, high speed vehicles will only stay in an RSU coverage for a short amount of time and it will lead to frequent handovers between the controllers, which is not desirable. In fact, a study has been conducted on the characteristics of communication contacts between vehicles and intersection for SDVN as depicted in the literature review. According

to [54], the CCDF of contact duration is given as follows,

$$f_{contact\ duration}(x) = b_1 * e^{-b_2 x} \quad (3.1)$$

where $b_1 = 1.4056$ and $b_2 = 0.7250$.

From the above equations, we can derive the average contact duration as follows.

$$\begin{aligned} E[x] &= \int_0^{\infty} x * b_1 * e^{-b_2 x} dx \\ E[x] &= \frac{b_1}{b_2^2} \end{aligned} \quad (3.2)$$

$$\text{Avg. contact duration} = E[x] = 2.6741 \text{min.}$$

This derived experimental contact duration between vehicles and RSUs in urban areas is a moderate value with an average of around 2.67 minutes. Yet, it is always better to have a higher contact duration with the SDN controller to avoid frequent handovers. So in terms of number of controllers, it is desirable to have less number of controllers with a higher controlling area.

3.3.1.2 Latency

In contrary, if few controllers are used, several OBUs which are far away (in terms of hops) from the controlling RSUs may undergo a considerable amount of delay in route setup. Therefore, when placing the controllers, it is vital to minimize the delay experienced by the OBUs because many VANET applications are delay sensitive. It causes for a trade-off between the number of controllers and latency.

3.3.1.3 Geographical Location and Vehicular Parameters

The geographical location and vehicular parameters associated with an RSU are also important considerations in placing the controllers. Higher priorities have to be given to the RSUs located near intersections in placing controllers, because vehicles usually tend to stop at intersections due to high traffic, and traffic light systems, if present. Also, the speeds of vehicles are not very high at these intersections. Basically, vehicles tend to use a considerable amount of time passing through the intersections with respect to other locations. So, if the controllers are placed at intersection RSUs, the OBUs at vehicles will have a higher direct contact with the controllers allowing them a better service in terms of delay.

It is vital to avoid placing controllers in the RSUs located at unrestricted roads with a higher mobility, because various studies have shown that high mobility affects the QoE of vehicles negatively [99]. The ideal locations for placing controllers are the locations where vehicles display a lower mobility.

3.3.1.4 Neighboring RSU's Location Status

In addition to above facts, it is equally important to consider the distance based status of the neighboring RSUs whom could be assigned to the considered local RSU controllers in deciding the controller placement. Consideration given to neighboring RSUs should scale down as the number of hops increases, i.e. high priority should be given to one-hop neighbors over two-hop neighbors.

As an example in Fig. 3.4³, candidate RSU 1 is located at a 4-way intersection while candidate RSU 2 is at a 3-way intersection. If we only consider the previous controller selection factor, which is the significance of geographical location, the selection for the controller would be candidate RSU 1. However, three out of candidate RSU 1's one-hop neighboring RSUs are in relatively unrestricted roads without intersections or traffic lights, where vehicles move fast. The vehicle OBUs within the coverage of these three RSUs which will be connected to the controller in RSU 1 (if selected) will not get a satisfactory QoE due to the high mobility of vehicles in these areas. So a significant portion of vehicles who are under one-hop neighbors will not get a satisfactory service. On contrary, if we place the controller

³We only consider one-hop neighboring RSUs for the purpose of simplicity in this explanation

Figure 3.4: Analysis of controller selection based on location status.

in candidate RSU 2, not only the vehicles under RSU 2 will get better QoEs, but also vehicles within its three one-hop neighboring RSUs which are located near multi-road intersections, will also experience better services. Basically, RSUs with high vehicular mobility around them should be connected to controlling RSUs as N-hop neighbors with high N values.

3.3.1.5 Workload Distribution

As discussed above, it is also important to evenly distribute the workload among local RSU controllers, so that the individual controllers will not be too stretched. This is vital especially in highly dense road networks such as urban. The number of normal RSUs that a controlling RSU controls can be taken as a measure of its workload as higher the number of RSUs, larger the coverage area, and more the number of vehicles.

3.3.2 Delay Analysis

The controller placement problem is associated with the delay generated over the links. Most of the time, PACKET IN messages transmitted by OBUs go through both wireless channel (V2I) and wired channel (I2I) before reaching the controller. Therefore, we need to model the delay components over the wired and wireless links in order to evaluate the total delay which we should minimize in placing the controllers. Therefore, in this subsection, we model the delay parameters associated with SDVN. For that, we follow the work of [100], where the delay associated with IEEE 802.11 networks have been analyzed in general.

Delay is an additive metric and thus, delay along a path is equal to the sum of delay components in each hop in a multi-hop data transmission. The total delay, D , in a single hop can be represented as in (3.3).

$$D = t_{trans} + t_q + t_{cont} + t_{proc} + t_{prop} \quad (3.3)$$

where t_{trans} is the transmission delay, t_q is the queuing delay, t_{cont} is the contention delay, t_{proc} is the processing delay and t_{prop} is the propagation delay.

3.3.2.1 Transmission Delay

Transmission delay refers to the time taken to transmit a complete packet on to the channel and it depends on the packet size and the data rate of the interface

as shown in (3.4).

$$t_{trans} = \frac{L \text{ (packet size)}}{R \text{ (data rate)}} \quad (3.4)$$

3.3.2.2 Queuing Delay

We assume that each OBU and RSU maintain a single finite queue, and it can be modeled using a M/M/1/N queuing model. The reason for using such a simple model is to evaluate the queuing delay at an upper bound so that our systems will be tested better. The packet arrival rate is assumed to be a Poisson distribution with a mean packet arrival rate λ and a service rate μ . So the delay due to queuing can be represented as in (3.5) with N as the maximum queue size and $\rho = \lambda/\mu$.

$$t_q = \frac{\rho[1 - \rho^N - N\rho^{N-1}(1 - \rho)]}{\mu(1 - \rho)(1 - \rho^{N+1})} \quad (3.5)$$

3.3.2.3 Contention Delay

In the proposed architecture, packets go through different contention periods depending on the MAC protocols. V2V and V2I communications use IEEE 802.11p standard in which the underlying MAC protocol is essentially CSMA with Collision Avoidance (CSMA/CA). Between RSUs, the communication is through Ethernet and controlled by the CSMA with Collision Detection (CSMA/CD). In order to model the contention delay for the two MAC protocols, a similar approach has been taken as in [100].

Let x be the number of re-transmissions suffered by a given packet, p be the collision probability of the channel and c be the maximum number of attempts before the packet is dropped. The re-transmission probability is given by,

$$P(x = k) = \begin{cases} p^k(1 - p), & k \leq c - 1 \\ p^k, & k = c \\ 0, & k > c \end{cases} \quad (3.6)$$

The expected number of re-transmissions, $n = E[x]$, for a given frame is,

$$\begin{aligned} n &= \sum_{k=1}^c kP(x = k) \\ n &= \left[\sum_{k=1}^{c-1} kp^k(1 - p) \right] + cp^c \end{aligned} \quad (3.7)$$

CSMA/CA - The IEEE 802.11p standard employs EDCA as its MAC protocol with the same nature of the CSMA/CA to provide channel access. Here, a node willing to transmit a packet will sense the medium first before transmission. If it is free for an Arbitration Inter-Frame Space (AIFS) [Access Class (AC)] time period, it will start to transmit the packet immediately. There are several Inter-Frame Space (IFS) parameters associated with EDCA [1] as shown in Fig. 3.5.

Figure 3.5: Several IFS relationships [1].

If the channel becomes busy during the AIFS, the node will defer the transmission to backoff. In backoff, the node first selects a random backoff from the interval $[0, CW - 1[AC]]$ where the initial Contention Window (CW) equals $CW_{min}[AC]$. The CW size doubles, if the subsequent transmission attempt fails until a maximum value of $CW_{max}[AC]$. Backoff value will decrease by a slot time, if the channel becomes free for a backoff slot time. When reaching a backoff value of 0, the packet will be sent immediately. After $c(= \log_2(\frac{CW_{max}}{CW_{min}} + 1))$ unsuccessful attempts, the node finally drops the packet.

In order to ensure differential QoS, the IEEE 802.11p MAC protocol accounts for the priority of the messages using different ACs. There are four available traffic categories with different priorities: Background traffic (BK or AC0), Best Effort traffic (BE or AC1), Video traffic (VI or AC2) and Voice traffic (VO or AC3). Different AIFSN (Arbitration Inter-Frame Space Number) and CW values are chosen for different ACs, as illustrated in Table 3.1 [1]. For control packet transmission such as Route Request (RREQ) and Route Reply (RREP) packets, we assume the nodes use the highest priority queue (VO class).

In order to simplify the analysis, we assume that RTS and CTS messages are not used to access the channel.

Table 3.1: Default EDCA parameter values in IEEE 802.11p [1].

AC	CW _{min}	CW _{max}	AIFSN
BK	15	1023	9
BE	7	1023	6
VI	3	15	3
VO	3	7	2

So to find the contention delay, first we need to calculate average number of backoff slots. Here, we take the mean value of random backoff time at each contention window as $CW/2$. The average number of backoff slots can be modeled as,

$$\begin{aligned}
 \overline{\text{backoff}} &= \sum_{k=0}^c P(x=k) \frac{\min(CW_{max}, 2^k CW_{min})}{2} \\
 &= \left[\sum_{k=0}^{c-1} p^k (1-p) \frac{2^k CW_{min}}{2} \right] + p^c \frac{CW_{max}}{2} \\
 &= (1-p) \frac{CW_{min}}{2} \left[\sum_{k=0}^{c-1} (2p)^k \right] + p^c \frac{CW_{max}}{2} \\
 &= (1-p) \frac{CW_{min}}{2} \frac{1 - (2p)^c}{1 - 2p} + p^c \frac{CW_{max}}{2}
 \end{aligned}$$

But, $CW_{max} = 2^{(c-1)} CW_{min}$,

$$\overline{\text{backoff}} = \frac{CW_{min}}{4} \left[\frac{2 - 2p - (2p)^c}{1 - 2p} \right] \quad (3.8)$$

There are two main reasons for the initiation of the backoff process in CSMA/CA protocol. The channel could be occupied by a transmission initiated by the same node from different queue or the channel could be occupied by a transmission initiated by a different node within its transmission range. This also depends on the Access Class of the packets. If the packets are from the Voice Class (Highest Priority), then, the probability of channel becoming busy during AIFS[AC] is quite low because the only interference can come from other node's Voice queue. A detailed analysis can be found in [100].

However, for the simplicity of analysis, it is assumed that there is a 50% probability for the channel to become busy during AIFS[AC] period. Also, following the work of [100], the mean value of AIFS[AC] is taken as the start point of the backoff process in the derivations. So, the final expression of the contention delay incurred

by the EDCA protocol can be stated as follows.

$$t_{cont} = \frac{3 * AIFS[AC]}{4} + \frac{\overline{\text{backoff}}[AC] * T_{slot}}{2} + nT_c \quad (3.9)$$

where T_c is the collision duration and T_{slot} is the slot time.

CSMA/CD - In the CSMA/CD protocol, an RSU who is willing to transmit a data, continuously senses the medium until it becomes idle (1-persistence). When the medium becomes idle, it immediately transmits the data and continues to sense the medium for possible collisions. If a collision is detected, the RSU jams the network and defers the process to backoff.

In the backoff, as in the CSMA/CA scheme, a random backoff slot is selected from a $[0 : CW - 1]$ interval where $CW_{min} = 2$. After each consequent collision, the contention window size doubles until a max of $CW_{max}(= 1024)$. The packet is dropped after $c(= 16)$ unsuccessful attempts. The expected backoff value of CSMA/CD protocol can be modeled in a way similar to that of the CSMA/CA protocol as follows.

Let, c be the maximum number of attempts and $c'(= \log_2(\frac{CW_{max}}{CW_{min}}) + 1)$ be the number of attempts until CW_{max} . The expected number of backoff value is given by,

$$\begin{aligned} \overline{\text{backoff}} &= \sum_{k=1}^c P(x=k) \frac{\min(CW_{max}; 2^{k-1}CW_{min})}{2} \\ &= \left[\sum_{k=1}^{c'-1} p^k (1-p) \frac{2^{k-1}CW_{min}}{2} \right] + \\ &\quad \left[\sum_{k=c'}^{c-1} p^k (1-p) \frac{CW_{max}}{2} \right] + p^c \frac{CW_{max}}{2} \\ &= p(1-p) \frac{CW_{min}}{2} \left[\frac{1 - (2p)^{c'-1}}{1 - 2p} \right] + \\ &\quad p^{c'} (1-p) \frac{CW_{max}}{2} \left[\frac{1 - p^{c-c'}}{1 - p} \right] + p^c \frac{CW_{max}}{2} \\ &= p(1-p) \frac{CW_{min}}{2} \left[\frac{1 - (2p)^{c'-1}}{1 - 2p} \right] + p^{c'} \frac{CW_{max}}{2} \end{aligned}$$

But, $CW_{max} = 2^{(c'-1)}CW_{min}$,

$$\overline{\text{backoff}} = p \frac{CW_{min}}{4} \left[\frac{2 - 2p - (2p)^{c'}}{1 - 2p} \right] \quad (3.10)$$

Therefore,

$$t_{cont} = \overline{\text{backoff}}[AC] * T_{slot} + n(T_c + IFG) \quad (3.11)$$

where IFG represents the Inter-Frame Gap which allows the RSU to switch from the transmit mode to the receive mode.

3.3.2.4 Processing Delay

The theoretical processing delay depends on the number of instructions in the corresponding process and the speed of the processor. Since the OBU and SDN controller processor speeds are around several hundreds of MHz and GHz, respectively, the theoretical delay calculation leads to negligible values. Even though several studies such as [101], have numerically evaluated the centralized SDN controller latency to be in $100 \mu s - 10 ms$ range, it is not suitable to incorporate them in our theoretical calculations as it may lead to ambiguous results. It should be further noticed that the chapter proposes a distributed architecture with fair workload distribution among controllers. Thus, the processing delay in reality would be a very low, especially, given that the existing controller hardware can support millions of client devices, simultaneously [98].

3.3.2.5 Propagation Delay

The propagation delay depends on the distance between the RSUs and the speed of the signal propagation as in (3.12).

$$t_{prop.} = \frac{d \text{ (distance)}}{v \text{ (channel speed)}} \quad (3.12)$$

The parameter values considered in the theoretical calculations are tabulated in Table 3.2. It should be noted that the tabulated values such as distance and packet rate, are of a general VANET instance and does not dictate nor reflect the overall operation of the vehicular network. The distance between vehicles and the vehicle-RSU in the routing path is considered as 100 m, so as to reflect the closest controller placement in a 200 m transmission range network.

3.4 Proposed Controller Placement Model

In this section, we derive the proposed controller placement model as a modified p-median facility location problem which is known as a NP-hard problem [103]. One important assumption is made which is that the RSUs are located uniformly over the road network with nearly equal distances between them.

3.4.1 Notations

3.4.1.1 Network Map

Lets assume that there is a given network graph, $G = (V, E)$, and a distance matrix D_S , where $d_{ij} \in D_S$ represents the distance from i^{th} RSU to j^{th} RSU in terms of number of hops. The set V represents the RSUs interconnected by the edges in E . The terms r and k represent the total number of RSUs and the number of controlling RSUs, respectively.

3.4.1.2 RSU Coefficients

The set, $C_i = \{C_1, C_2, \dots, C_r\}$ refers to the set of coefficient values of RSUs where $C_i \forall i \in V$ is assigned depending on the RSU's strategic location and vehicular statistics around the location as in (3.13).

Table 3.2: Theoretical parameter values used in the delay analysis.

Parameter	Value
Distances	Distance between two vehicles and between a vehicle and RSU in the routing path = 100 m, Distance between neighboring RSUs = 400 m
Propagation Speeds	Wireless channel = $3 * 10^8 ms^{-1}$, Wired channel = $2 * 10^8 ms^{-1}$
Data Rates	DSRC = 6 Mbps, Ethernet = 100 Mbps
Packet Sizes	$L_{RREQ} = 24$ bytes, $L_{RREP} = 20$ bytes [27], $L_{Packet_In} = 32$ bytes, $L_{Flow_Mod} = 56$ bytes (OpenFlow 1.3 [102]), $L_{Data} = 1000$ bytes, $L_{Beacon} = 400$ bytes
Queuing Parameters	$\lambda = 50$ packets/sec., $N = 15$, $L = 400$ bytes
CSMA/CA Parameters	$T_{slot} = 13 \mu s$, $SIFS = 32 \mu s$, $p = 0.3$, standard contention window values [100]
CSMA/CD Parameters	$T_{slot} = 51.2 \mu s$, $IFG = 9.6 \mu s$, $p = 0.3$, standard contention window values [100]

Let R_n be the number of road segments passing through the RSU, V_{vp} be the average vehicle passing rate and V_{vs} be the average vehicle speed. Then the RSU weight coefficient is given by,

$$C_i = p * \frac{R_n * V_{vp}}{V_{vs}} \quad (3.13)$$

where p is a negative constant.

Incentive behind above weight assignment is to reward the selection of candidate RSUs as controllers which are located in high traffic areas (Ex: candidate RSU 1 in Fig. 3.4). If the candidate RSU is located in a multi-road intersection with a relatively higher vehicle passing rate and a lower average vehicle speed, then placing a controller at this candidate RSU can be beneficial in terms of latency because more OBUs get to directly consult the controller. The RSUs at an intersection with a higher number of roads connected to the intersection will have a high priority. For an example, a higher priority is given to a 4-way intersection over a 3-way intersection as indicated by (3.13). Further, it builds up a stable southbound interface (interface between the control and data plane) with the controller for a considerable time, as vehicles move slowly in these areas.

3.4.1.3 Delay Parameters

D_{PI} and D_{FM} are used to indicate the time to transmit a PACKET IN message and a FLOW MOD message, respectively. These values are calculated based on the previous delay analysis, and can be formulated as follows.

$$D_{PI|FM} = (t_{trans} + t_q + t_{cont} + t_{proc} + t_{prop})_{PI|FM} \quad (3.14)$$

The notations, D_W and D_{WL} , (Fig. 3.6) are the two interpretations used to simplify the delay analysis for the wired and wireless transmission, respectively, as shown in below equations.

$$D_{WL} = (D_{PI} + D_{FM})_{Wireless} \quad (3.15)$$

$$D_W = (D_{PI} + D_{FM})_{Wired} \quad (3.16)$$

3.4.1.4 Controller Span

The span of a controller, in terms of number of consecutive neighboring RSUs outwards from the respective controller including the controlling RSU itself is

represented by N_s . This value is associated with the values of D_W and D_{WL} (3.21, 3.22), and also depends on the considered delay constraint in the optimization problem.

If $N_s = 1$, the local controller only controls vehicle OBUs inside its own RSU coverage. So there will be an SDN controller at each RSU. If $N_s = 2$, the controller controls the OBUs up to a span of one adjacent RSU as shown in Fig. 3.6. The maximum allowed controller span is indicated by S_{max} .

Figure 3.6: Positioning of normal RSUs with respect to the controlling RSU.

3.4.1.5 Controller Workload

The term, y_j is used to indicate the workload of controller j and it is measured in terms of number of RSUs that the controller j governs.

3.4.1.6 Controller Selection

Let, x_{ij} represents the association between a normal RSU and a controlling RSU as follows.

$$x_{ij} = \begin{cases} 1, & \text{if the RSU } i \text{ is attached to the controller } j \\ 0, & \text{otherwise} \end{cases} \quad (3.17)$$

The term x_{jj} is a special case of x_{ij} , and it indicates whether the j^{th} RSU is a controller or not.

$$x_{jj} = \begin{cases} 1, & \text{if the RSU } j \text{ is selected as a controller} \\ 0, & \text{otherwise} \end{cases} \quad (3.18)$$

We choose k, N_s, y_j and x_{ij} as our decision variables in the optimization problem.

Further, we list all notations used in the controller placement model as a summary in Table 3.3.

Table 3.3: Notations used in the ILP model.

Parameter	Description
V	Set of RSUs
E	Set of Edges
N_s	The span of the controller, in terms of number of consecutive neighboring RSUs outwards from the respective controller including the controlling RSU
C_c	Weight coefficient for number of controllers
C_i	Weight coefficient of RSU, where $C_i < 0$
C_w	Weight coefficient for workload distribution
d_{ij}	Distance between i^{th} and j^{th} RSUs, in terms of minimum number of hops
D_{PI}	Time to transmit a PACKET IN message
D_{FM}	Time to transmit a FLOW MOD message
D_{AODV}	Minimum route discovery delay in AODV (for 2 hops)
x_{ij}	Boolean flag to represent RSU i 's connection with controller j
y_j	Workload of controller j , in terms of number of RSUs
k	Number of controllers
r	Number of RSUs

3.4.2 Objective Function

Our objective of the controller placement is to minimize the number of controllers required to cover a given network map while placing them at the significantly located RSUs. Further, we encourage a fair distribution of workload among local controllers as well. Based on the aforementioned requirements, the objective function can be formulated as in (3.19).

$$\text{minimize} \quad C_c k + \sum_{i=1}^r \sum_{j=1}^r (N_s - d_{ij}) C_i x_{ij} + C_w \sum_{j=1}^r \sum_{j'=1}^r (y_j - y_{j'})^2 \quad (3.19)$$

Out of the multiple objectives in (3.19), the major goal is to minimize the number of controllers (k) required to cover the given network and thus, the weight coef-

ficient related to the number of controllers, C_c , is given as a very high positive value. This helps the term, number of controllers, to make a significant impact in the objective function.

As the objective function is a minimization, the weight coefficients of RSUs (C_i) are selected as negative values and thus, the term $(N_s - d_{ij}).C_i$ is used to give higher priorities to the controlling RSUs while considering the neighboring RSUs' status connected to that particular controlling RSU with decreasing priorities. i.e. a higher priority is given to the one-hop neighbors over two-hop neighbors.

We can consider another possible controller placement method by considering all RSUs as equally significant. This can be used as a comparative measurement to evaluate the importance of prioritization of RSUs. If we place the controllers without considering RSU weight coefficients, the objective function is changed from (3.19) to (3.20) and all other constraints remain the same.

$$\text{minimize } C_c k + \sum_{i=1}^r \sum_{j=1}^r d_{ij} x_{ij} + C_w \sum_{j=1}^r \sum_{j'=1}^r (y_j - y_{j'})^2 \quad (3.20)$$

As stated in section 3.3.1.5, we intent have a fair distribution of workload among the local controllers. For that, we minimize the difference in the number of RSUs that each controller controls, with $\sum_{j=1}^r \sum_{j'=1}^r (y_j - y_{j'})^2$ term in the objective function. Yet, the workload distribution coefficient, C_w is assigned a lower positive value than C_c to reflect it's comparatively lower priority.

3.4.3 Constraints

As mentioned previously, there is a trade-off between the number of controllers and the latency. When the number of controllers in the system is minimized, the latency experienced by OBUs start to increase, as graphically represented by Fig. 3.6. In the proposed controller selection model, we chose to constraint this latency from increasing higher than a threshold value.

The time taken to transmit a data packet between a source and a destination in VANET undergoes two phases. The first one is the route setup phase where the source node employs either a broadcast or any other similar mechanisms to find the route to the destination. The second one is the actual data transmission phase where the discovered route is used to transmit the data. Given that two different routing protocols transmit a data between the same source and destination pair

along the same path, the difference in delay occurs in the route setup.

In the optimization problem, this route setup time of the proposed SDVN system is constrained by the minimum route setup time of conventional VANETs in the multi-hop scenario. The Ad hoc On-Demand Distance Vector (AODV) [27] is selected as the reference routing protocol for the VANETs because it is a well-studied routing protocol with a similar route discovery phase to the SDVN. The minimum route setup delay in AODV, D_{AODV} occurs when the source and the destination are apart from each other by 2 hops in the multi-hop data transmission. In the proposed SDVN operation, the route setup delay is equal to the time to send a PACKET IN message to the control plane and receive the respective FLOW MOD message. When the source and destination are 2 hops away from each other, it is high likely that the local RSU controllers possess the required topology information to route packets between these 2 hops. Hence, it is assumed that the local controller-regional controller interface is not used in this case.

The minimum route setup time of the proposed system further depends on the position of the vehicle with respect to the position of the controller. Following up on this, we consider two delay aspects in the constraints.

Worst Case Delay Constraint,

$$D_{WL} + (N_s - 1)D_W \leq D_{AODV} \quad (3.21)$$

Average Case Delay Constraint,

$$D_{WL} + \frac{2}{2N_s - 1} \sum_{i=1}^{N_s} [(i - 1)D_W] \leq D_{AODV} \quad (3.22)$$

In (3.21), the worst case route setup delay which occurs when the vehicle is under the coverage of the furthest away RSU, is bounded by the minimum route setup delay in the AODV (Fig. 3.6). In the average case (3.22), the vehicle is considered to be with all the RSUs which are under the coverage of the same controller with equal probabilities. So the delay constraint is written by considering all the possible delay values that can occur when the vehicle is with each RSU.

By substituting the parameter values of Table 3.2, in theoretical delay expressions (3.4, 3.5, 3.9, 3.11, and 3.12), we can get a numerical evaluation for D_W (3.15) and D_{WL} (3.16). With that, we can further simplify the worst case (3.21) and

average case (3.22) delay constraints in the format of (3.23), where S_{max} is equal to **2.55** in the worst case and **3.68** in the average case.

$$N_s \leq S_{max} \quad (3.23)$$

We include the latency factor in the constraints rather than embedding in the objective function for the minimization and thus, the delay constraints are never to be violated in placing the controllers. Apart from that, few others constraints are considered as follows.

$$(\forall i, j) \quad d_{ij}x_{ij} \leq N_s - 1 \quad (3.24)$$

Equation (3.24) defines the controller span in terms of number of hops. In general sense, it restricts the selected controller RSUs governing conventional RSUs which are more than $N_s - 1$ hops away.

$$(\forall i, j) \quad x_{ij} \leq x_{jj} \quad (3.25)$$

Equation (3.25) indicates that a non-median (RSU), i can only be associated with a median (controller), j .

$$\sum_{j=1}^r x_{jj} = k \quad (3.26)$$

$$(\forall i) \quad \sum_{j=1}^r x_{ij} = 1 \quad (3.27)$$

Equation (3.26) accumulates the controlling RSUs and equates it to the number of controllers term, k . By the proposed model, a single non-controlling RSU and its coverage area is controlled by only one controlling RSU and it is modeled by (3.27).

$$(\forall j) \quad y_j = \sum_{i=1}^r x_{ij} \quad (3.28)$$

The controller workload, y_j is modeled as in (3.28). The equation essentially counts all the RSUs which are assigned to the controller RSU, j as its workload.

To summarize the controller placement problem; our optimization function consists of three sub-objectives (3.19). The main objective is to minimize the number of controllers, which is required to serve the network. The second term allocates controlling regions for candidate controllers. Here, we also prioritize several candidates in the controller selection based on their tactical location. From the third term, we minimize the discrepancy in workload distribution among candidate controllers. In addition, we have our main constraint (3.24), limiting the controller span or in other terms, the extent of the region that a controller can control. The controller span is influenced by the delay constraints in (3.21 and 3.22). To simply put, we constraint the controlling regions of controllers to avoid large delays, as this is one of our design goals in HD-SDVN system model.

3.4.4 Solution Scheme

As mentioned above, we have not used the number of controllers (k) as a hard constraint as in the conventional p-median facility location problem [103]. In fact, we have included it in the objective function as a soft constraint. Therefore, the model will always converge to a feasible solution for any given network map. The worst possible solution we can get is that all the RSUs are selected as controllers.

In terms of optimality, the objective function of the controller placement optimization model with prioritized RSUs is not convex, which can be verified by using the second derivative test with Hessian matrix. One of the Eigenvalues of the Hessian matrix is negative and thus, the function is concave. The results acquired by solving the optimization problem (3.19) directly via any Integer Quadratic Programming (IQP) or Mixed Integer Quadratic Programming (MIQP) solver may not be optimal. In order to address this issue, we employ a heuristic approach as follows.

The concavity comes from the non-linear term, $N_s * x_{ij}$. Therefore, in order to convert the function into a convex format, we intend to fix the controller span, N_s . As our main objective in placing the controllers is to minimize the number of controllers required to cover the given network map, C_c is very high. It is clear that, to minimize the number of controllers, the controller span (N_s) should be maximized. Further, the maximum controller span is bounded by the delay constraints (3.23). So we assign the highest possible controller span to N_s without violating the delay constraints as in (3.29).

$$N_s = \lfloor S_{max} \rfloor \quad (3.29)$$

Now, the controller span, N_s , is no longer a decision variable. Hence, the overall optimization problem is a convex IQP problem, and can be solved optimally via a proper MIQP solver such as CPLEX [104]. The overall controller placement selection problem for prioritized RSUs can be summarized as below.

$$\underset{k, x_{ij}, y_j}{\text{minimize}} \quad C_c k + \sum_{i=1}^r \sum_{j=1}^r (N_s - d_{ij}) C_i x_{ij} + C_w \sum_{j=1}^r \sum_{j'=1}^r (y_j - y_{j'})^2 \quad (3.30a)$$

$$\text{subject to} \quad (\forall i, j) \quad d_{ij} \cdot x_{ij} \leq N_s - 1 \quad (3.30b)$$

$$(\forall i, j) \quad x_{ij} \leq x_{jj} \quad (3.30c)$$

$$\sum_{j=1}^r x_{jj} = k \quad (3.30d)$$

$$(\forall i) \quad \sum_{j=1}^r x_{ij} = 1 \quad (3.30e)$$

$$(\forall j) \quad y_j = \sum_{i=1}^r x_{ij} \quad (3.30f)$$

$$(\forall i, j) \quad x_{ij} = 0, 1 \quad (3.30g)$$

$$k, y_j \in \mathbb{Z}^+ \quad (3.30h)$$

3.5 Results and Discussion

In this section, we apply the above controller placement model to a real world problem and compare the performance of the proposed approach with several reference methods. The simulation study consists of two parts.

In the first part, we apply the proposed controller placement model with and without proper RSU coefficients as in (3.19) and (3.20), respectively, in a real world map for both average case delay constraint and worst case delay constraint. In addition, we generate another set of controller placements randomly as a comparative measurement.

In the second part, we simulate several custom traffic data sets on the network map against the above mentioned controller placements and compare the performance

of different systems. The custom traffic data sets have been composed based on the local knowledge that we have on the road network and also based on random number generators. The incentive behind using the custom traffic data sets is to test the significance of proposed candidate RSU prioritization method based on their strategic location in the local controller selection. We also make a comparative study on the latency, PDR, and packet overhead generated by the proposed HD-SDVN system against conventional vehicular networks and existing Internet based SDVN systems.

3.5.1 Simulation Model

We applied the above controller placement model for a simplified version of the campus map of Nanyang Technological University (NTU) in Singapore. The map is covered with a total of 19 RSUs as shown in Fig. 3.7.

Figure 3.7: Simulation environment – NTU campus map.

Figure 3.8: Controller placement results for the worst delay constraint.

Figure 3.9: Controller placement results for the average delay constraint.

The controller placement model for the above scenario consists of 381 decision variables and 722 inequality constraints and 39 equality constraints. We use the CPLEX Optimizer software to solve the two controller placement problems under different delay constraints. The obtained results **with** differential coefficients are shown in Fig. 3.8a and 3.9a for the worst case delay constraint and average case delay constraint, respectively. The Fig. 3.8b and 3.9b show the results for the same controller placement models **without** proper coefficients. The Fig. 3.8c and 3.9c show the **random** controller placement results using 6 and 3 controllers, respectively.

As the next step, we measure the performance of each controller placement method in terms of latency using the NS-3 simulator [105]. In the simulation, the road network is a compressed version of the NTU campus road network, which span

over nearly a $1500 * 1500 \text{ m}^2$ area as shown in Fig. 3.7. As mentioned previously, we generate a set of custom mobility scenarios which resemble the actual traffic patterns on the NTU campus road network. The vehicle speed profiles are generated based on the Weibull random number generator with different shape and scale values depending on the road environment. Intersections deploy a lower speed profile and unrestricted straight roads entertain a higher speed profile with a maximum speed of 90 kmh^{-1} . Nodes which include both OBUs and RSUs broadcast Hello messages and update the controllers periodically at each 3 seconds. We selected these higher interval values to test the proposed system at extreme ends.

For the controller placement comparison, we measure the average route setup delay, which is the time taken to discover or/and setup the route from a source to a destination, and the average end to end delay, which is the time taken to find the route and transmit a packet from a source to a destination, for each type of controller placement against varying number of vehicles starting from 10 to 80 in an increment of 10. At each run, half of the OBUs acts as the sources and sends packets to the other half of the OBUs. The packet arrival is implemented using a Poisson distribution with a mean of 3 seconds. Each point in the graphs is an average over 50 simulation runs with different random numbers without outliers. The results for the worst case delay constraint and average case delay constraint are shown in Fig. 3.10 and 3.11, respectively.

Finally, we compare the performance of the proposed system under the Worst Delay Constraint (WDC) against conventional VANETs, existing Internet based SDVN systems and also against reactive operation mode of SDVNs. For the VANET, we employ conventional routing protocols of AODV and OLSR [65]. For the existing SDVN, we recreate a simulation model for the SDVN architecture from [7] in NS-3 and connect a centrally placed SDN controller with OBUs via LTE interfaces. Then, we test the performance of these systems following the same mobility patterns. The performance has been measured in terms of average end to end delay and PDR by varying the number of vehicles, and the results are shown in Fig. 3.12a and 3.12b, respectively. We further test the total packet overhead of the proposed proactive based operation mode against existing reactive operation mode by varying the average vehicle speed at unrestricted roads with number of vehicles fixed to 40. The results are shown in Fig. 3.12c. The parameters and values used in the simulation are tabulated in Table 3.4.

Table 3.4: Simulation parameters of HD-SDVN.

Parameter	Value
Simulators	NS-3.25, NS-3-dev
Simulation time	600 seconds (at each run)
Simulation area	Simplified model of NTU-Campus Map ($1500 * 1500 m^2$)
Transmission type	UDP
Data size	1000 bytes/packet
Wireless network	V2V and V2I = DSRC, Vehicle to Base station = LTE
Transmission speed	Wireless = 6 Mbps, Wired = 100 Mbps, Internet = 100 Gbps
Packet Intervals	Hello message = 3 seconds, Status update message = 3 seconds
Transmission range	DSRC ~ 200 m
Propagation model	Friis Propagation Model
Propagation delay	RSU backbone = $100 \mu s$, Internet backbone = 20 ms

3.5.2 Result Analysis

According to Fig. 3.10 and 3.11, it is clear that our proposed controller placement models with and without coefficients show considerably lower delays than those by random controller placement in each run of the simulation. In random controller placement, as we can see from Fig. 3.8c and 3.9c, several non-controlling RSUs are assigned to the controlling RSUs which are far away in terms of hops in comparison to other controller placement schemes with the same number of controllers. Therefore, OBUs under the coverage of these non-controlling RSUs experience slightly higher delays compared to other OBUs. It highlights the importance of placing controllers with proper control on the delay induced in the systems.

Again, slightly lower delays in both route setup (Fig. 3.10a and 3.11a) and end to end delay (Fig. 3.10b and 3.11b) are achieved by the proposed system with coefficients than that by the controller placement method without proper coefficients. It has proved our hypothesis that prioritizing the RSUs based on their location and statistics is important to the performance of the system in terms of latency. It is because the controllers are placed at the locations where most of vehicles show lower vehicular speeds and long vehicular traffic queues.

Previous research has shown that OLSR can give a better performance in terms

Figure 3.10: Latency results under the worst case delay constraint.

Figure 3.11: Latency results under the average case delay constraint.

of delay in VANETs because it does not require any route setup time [106]. Yet, it struggles when it is applied to long source-destination pairs as the protocol needs a considerable time to converge. It is clearly shown in the results as a very low latency (Fig. 3.12a) and the lowest PDR (Fig. 3.12b) have been displayed by the OLSR scheme. On the other hand, the AODV exhibits a high successful packet delivery. However, it struggles to maintain that PDR as the number of vehicles increases (Fig. 3.12b). This is due to the fact that the AODV's broadcast approach in finding routes makes the network more congested when the number of vehicles increases. This can be stated as the root cause for the high and increasing latency of the AODV scheme (Fig. 3.12a).

Unlike the conventional VANETs, both SDVN systems have been able to maintain a promising PDR (Fig. 3.12b) against increasing number of vehicles, due to the global view they possess over the network. However, the existing SDVN systems with the control plane on the Internet (SDVN - Internet) undergo a considerable

Figure 3.12: Performance comparison.

amount of delay in the operation as shown in Fig. 3.12a due to its way of placing controllers and reactive based FI operation. In fact, these latency results for the Internet based SDVN systems can go up even higher depending on the placement of the control server in the Internet Service Provider (ISP) network. The latest statistics show that the round trip delay of the communication over the Internet between a single pair of a client and a server is around 100 ms on average [107,108]. Thus, it is possible that a much higher actual latency could reach for the existing Internet based SDVN systems.

However, the upper bound delay requirements of active road safety applications and cooperative traffic efficiency applications in VANET are around 100 ms and 200 ms, respectively [12]. Therefore, this type of SDVN architectures prompts serious doubts on their feasibility in above areas.

In contrast, the proposed HD-SDVN system with coefficients (WDC) has managed to excel in both latency and PDR at all vehicle density levels. In terms of latency, our system which displays a delay range of 5 - 20 ms, can be applied in all the VANET applications with the excellent QoS. This is mainly due to the controlled way of placing controllers closer to vehicles. The latency is even lower than that achieved by the OLSR scheme in most cases (Fig. 3.12a). Further, the collective generation of global network view of the proposed HD-SDVN has managed to achieve a very high successful PDR, which is even slightly higher than the PDR by the Internet based SDVN (Fig. 3.12b). We believe that the low route setup time of the HD-SDVN scheme is the root cause for the PDR improvement. Since the communication links in VANETs are highly vulnerable in their existence, faster the accurate route information is supplied to the OBUs, higher the successful packet delivery rate.

Further, both Fig. 3.12a and 3.12b can be viewed as a comparison of performances between different communication modes of the proposed system. The performance of HD-SDVN is dominated by the DSRC mode as the network was almost covered with RSUs. Thus, the HD-SDVN profile can be considered as the performance of DSRC mode. SDVN-Internet lines indicate the performance with a sole LTE interface, and thus, the LTE mode. Both VANET-AODV and VANET-OLSR depict the performances with distributed routing protocols. Note that, the proposed system also falls-back to a similar distributed protocol (GPSR), when both DSRC and LTE coverages are not available. Thus, even though not a direct representation, VANET-AODV and VANET-OLSR performances can be approximated to the backup mode performance. To this end, both Fig. 3.12a and 3.12b exhibit a performance comparison between different communication modes.

In terms of communication overhead, the existing reactive based operation induces a very high overhead with a diminishing increment rate (Fig. 3.12c) when average vehicle speed increases, due to multiple consultation of the control plane in the multi-hop routing. If the control-data plane interface is composed of expensive LTE links as that in the Internet based SDVN, it further accounts for a huge communication cost as well. However, the proposed proactive based operation incurs a very low communication overhead (Fig. 3.12c) due to its limited consultation of the control plane, and further does not generate a significant communication cost as a result of the local placement of controllers and use of inexpensive DSRC links in the southbound interface.

Our main objective in introducing the hierarchical distributed SDVN architecture was to exploit the advantages of SDN in VANET without violating the stringent latency requirement of vehicular networks. So as it can be seen from the above results, the proposed architecture, the controller placement model, and the operation help to achieve a competitive and in most cases slightly lower latency level than conventional VANET and show a remarkable latency reduction over existing Internet based SDVN. However, these improvements can be fully achieved only by the vehicles who are under RSU coverage. Further research is required on the issue of coverage and disconnected vehicles.

As a closing note, we have summarized how the key features of SDVN have been achieved in our proposed HD-SDVN and in existing SDVN architectures comparatively in Table 3.5.

⁴Flexibility & Programmability

⁵Communication Cost

Table 3.5: A summary of comparison of SDVN systems with respect to key features.

Model \ Feature	Flexib. & Program. ⁴	Comm. Cost ⁵	Scalability	Global Network View	Latency
I. Ku et al. [7] SD-VANET	Yes (Full)	Very High	Low	Available	High
H. Zongjian et al. [14] SDVN	Yes (Full)	High - Medium	Low	Available	Very High
P. Duan et al. [41] Cyber Physical System	Yes (Partial)	Low	Medium	Partially Available	Low
L. Yu-Chun et al. [42] GeoBroadcast	Yes (Partial)	Low	Medium	Partially Available	Low
A. Kazmi et al. [46] DeVANET	Yes (Full)	High	High	Collectively Available	High
H. Li et al. [29] Control Plane Optimization	Yes (Full)	Medium	Low	Available	Medium
Q. Zheng et al. [52] Radio Resource Scheduling	Yes (Partial)	High	High	Collectively Available	Medium
N.B. Truong et al. [24] Fog Comput.	Yes (Partial)	Low	High	Partially Available	Low
<i>HD-SDVN</i> (Proposed System)	Yes (Full)	Low	High	Collectively Available	Low

3.6 Summary

In this chapter, we have proposed a novel hierarchical distributed SDVN architecture with a detailed controller placement model to optimize the placement of the controllers while preserving the stringent delay constraints in vehicular networks. The architecture splits the control plane into two levels and the top tier is placed on the Internet while the bottom tier is distributedly placed at the RSU level. The proposed controller placement model for the RSU level places local SDN controllers scrutinizing several factors such as number of controllers, latency, significance of the RSU's geographical location, workload distribution and the vehicular statistics around the RSU location. Via extensive simulation experiments, we have shown that our proposed controller placement method generates a lower

latency compared to other existing controller placement methods. It is also shown that our HD-SDVN architecture with the proposed controller placement method outperforms the existing Internet based SDVN and conventional VANET with the lowest latency while maintaining a high PDR and a lower communication overhead.

Chapter 4

Routing in Software Defined Vehicular Network under Ideal Conditions

4.1 Introduction

In this chapter, we aim to explore the enhanced effect of SDN on information dissemination in vehicular network through an intelligent routing framework [109–111]. The routing framework consists of a centralized routing algorithm and an FI scheme.

The HD-SDVN architecture and operation introduced in chapter 3 is taken as the system model for this work [75, 90]. One important assumption made in this chapter is that the complete global network view over the road network is available at the control plane.

The routing protocol considers multiple vehicular network attributes, link stability and travel time, in its algorithm. The primary focus is to find multiple paths with lower travel times and with enough stability to deliver a given number of packets. To the best of our knowledge, this is the first piece of research work to address the link stability through an SDVN approach. The motivation for the problem formulation is taken from Air Traffic Flow Management (ATFM) where flights are routed and scheduled in the airspace under link and sector capacity constraints [112, 113]. Identically, here we route individual packets between source and destination pairs under link capacity constraints.

However, compared to ATFM, it is more challenging in vehicular networks due to the following reasons.

1. Links are short-lived and thus, the capacity of a link is a function of time¹.
2. Wireless channel access imposes limitations on continuous packet transmission.
3. Combinational patterns of wired and wireless links make it hard to accurately estimate the packet arrival times and link capacities.

The routing rules also called as flow rules are delivered to the respective data plane nodes via an FI scheme. For an FI operation to be effective, the extent of communication between the control plane and data plane, and the time spent on delivering the flow rules should be minimized while maximizing the number of flow rules successfully enforced in the respective nodes. To achieve these goals, we introduce an effective FI scheme based on source routing and selective route caching.

The rest of the chapter is organized as follows. In section 4.2 and 4.3, we present the link stability based routing protocol and source routing based FI scheme, respectively. Section 4.4 discusses the results and finally, Section 4.5 concludes the chapter.

4.2 Link Stability based Packet Routing Algorithm

4.2.1 Problem Statement

4.2.1.1 Background of the Problem

The general trend in vehicular networks has been to route the packets in the shortest possible paths. However, this approach might not be ideal as in some situations the lifetime² of the path may not be enough to deliver the packets. By the time packets reach an intermediate node, the opening window of the link can

¹It should be noted that, here we refer the capacity of a link as its capability to deliver packets through its path satisfying a given minimum SNR level and thus, different from the channel capacity. Therefore, the link capacity can be defined as the number of packets that can be delivered through the link before it terminates. A link is considered as terminated at the point required receiver sensitivity level (-82 dBm, given that the data rate is 6 Mbps, and the bandwidth is 10 MHz) is failed to achieve.

²Please note that by means of lifetime, we refer to the residual lifetime of the considered link.

disappear leading to unsuccessful transmissions [26]. In fact, studies have found that, in VANET, the route lifetime sharply drops when a path consists of three to four hops and has a high probability of causing a route error [25].

This becomes severe, when a considerable time is taken to transmit the packets which could be a consequence of a set of packets to be transmitted and/or selection of a long route, as vulnerability increases in both cases. It is highly probable that the packets at the end of the sequence may not be able to reach the destination as a result of a route breach due to continuous vehicle movements, leading to a lower PDR [26]. Moreover, the system is compelled to find new routes frequently, which could result in higher latency as well. Therefore, it is imperatively important to analyze whether a selected path can deliver a given number of packets over the multiple hops before the route is broken.

In the network map instance shown in Fig. 4.1, at a macroscopic level, if the lifetime of the link 6-7 in the shortest path is 7 ms, and the total time it takes for a packet to travel a single hop is 5 ms, it is highly likely that the packet will not be able to reach node 7, as at the time packet reaches node 6 only a 2 ms of opening remaining in the link 6-7, which is not sufficient compared to a travel time of 5 ms.

Figure 4.1: An instance of a vehicular network.

Moreover, if there are multiple packets to be transmitted, a single path may not be adequate. As per the current example, if the lifetime of the same link, 6-7 is to be 12 ms, only a single packet can be delivered via the path, S-6-7-D and hence, not ideal. A better option would be to transmit the 1st packet via S-6-7-D and the rest of the packets through S-2-3-4-5-D.

Furthermore, a bulk of packets cannot be transmitted consecutively right after the other in a wireless environment, because of the employed channel access mechanisms [6]. VANET uses EDCA as the MAC protocol [1, 6]. EDCA relies on CSMA/CA for providing channel access. In CSMA/CA, nodes sense the channel first for ongoing transmissions before starting theirs. After initiating a transmission, the consecutive packets even from the same node have to wait, at least until the ongoing packet completely reaches the next hop. Even after that if the next consecutive link is wireless, we cannot accurately predict the transmission order as it depends on the random backoff procedure. For an instance, in the network shown in Fig 4.1, only a single packet transmission can take place at a time in the links of 3-4, 4-5, 5-8 without causing packet collisions. This is the same for the 1-6, 6-7, 7-8 links as well³.

The channel access mechanism will not be a concerning factor, if the inter-arrival time of packets (gap between consecutive packets) is reasonably high. Further, if the next link is wired, then again, the channel access mechanism will not be an issue due to the different physical characteristics of two channels. The consecutive links 1-2 and 2-3 shows such a situation. Here, a packet can be transmitted in both links at the same time, as the channels of the two links are different from each other (1-2 : wireless, 2-3 : wired).

Overall, the consecutive packet transmission will depend upon the interval between consecutive packets and the type of channel and their setup.

4.2.1.2 Possible Solutions

If the controller is given information about the number of packets needed to deliver and the gap between the consecutive packets, one option is to select the most stable path as generally followed by research that include stability in their routing protocols [64, 79]. Even though this approach may successfully deliver more packets, it might not be effective in terms of latency, as it can lead to longer routes. Given that the vehicular network topology is rapidly changing, there is also a possibility for this approach to not to find a feasible route.

Alternatively, we can always find the shortest path without taking the route stability into consideration, as followed by many studies in this area. However, as depicted earlier, packets can struggle to reach the destinations with this approach

³Please note that here, we consider the nodes shown in dotted lines are the ones who are in the transmission range of each other.

displaying a relatively low PDR. Though, the shortest path generally implies a lower latency, the fact that the nodes will be compelled to repeatedly consult the controller as the supplied flow rules expire, can limit the latency reduction. Moreover, frequent consultation of the controller generates a higher routing overhead, and a higher communication cost.

In our routing scheme, we balance this trade-off between latency and PDR by routing individual packets in the shortest possible paths (in terms of travel time), which are sufficiently stable in their existence. Different paths will transmit a different number of packets depending on their route valid times.

In order to realize a low complexity system, we make several assumptions and base rules as listed below.

- Hello and status messages use different frequency bands, and thus, the effect on data transmission is minimal (An uncertainty component pertaining to these events is added in the packet transmission time).
- In order to account for the uncertainty affiliated with wireless channel access, we consider the worst case transmission time in links following robust optimization.
- Only existing links in the system are considered in the formulations, as it is not reliable to explore future connections that could establish during the data transmission.

4.2.2 Problem Formulation

4.2.2.1 Notations

Given a network graph, $G = (V, E)$ where V and E refer to the set of nodes, and edges connecting the nodes, respectively, our aim is to successfully deliver a given number of packets (P) between a source node, $s \in V$ and a destination node, $d \in V$, with minimum travel times and without violating the dynamic link constraints. Each link from node i to j (refer to as ij) is associated with two metrics. First is the cost metric, C_{ij} , and it indicates the total transmission time of the data packet in link ij . The second metric T_{ij} , refers to the link lifetime at the moment flow rules reach the source node.

We use t_{hop}^w and t_{hop}^{wl} to represent the total time to send a packet within a single wired and wireless hop, respectively (hop time). We mainly include the transmis-

sion ($t_{trans.}^x$), propagation ($t_{prop.}^x$), queuing (t_q^x) and contention ($t_{cont.}^x$) times in hop time (x represents the link type as in wired or wireless), and neglect the processing time. Detailed expressions of the above terms are included in the previous chapter.

It is important to note that in reality, both t_{hop}^w and t_{hop}^{wl} are affected by other ongoing transmissions. However, as some transmissions do not require the controller's intervention, it is difficult to accurately analyze the effect from ongoing transmissions on the current transmission. Thus, we model t_{hop}^w and t_{hop}^{wl} with added uncertainties as follows.

$$t_{hop}^w = t_{trans.}^w + t_{prop.}^w + t_q^w + t_{cont.}^w + \Delta t^w \quad (4.1a)$$

$$t_{hop}^{wl} = t_{trans.}^{wl} + t_{prop.}^{wl} + t_q^{wl} + t_{cont.}^{wl} + \Delta t^{wl} \quad (4.1b)$$

where Δt^w and Δt^{wl} refer to the uncertainty associated with wired and wireless packet transmission, respectively. The upper bounds of (4.1) are estimated by the controller at each run depending on the parameter values such as packet size, and the detailed expressions are depicted in the previous chapter (section 3.3.2).

We use X_{ij} to indicate the packet flow (number of packets) on link ij , and Y_{ij}^p to indicate the p^{th} packet's path through link ij . $H_{s,r}^{p,w}$ and $H_{s,r}^{p,wl}$ represent the number of wired and wireless hops, respectively, from the source node, s to an intermediate node, r via the estimated route for the packet p . $T_{s,r}^p$ refers to the time p^{th} packet takes to reach an intermediate node, r from the source node, s . $N(=|V|)$ and $L(=|E|)$ refer to the number of nodes and links in the network, respectively. A summary of notations is available in Table 4.1.

4.2.2.2 Objective function

Our objective is to route the individual packets in the vehicular wireless network with the least amount of delay and error rate, thus, the the objective function can be formulated as in (4.2).

$$\text{minimize} \quad \sum_{i=1}^N \sum_{j=1, j \neq i}^N \left\{ C_t t_{hop}^{ij} X_{ij} + C_v V_{eff}^{ij} X_{ij} \right\} \quad (4.2)$$

Here, t_{hop}^{ij} will be equivalent to t_{hop}^w or t_{hop}^{wl} depending on the link type. Further, V_{eff}^{ij} refers to the effective velocity associated with the link ij and can be estimated

Table 4.1: A summary of notations

Parameter	Description
V	Set of nodes in the network ($N(= V)$)
E	Set of links (edges between nodes) in the network ($L(= E)$)
P	Set of packets to be delivered
C_{ij}	Total transmission time of the data packet in link ij (Cost of link ij)
T_{ij}	Link lifetime at the moment flow rules reach the source node
t_{hop}^w, t_{hop}^{wl}	Total time to send a packet within a single wired (w) and wireless (wl) hop
$t_{trans.}^x$	Transmission time component in hop time (x represents the link type as in wired or wireless)
$t_{prop.}^x$	Propagation time component in hop time (x represents the link type as in wired or wireless)
$t_{cont.}^x$	Contention time component in hop time (x represents the link type as in wired or wireless)
t_q^x	Queuing time component in hop time (x represents the link type as in wired or wireless)
$\Delta t^w, \Delta t^{wl}$	Uncertainty associated with wired (w) and wireless (wl) packet transmission
X_{ij}	Packet flow (number of packets) on link ij
Y_{ij}^p	A boolean to represent p^{th} packet's path through link ij
$H_{s,r}^{p,w}, H_{s,r}^{p,wl}$	Number of wired ($x = w$) or wireless ($x = wl$) hops, from the source node, s to an intermediate node, r via the estimated route for the packet p
$T_{s,r}^p$	Time p^{th} packet takes to reach an intermediate node, r from the source node, s
N_i^{in}, N_i^{out}	Set of neighbors with incoming links to and out going links from node i , respectively
$L_{s,r}^{u,w}, L_{s,r}^{u,wl}$	Set of all wired (w) and wireless (wl) upstream links of node r from node s , respectively
t_{hop}^{ij}	Packet transmission time in the considered link, ij and will be equivalent to t_{hop}^w or t_{hop}^{wl} , depending on the link type
V_{eff}^{ij}	Effective velocity associated with link ij
t_{gap}	Inter-arrival time between consecutive packets
U_{ij}^p	Capacity of link ij at the time packet p reaches node i

as in (4.3).

$$V_{eff}^{ij} = \sqrt{\left[v_i^{\overrightarrow{ij}}\right]^2 + \left[v_j^{\overrightarrow{ji}}\right]^2} \quad (4.3)$$

where $v_i^{\overrightarrow{ij}}$ indicates the velocity component of node i in the direction of node j .

The terms, C_t and C_v refer to the coefficients of total transmission time and effective velocity, respectively, and $C_t \gg C_v$. Hence, our primary focus is on delivering packets with the least travel time which we define as the shortest path in the context of this chapter. If multiple paths exist with same travel time, the scheme goes for the path with less mobility. As empirically evaluated in [6, 114], at higher effective velocities, the Doppler spread can have a considerable negative impact on error rate, especially, when the packet size is high. Thus, we discourage the selection of such highly dynamic links in the route selection.

4.2.2.3 Constraints

- **Routing**

The following equations (4.4, 4.5, 4.6, and 4.7) are based on the flow conservation concept, and responsible for routing the n packets in the network. Equation (4.7) helps to avoid loops in the routing path, pertaining to the bi-directionality of links.

$$(\forall p \in P) \quad \sum_{\forall j \in N_s^{out}} Y_{sj}^p = 1 \quad (4.4)$$

$$(\forall k \in V)(\forall p \in P) \quad - \sum_{\forall i \in N_k^{in}} Y_{ik}^p + \sum_{\forall j \in N_k^{out}} Y_{kj}^p = 0 \quad (4.5)$$

$$(\forall p \in P) \quad - \sum_{\forall i \in N_d^{in}} Y_{id}^p = -1 \quad (4.6)$$

$$(\forall p \in P)(\forall i, j \in V) \quad Y_{ij}^p + Y_{ji}^p \leq 1 \quad (4.7)$$

where N_i^{in} and N_i^{out} refer to the set of neighbors with incoming links to and out

going links from node i , respectively.

$$(\forall i, j \in V) \quad X_{ij} = \sum_{\forall p \in P} Y_{ij}^p \quad (4.8)$$

Equation (4.8) accumulates the number of individual packets in the link ij and equates it to the packet flow of that link.

• Timing

Equation (4.9a) and (4.9b) calculate the number of wired and wireless hops, respectively, from the source, s to an intermediate node, r for the packet p . The binary variable Y_{ij}^p will be 1 in the links where packet p travels leading up to the node r .

$$(\forall p \in P)(\forall r \in V) \quad H_{s,r}^{p,w} = \sum_{\forall ij \in L_{s,r}^{u,w}} Y_{ij}^p \quad (4.9a)$$

$$(\forall p \in P)(\forall r \in V) \quad H_{s,r}^{p,wl} = \sum_{\forall ij \in L_{s,r}^{u,wl}} Y_{ij}^p \quad (4.9b)$$

where $L_{s,r}^{u,w}$ and $L_{s,r}^{u,wl}$ indicate the set of all wired and wireless upstream links of node r from node s , respectively. To increase the number of potential routes, the links are selected from a larger circular region⁴ without considering the vector direction.

The packet p 's arrival time to node r , $(T_{s,r}^p)$ primarily depends on the inter-arrival time between consecutive packets, t_{gap} . If t_{gap} is considerably higher than t_{hop}^{wl} , then, the packet transmission will not be disturbed by preceding and succeeding transmissions as once the next packet is due, the previous packet will be far away from the current node's coverage. Further, it will not be affected by the hidden node problem as well. Thus, we formulate $T_{s,r}^p$ when $t_{gap} \gg t_{hop}^{wl}$ as in (4.10).

$$T_{s,r}^p = (p-1)t_{gap} + H_{s,r}^{p,w} * t_{hop}^w + H_{s,r}^{p,wl} * t_{hop}^{wl} \quad (4.10)$$

When t_{gap} is very small, it becomes very complex to accurately model the time a packet takes to reach a given node due to the involvement of randomness. How-

⁴A circular region of $2R$ (radius) with the center in the middle of source-destination. Here, R refers to the distance from the source to the middle point.

ever, adopting robustness, we make the following assumptions to narrow down the possible node patterns to few cases.

- First and last links of the path are wireless. This limits the application of the work to transmissions starting from vehicles and ending in vehicles.
- Two neighboring nodes in a selected path are on the edges of each other's transmission range as the model selects the shortest paths. Thus, two nodes in a computed path that both have a common neighbor will not be in each other's transmission range.

With the above assumptions, we narrow down all possible node patterns to three, which we discuss below case by case.

Case 1: $H_{s,d}^{p,w} = 0$, $H_{s,d}^{p,wl} = 2$

Figure 4.2: Graphical illustration of case 1 scenario.

This is the most basic case in multi-hop data transmission scenario where the source and destination nodes are within 2 hop distance as shown in Fig. 4.2. Here, both hops have to be wireless due to the above assumptions. As randomness is involved in packet transmission, we assume that the next packet will start to transmit only after the previous packet reaches the destination. Thus, $T_{s,r}^p$ can be expressed as in (4.11).

$$T_{s,r}^p = H_{s,r}^{p,wl} * t_{hop}^{wl} + 2(p-1)t_{hop}^{wl} \quad (4.11)$$

Case 2: $H_{s,d}^{p,w} \geq 1$, $H_{s,d}^{p,wl} = 2$

Figure 4.3: Graphical illustration of case 2 scenario.

In this case, routing path consists of multiple wired links in between the first and last wireless links as indicated in Fig. 4.3. Therefore, the packet transmission will

not be disturbed by the preceding and succeeding packets. Hence, the packets can be continuously transmitted with a gap of t_{hop}^{wl} between them.

$$T_{s,r}^p = H_{s,r}^{p,w} * t_{hop}^w + H_{s,r}^{p,wl} * t_{hop}^{wl} + (p-1)t_{hop}^{wl} \quad (4.12)$$

Case 3: $H_{s,d}^{p,wl} \geq 3$

Figure 4.4: Graphical illustration of case 3 scenario.

In cases where there are more than 3 wireless links as shown in Fig. 4.4, it is difficult to accurately predict the time a packet arrives at an intermediate node as it depends on the orientation of the network. However, in order to achieve robustness, we plan for the worst case which is a minimum of 3 wireless links sequentially located in the path. This is the bottleneck of the path in terms of continuous packet flow.

Due to the employed channel access mechanisms, only a single packet can be transmitted at a time in 3 consecutive wireless links without a possible collision. Yet, the way packets are transferred in these links is unpredictable. But it is safe to say that, there should be a gap of $3t_{hop}^{wl}$ in between consecutive packets in order to avoid collisions at all scenarios. A packet collision situation with a smaller packet gap is shown in Fig. 4.5.

Figure 4.5: Illustration of a collision situation under a packet gap of $2 * t_{hop}^{wl}$.

Therefore, in this case, $T_{s,r}^p$ can be expressed as,

$$T_{s,r}^p = H_{s,r}^{p,w} * t_{hop}^w + H_{s,r}^{p,wl} * t_{hop}^{wl} + 3(p-1)t_{hop}^{wl} \quad (4.13)$$

The case 2 is a deterministic expression of $T_{s,r}^p$ as it does not involve any uncertainty. However, in both case 1 and 3, we derived the expressions for the worst cases, thus robust.

• Capacity

The capacity of a link is a function of time, however, in this problem, it is only considered at several discrete points (at packet arrival times).

The capacity of link ij at the time packet p reaches node i , U_{ij}^p , can be expressed as in (4.14).

$$(\forall i, j \in V)(\forall p \in P) \quad U_{ij}^p = \max(0, \lfloor \frac{T_{ij} - T_{s,i}^p}{t_{hop}^{ij}} \rfloor) \quad (4.14)$$

where t_{hop}^{ij} refers to the packet transmission time in the considered link, ij . Depending on the medium of the link, it will be equal to t_{hop}^w or t_{hop}^{wl} .

Finally, the capacity constraint of link ij at the time packet p reaches node i can be represented as in (4.15).

$$(\forall i, j \in V)(\forall p \in P) \quad Y_{ij}^p \leq U_{ij}^p \quad (4.15)$$

4.2.3 Solution Approaches

We can solve the above routing problem globally using a linear optimization solver. This method maybe applicable for smaller networks. However, the solution time and complexity increase with the network size, raising concerns on the applicability of such a direct method. Moreover, in several situations, the network may not be able to find routes which are stable enough to deliver all the packets. In these situations, the optimization solver will result in an infeasible output. Yet, it is possible that there exist some stable routes that can deliver a limited number of packets out of the total packets. In order to retrieve this partial routing information, the optimization problem has to be executed step wisely decrementing the number of packets until a proper result is found. This yields a very high processing time.

In most of the existing routing algorithms, this processing time is neglected, which is a crucial factor in the efficiency of an algorithm. This is further important in highly dynamic networks such as vehicular, as the links only exist for a limited

amount of time. Therefore, it is of most importance that the routing algorithm computes and delivers routing information to the respective nodes as quickly as possible.

Therefore, in this subsection, we introduce an alternative incremental packet allocation scheme which can effectively solve the above routing problem. Later, we compare the computation time of both global and incremental optimization schemes against vehicle density and number of packets.

4.2.3.1 Global Optimization

In this subsection, we discuss how to solve the above routing problem as a global optimization problem using a linear solver such as Gurobi [115]. But first, we need to transform several formulations into linear form.

The $T_{s,r}^p$ expressions under channel restrictions (4.11, 4.12, 4.13), and link capacity representation (4.14) are formulated using logical conditions, and thus, need to be converted into linear form. For that, we use mixed logic constraints [116], which is a mathematical technique used in the translation of logical and nonlinear conditions into linear form.

We introduce three binary variables (b_1, b_2, b_3) as follows to assist in generalizing and linearizing the $T_{s,r}^p$ expression.

$$\text{Case 1 : } (H_{s,r}^{p,wl} + H_{s,r}^{p,w} \leq 2) \iff [b_1 = 1] \quad (4.16a)$$

$$\text{Case 2 : } (H_{s,r}^{p,wl} \leq 2 \wedge H_{s,r}^{p,w} \geq 1) \iff [b_2 = 1] \quad (4.16b)$$

$$\text{Case 3 : } (H_{s,r}^{p,wl} \geq 3) \iff [b_3 = 1] \quad (4.16c)$$

The mixed logic constraints can be directly applied to case 1 and 3 in the following manner.

$$g(\text{case 1}) = H_{s,r}^{p,wl} + H_{s,r}^{p,w} - 2 \quad (4.17a)$$

$$g(\text{case 3}) = 3 - H_{s,r}^{p,wl} \quad (4.17b)$$

In general, case i ($i \in \{1, 3\}$) can be linearized as,

$$g(\text{case } i) \leq M_i(1 - b_i) \quad (4.18a)$$

$$g(\text{case } i) \geq \epsilon + (m_i - \epsilon)b_i \quad (4.18b)$$

where $M_i (\triangleq \max : g(\text{case } i))$ are sufficiently big and $m_i (\triangleq \min : g(\text{case } i))$ are small integers. ϵ is the machine precision, beyond which the constraint is violated ($\epsilon > 0, \epsilon \sim 0$).

For case 2, we introduce another two auxiliary binary variables as b_2^1, b_2^2 such that $b_2 = b_2^1.b_2^2$ and,

$$(H_{s,r}^{p,w} \geq 1) \iff [b_2^1 = 1] \quad (4.19a)$$

$$(H_{s,r}^{p,wl} \leq 2) \iff [b_2^2 = 1] \quad (4.19b)$$

The non-linear AND logic expression, ($b_2 = b_2^1.b_2^2$) requires further linearization as follows.

$$-b_2^1 + b_2 \leq 0 \quad (4.20a)$$

$$-b_2^2 + b_2 \leq 0 \quad (4.20b)$$

$$b_2^1 + b_2^2 - b_2 \leq 1 \quad (4.20c)$$

The final representation of $T_{s,r}^p$ is formulated as in (4.21).

$$T_{s,r}^p = H_{s,r}^{p,wl} t_{hop}^{wl} + H_{s,r}^{p,w} t_{hop}^w + 2(p-1)t_{hop}^{wl} b_1 + (p-1)t_{hop}^{wl} b_2 + 3(p-1)t_{hop}^{wl} b_3 \quad (4.21)$$

In addition to timing constraints, the link capacity representation (4.14) also requires to be linearized. In order to do that, we introduce an auxiliary variable, $\gamma(r, p)$, as follows.

$$f(r, p) = \frac{T_{ij} - T_{s,r}^p}{t_{hop}^{ij}} \quad (4.22)$$

$$[f(r, p) \geq 0] \iff [\gamma(r, p) = 1] \quad (4.23a)$$

$$[f(r, p) < 0] \iff [\gamma(r, p) = 0] \quad (4.23b)$$

Linearization of the above logical equivalences follows the same format as in (4.16, 4.17, 4.18). However, to save space, detailed mathematical expressions are excluded from this chapter. Nevertheless, it leads the formulations to $U_{ij}^p = \lfloor f(i, p) \rfloor . \gamma(i, p)$. Here, the floor function can be mathematically formulated as

follows.

$$U_{ij}^p \leq \frac{T_{ij}}{t_{hop}^{ij}} \gamma(i, p) - \frac{1}{t_{hop}^{ij}} T_{s,i}^p \gamma(i, p) \quad (4.24a)$$

$$U_{ij}^p > \frac{T_{ij}}{t_{hop}^{ij}} \gamma(i, p) - \frac{1}{t_{hop}^{ij}} T_{s,i}^p \gamma(i, p) - \gamma(i, p) \quad (4.24b)$$

Still, there exists a nonlinear term in the formulations, $(T_{s,i}^p \cdot \gamma(i, p))$, and needs to be linearized using mixed logic constraints, similar to (4.20).

$$\tau(i, p) = T_{s,i}^p \delta(i, p) \quad (4.25a)$$

$$\tau(i, p) \leq W \delta(i, p) \quad (4.25b)$$

$$\tau(i, p) \geq w \delta(i, p) \quad (4.25c)$$

$$\tau(i, p) \leq T_{s,i}^p - w(1 - \delta(i, p)) \quad (4.25d)$$

$$\tau(i, p) \geq T_{s,i}^p - W(1 - \delta(i, p)) \quad (4.25e)$$

where W and w represent the maximum and minimum bounds of $\tau(i, p)$, respectively. Effectively they represent the maximum and minimum bounds of $T_{s,i}^p$ as $\delta(i, p)$ is a binary variable.

The final linear representation of U_{ij}^p is shown in (4.26).

$$U_{ij}^p \leq \frac{T_{ij}}{t_{hop}^{ij}} \gamma(i, p) - \frac{1}{t_{hop}^{ij}} \tau(i, p) \quad (4.26a)$$

$$U_{ij}^p > \frac{T_{ij}}{t_{hop}^{ij}} \gamma(i, p) - \frac{1}{t_{hop}^{ij}} \tau(i, p) - \gamma(i, p) \quad (4.26b)$$

Overall formulation of the above packet routing problem adopts features from maximum flow problem and minimum cost flow problem. We consider $T_{s,r}^p$ as an integer in the units of milliseconds and thus, all the decision variables are integers (including binary). Moreover, as all the formulations, including the objective function and constraints, are linearized, the overall problem is an ILP problem.

4.2.3.2 Incremental Optimization

In this subsection, we introduce the proposed incremental packet allocation scheme which can solve the previously modeled routing problem in a lower time complexity [117]. The incremental packet allocation scheme is shown in pseudo-code in Algorithm 1. The basic motivation behind this scheme is to allocate packets as

much as possible to the consecutive shortest paths without exceeding the lifetime based link capacities.

Algorithm 1 Incremental packet allocation scheme

Inputs: P - Number of packets, S - Source node, D - Destination node, N_{max} - Maximum permissible paths, T - Link lifetime array

Output: results

```

1:  $(cP, n, LSI) \leftarrow 0$   $\triangleright$   $cP$  - current packet,  $n$  - explored paths,  $LSI$  - last
   success path index
2: while  $(cP < P) \wedge (n < N_{max})$  do
3:    $path[n] \leftarrow K\_BiDirectionalDijkstra(S, D, n+1)$   $\triangleright$  find the  $(n+1)^{th}$ 
   shortest path using the Bi-directional Dijkstra algorithm and Yen's SPA
4:    $ij \leftarrow findBottleneckLink(path[n])$ 
5:   if  $feasibilityCheck(path[n], ij, S, T)$  then
6:      $p' \leftarrow cP$ 
7:     while  $\frac{T_{ij} - T_{s,i}^{p'}}{t_{hop}^{ij}} > 1$  do
8:        $p'++$ 
9:     end while
10:     $result + \leftarrow (p' - cP, path[n])$ 
11:     $cP \leftarrow p'$ 
12:     $LSI \leftarrow n$ 
13:  end if
14:   $n++$ 
15: end while
16: if  $(cP < P)$  then
17:    $result + \leftarrow (P - cP, path[LSI])$ 
18: end if
19: return result

```

As illustrated in the algorithm, we iteratively find the n^{th} shortest path with n incremented at each step. A combination of Bi-directional Dijkstra Shortest Path Algorithm (SPA) and K-shortest path algorithm [33,34] is used to find successive shortest paths. Moreover, we use $(C_t t_{hop}^{ij} + C_v V_{eff}^{ij})$ as the cost metric to empower both lower latency and error rate in route selection.

As links in a vehicular environment are bidirectional, we can employ the bidirectional SPA. In this algorithm, rather than exploring nodes only from the source node, we start to explore from both source and destination nodes as shown in Fig. 4.6b. The node exploration continues until both circles meet in a middle node, and the shortest path is found by extracting the minimum distance from different nodes in two circles. With respect to the conventional Dijkstra SPA, the bi-directional SPA explores fewer number of nodes, and thus, faster, especially

when the source and destination nodes are far away from each other as graphically indicated in the Fig. 4.6.

Figure 4.6: Node exploration in Dijkstra algorithms (circles indicate the nodes explored in finding the shortest path).

In order to find successive paths, we use a modified version of the Yen's SPA [34]. Here, we find the first shortest path using the bi-directional SPA and then, in order to find the next shortest path, we delete the individual links step-wise in the first shortest path and look for any new shortest paths from the resulting networks and store them in a shortest path pool. Then, the path with the lowest cost is picked from the shortest path pool as the next shortest path. Similarly, to find the K^{th} shortest path, we delete the individual links iteratively from the $K - 1$ shortest path, and find the shortest paths from the resulting networks and insert them into the shortest path pool, if they are not already in. Thereafter, the path with the least cost will be selected from the pool, if not already selected as one of the previous $K - 1$ shortest paths, as the next best (K^{th} , in this case) shortest path.

Next, before assessing the stability of the path, we extract out the bottleneck link of the path, as it is a qualified representation of the path's stability. It can be found out by estimating the remaining link lifetimes at the times the considered packet reaches the starting node of the links, and finding the link with the minimum value as formulated in (4.27).

$$\text{link } ij \text{ index} = \frac{T_{ij} - T_{s,r}^p}{t_{hop}^{ij}} \quad (4.27a)$$

$$\text{bottleneck link index} = \min : (\text{link } ij \text{ index}) \quad (4.27b)$$

where p = number of currently allocated packets + 1.

If the bottleneck link satisfies the previously modeled link capacity constraints, then, it can be concluded that all the links in the path successfully satisfy the link capacity constraint. This yields a competitive edge over the global optimization in which all the links are assessed by the link capacity constraints, and further, one of the main reasons for the higher computational efficiency.

If the selected path passes the feasibility test, we allocate packets as much as possible to the corresponding route until the link capacity is not exceeded as shown by the lines 7, 8 in Algorithm 1. Then, if there are still packets to be allocated, the algorithm goes for a higher shortest path by incrementing n . The process continues until either all the packets are allocated or the maximum permissible paths are explored. If the maximum permissible path limit (N_{max}) is reached, the remaining packets will be allocated to the last successfully explored path, as it is the most stable path with the highest probability of successfully transmitting the packets. With respect to the global optimization scheme, the incremental scheme is capable of finding a finite set of packet allocated routes in a less time complexity.

The time complexity of Algorithm 1 is derived with respect to the best case and the worst case as follows. In the best case, all the packets are assigned to the first shortest path assuming it is stable enough to deliver all of them. Thus,

$$\begin{aligned} \text{Best case time complexity} &= O(1)(O(\mathcal{Y}) + O(\mathcal{B}) + O(\mathcal{F} * P)) \\ &= O(\mathcal{Y} + \mathcal{B} + \mathcal{F} * P) \end{aligned} \quad (4.28)$$

Here, $O(\mathcal{Y})$ refers to the complexity of Yen's K-shortest path algorithm, $O(\mathcal{B})$ refers to the complexity of finding the bottleneck link, $O(\mathcal{F})$ indicates the complexity of feasibility check, and P refers to the number of packets.

In the worst case, all permissible paths are explored to route the packets. Thus,

$$\begin{aligned} \text{Worst case time complexity} &= O(N_{max})(O(\mathcal{Y}) + O(\mathcal{B}) + O(\mathcal{F} * 1)) \\ &= O(N_{max}(\mathcal{Y} + \mathcal{B} + \mathcal{F})) \end{aligned} \quad (4.29)$$

where, N_{max} is the maximum permissible paths.

The time complexity of Yen's algorithm, $O(\mathcal{Y}) = O(KN(M + N \log N))$ [118], where K is the degree of the shortest path as in 1^{st} , 2^{nd} , etc. M is the number of edges and N is the number of nodes in the network. Finding the bottleneck link

is similar to finding a minimum value from an array, and thus, $O(\mathcal{B}) = O(N_{size})$, where N_{size} is the path length. In feasibility check, we compare the lifetime of the bottleneck link with the travel time of the corresponding packet leading up to that link. Thus, a constant complexity can be assumed. $O(\mathcal{F}) = O(1)$.

4.3 Source Routing based FI Scheme

In SDVN, when a node receives a packet, it first checks for a matching entry in its flow table. If a match is found, the corresponding action will be performed. If not, a PACKET IN message which includes information about the source and destination nodes, is generated and forwarded to the controller. In response to a PACKET IN message, the control plane computes the required route using the previously described routing algorithm. The contribution of this section lies in the way this routing information is delivered to the respective nodes.

4.3.1 Route Valid Time Estimation

Before sending the flow rules, the control plane will estimate whether the flow information should be stored at intermediate nodes, based on the valid time of the route. The valid time of a route is different at different nodes in the path. At a considered node, the valid time is equal to the minimum link lifetime of its all downstream links in the path as expressed in (4.30).

$$\begin{aligned} \text{Route valid time at node } r, \quad T_r^v &= \min.(\tau_{ij}) \\ \text{where } ij &\in L_r^D \end{aligned} \tag{4.30}$$

where L_r^D refers to the set of downstream links in the path from node r and τ_{ij} represents the lifetime of link ij .

Figure 4.7: Illustration of vehicle interconnection and velocity profiles.

The lifetime of a link primarily depends on the current position of the two nodes

that formed the link, and their relative velocities. It can be modeled as a simple sub-optimization problem as in (4.31) with Fig. 4.7 as the reference.

$$\begin{aligned}
 &\text{maximize: } (\Delta\tau_{ij}) \\
 &\text{subject to } \{x_i - x_j + [v_i^x(t_o) - v_j^x(t_o)]\Delta\tau_{ij} + \frac{a_i^x - a_j^x}{2}\Delta\tau_{ij}^2\}^2 + \\
 &\quad \{y_i - y_j + [v_i^y(t_o) - v_j^y(t_o)]\Delta\tau_{ij} + \frac{a_i^y - a_j^y}{2}\Delta\tau_{ij}^2\}^2 \leq D_R^2(\bar{x}, \bar{y})
 \end{aligned} \tag{4.31}$$

where $\Delta\tau_{ij}$ refers to the residual lifetime of link ij , t_o indicates the time of the last received update, and x_i, x_j and y_i, y_j represent the x and y coordinates of the two nodes. $D_R(\bar{x}, \bar{y})$ indicates the statistically estimated transmission range of the nodes at $\bar{x}$ and $\bar{y}$, in which $\bar{x} = \frac{x_i + x_j}{2}$, $\bar{y} = \frac{y_i + y_j}{2}$.

The effective transmission range is affected by the obstacles in that particular area. Thus, the use of a constant value for the maximum transmission range, may lead to incorrect estimations of link lifetime. To mitigate this effect, we statistically profile the maximum effective transmission range with respect to the position and generate Probability Density Functions (PDFs), and contour maps.

The terms, a_i^x and a_i^y refer to the x and y components of node i 's acceleration, and they can be calculated as, $a_i^x = \frac{v_i^x(t_o + \Delta\tau_{ij}) - v_i^x(t_o)}{\Delta\tau_{ij}}$, $a_i^y = \frac{v_i^y(t_o + \Delta\tau_{ij}) - v_i^y(t_o)}{\Delta\tau_{ij}}$.

Further, $v_i^x(t)$, $v_i^y(t)$ represent the projected x and y velocity components of node i at time t . Here, the vehicle speed is not treated as a deterministic constant, and rather, its variable nature is acknowledged. Differentiating from the existing studies, which model the speed as a stochastic variable with a probabilistic approach [119], we exploit the knowledge that the control plane has over the vehicles' recent speed profiles. Thus, our approach is to predict the vehicle's speed profile in near future using curve fitting (regression) models. Existing literature shows that a vehicle's speed is generally affected by the speed of vehicles in-front [119]. Given that, the proposed SDVN system owns those information, we include them in the regression model as well, however, with a less priority. The set of vehicles that is interest to a considered vehicle's speed is selected based on the location information and velocity vectors as shown in Fig. 4.8.

From the vehicles within the given boundary, those who show positive dot products in terms of placement ($\vec{V}_1 \cdot \vec{AB} > 0$) and velocity ($\vec{V}_1 \cdot \vec{V}_2 > 0$) with the considered vehicle are selected as vehicles with interest.

We use both linear and nonlinear regression models in predicting the velocity of

Figure 4.8: Selection of candidate vehicles.

the vehicle. The fundamental concept is to select the function with the least sum of error. The sum of error can be represented as in (4.32).

$$\text{Sum of error} = \sum_{i=1}^n C_i [Y_i - f(t_i)]^2 \quad (4.32)$$

where n refers to the number of data points, $f(t)$ is the predicted velocity profile, Y_i is the recorded velocity of the considered vehicle at time t_i , and C_i is the weight of the i^{th} data point. C_i 's value depends on the vehicle that the corresponding data point belongs to. If it is from the considered vehicle, its value will be high. C_i will be low if it is from a vehicle away from the considered vehicle.

In linear regression, $f(t) = at + b$, while in nonlinear regression, only a 2nd order polynomial function, $f(t) = a_0 + a_1t + a_2t^2$, is considered. The sum of error expression, which is to be minimized is partially differentiated with respect to different coefficients, and solved for the minimum.

After getting the best-fitted curve and line, the corresponding sum of errors are calculated and the function with the least sum of error is selected as the velocity profile of the considered vehicle. An example case is shown in Fig. 4.9.

The problem in (4.31) can be simply solved by converting the inequality constraint into an equality and then solving the resulting 4th order polynomial equation shown

Figure 4.9: Projection of vehicle velocity.

in (4.33).

$$\begin{aligned}
 & \Delta\tau_{ij}^4[A_{ij}^{x^2} + A_{ij}^{y^2}]/4 + \Delta\tau_{ij}^3[V_{ij}^x(t_o)A_{ij}^x + V_{ij}^y(t_o)A_{ij}^y]/2 + \\
 & \Delta\tau_{ij}^2[V_{ij}^x(t_o)^2 + V_{ij}^y(t_o)^2 + (X_{ij}A_{ij}^x + Y_{ij}A_{ij}^y)/2] + \\
 & \Delta\tau_{ij}[2X_{ij}V_{ij}^x(t_o) + 2Y_{ij}V_{ij}^y(t_o)] + X_{ij} + Y_{ij} - D_R^2(\bar{x}, \bar{y}) = 0
 \end{aligned} \tag{4.33}$$

where $x_i - x_j = X_{ij}$, $y_i - y_j = Y_{ij}$, $v_i^x(t_o) - v_j^x(t_o) = V_{ij}^x(t_o)$, $v_i^y(t_o) - v_j^y(t_o) = V_{ij}^y(t_o)$, $a_i^x - a_j^x = A_{ij}^x$, and $a_i^y - a_j^y = A_{ij}^y$.

We use Ferrari's method [120] to solve the quartic equation in (4.33). Ferrari's method reduces the 4th degree of the original quartic equation into one cubic or two quadratic equations and derives the final answer. Thus, the time complexity of Ferrari's method (or even link lifetime estimation) can be taken as constant, $O(1)$. Its usage is shown in Algorithm 2, where the estimation of route valid time at intermediate nodes is illustrated.

The route caching decision is taken by the controller based on the evaluation of route valid times. If the route shows an enough valid time at a particular node ($>$ data transmission time from the considered node to the destination + time to deliver the flow rules to that node), the decision is taken to cache the route information as shown in (4.34).

Algorithm 2 Computation of path (route) valid time

Input: $n[N]$, $p[N, N]$, $q[N, N]$, $r[N, N]$, $s[N, N]$, $u[N, N]$ $\triangleright n[N]$ - set of nodes in the computed path. p , q , r , s , and u are the coefficients of 4th to 1st order terms, respectively, of the quartic equation in (4.33).

Output: $T_V[N]$ $\triangleright$ Path valid times at each node

```

1:  $T_{low} \leftarrow M$   $\triangleright M$  - very large positive value
2:  $T[i, i + 1] \leftarrow 0$   $\triangleright$  Link lifetime array
3: for ( $i = (N - 1) \rightarrow 1$ ) do
4:   if ( $p == 0 \wedge q == 0 \wedge r == 0 \wedge s == 0$ ) then
5:      $T[i, i + 1] \leftarrow T_c$   $\triangleright$  Wired: High constant time
6:   else
7:      $T[i, i + 1] \leftarrow \text{FerrariMethod}(p[i, i + 1], q[i, i + 1], r[i, i + 1], s[i, i + 1], u[i, i + 1])$   $\triangleright$  Wireless: Solution of the quartic equation (4.33).
8:   end if
9:   if ( $T_{low} > T[i, i + 1]$ ) then
10:     $T_V[i] \leftarrow T[i, i + 1]$ 
11:     $T_{low} \leftarrow T[i, i + 1]$ 
12:   else
13:     $T_V[i] \leftarrow T_{low}$ 
14:   end if
15: end for
16: return  $T_V$ 

```

Route cache at node r , $U(T_V[r] - (D_{c,r} + 2D_{r,d}))$

$$= \begin{cases} 1, & \text{if } T_V[r] \geq D_{c,r} + 2D_{r,d} \\ 0, & \text{otherwise} \end{cases} \quad (4.34)$$

where $U(\cdot)$ denotes the unit step function, $D_{r,d}$ is the time taken to transmit a packet from node r to destination node d , and $D_{c,r}$ is the time to deliver the flow information from the controller to the node r via source routing. It should be noted that $D_{c,r}$ encompasses the inter-arrival time of packets as well.

The resulting route valid time at node r ,

$$T_V^C[r] = T_V[r] \cdot U(T_V[r] - (D_{c,r} + 2 * D_{r,d})) \quad (4.35)$$

The motivation is to save the flow rules only at the nodes which are associated with sufficient route valid times to transmit at least one more packet. Overall, it cuts down unnecessary route caching and helps to lower the communication delay with less work at garbage collection.

4.3.2 Source Routing

The computed routing information is then delivered to the respective nodes using a conventional routing mechanism known as source routing, which is used in protocols such as DSR [28]. All the path information will be embedded in a modified FLOW MOD message ($FLOW\ MOD^*$) and send to the route requested node (usually the source node).

Upon receiving the $FLOW\ MOD^*$ message which includes sequences of information of intermediate nodes and valid times, the source node will extract the information related to the first path and embeds in the buffered packet and forward to the next hop. The remaining path information (if any) will be used at the designated packet due times.

When the path appended data packet (data*) is transversed in the network, all the intermediate nodes will check the corresponding route valid times in the header and if it is positive, the respective IP address and route timeout, which we refer to as a *segment*, will be stored in their flow tables. Moreover, they will remove the path information related to the visited nodes and shift the header before forwarding to the next hop, and thus, this mechanism generates a lower average routing overhead⁵. The overall process is graphically illustrated in Fig. 4.10.

Figure 4.10: Proposed FI operation.

4.3.3 OpenFlow Realization

In order to make this process a realization, we introduce several modifications to the existing OpenFlow protocol. The current OpenFlow protocol will update the

⁵The routing overhead is defined as the total packet size utilized in delivery of routing information to the required nodes in the respective path(s).

flow table and send the packet to the next hop without any alteration in response to a FLOW MOD [18]. However, we modified this protocol by adding new action types such that upon receiving a *FLOW MOD**, the node will first extract the information related to the first route and then, the first segment (IP and valid time) of the extracted route will be saved in its flow table and the remaining segments will be appended in the option field of the corresponding buffered data packet. Next, the modified data packet (data*) will be forwarded in the data plane. The remaining routing information (if any) will be temporarily buffered to process in due times.

Moreover, we introduce new protocol ids in the protocol field of the IPv4 packet format so that a data* packet can be detected by the OpenFlow protocol. If a match is found for a data* packet, the corresponding action is to extract the first segment of the routing information from the option field, which reflects the next hop information, and save it in its flow table, similar to FLOW MOD* operation (refer Fig. 4.10).

In case of a route breakdown, the respective node will consult the assigned controller for route information and the above process will be re-initiated.

4.3.4 Path Encoding

In this subsection, we discuss the packet formatting related to the aforementioned operation. We specify routing as the *reason* in the PACKET IN header and embed the number of packets required to be routed in the *data* field, and thus, no alteration to the PACKET IN format is required.

As the computed paths are appended in the FLOW MOD message, the classic FLOW MOD format is altered to *FLOW MOD** as shown in Fig. 4.11. We no longer require the *idle timeout*, *out port* and *out group* fields in the FLOW MOD message. Rather, we have a common *hard timeout* and a sequence of segments in the *FLOW MOD** message. Additionally, we include an attribute called *number of paths* in the *FLOW MOD** header, as these messages can carry multiple path information.

In conventional wired networks, differentiable port numbers can be used to indicate specific next hop information [31]. However, vehicular networks do not have this opportunity and thus, we are compelled to use IP addresses to indicate next hop information. Yet, it can be justified by the fact that communication in VANET is usually within a few hops in compared to wired networks.

Figure 4.11: Packet format of *FLOW MOD** message.

4.4 Performance Evaluation

In this section, we comparatively evaluate the performance of our proposed routing framework in terms of effectiveness of the results (time complexity and accuracy) and simulation metrics such as packet reception rate, latency and overheads.

4.4.1 Agility and Accuracy

First, we compare the time complexity and the accuracy of the proposed incremental optimization based routing algorithm against the global optimization scheme and basic Dijkstra SPA. We considered different instances of vehicular networks with varying vehicle density and number of packets and executed the above mentioned three algorithms against multiple source-destination pairs. The considered network map is a simplified version of the NTU, Singapore campus map with 20 RSUs covering the entire area as shown in Fig. 3.7. The number of nodes was changed from 30 to 100 in steps of 10 in both Figs of 4.12 and the number of packets was changed from 1 to 16 in Fig. 4.12a and was fixed at 16 in Fig. 4.12b.

Figure 4.12: Effectiveness of different solution approaches.

The Gurobi commercial optimizer [115] was used to solve the global linear interpretation of the routing problem and C++ language was used to implement and solve both incremental optimization scheme and Dijkstra SPA. A workstation with a Intel-core i7 3.40 GHz processor, and a 8 GB RAM was used in performing the experiments. The corresponding results are shown in Fig. 4.12. The computational time of the Global optimization scheme spans from around 220 ms to 100 sec. To avoid anomalies, it is represented as a plane at the top.

It is important to note that both Dijkstra SPA and incremental optimization scheme display processing delays in scale of milliseconds, while global optimization goes even up to few hundreds of seconds (Fig. 4.12a). Such a high latency profile is not applicable in vehicular networks as the latency requirement for multi-hop data transmission is generally under 500 ms [12]. The proposed incremental optimization scheme is well within the required limits and thus, satisfactorily meets the requirement. Even though the shortest path approach exhibits a very low processing delay, inconsideration of number packets and stability of routes have adversely affected the packet reception (Fig. 4.12b). Meanwhile, the proposed computational strategy almost aligns with the global strategy in terms of number of dropped packets and objective value. However, at several data points (when the number of nodes is 40, 80, and 100), the objective value of the incremental approach is lower than that of global optimization. Yet, observe that, the number of dropped packets of the former is higher than that of the latter. Hence, the two plots in Fig. 4.12b should be read together to extract the correct interpretation.

4.4.2 Simulation Scenarios

In this section, we simulate our optimized routing framework in NS-3 [105] and measure its performance against an existing SDVN routing protocol and conventional VANET routing protocols such as AODV and ROMSGP [64]. We use PDR, average end to end delay, routing overhead and state information overhead (total network state information acquired by the control plane) as the performance metrics.

4.4.2.1 Simulation Configuration

We use the same simplified NTU, Singapore campus map with 20 scattered RSUs as the simulation environment which covers an approximate area of $1500 \times 1500 m^2$. Realistic mobility traces are extracted from SUMO [121] for each simulation sce-

nario and fed into NS-3 [105]. Hello and beacon message interval is set to 250 ms. Data transmission follows a Poisson distribution with a mean of 4 sec. Each transmission is associated with multiple packets which are randomly selected based on a uniform distribution between 1-10. The packet arrival interval is set to 100 ms. Additional parameters and values are tabulated in Table 4.2.

Table 4.2: Simulation parameters of link stability based routing framework.

Parameter	Value
Simulation time	500 seconds per run with 25 runs per each point
Packet sizes	Packet In = 32 bytes, Flow Mod = 56+ bytes, Data = 1500 bytes
Transmission	Type = UDP, Range = 200 m (DSRC)
Propagation model	Friis Propagation Loss model
Wireless network	DSRC (IEEE 802.11p) = 6 Mbps
Wired network (RSU-RSU)	Ethernet (IEEE 802.3) = 100 Mbps, Propagation delay = 1 ms
Wired network (Controller-RSU)	Broadband = 1 Gbps, Propagation delay = 10 ms
Internet backbone	Data Rate = 10 Gbps, Propagation delay = 10 ms

In order to make the simulations more realistic, we impose a random processing delay for computing the route information at the control plane. For that, we first generated the best-fitted processing delay graphs using our sample data at each number of packets and nodes (Fig. 4.12a). All the best-fitted graphs resulted in Gaussian distributions. Thus, processing times were randomly selected using corresponding Gaussian distributions.

The overall simulation results are taken by varying the number of vehicles (vehicle maximum speed fixed at 20 ms^{-1}) and vehicle speed (number of vehicles fixed at 60). The results are shown in Fig. 4.13, 4.14, 4.15a and 4.15b for PDR, average end to end delay, routing overhead and state information collection overhead, respectively. The Fig. 4.16a and 4.16b depict theoretical comparisons of different attributes of FI schemes. Packet overhead and latency overhead respectively refer to the additional packet size and delay consumed in delivery of flows.

Figure 4.13: PDR comparison.

Figure 4.14: Latency comparison.

4.4.2.2 Results Analysis

According to the PDR comparison shown in Fig. 4.13, as expected, the proposed optimized routing scheme displays the highest packet reception ratio at all vehicle density levels and speeds. AODV and shortest path approach have struggled to maintain competitive PDRs due to the inconsideration of stability. Since the ROMSGP utilizes the most stable path, it has maintained a competitive packet reception rate against increasing vehicles and mobility. Yet, inefficiencies in distributed routing schemes and unavailability of global network information have limited its PDR rate. Overall, the proposed routing framework has a competitive edge with a 15–20% PDR gap over the shortest path approach in both graphs (Fig. 4.13a, 4.13b). In depth analysis over stability of routes in the proposed framework can be stated as the root cause for the accomplished PDR improvement.

Figure 4.15: Experimental comparison of routing attributes.

Figure 4.16: Theoretical comparison of FI attributes.

In terms of latency (Fig. 4.14), AODV clearly suffers a very high delay profile due to their broadcast approach in finding routes. The latency performance gets poorer with increasing number of vehicles and speed, as in both cases, more RREQ packets are broadcasted in the network, making the network more congested. Even though, ROMSGP utilizes a broadcasting approach in finding routes, the higher stability in selected routes causes less route discovering cycles, and it has helped to exhibits a lower competitive latency. The proposed routing framework presents very low delay patterns against both number of vehicles and speed, mainly owing to the source routing based FI scheme, as it minimizes the extent of communication with the control plane (Fig. 4.16a). Accurate consideration of link dynamics in the routing algorithm is also equally important in achieving the lower delay profile⁶,

⁶Observe that the latency measurements in Fig. 4.14 represent the whole operational delay in routing, including route request, route computation (note that, a processing delay is added

as it selects multiple routes beforehand which are stable enough to transmit all the given packets. However, in the shortest path approach, the nodes only receive single path information and are compelled to consult the controller frequently when the route expires, leading to slightly higher communication delays as verified by Fig. 4.14a and 4.14b.

This is further evident in the experimental routing overhead shown in Fig. 4.15a. According to the graph, the shortest path scheme generates a considerable routing overhead when the vehicle speed increases as the links become vulnerable in their existence with high mobility leading to shorter route valid times. Thus, the nodes are compelled to consult the controller frequently. In the case of ROMSGP, its broadcast based route discovery process is the main cause for the extremely high overhead. However, the proposed routing framework displays a significantly low routing overhead due to its source routing based FI scheme and consideration over link dynamics, as both of them help to minimize the controller consultation.

As shown in Fig. 4.16a, the proposed FI mode yields the lowest forwarding state distribution against increasing number of hops. The underlying reason is that the controller sends all the path information to the data plane in a single transmission. However, in other FIs, packet overhead linearly increases with higher slopes against the number of hops, as the controller separately delivers flows to each node. Even though, in 4.16b, the proactive FI displays a very low constant latency overhead, architectural challenges limit its adaptability in SDVN.

The yielded performance improvement is further helped by the offloading of state information collection to RSUs, as it takes out the periodical status message transmission of the vehicles, as empirically shown in Fig. 4.15b. Overall, the results have comparatively validated the effectiveness of the proposed routing framework and its practicality in the vehicular domain.

4.5 Summary

In this chapter, we proposed a novel routing framework for SDVN scrutinizing the stability of routes in addition to the shortest travel time. The routing problem is modeled as an ILP problem. We further introduced an incremental packet allocation scheme to solve the routing problem in a less time complexity. The resulting framework routes the packets via multiple routes along the shortest possible paths

resembling the computation time in Fig. 4.12a), flow instantiation, and actual data transmission.

with enough stability. Additionally, a source routing based FI operation was introduced to efficiently deliver the routing information to the respective nodes. The proposed FI scheme encapsulates all the routing information in a modified flow mod message and sends only to the route requested node, minimizing the extent of communication between the control and data planes. Based on the route valid times, the path information is only cached at several selected nodes to avoid unnecessary route caching. Via simulation results, we have shown that the proposed routing framework improves the performance of multi-hop data transmission in vehicular networks in multiple-dimensions with respect to both existing VANET and SDVN routing protocols.

Chapter 5

Routing in Software Defined Vehicular Network under Uncertain Conditions

5.1 Introduction

In the previous chapter, we assumed that the control plane possesses adequate network information on the vehicular network during the proposed routing framework [109–111]. The control plane gathers network information through the periodical beaconing of status via LTE in existing SDVN systems and with the help of RSU agents in our proposed HD-SDVN. However, as mentioned in chapter 1, the required infrastructure support for the above operation may not exist at all areas, and moreover, the received information soon become obsolete due to the highly dynamic nature of vehicular networks, inducing a sense of uncertainty in the selecting of optimal routes. This is further worsened by the diversity in network conditions, especially by the occurrence of sparse network conditions pertaining to sub-urban and rural areas. Thus, in this chapter, we aim to drop the assumption of an ideal network and incorporate the aforementioned practical realizations in an extended version of the routing framework in the previous chapter [122, 123].

Outdated network information and gaps in the network topology can impose challenges in finding stable and connected paths. These black holes are handled through a combination of broadcasting based distributed routing phase and unicast based centralized routing phase. The sparse network conditions are handled through an improved SCF routing scheme that explores future connectivity be-

tween vehicular nodes with least store time utilizing the available global network view. Similar to the previous routing model, if an accurate image on the network is available, we follow an optimization based centralized routing protocol based on unicast forwarding.

Thus, the resulting model is a tribrid routing protocol which consists of unicast, broadcast and SCF techniques for a general vehicular network.

The remainder of the chapter is structured as follows. In section 5.2 and 5.3, we layout the fundamentals related to the routing problem and formulate the routing problem on a case by case basis, respectively. Section 5.4 analyses the results, and finally, section 5.5 concludes this chapter.

5.2 Problem Statement

In this section, we lay out the fundamentals related to the routing problem under uncertain network conditions. As in the previous routing model, this chapter also uses the HD-SDVN architecture and operation introduced in chapter 3 as the underlying system model [75,90].

As mentioned earlier, vehicular links are highly unstable and their connectivity exists only for a limited time due to the associated high mobility of vehicles. Therefore, the link connectivity should be a decisive factor in selecting a route in addition to the latency associated with the route [26]. Overall, all the links in a selected route should exist from the route discovered time, until the time the packet(s) passes each link and reaches the destination. We have shown the importance of link connectivity in routing through multiple example scenarios in the previous chapter.

The current literature to overcome the link stability problem is based on the selection of the most stable path. However, as mentioned above, this approach could lead to higher delays due to the possible selection of longer routes. Alternatively, the proposed system focuses on optimizing the latency associated with the route without violating the necessary link stability requirement.

Yet, due to certain circumstances, a feasible route may not be found using the above method. The concerning circumstances that we focus on this chapter are listed below.

1. Obsolete network information.

2. Unavailability of network information.
3. Sparse network conditions.

The above factors individually or in combinations can cause black holes, restricting the connectivity between the source and the destination as shown in Fig. 5.1. Therefore, they may develop situations such as,

1. Inability to find a route between the source and the destination due to unavailability of network information.
2. Inability to accurately estimate the stability of paths owing to outdated network information.

Figure 5.1: Disturbance in network information.

This barrier (disturbance) could be found at different locations in the network as shown in Fig. 5.1, and different treatments are applied based on the location of the barrier. For an example, if the barrier is found at the location **I**, it implies that the source node has no neighbors, whereas if the barrier is at location **II**, it implies that the connectivity between the source and the destination is breached at some point either by obsolete or no network information or sparse network conditions. The barrier at location **III** refers to the situation where the destination's location is unknown. This is most likely due to the unavailability of network information.

The disturbance at location **I** and **II** are handled by SCF as accurate information including mobility and location of the nodes adjacent to the barrier is available. However, at location **III**, the SCF might not be the ideal solution as updated mobility and location information of the destination is not available, and thus, the computation of store time in this situation can lead to inaccurate estimations. Therefore, we incorporate a limited broadcast cycle in the area with no network information, assuming packets will reach the destination in a few broadcast hops.

5.3 Problem Formulation

In this section, we formulate the tribrid routing problem addressing the issues mentioned in the problem statement. Similar to the previous chapter, our focus here is to route a given set of packets in the shortest possible paths without violating the stability requirements. In case of uncertain conditions, we incorporate several distributed routing techniques. Overall, we make use of the following routing concepts in the system.

1. Unicast based centralized routing
2. Broadcast based distributed routing
3. Store, carry and forward routing

5.3.1 Notations

Given a network graph, $G = (V, E)$ where V and E refer to the set of nodes (OBUs and RSUs), and edges connecting the nodes, respectively, the source node, $s \in V$ and the destination node, $d \in V$, our objective is to successfully deliver a given set of packets, $P(|P| = n_p)$ with least delivery times without exceeding the dynamic link capacity values and delay tolerances of the packets including the effect of uncertainty. Each link, ij , is associated with three metrics, the lifetime (τ_{ij}^s), effective velocity (v_{ij}^{eff}) and transmission time ($t_{ij}^{hop} = \text{hop time}$).

The lifetime of link ij (τ_{ij}^s) refers to the residual lifetime of the link at the moment flow rules reach the source node. v_{ij}^{eff} refers to the effective velocity associated with the link ij , where $v_{ij}^{eff} = \sqrt{\left[v_i^{\rightarrow ij}\right]^2 + \left[v_j^{\leftarrow ji}\right]^2}$ [6]. Here, $v_i^{\rightarrow ij}$ indicates the velocity component of the node i in the direction of the node j .

Depending on the link type, the hop time t_{ij}^{hop} , will either be t_w^{hop} (wired) and t_{wl}^{hop} (wireless). Detailed expressions for the hop time are depicted in chapter 3 under section 3.3.2. Similar to the previous chapter, here, the hop time is associated with a certain level of uncertainty as in (4.1).

Y_{ij}^p is the main decision variable used in the model, which indicates the packet p 's trajectory through the link ij . Moreover, $H_{s,r}^{w,p}$ and $H_{s,r}^{wl,p}$ represent the number of wired and wireless hops, respectively, from the source node, s to an intermediate node, r via the estimated route for the packet p . $T_{s,r}^p$ refers to the time taken by the p^{th} packet to reach an intermediate node, r from the source node, s via the estimated path. N refers to the number of nodes in the network. Further, N_i^{in}

and N_i^{out} refer to the set of neighbors with incoming links to node i and outgoing links from node i , respectively. ϵ_{max}^p refers to the delay tolerance margin of the p^{th} packet. The notations used in this problem are summarized in Table 5.1.

Table 5.1: A summary of notations

Parameter	Description
V, E, P	Set of nodes, edges and packets, respectively
n_p	Number of packets
τ_{ij}^s	Residual lifetime of link ij at the moment flow rules reach source node s
v_{ij}^{eff}	Effective velocity associated with the link ij
t_{ij}^{hop}	Packet transmission time through link ij (Hop time)
$v_i^{ij \rightarrow}$	Velocity component of node i in the direction of the node j
t_w^{hop}, t_{wl}^{hop}	Hop time (w - wired, wl - wireless)
Y_{ij}^p	Indicator of packet p 's trajectory through link ij
$H_{s,r}^{w,p}, H_{s,r}^{wl,p}$	Number of wired ($x = w$) or wireless ($x = wl$) hops from source node s to intermediate node r via the estimated route for packet p
$T_{s,r}^p$	Time taken by p^{th} packet to reach intermediate node r
N_i^{in}	Set of neighbors with incoming links to node i
N_i^{out}	Set of neighbors with outgoing links from node i
ϵ_{max}^p	Delay tolerance margin of p^{th} packet
v_{ij}^{eff}	Effective velocity associated with link ij
$L_{s,r}^{u,x}$	Set of all upstream wired ($x = w$) and wireless ($x = wl$) links of node r from source node s
t_{gap}	Time gap between consecutive packets
$\tau_{rr'}^{r,p}$	The residual lifetime of a common link ($r - r'$) at the moment packet p reaches the starting node of that link
t_{ij}^{scf}	Store time of link ij

5.3.2 Centralized Unicast Routing

In this part, we assess the connectivity related stability of the routes. One main assumption made here is that updated network information is available at the control plane.

The proposed model routes a given set of packets via unicast in the vehicular network between the source and the destination under link stability and latency based QoS constraints. The centralized unicast routing model is similar to the routing framework introduced in the previous chapter, and thus, detailed information is excluded in this section.

5.3.2.1 Objective Function

The objective of this model is to maximize the delivery of packets by routing them in short and stable paths, and thus, the objective function can be formulated as in (5.1).

$$\text{minimize} \quad \sum_{i=1, i \neq j}^N \sum_{j=1}^N \sum_{p=1}^{n_p} [C_t t_{ij}^{hop} + C_v v_{ij}^{eff}] Y_{ij}^p - \sum_{p=1}^{n_p} \sum_{\forall i \in N_d^{in}} C_p Y_{id}^p \quad (5.1)$$

The term, t_{ij}^{hop} is used to incentivize the selection of shortest paths (paths with least travel times), and v_{ij}^{eff} to encourage less dynamic links, similar to the previous routing framework.

Moreover, the last term, $\sum_{p=1}^{n_p} \sum_{\forall i \in N_d^{in}} C_p Y_{id}^p$ is used to maximize the number of delivered packets. The values of the coefficients are selected as $C_p \gg C_t \gg C_v$ in order to symbolize their respective priorities.

5.3.2.2 Constraints

Most of the constraints of the centralized routing section, including routing and feasibility assessment are similar to the previous routing framework. Yet, in order to have a complete framework for this chapter, we re-introduce them with minimal explanations as follows.

- **Successive shortest paths**

Similar to the previous routing model, the following equations model the routing of the packets in the network. Yet, the constraints (5.2) and (5.4) are modeled as inequalities allowing packet cancellation options, as it is probable that all the packets may not be able to find stable paths to reach the destination.

$$(\forall p \in P) \quad \sum_{\forall j \in N_s^{out}} Y_{sj}^p \leq 1 \quad (5.2)$$

$$(\forall k \in V)(\forall p \in P) \quad - \sum_{\forall i \in N_k^{in}} Y_{ik}^p + \sum_{\forall j \in N_k^{out}} Y_{kj}^p = 0 \quad (5.3)$$

$$(\forall p \in P) \quad \sum_{\forall i \in N_d^{in}} Y_{id}^p \leq 1 \quad (5.4)$$

The equation (5.5) avoids loops in the path that can occur due to the bidirectional nature of the links in the vehicular networks.

$$(\forall i, j \in V)(\forall p \in P) \quad Y_{ij}^p + Y_{ji}^p \leq 1 \quad (5.5)$$

The above problem can be solved with the help of Bidirectional Shortest Path Algorithm (SPA) [124] with the cost metric (C_{ij}) as $C_t t_{ij}^{hop} + C_v v_{ij}^{eff}$. The shortest path algorithm is more efficient in solving the routing problem, as it does not scrutinize all the links in the network to find the optimal path unlike with an optimization solver. Furthermore, as the links in the vehicular environment are bidirectional, we can employ the bi-directional SPA as in the previous chapter.

• Feasibility assessment

The above extracted routes are then gone through a feasibility assessment to check the stability and the latency requirement for a given packet (the packets are incrementally considered from the given set P). For that, we formulate the variables related to packet arrival times as follows.

$$(\forall r \in V) \quad H_{s,r}^{w,p} = \sum_{\forall i,j \in L_{s,r}^{u,w}} Y_{ij}^p \quad (5.6)$$

$$(\forall r \in V) \quad H_{s,r}^{wl,p} = \sum_{\forall i,j \in L_{s,r}^{u,wl}} Y_{ij}^p \quad (5.7)$$

Equation (5.6) and (5.7) calculate the number of wired and wireless hops, respectively, from the source s to an intermediate node r for the packet p using the path variable Y_{ij}^p . $L_{s,r}^{u,x}$ indicates the set of all upstream wired ($x = w$) and wireless ($x = wl$) links of node r from s .

The packet arrival time to an intermediate node r , $T_{s,r}^p$ depends on the gap between consecutive packets, (t_{gap}). If $t_{gap} \gg t_{wl}^{hop}$, $T_{s,r}^p$ can be modeled as in (5.8).

$$T_{s,r}^p = (p-1)t_{gap} + H_{s,r}^{w,p}t_w^{hop} + H_{s,r}^{wl,p}t_{wl}^{hop} \quad (5.8)$$

When the packets are closely timed ($t_{gap} \approx t_{wl}^{hop}$), continuous packet transmission will be disturbed by channel access mechanisms, especially when there are multiple wireless consecutive links in a selected path as shown in Fig. 5.2. Multiple packets can not be transmitted at the same time in these consecutive wireless links without realizing a packet collision, and further, transmission patterns (packet order and timing) depend on the timings of the RTS/CTS packets. However, assuming the worst case in terms of timing, $T_{s,r}^p$ can be formulated as in (5.9).

Figure 5.2: Multiple wireless links in a selected path.

Let, the maximum number of consecutive wireless links in a selected path be n_b .

$$T_{s,r}^p = \begin{cases} n_b(p-1)t_{wl}^{hop} + H_{s,r}^{w,p}t_w^{hop} + H_{s,r}^{wl,p}t_{wl}^{hop}; & n_b < 3 \\ 3(p-1)t_{wl}^{hop} + H_{s,r}^{w,p}t_w^{hop} + H_{s,r}^{wl,p}t_{wl}^{hop}; & n_b \geq 3 \end{cases} \quad (5.9)$$

The residual lifetime of a common link ($r-r'$) at the moment the packet p reaches the starting node of that link ($\tau_{rr'}^{r,p}$) can be expressed as,

$$(\forall rr' \in E) \quad \tau_{rr'}^{r,p} = \tau_{rr'}^s - T_{s,r}^p \quad (5.10)$$

where $\tau_{rr'}^s$ is the residual lifetime of link rr' at the moment flow rules reach the source node. The link lifetime estimation process is discussed in depth in the previous chapter, hence, will not be analyzed in depth in this chapter.

For a route to be feasible, $\tau_{rr'}^{r,p}$ of all the links in the path should be higher than the transmission time of the packet in the corresponding link ($r-r'$). The condition for a stable transmission in link rr' for the packet p is formulated as in (5.11).

$$(\forall rr' \in E) \quad \tau_{rr'}^{r,p} > t_{rr'}^{hop} \quad (5.11)$$

However, the above assessment need not be performed for all the links in the path. It can be narrowed to one necessary and sufficient condition, which is the

assessment of the bottleneck link of the path. The bottleneck link is a qualified representation of the path's stability. If the bottleneck link passes the feasibility test for the packet p , it can be safely concluded that all the links in the path are stable for the packet p . The bottleneck in a path is the link with the minimum residual lifetime at the moment the respective packet reaches the starting node of the link, as expressed in (5.12).

$$\text{Bottleneck link} \rightarrow \text{link with min:}(\tau_{rr'}^{r,p}) \quad (5.12)$$

If the bottleneck link passes the feasibility assessment, we allocate packets as much as possible to the selected path, until the feasibility criteria is satisfied.

If the bottleneck link does not satisfy the stability criteria in (5.11), the model goes for successive shortest paths. In order to find successive paths, we use a modified version of the Yen's SPA [34]. As we cannot accurately predict how many paths are required to be tested to find a feasible set of paths to transmit the given set of packet, we opt to find the next best shortest path incrementally rather than finding k-shortest paths at once. The feasibility study will be performed on each successive shortest path until either all the packets are allocated or the maximum permissible route exploration limit is reached.

We further check whether the selected paths can deliver the given set of packets within their delay tolerance margins from the condition (5.13).

$$T_{s,d}^p < \epsilon_{max}^p \quad (5.13)$$

The unicast routing protocol assumes the availability of accurate and updated network information. However, as stated previously, in practical scenarios, this is not always realized. In these scenarios, we apply different treatments based on the location of the disturbance in the network information, as discussed in the next subsection.

5.3.3 Distributed Broadcast Routing

We apply broadcast routing in two network instances as follows.

- **Case 1:** Computed path's stability is under scrutiny

Even though a considered route in the unicast approach may satisfy the stability

criteria, uncertainties involved in vehicular network can cause deviations from the estimated link lifetime, and thus, we acknowledge the possibility of inaccurate assessment of link stability. Therefore, if any link merely satisfies the stability criteria such that, $(t_{rr'}^{hop} < \tau_{rr'}^s - T_{s,r}^p < t_{rr'}^{hop} + \delta)$ where δ indicates the allowed uncertainty in link lifetime estimation, rather than unicast through the corresponding link, we incorporate broadcast with a Time To Live (TTL) counter of 2. As a TTL of 2 is allowed, the packets can reach the end node of the corresponding link via an intermediate node. Respectively, the condition for unicast transmission can be stated as, $(t_{rr'}^{hop} + \delta < \tau_{rr'}^s - T_{s,r}^p)$.

For an example, in Fig. 5.3, if the lifetime of link AB is under scrutiny, we will transmit the packet until A via unicast and broadcast the packet from A, expecting that via few broadcast hops, the packet will be received by the node B. The intuition here is to treat the link AB as a black hole and expect other nodes' support on reaching the node B. Unlike conventional VANET routing protocols such as AODV where the path from S to D is discovered entirely based on broadcasting, here we only exploit a limited broadcasting cycle.

Figure 5.3: Inclusion of broadcasting in a route.

- **Case 2:** Updated network information of the destination is not available

This is related to the disturbance implicated by the location **III** in Fig. 5.1. In this case, we find another node (agent node) that can replace the destination node in the centralized routing based on the destination node's last known location information as shown in Fig. 5.4.

This is executed by first calculating the distance between each node in a potentially interested area and the estimated location of the destination node, and then by finding the minimum out of all the distance values. Then, we find a stable short path from the source node to the new agent node which replaces the destination node. If a feasible path is not found, the procedure is repeated until an agent node is found with a stable path from the source node. The overall procedure is

Figure 5.4: Destination in a black hole.

summarized in Algorithm 3.

Algorithm 3 Selection of the agent node

Input: *source* - source node, *dest* - destination node

Output: *agent*, *path* $\triangleright$ agent node that replaces the destination and its path from the source

```

1: path  $\leftarrow$  null
2: while ( path == null )
3:   minDist  $\leftarrow$  M  $\triangleright$  M - a large value
4:   destLoc  $\leftarrow$  FindLoc(destLoc, destVec, dueTime)  $\triangleright$  estimates the new location of a given node based on its past location, velocity and time elapsed
5:   for (  $\forall$  node  $\in$  {V} - dest )
6:     nodeLoc  $\leftarrow$  FindLoc(nodeLoc, nodeVec, dueTime)
7:     dist  $\leftarrow$  FindDist(nodeLoc, destLoc)  $\triangleright$  estimates the distance between given two nodes
8:     if ( dist < minDist  $\wedge$  node  $\notin$  exploredAgents )
9:       agent  $\leftarrow$  node
10:      minDist  $\leftarrow$  dist
11:    end if
12:  end for
13:  path  $\leftarrow$  FindPath(source, agent)  $\triangleright$  finds the shortest stable path between given nodes via unicast
14:  if (path == null)
15:    exploredAgents  $\leftarrow$  {exploredAgents, agent}
16:  end if
17: end while

```

Upon finding an agent node with a stable path, packets are forwarded until the agent node via unicast (or in a combination of both unicast and broadcast, if case 1 occurs). Then, a limited broadcasting cycle is used to forward the packets in the black hole network (network area with unknown network information). The

agent node will start broadcasting the packets towards the estimated location of the destination, which essentially reflects a directional broadcast. The TTL count is adaptively set by the controller in the flow rules, and it is computed by dividing the estimated distance between the agent and the destination nodes by the transmission range of IEEE 802.11p given by the 2-ray ground reflection model.

The main idea behind the above approach is to limit the broadcasting in the network by transmitting the packets to a closer node to the destination in an uncertain area. As the broadcasting starts from a closer node, a limited broadcasting cycle will incur in the network minimizing the congestion unlike in a conventional broadcasting routing protocol.

In addition to the above mechanism, we instruct the nodes in the black hole whom received the broadcasted data packets and the agent node to temporarily hold the packets until their delay tolerance margins (ϵ_{max}^p) expire as, if the destination node comes into the transmission range of one of these nodes inside the above time period, a packet forwarding is possible. This acts as a backup mechanism since the broadcasting based approach cannot necessarily guarantee a successful transmission, especially as the area with the destination is a black hole with limited network information.

5.3.4 Store, Carry and Forward (SCF) routing

The SCF technique is solely used in places where a feasible path is not realized between the source and the destination, yet, location information about intermediate nodes are known. These scenarios refer to the location **I** and **II** in Fig. 5.1. SCF routing can yield better results in these situations as location information of nodes besides the barrier are known, and thus, an accurate estimation of SCF is possible.

The general practice in sparse network conditions is to store the packet until another node comes into its transmission range and then forward the packet to the new node expecting a route towards the destination. Few drawbacks can be found in this approach as discussed below.

The main issue is that the newly encountered node may not contain a path to the destination, and further route discovery setup can be needed. Critically, forwarding packets to a node which is far away from the destination can lead to longer routes and thus, higher delays and vulnerability on packet reception.

Vehicular nodes detect an incoming node into its transmission range by exchange of beacon packets, which occurs in periodic cycles. Therefore, node detection can be associated with a maximum delay up to the periodic time of beacon message exchange. This can also cause for a slight latency increment in SCF routing.

Alternatively, the objective here is to find the path with the least travel time including the store and forward time. In order to find it, we include the store time (t_{ij}^{scf}) of the link, if any, in the cost metric, as represented in (5.14). Then, the paths are found using a combination of bi-directional Dijkstra and successive shortest path algorithms similar to the unicast approach.

$$C_{ij} = C_t(t_{ij}^{hop} + t_{ij}^{scf}) + C_v v_{ij}^{eff} \quad (5.14)$$

In addition to the existing connections here, we explore possible future connections between nodes as well. However, the link travel time (transmission time + store time) should be within the delay tolerance margin of the packet. The amount of time a packet should be buffered until the ideal node comes into its transmission range is computed as follows with Fig. 5.5 and 5.6 taken as references.

Figure 5.5: A general instance of two vehicular nodes in the network.

The squared Euclidean distance between the nodes at a given future point in time can be defined as,

$$d_{ij}^2 = [P_{X_i} - P_{X_j} + (V_{X_i} - V_{X_j})t]^2 + [P_{Y_i} - P_{Y_j} + (V_{Y_i} - V_{Y_j})t]^2 \quad (5.15)$$

where P_{X_i}, P_{Y_i} and V_{X_i}, V_{Y_i} refer to the position and velocity components of node i , respectively, and t a general point in future time. Further, to maintain simplicity, let, $P_{X_i} - P_{X_j} = P_{X_{ij}}$, $P_{Y_i} - P_{Y_j} = P_{Y_{ij}}$ and $V_{X_i} - V_{X_j} = V_{X_{ij}}$, $V_{Y_i} - V_{Y_j} = V_{Y_{ij}}$.

Let, the time at two nodes comes into each other's transmission range be t' , and in turn d_{ij} be D_R . Then, an expression for t' can be derived as follows.

$$t'^2(V_{X_{ij}}^2 + V_{Y_{ij}}^2) + 2t'(P_{X_{ij}}V_{X_{ij}} + P_{Y_{ij}}V_{Y_{ij}}) + P_{X_{ij}}^2 + P_{Y_{ij}}^2 - D_R^2 = 0 \quad (5.16)$$

where D_R refers to the transmission range of a node via IEEE 802.11p. Finally, the packet store time can be evaluated as follows.

$$t_{ij}^{scf} = \begin{cases} 0, & \text{if } d_{ij} < D_R \\ t' + \lambda, & \text{if } d_{ij} > D_R \wedge t' \in \mathbb{R} \\ \bar{M}, & \text{otherwise} \end{cases} \quad (5.17)$$

Here, a λ time is allowed for the vehicles to cross the transmission boundary. $\bar{M}$ is a large value, and it helps to eliminate node pairs that will not be met in near future.

Unlike in conventional SCF, we do not necessarily forward the packet to the first encountered node, rather select an optimal route with the least travel time. However, it is imperatively important to assess whether all the connections exist at the times packets pass through the corresponding links.

Figure 5.6: A path with updated travel times for SCF.

For an example, in Fig. 5.6, connectivity of the link 2-3 does not exist at the route setup time, and both nodes will come into each other's transmission range at a time in future. Therefore, a store time is associated with that link, in addition to the transmission time. For a path to be feasible, the residual lifetime of all *connected* links should be higher than the total time the corresponding packets take to reach the end node of those links as modeled in (5.11). In this instance, the lifetime of link 3-4 should be higher than the summation of all the travel times in preceding links and the current link. This is a vital condition in SCF routing more than in unicast routing, as packet store times are likely to dominate the residual lifetimes of links.

In this case, the expressions for the packet arrival times at intermediate nodes are slightly different from the centralized approach due to the inclusion of packet store times.

For a connected link $rr' \in E$, if $t_{gap} \gg t_{wl}^{hop}$,

$$T_{s,r}^p = (p-1)t_{gap} + \sum_{\forall ij \in L_{s,r}^u} (t_{ij}^{hop} + t_{ij}^{scf})Y_{ij}^p \quad (5.18)$$

where $L_{s,r}^u$ indicates the set of all upstream links of node, r from the source node, s .

If $t_{gap} \approx t_{wl}^{hop}$,

$$T_{s,r}^p = \begin{cases} n_b(p-1)t_{wl}^{hop} + \sum_{\forall ij \in L_{s,r}^u} (t_{ij}^{hop} + t_{ij}^{scf})Y_{ij}^p; & n_b < 3 \\ 3(p-1)t_{wl}^{hop} + \sum_{\forall ij \in L_{s,r}^u} (t_{ij}^{hop} + t_{ij}^{scf})Y_{ij}^p; & n_b \geq 3 \end{cases} \quad (5.19)$$

The SCF routing scheme is summarized in Algorithm 4.

Algorithm 4 SCF routing

```

1: pass  $\leftarrow false$ 
2: newNetwork  $\leftarrow$  UpdateCostMetric(oldNetwork, delay tolerance)
3: while ( ! pass )
4:   path  $\leftarrow$  SuccessiveBiDirectionalSPA(newNetwork, source, dest)  $\triangleright$  finds
     the successive shortest paths between a given source and destination pair
5:   pass  $\leftarrow$  CheckFeasibility(path, packet)  $\triangleright$  assesses the feasibility of a
     given path under the stability and QoS constraints
6: end while

```

The resulting flow rules are supplied to the data plane elements following the source routing based FI operation introduced in chapter 4. Therefore, all the path information is first forwarded to the source node via the southbound interface and subsequently to the respective nodes in the path via the data plane communication along with the data packets. In order to facilitate the new improvements, we incorporate the transmission method in the packet format as depicted in Fig. 5.7.

Figure 5.7: Inclusion of next hop information in Flow Mod packet (*please note that Tx Method only require 2 bits and the characters are for the explanation purposes only).

5.4 Performance Evaluation

In this section, we comparatively analyze the performance of the proposed tribrid routing protocol using the NS-3 simulator [105]. A shortest path based SDVN routing protocol (SHORTEST), most stable path based distributed VANET routing protocol (ROMSGP [64]) and a modified store, carry and forward based distributed VANET routing protocol (SCF [73]) are chosen for the comparative study.

5.4.1 Simulation Configuration

We used a simplified version of the NTU, Singapore campus map with 15 scattered RSUs as the simulation environment as shown in Fig. 5.8 [125]. The simulation area covers around $1500 \times 1500 \text{ m}^2$. The road network traces are extracted from the OpenStreetMap [126] and simplified by removing the pedestrian lanes using the NETEDIT tool [121]. The vehicles are allowed to loop randomly in the road network and mobility traces are captured over a 500 sec. time duration via the SUMO software tool [121]. The total RSUs do not cover all the road network and thus, some areas create black holes in the network. All the results except Fig. 5.9b are taken without the top tier of the control plane as it would be an unfair comparison given that other routing protocols do not rely on a centralized server.

Hello and beacon message interval is set to 500 ms. At each simulation run, quarter of the vehicles send randomly selected set of packets to randomly selected destinations. The number of packets is selected based on an uniform distribution in between 1-8. The interval between set of packets is based on a Poisson distribution with a mean of 4 seconds. The upper bound latency requirement of packets is taken as 5 sec. Additional parameters and values used in the simulation are tabulated in Table 5.2.

Figure 5.8: Simulation network (NTU, Singapore campus map).

We first analyze the improvement brought by the proposed treatments such as consideration of stability, broadcasting based routing and SCF routing in Fig. 5.9a. The plot labeled as SHORTEST only focuses on the shortest path, whereas STABLE graph scrutinizes the link stability/connectivity in addition to the shortest distance. The TRIBRID graph, which is the proposed routing framework of this work, includes broadcasting based distributed routing and SCF routing under uncertain/disconnected network conditions, in addition to the link stability.

As mentioned above, most of the results are taken only with local RSU level controllers, and yet, the proposed HD-SDVN architecture exploits a top tier of controllers with the southbound as LTE. Fig. 5.9b depicts the results with the complete HD-SDVN architecture.

Moreover, we use the SCF success rate, which indicates the percentage of packets successfully delivered via the SCF technique (Fig. 5.10), as a metric to evaluate the accuracy of the proposed path selection scheme under the sparse network conditions and disturbances in network information. Finally, the key performance

Table 5.2: Simulation parameters of connectivity aware tribrid routing framework.

Parameter	Value
Simulators	NS-3.25, NS-3-dev
Simulation time	400 seconds (at each run)
Transmission type	UDP
Data size	1500 bytes/packet
Wireless network	V2V and V2I = DSRC
Transmission speed	DSRC = 6 Mbps, Ethernet = 100 Mbps, Broadband = 1 Gbps
Transmission range	DSRC $\sim$ 200 m
Propagation model	Two ray ground propagation loss model
Propagation delay	RSU backbone = 100 μ s

metrics such as the PDR (Fig. 5.11a), average end to end delay (Fig. 5.11b), average routing overhead, which refers to the additional packets utilized in discovering the routes per a single data packet delivery (Fig. 5.11c), and the average hop count, which symbolizes the average length of the selected routes (Fig. 5.11d), are comparatively evaluated. All the results are taken by varying the number of vehicles in the network from 20 to 100 in increments of 10.

5.4.2 Result Analysis

Fig. 5.9a and 5.9b show the effects of the proposed improvements of the routing framework in stepwise. According to Fig. 5.9a, the shortest path approach in SDVN displays a lower and degrading packet reception rate against increasing vehicle density. The reason for the degradation can be stated as the domination of the traffic density in a limited area which led to overwhelming communication traffic. However, each proposed improvement has upheld the PDR from the shortest path level which is used as a baseline in here. The consideration of link stability in route selection has increased the PDR by a considerable margin. Moreover, the inclusion of broadcasting under uncertain network conditions and optimized packet storing and path selection based on the residual link lifetime, which is a treatment for the disturbances in network information, have collectively yielded a slight improvement in packet reception as well. As discussed repeatedly, it is

Figure 5.9: PDR improvement with respect to applied treatments.

Figure 5.10: Effectiveness of the SCF scheme.

important to assess whether the residual lifetime of successive links in a selected path can carry the packet towards the destination, especially when the preceding links are associated with buffering of packets. This effect is explicitly analyzed in Fig. 5.10, in which the proposed SCF scheme dominates the tested existing SCF scheme by a significant margin.

The local RSU controllers do not cover the entire road network, and thus, in some areas, the vehicles do not have the capability to contact the control plane in Fig. 5.9a. Therefore, the packets under these circumstances failed to even discover a route. But, with the top tier control plane, these packets have better opportunities to find paths and reach the corresponding destinations as experimentally verified in Fig. 5.9b. Overall, a considerable PDR increment at around 9-10% is extracted in the simulated vehicular network with the complete HD-SDVN architecture.

Figure 5.11: Performance comparison.

Fig. 5.11a which compares the packet reception effectiveness of the proposed routing protocol with respect to existing protocols, further verifies the accuracy of the protocol. The SCF protocol which targets sparse network conditions has immensely struggled under dense network conditions (when the number of vehicles is increased), due to the congestion created by the broadcast of data packets. Similarly, the most stable path based protocol (ROMSGP) has failed to display a competitive PDR due to the broadcasting of route discovery packets, and also exclusion of packet storing in sparse network areas. However, the proposed routing protocol scrutinizes the link stability, and acknowledges the generality in vehicular networks and thus, has managed to excel over existing routing protocols by significant margins.

The primary focus of our routing protocol is on the maximization of delivery of packets within given latency based QoS. However, the protocol additionally thrives on minimization of the average end to end delay of packets by transmitting them via the shortest possible paths even under packet storing scenarios. Fig.

5.11b provides proof of it as the proposed protocol has managed to keep a competitive latency with the shortest path and most stable path protocols. When the vehicle density is low, which further empowers sparsity in the network, the proposed scheme has to depend on storing and carrying of packets, and thus, a higher latency is inevitable. However, when the vehicle density increases, the proposed model has displayed a competitive latency which is lower than the most stable path based approach. It is evident that the SCF and shortest path approach display the highest and lowest latency profiles due to their high dependency on storing of packets and usage of shortest paths without considering connectivity of links, respectively. Fig. 5.11d agrees with the above explanation as understandably the average hops in the selected routes via the shortest path and most stable path have exhibited the lowest and highest values, respectively. As expected, the proposed scheme balances the average hop count closer to the lower bound.

In terms of routing overhead, both SDVN studies have displayed lower packet utilizations in finding routes as the route setup generally includes a PACKET IN message (the packet requesting the flow rules) transmission to the controller and reception of a FLOW MOD message (the packet with the flow rules) only. However, the FLOW MOD messages in the proposed scheme include additional routing information such as transmission method, etc. and thus, a moderately higher overhead. Meanwhile, the distributed schemes rely on the broadcasting of route discovery packets and thus, higher overheads. The SCF scheme distributes copies of the data packet which are larger in size, instead of route request packets and therefore, a significantly high routing overhead.

5.5 Summary

In this chapter, we introduced a tribrid routing protocol for a general SDVN instance incorporating unicast, broadcast and store, carry and forward routing concepts. The centralized section of the routing problem is modeled following an incremental algorithm, where the successive shortest paths are validated for the stability and latency based QoS for a given set of packets. Appropriate distributed routing techniques as in broadcast and SCF, are applied to the uncertainties and disturbances in the network information on a case by case basis. The proposed protocol makes use of limited broadcasting cycles in uncertain network conditions and optimal SCF in sparse network conditions. The simulation results comparatively verified the improvement extracted by the proposed routing framework.

Chapter 6

Cooperative Data Routing and Scheduling in Software Defined Vehicular Network

6.1 Introduction

The previous two chapters have depicted how a software defined approach can be utilized to enhance routing in vehicular networks [111,123]. The respective routing frameworks in the previous chapters have scrutinized multiple attributes mainly the link stability, yet, under different network conditions, for an optimal selection of paths to route a given set of packets. The logically centralized approach enabled by SDVN can be used not only to find optimal paths, but also to address exact timings of packets to achieve an even better performance. Thus, in this chapter, we acknowledge both space and time dimensions in packet transmission [127,128].

As mentioned under the literature review in chapter 2, scheduling in VANET generally refers to setting up of the order of the packets to be delivered by a service RSU to the passing vehicles. Most studies narrow down to the single-hop use of the I2V channel mode and ignore the availability of the V2V channel that can be simultaneously scheduled with the I2V. Recently, K. Liu et al. [85] highlighted the above possibility through a cooperative data scheduling study that scrutinizes both I2V and V2V channels, still for single hop transmissions. Extending [85] and targeting a wider application scope, we propose a Cooperative Data Routing and Scheduling (CDRS) scheme with the support of SDVN for general multi-hop data transmission through a global approach scrutinizing diverse channel modes,

packet deadlines, and link dwell times. The proposed primary model is then further customized to adapt to the conventional data dissemination inside an RSU domain application scenario.

The primary problem is formulated as an ILP optimization problem with the objective of maximizing the number of served data while minimizing their weighted service delays. All possible packet conflicts, packet deadlines, and link dwell times are modeled as constraints. As each problem is executed inside a scheduling period (referred to as SP), the associated computational complexity needs to be drastically reduced. For that, we propose an incremental optimization based solution approach for the primary routing and scheduling problem and MIS based approach for the derived model inside an RSU domain.

The rest of this chapter is organized as follows. Section 6.2 presents the problem motivation, and preliminaries. In section 6.3, we formulate the cooperative data routing and scheduling problem. Section 6.4 presents specifics related to implementation and solution generation. Section 6.5 depicts the results, and finally, section 6.6 concludes the chapter.

6.2 Motivation and Preliminaries

6.2.1 Problem Motivation

In this subsection, we highlight the importance of giving consideration to the global picture in allocating channels, and in general, routing and scheduling via multiple network instances (Fig. 6.1). A conceptual motivation is taken from [45], where they cooperatively schedule heterogeneous transmitting resources available in VANET to minimize the overall communication cost.

Figure 6.1: Example vehicular network instances.

In network instance A shown in Fig. 6.1a, if the lifetime (dwell time) of link 1-3 is 1 time unit, the channel access should be granted to packet P1 to use link 1-3 at time step 1. Similarly, in network instance B shown in Fig. 6.1b, if packet P1 has a delay tolerance of 1 time unit, the ideal option is to grant the channel access to P1 at link 1-2 at time step 1. In both above example networks, simultaneous transmissions are not possible without realizing a collision. However, as mentioned above, EDCA grants channel access ignoring these concerning parameters, thus, there is a possibility to not to realize the above ideal options.

Moreover, the developed contention for channel access can be better managed by optimizing the availability of diverse communication channels. Let us elaborate it via a toy example as follows. In the network instance shown in Fig. 6.1c, the system has two source-destination pairs with two corresponding data items as S1-D1 (data item P1) and S2-D2 (data item P2). The obvious approach is to transmit both packets in direct links. However, due to the broadcast nature of wireless links, two simultaneous transmissions cannot occur in links, 2-3 and 4-5 without a packet collision. Therefore, only a single transmission can occur in these links at a given time, and it will be decided based on the timing of the RTS/CTS packets. Thus, any one of those packets could be transmitted.

However, if dwell times of both direct links are equal to 1 time unit, then we can utilize the RSU as an intermediate node for one packet and use the direct link for the other packet. Here, two packets can be transmitted at the same time as two transmissions are in different channel modes, V2V and V2I. Generally, I2V and V2V uses two different service channels and thus, no collision will occur. Detailed information about transmission modes and corresponding channels will be discussed in next sections. Packet selection for the direct transmission can be based on the type of the packet as in real-time or non-real-time.

This approach is not possible with current distributed VANET architecture, as nodes do not have global network information in granting channel access, and rather works at a microscopic level with the purpose of avoiding and resolving packet collisions. But, SDVN provides the necessary infrastructure to globally optimize the packet routing and scheduling as depicted under the system model.

6.2.2 Preliminaries

Following preliminary assumptions and facts are made to realize a low complex system model.

- Similar to the previous cases, the HD-SDVN architecture and operation introduced in chapter 3 is used as the underlying system model in this chapter.
- In the proposed model, time is considered in discrete time units, and thus, a discrete time model. One unit of time, which is also referred to as a SP, consists of a beacon message exchange period, controller update period, FLOW MOD (execution command) message reception period, and the actual data transmission period. The size of the SP will be mostly dominated by the actual data transmission time, as it comparatively consumes a larger portion as depicted in the following simple example.

Given that, BSM size = 39 Bytes, controller update message size = 10 Bytes [85], and flow mod message size = 56 Bytes, the total time to transmit overhead packets = $(39+10+56)*8 \text{ bits}/6 \text{ Mbps} = 0.14 \text{ ms}$. If the total size of a data packet is 5 KB [85], its total transmission time = $500*8 \text{ Kb}/6 \text{ Mbps} = 666 \text{ ms}$.

- DSRC works in the 5.9 GHz band and as per IEEE 1609.4 [6], it is divided into 6 service channels and 1 control channel. We assign 3 service channels to the different communication modes of DSRC which are V2V, I2V, and V2I. The control channel is used to connect with the SDN control plane. Depending on the instruction received by the controller, the nodes will tune into one of the modes at the data transmission period of each SP.
- The existing literature uses the V2I mode as the control channel mainly to send requests to the RSU. However, in this study, the V2I channel can be used with both data and control packet transmissions, given that RSUs are selected as local controllers according to the adapted HD-SDVN architecture. In this situation, the data transmission uses a SCH and control packet uses the CCH.
- Unlike in conventional data dissemination within an RSU domain, in which the RSU has all the data items, here, in the primary data routing and scheduling model (section 6.3), we assume that the data required by the destination is only stored in one source node. Thus, a single source-single destination. The primary model will be extended to the data dissemination under RSU scenario in section 6.4.
- We further assume that all source-destination pairs are vehicles. However, it does not confine the scope of the model, and rather a simplification to make this chapter concise.

- Note that, the proposed general routing and scheduling model will not affect the reliability of conventional data dissemination from an RSU to passing vehicles. The proposed model covers a much broader scope, where all data transmissions in a considered area are optimized for an improved delivery of packets with a lower service delay. For instance, if there are requests for transmissions from a vehicle to another vehicle and also from an RSU to a passing vehicle. In this case, as the RSU to vehicle transmission has lower dwell time in packets and also in links, a higher probability is there to initiate that transmission first, and the other in parallel, if there is no conflict, or else, subsequently. Therefore, the effect on specific I2V transmissions from V2V transmissions is minimum, as I2V transmissions generally have a lower dwell time and thus, a higher priority.

6.3 Proposed Routing and Scheduling Framework

In this section, we formulate the data routing and scheduling problem here as an optimization problem. Similar to previous two chapters, the motivation for the problem formulation is taken from the ATFM problem [113] where flights are routed and scheduled under in-trail separation and sector capacity constraints. Similarly, here we route and schedule data packets under packet conflict avoidance, link lifetime and packet deadline constraints.

6.3.1 Network Model

Given a network graph, $G = (V, E)$ where V and E refer to the set of nodes (nodes include both OBUs and RSUs in the network), and edges connecting the nodes, respectively, source node set, $S \subset V$ and corresponding destination node set, $D \subset V$, our main objective is to maximize the number of delivered data packets from set P , with the least average service time. The p^{th} packet is mapped to the corresponding source and destination as s^p and d^p , respectively.

We define four binary variable sets for different channels as in $V2V \rightarrow X$, $I2V \rightarrow Y$, $V2I \rightarrow Z$, $I2I \rightarrow Q$. In general, $X_{i,j,t}^p$ will be 1, if the p^{th} packet is transmitted through the link ij at time step t via the V2V channel mode. The same modeling style is applied to other channel modes as well.

The broadcast nature of wireless transmission is indicated by $\bar{X}$, $\bar{Y}$, $\bar{Z}$, and $\bar{Q}$ complementing variables. For an example, in the V2V mode, a transmission initi-

ated from node i to node j is overheard by node i 's all neighboring vehicle nodes. Therefore, $\bar{X}_{ikt}^p$ of all k vehicle neighbors of node i (except j) will be 1 in this case. This is similar for all the other channel modes, except for I2I when a point-to-point interface is used. However, if CSMA/CD is used, the same format is employed.

Further, N , N^V and N^R refer to the sets of all nodes, vehicle nodes and RSU nodes, respectively, in the network. $N_{in,i}^V$ and $N_{out,i}^V$ indicate the sets of neighboring vehicles with incoming links to node i and outgoing links from node i , respectively. Similarly, $N_{in,i}^R$ and $N_{out,i}^R$ indicate the sets of neighboring RSUs with incoming links to node i and outgoing links from node i , respectively. The delay tolerance margin of the p^{th} data packet is given by T^p , whereas the dwell time of the ij link is indicated by T^{ij} . The total time horizon considered in the model is indicated by T .

Further, the primary notations used in the model including the above depicted notations are summarized in Table 6.1.

Table 6.1: A summary of notations.

Notations	Description
V and E	set of nodes and edges, respectively, in the network
$S \subset V$ and $D \subset V$	source node and corresponding destination node sets, respectively, in the network
X_{ijt}^p	traveling state of p^{th} data packet through link ij at time unit t via V2V ($I2V \rightarrow Y$, $V2I \rightarrow Z$, $I2I \rightarrow Q$)
$\bar{X}_{ikt}^p$	overheard state of p^{th} data packet in link ij at time unit t via V2V ($I2V \rightarrow \bar{Y}$, $V2I \rightarrow \bar{Z}$, $I2I \rightarrow \bar{Q}$)
N^V and N^R	sets of vehicle nodes and RSU nodes, respectively, in the network
$N_{in,i}^V$ and $N_{out,i}^V$	sets of neighboring vehicles with incoming links to node i and outgoing links from node i , respectively
$N_{in,i}^R$ and $N_{out,i}^R$	sets of neighboring RSUs with incoming links to node i and outgoing links from node i , respectively
T^p	delay tolerance margin of the p^{th} data packet
T^{ij}	dwell time of the link ij
T	total time horizon considered in the model

6.3.2 Objective

As mentioned above, the main objective of the model is to maximize the number of delivered data packets. The total number of data packets arrived at destinations ($F(\cdot)$) can be modeled as in (6.1), assuming the destinations are vehicle nodes. However, it can be easily extended to RSUs as destinations as well by replacing the X and Y with Z and Q .

$$F(\cdot) = \sum_{\forall t \in \{t_s^p : t_e^p\}} \sum_{\forall p \in P} \left[\sum_{\forall i \in N_{in,dp}^V} X_{idpt}^p + \sum_{\forall i \in N_{in,dp}^R} Y_{idpt}^p \right] \quad (6.1)$$

Here, t_s^p and t_e^p refer to the arrival time of the data packet p at the source node, and the maximum live time of the packet p , respectively. Therefore, $t_e^p = t_s^p + T^p$.

Another sub objective is to minimize delays of the arrived packets. For that, we first estimate the arrival time of the packets, and subsequently the delay as follows.

$$t_d^p = \sum_{\forall t \in \{t_s^p : t_e^p\}} (t + 1) \left[\sum_{\forall i \in N_{in,dp}^V} X_{idpt}^p + \sum_{\forall i \in N_{in,dp}^R} Y_{idpt}^p \right] \quad (6.2)$$

where t_d^p indicates the arrival time of the packet p at the corresponding destination.

$$\text{Delay of the } p^{\text{th}} \text{ packet, } L^p = t_d^p - t_s^p \quad (6.3)$$

The final expression of the objective function can be represented as in (6.4).

$$\text{minimize : } -C_T F(\cdot) + \sum_{\forall p \in P} (C^p L^p) \quad (6.4)$$

As the major objective is to maximize the delivery of packets, its coefficient (C_T) is assigned a very large negative value. Since, we prioritize the real-time traffic over non-real-time traffic in terms of latency, as former generally require a very low latency than latter, a higher coefficient value is used for real-time traffic over non-real-time traffic. Overall, $|C_T| \gg \sum_{\forall p \in P} C^p$.

6.3.3 Constraints

6.3.3.1 Routing

The packet dispatching and arriving constraints can be formulated as in (6.5) and (6.6), respectively. The inequality in these constraints allows either dispatching of the packets or opting out (cancel), as transmission of all packets entirely depends on the network instance.

$$(\forall p \in P) (\forall s^p \in S)$$

$$\sum_{\forall t \in \{t_s^p : t_e^p\}} \sum_{\forall j \in N_{out, s^p}^V} X_{s^p j t}^p + \sum_{\forall t \in \{t_s^p : t_e^p\}} \sum_{\forall j \in N_{out, s^p}^R} Z_{s^p j t}^p \leq 1 \quad (6.5)$$

$$(\forall p \in P) (\forall d^p \in D)$$

$$\sum_{\forall t \in \{t_s^p : t_e^p\}} \sum_{\forall i \in N_{in, d^p}^V} X_{i d^p t}^p + \sum_{\forall t \in \{t_s^p : t_e^p\}} \sum_{\forall i \in N_{in, d^p}^R} Y_{i d^p t}^p \leq 1 \quad (6.6)$$

Equation (6.7) and (6.8) build up the relationship between incoming packets and outgoing packets at intermediate nodes, i.e. an incoming packet should go out of the intermediate node in a subsequent time step.

$$(\forall p \in P) (\forall k \in \{N^V - s^p - d^p\})$$

$$\begin{aligned} \sum_{\forall t' \in \{t_s^p : t_e^p\}} (t' + 1) [\sum_{\forall i \in N_{in, k}^V} X_{i k t'}^p + \sum_{\forall i \in N_{in, k}^R} Y_{i k t'}^p] \leq \\ \sum_{\forall t'' \in \{t_s^p : t_e^p\}} t'' [\sum_{\forall j \in N_{out, k}^V} X_{k j t''}^p + \sum_{\forall j \in N_{out, k}^R} Z_{k j t''}^p] \end{aligned} \quad (6.7)$$

$$(\forall p \in P) (\forall k \in N^R)$$

$$\begin{aligned} \sum_{\forall t' \in \{t_s^p : t_e^p\}} (t' + 1) [\sum_{\forall i \in N_{in, k}^R} Q_{i k t'}^p + \sum_{\forall i \in N_{in, k}^V} Z_{i k t'}^p] \leq \\ \sum_{\forall t'' \in \{t_s^p : t_e^p\}} t'' [\sum_{\forall j \in N_{out, k}^V} Y_{k j t''}^p + \sum_{\forall j \in N_{out, k}^R} Q_{k j t''}^p] \end{aligned} \quad (6.8)$$

6.3.3.2 Broadcast Nature of Wireless Transmission

Due to the common channel used in wireless communication, an on-going transmission is overheard by all other nodes in the respective transmission range. It is modeled by the set of equations in (6.9) with respect to corresponding mediums.

$$(\forall p \in P) (\forall t \in \{t_s^p : t_e^p\})$$

$$X_{ijt}^p \leq \bar{X}_{ikt}^p; \quad \forall i \in N^V, \forall k \in \{N_{out,i}^V - j\} \quad (6.9a)$$

$$Y_{ijt}^p \leq \bar{Y}_{ikt}^p; \quad \forall i \in N^R, \forall k \in \{N_{out,i}^V - j\} \quad (6.9b)$$

$$Z_{ijt}^p \leq \bar{X}_{ikt}^p; \quad \forall i \in N^V, \forall k \in \{N_{out,i}^R - j\} \quad (6.9c)$$

If the RSU-RSU interface uses CSMA/CD,

$$(\forall i \in N^R)(\forall k \in \{N_{out,i}^R - j\}) \quad Q_{ijt}^p \leq \bar{Q}_{ikt}^p; \quad (6.9d)$$

It should be noted that, if X_{ijt}^p is 1, its identical $\bar{X}_{ijt}^p$ will be 0.

6.3.3.3 Collision Avoidance

Packet collision occurs when a node receives multiple transmissions at the same time as shown in Fig. 6.2a, b and c. Equation (6.10a) is when a vehicle OBU is tuned into the V2V mode. In this case, at time t node i is receiving a data via V2V, and if another transmission starts directing to node i or if node i overhears a transmission from one of its neighbors, a packet collision will occur. Therefore, we constraint this type of simultaneous transmissions with (6.10a). However, if node i overhears multiple transmissions not directing to itself, even though a packet collision occurs at node i 's receiver antenna, it will not be an impactful event, as the transmissions are not intended for node i . These events are excluded from the constraints.

Figure 6.2: Conflicts at packet reception.

Similar constraints are deployed when the vehicle OBU is tuned into the I2V mode

(6.10b) and when an RSU's wireless interface is tuned into V2I mode (6.10c).

$$\forall t \in T, \forall i \in N^V,$$

$$(\forall j \in N_{in,i}^V) \quad \sum_{\forall p \in P} \left[\left(\sum_{\forall k \in N_{in,i}^V} X_{kit}^p \right) + \bar{X}_{jit}^p \right] \leq 1 \quad (6.10a)$$

$$(\forall j \in N_{in,i}^R) \quad \sum_{\forall p \in P} \left[\left(\sum_{\forall k \in N_{in,i}^R} Y_{kit}^p \right) + \bar{Y}_{jit}^p \right] \leq 1 \quad (6.10b)$$

$$\forall t \in T, \forall i \in N^R,$$

$$(\forall j \in N_{in,i}^V) \quad \sum_{\forall p \in P} \left[\left(\sum_{\forall k \in N_{in,i}^V} Z_{kit}^p \right) + \bar{Z}_{jit}^p \right] \leq 1 \quad (6.10c)$$

In the same way, if the RSU-RSU interface is via CSMA/CD, then collision avoidance in the I2I mode is given by (6.10d).

$$(\forall j \in N_{in,i}^R) \quad \sum_{\forall p \in P} \left[\left(\sum_{\forall k \in N_{in,i}^R} Q_{kit}^p \right) + \bar{Q}_{jit}^p \right] \leq 1 \quad (6.10d)$$

It should be noted that above collision avoidance constraints handle both hidden node and exposed node problems as well.

Similar to collision avoidance, a single interface can only successfully receive one packet at a given time and it is modeled in (6.11). Anymore receptions will lead to unsuccessful events as resembled in the network instance shown in Fig. 6.2d.

$$(\forall t \in T) (\forall i \in N^V)$$

$$\sum_{\forall p \in P} \left[\sum_{\forall k \in N_{in,i}^V} X_{kit}^p + \sum_{\forall k \in N_{in,i}^R} Y_{kit}^p \right] \leq 1 \quad (6.11)$$

6.3.3.4 Multiple Transmission Avoidance

Another conflict that should be avoided in network communication is multiple transmissions in the same link in opposite directions at a given time. This is graphically illustrated in Fig. 6.3a, and constrained in (6.12).

$(\forall i, j \in N) (\forall t \in T)$

$$\sum_{\forall p \in P} X_{i,j,t}^p + \sum_{\forall p' \in P} X_{j,i,t}^{p'} \leq 1 \quad (6.12a)$$

$$\sum_{\forall p \in P} Y_{i,j,t}^p + \sum_{\forall p' \in P} Z_{j,i,t}^{p'} \leq 1 \quad (6.12b)$$

$$\sum_{\forall p \in P} Z_{i,j,t}^p + \sum_{\forall p' \in P} Y_{j,i,t}^{p'} \leq 1 \quad (6.12c)$$

$$\sum_{\forall p \in P} Q_{i,j,t}^p + \sum_{\forall p' \in P} Q_{j,i,t}^{p'} \leq 1 \quad (6.12d)$$

Figure 6.3: Looping situations.

Also, a single interface can only transmit a single data at a time in any channel mode as indicated by (6.13) and (6.14). Only an RSU can transmit multiple data items at a given time since it consists of multiple interfaces. $\forall t \in T$,

$$(\forall i \in N^V) \quad \sum_{\forall p \in P} \left[\sum_{\forall j \in N_{out,i}^V} X_{ijt}^p + \sum_{\forall j \in N_{out,i}^R} Z_{ijt}^p \right] \leq 1 \quad (6.13)$$

$$(\forall i \in N^R) \quad \sum_{\forall p \in P} \sum_{\forall j \in N_{out,i}^V} Y_{ijt}^p \leq 1 \quad (6.14)$$

6.3.3.5 Loop Avoidance

In routing, a key requirement is to avoid packet looping situations. A data packet transverse from node i to j , should not come back from node j to i at any time in future as illustrated in Fig. 6.3b, and it is formulated in (6.15).

$$(\forall p \in P) (\forall i, j \in N)$$

$$\sum_{\forall t \in T} X_{ijt}^p + \sum_{\forall t' \in T} X_{ijt'}^p \leq 1 \quad (6.15a)$$

$$\sum_{\forall t \in T} Y_{ijt}^p + \sum_{\forall t' \in T} Z_{ijt'}^p \leq 1 \quad (6.15b)$$

$$\sum_{\forall t \in T} Z_{ijt}^p + \sum_{\forall t' \in T} Y_{ijt'}^p \leq 1 \quad (6.15c)$$

$$\sum_{\forall t \in T} Q_{ijt}^p + \sum_{\forall t' \in T} Q_{ijt'}^p \leq 1 \quad (6.15d)$$

6.3.3.6 Link Dwell Time

As repeatedly emphasized in the literature and two previous chapters,, residual lifetime of a link, which is referred to as link dwell time, is a key attribute to be considered in routing. In this model, we remove all the variables which are associated with higher time units than the links dwell time as in (6.16). Therefore, only the links with active time slots will be inserted in to the optimization problem. The link lifetime is evaluated using the procedure introduced in section 4.3.1 in chapter 4.

$$(\forall p \in P) \quad X_{ijt}^p = 0; \quad \forall t > T^{ij} \quad (6.16)$$

The above equation is applied to all the channel modes, except I2I (wired links' dwell time is considered as infinite) in the same format. However, to save space, we only formulated the equation in the V2V mode.

6.3.3.7 Packet Dwell Time

Similarly, variables associated with higher time slots than the packet dwell time are considered as unusable by setting to 0 as in (6.17).

$$(\forall i, j \in N) \quad X_{ijt}^p = 0; \quad \forall t > T^p \quad (6.17)$$

The above equation is applied to all the channel modes which satisfy the condition

$(\forall t > T^p)$.

All the constraints and objective function of the routing and scheduling problem are linearly formulated, and thus, an ILP problem.

6.4 Implementation Details and Solution Mechanisms

6.4.1 Data Dissemination under an RSU Domain

The above routing and scheduling model can be extended to an especial use case of data dissemination under an RSU domain. In fact, in VANET literature, scheduling generally refers to this application scenario. The uniqueness in this domain can be discussed with respect to the ownership of the data, and most of the studies consider that the sole ownership of the data lies with the RSU. However, as depicted in the literature, a recent study [85] has make use of the availability of data items at vehicle nodes as well.

In terms of key features, here, the time duration that vehicles are connected with the RSU is limited. Unlike in the previous case, the data packets are categorized with respect to a data type, and thus, a data packet transmitted by an RSU/vehicle can be retrieved by multiple vehicles which require that data type, hence, multiple sources-multiple destinations. The above mentioned features are reflected in the following mathematical modifications.

Let, N_p^V be the vehicle set that has the data p , and PT be the set of data types. The V2I (Z) channel mode is not used in this case, as the RSU already contains all types of packets, and thus, there is no need for a data packet to be transmitted from a vehicle to the RSU. Therefore, in all constraints, terms related to Z will be removed. Moreover, several modifications are made to above constraints as specified below.

The dispatching constraint in (6.5) will be changed as follows to allow multiple sources and multiple-transmissions. In basic sense, a node can transmit a data to multiple nodes in its range that require the corresponding data type.

$(\forall p \in P) (\forall i \in N_p^V)$

$$(\forall j \in N_{out,i}^V) \sum_{\forall t \in \{t_s^p : t_e^p\}} X_{ijt}^p \leq 1 \quad (6.18)$$

$(\forall p \in P)$ (r represents the RSU id),

$$(\forall j \in N_{out,r}^V) \sum_{\forall t \in \{t_s^p : t_e^p\}} Y_{rjt}^p \leq 1 \quad (6.19)$$

In the same way, constraint for routing at intermediate nodes (6.7) will be changed as follows to allow multiple-transmissions.

$(\forall p \in P) (\forall k \in \{N^V - s^p - s^p\}) (\forall j \in N_{out,k}^V)$

$$\sum_{\forall t' \in \{t_s^p : t_e^p\}} (t' + 1) \left[\sum_{\forall i \in N_{in,k}^V} X_{ikt}^p + \sum_{\forall i \in N_{in,i}^R} Y_{ikt}^p \right] \leq \sum_{\forall t'' \in \{t_s^p : t_e^p\}} t'' X_{kjt}^p \quad (6.20)$$

Similarly, (6.13) and (6.14) are respectively modified as follows for the same purpose.

$(\forall i \in N^V) (\forall k \in N_{out,i}^V) (\forall t \in T) (\forall p, p' \in P)$

$$X_{ikt}^p + \sum_{\forall j \in N_{out,i}^V} X_{ijt}^{p'} \leq 1; \quad PT(p) \neq PT(p') \quad (6.21)$$

$(\forall i \in N^R) (\forall k \in N_{out,i}^V) (\forall t \in T) (\forall p, p' \in P)$

$$Y_{ikt}^p + \sum_{\forall j \in N_{out,i}^V} Y_{ijt}^{p'} \leq 1; \quad PT(p) \neq PT(p') \quad (6.22)$$

All other constraints including collision avoidance, broadcast nature will prevail as same with the exception of removal of Z components.

6.4.2 Solution Mechanisms

Both above models are formulated with respect to a global prediction horizon. However, at the implementation stage, the prediction horizon is small and will be shifted by a SP at the end of each SP. Thus, at an intermediate step, the network consist of packets already scheduled, packets already delivered to respective destinations, packets in transition and packets to be scheduled in this SP.

More importantly, as both proposed models are repeatedly executed at each SP, the computational complexity associated with the problems should be reduced to a realizable level. At the same time, the quality of the solutions should be kept at an acceptable level. Thus, the following solution approaches are proposed to

achieve the above discussed trade-off.

6.4.2.1 Solution Approach for Routing and Scheduling Scenario

For the general routing and scheduling problem depicted in section 6.3, if the unmodified problem is solved using any optimization solver such as Gurobi [115], it will take a considerable time to reach the global optimality (0.01% default optimality gap) as all the aspects and options are considered in the global solution approach. Alternatively, we propose to introduce packets incrementally [117] to the problem and thus, reaching the respective local optimality will be much faster. However, we acknowledge that this approach would not probably reach to the same global optimal point. In order to close the gap with the global optimality, we propose to order the packets based on the packet dwell time, packet arrival time, and packet type which collectively reflects the priority order in the objective function and then, solve the problem incrementally on top of the previously fixed network, as summarized in Algorithm 5.

In brief, the solution approach first ranks the packets (line 1) as mentioned above and then, each packet is introduced into the problem from the ranked packet set incrementally. For each packet, the system finds K shortest paths, where K is set to a lower value (2 or 3) to reduce the solution time. Then, each path is tested for the feasibility as depicted below in the description of **isFeasible** function.

The function **isFeasible**(path, packet) is an accumulation of several functions and code segments, compressed in the algorithm to save space. It basically makes use of two main functions of **isLinkExist**(link, time) and **isConflict**(network, link, time). Aforementioned functions respectively check whether a given link exist at a given time and whether assigning a packet to the given link at the given time creates any conflict with the current network. Overall, **isFeasible** function validates whether a given path can be used to deliver a given packet within the packet's dwell time without any possible transmission conflicts which are reflected by the constraints in the proposed model.

If the packet finds a feasible path, the system then finds the lowest delay the packet can yield via that path. This is executed for each path found from the K shortest path algorithm. Finally, the path with the lowest delay is selected as the ideal path for the respective packet and the packet assigned path, refer to as a tuple, is inserted into the network graph (line 12). This is repeatedly executed for each packet in P^{ranked} . The computational complexity of the above approach can

Algorithm 5 Incremental optimization based solution approach at an intermediate SP

Inputs: $P, P^{arr}, P^{type}, P^{dwell}, S, D, G^C$ $\triangleright$ P - packets arrived at the current SP and forwarded from a previous SP, P^{arr} - packet arrival times, P^{type} - packet types, P^{dwell} - packet dwell times, S - source node set, D - destination node set, G^C - current state of the network

Output: $G^F, ToNext$ $\triangleright$ G^F - state of the network at the end of this SP, $ToNext$ - packets forwarded to the next SP

```

1:  $P^{ranked} \leftarrow \text{rankPackets}(P, P^{arr}, P^{dwell}, P^{type})$   $\triangleright$  priority order :  $P^{arr} > P^{dwell} > P^{type}$ 
2:  $G^F, ToNext \leftarrow [ ]$ 
3: for (  $\forall p \in P^{ranked}$  )
4:    $paths \leftarrow \text{K\_ShortestPaths}(s^p, d^p)$   $\triangleright$  find the K-shortest paths for the packet p
5:    $delay \leftarrow M$   $\triangleright$  M - a large value
6:    $assigned \leftarrow \text{false}$   $\triangleright$  packet not assigned to a path yet
7:   for (  $\forall \text{path} \in paths$  )
8:      $G' \leftarrow G^F$   $\triangleright$  temporary save networks status
9:     if (  $\text{isFeasible}(\text{path}, p)$  )
10:      if (  $delay > \text{getDelay}(\text{path}, p)$  )  $\triangleright$  if the feasible path has a lower delay for the packet p
11:         $delay \leftarrow \text{getDelay}(\text{path}, p)$ 
12:         $G' \leftarrow [G'(\text{path}, p)]$ 
13:         $assigned \leftarrow \text{true}$ 
14:      end if
15:    end if
16:     $G^F \leftarrow G'$   $\triangleright$  save back the updated network
17:  end for
18:  if (  $! assigned$  and  $\text{isForwardable}(p)$  )  $\triangleright$  if an unassigned packet has enough dwell time to be scheduled in the next SP, pass to the next SP
19:     $ToNext \leftarrow [ToNext \ p]$ 
20:  end if
21: end for

```

be derived as follows.

Computational complexity of Algorithm 5 (CC_{INC}) =

$$O(N_p * K * L * 2N_e) + O(N_p[N_e + N_n * \log N_n + K])$$

where L is the average path length in K shortest paths, N_p is number of packets, N_e is the number of edges, and N_n is the number of nodes in the network. The

above expression can be further simplified to,

$$\begin{aligned} CC_{INC} &= O(N_p * K * L * N_e) + O(N_p[N_e + N_n * \log N_n + K]) \\ CC_{INC} &= O(N_p[K * L * N_e + N_n * \log N_n]) \end{aligned} \quad (6.23a)$$

6.4.2.2 Solution Approach for Data Dissemination within an RSU Domain Scenario

For this application scenario, we propose to use a graph theory approach based on the generation of Maximum Weighted Independent Set¹ (MWIS) [113, 129] [85]. First, the vehicular network including its transmission behavior is converted into an adjacency graph. For that, all possible packet delivering patterns at all-time steps (inside the prediction horizon) are explored and taken as nodes in the graph. It should be noted that, a packet delivering pattern which is selected as a node, should also be feasible in terms of packet dwell time and link dwell time. Moreover, nodes are weighted based on the corresponding packet's arrival time (t_s^p), dwell time (T^p), type (p^{ty}) and time unit associated with the node (n^t) as shown below.

$$\text{Weight of node, } n = C_a t_s^p + C_d T^p + C_c p^{ty} + C_t n^t \quad (6.24)$$

where the coefficient values are set in the order of $C_a > C_d > C_c > C_t$.

Any possible conflict between two packet delivering patterns is taken as an edge. This also includes any two nodes that belong to the same packet. Similarly, any duplicate receptions are also connected as edges to avoid been included in the final result. An example network instance and resulting adjacency graph are shown in Fig. 6.4 and 6.5, respectively.

In Fig. 6.5, any two nodes that do not have an edge between them are independent and thus, those two simultaneous transmissions are possible in the network without any collision. Now, the optimization problem is transformed into a MIS finding problem and for that we use the approach suggested in [130].

The computational complexity associated with the depicted solution approach can

¹MWIS approach can in fact yield better results than the incremental approach as the global network status is considered. However, the computational complexity of this approach depends on the number of nodes and edges in the graph. In the former routing and scheduling problem, as a packet moves through multiple links at different time steps, the resulting graph becomes very large, thus, difficult to realize an applicable solution time using the MWIS method. However, in this application scope, as the communication range is limited to a transmission range of an RSU, the resulting adjacency graph is likely to be smaller, thus, more applicable.

Figure 6.4: A vehicular network instance (Shaded rectangular boxes and empty circles next to nodes indicate the data items that the corresponding nodes acquired and required, respectively).

Figure 6.5: Generated adjacency graph based on the network instance in Fig. 6.4.

be decomposed to the computational complexity of exploring all possible packet deliverable patterns together with all possible conflicts and the computational complexity of finding MWIS [131]. Thus,

$$\begin{aligned} \text{Computational complexity of MWIS approach } (CC_{MWIS}) = \\ O(N_p^s[N_e + N_n * \log N_n + K] + [N_p^s]^2 * T^2) + O([N_p^s]^2 * T^2) \end{aligned}$$

where N_p^s is the number of packets at source nodes.

As prediction horizon, T and number of paths, K are constants,

$$CC_{MWIS} = O(N_p^s[N_e + N_n * \log N_n + K] + [N_p^s]^2 * T^2) \quad (6.25a)$$

6.5 Performance Evaluation

In this section, we comparatively evaluate the performance of the proposed general routing and scheduling model (**scenario 1**) and its derivative for the data dissemination scenario under RSU domains (**scenario 2**) with respect to two simulation studies.

We first evaluate the performance of both scenarios via multiple network instances using Matlab, Gurobi (optimization solver) and C++ as simulation platforms/software (simulation study I). Later, in order to realize the practicality in the real world, we generate a grid type road network in NS3 [105] with realistic mobility traces extracted from SUMO [121], and test the general data transmission cases between multiple source-destination pairs (simulation study II).

As comparative studies for scenario 1, we use two SDVN routing protocols, Shortest Path Approach (SPA) [63], and Link Stability based Protocol (LSP) [109]. For scenario 2, we use FCFS, MRF and a recent Cooperative Data Dissemination (CDD) protocol which uses both I2V and V2V [85], as comparative schemes. The proposed cooperative data routing and scheduling scheme in both application scenarios is referred to as CDRS.

Following metrics are used to numerically evaluate the performance of the systems.

1. Number of packets dropped due to link breakages (n_{lb}) (Fig. 6.7a and 6.8a).
2. Deadline Missed Ratio (DMR) (Fig. 6.7b and 6.8b): The percentage of packets missed their packet deadlines out of all the served packets.
3. Effective Service Ratio (ESR) (Fig. 6.7c and 6.8c) or Packet Delivery Ratio (PDR) (Fig. 6.11): The ratio between the number of packets served within their packet deadlines and the total number of requests.
4. Service Delay (SD) (Fig. 6.7d and 6.8d): The average delay of the served data packets.
5. Service Delay of Real-time traffic (SDR) and non-real-time traffic (SDNR) (Fig. 6.9): The average service delays of real-time data and non-real-time data, respectively.
6. Solution time (Fig. 6.10): The average time it takes to solve a single SP.
7. Solution quality (Fig. 6.10): Closeness of the derived solution via the proposed solution approach, to the global optimality.

Each point on the graphs is an average over 20-25 simulation runs with different network instances.

6.5.1 Simulation Configuration

6.5.1.1 Simulation Study I

In this case, we randomly generate multiple network instances of a vehicular network as simulation environments using MATLAB. A workstation with a Intel-core i7 3.40 GHz processor, and a 8 GB RAM was used in performing the experiments. The results are taken by incrementing the number of nodes in the network from 5 to 25 in increments of 5. The number of RSUs out of total number of nodes are also randomly selected for scenario 1 and set to 1 in scenario 2. The number of packets is randomly generated based on a normal distribution with mean and standard deviation (σ) set to number of nodes - 1 and 4, respectively. In both scenarios, each data will either belong to real-time traffic class or non-real-time traffic class. Moreover, each data packet will be assigned to a randomly selected packet type in scenario 2.

The total time horizon (T) is set to the number of packets in each scenario. The packet arrival time is randomly selected based on a uniform distribution between 1 and $\lfloor (T/2) \rfloor$. Higher packet deadlines are assigned to the non-real-time packets based on an uniform random number generator between the packet arrived time (t_s^p) and $(t_s^p + \lfloor (T/3) \rfloor)$. For real-time packets, the upper limit of the random number generator is set to $\lfloor (T/5) \rfloor$. Dwell times of existing links are randomly set based on a normal distribution with mean and σ equal to the number of nodes - 1 and 4, respectively.

In each scenario, all the source and destination pairs are randomly selected. However, in scenario 2, the RSU is also selected as a potential source in addition to the randomly selected vehicle nodes. SP is taken as 1 second. All the performance metrics are quantified this simulation study and shown in Fig. 6.7, 6.8, 6.9, and 6.10. The purpose of the above randomization of key attributes is to evaluate the performance under diverse vehicular network instances.

6.5.1.2 Simulation Study II

The above simulation study projects performances in multiple controlled vehicular environments. Even though, it is relatively easier to analyze such controlled environments, it lacks a realization on the practicality of the proposed framework.

Figure 6.6: Simulated grid type vehicular network.

Thus, to bridge this gap, we first generate realistic mobility patterns on a grid type road network (Fig. 6.6a and 6.6b) using SUMO [121]. Then, these mobility traces are fed into NS3 and on top of the resulting vehicular network, communication channels, protocols, and interfaces are developed.

Here, we have implemented the HD-SDVN controller architecture, which was introduced in chapter 3, with only the bottom control layer. The RSU located at the central intersection in Fig. 6.6a is selected as the controller RSU. Packet arrival follows a Poisson distribution with a mean of 2 sec. and the packet size is set to 1.5 KB. Thus, the size of a SP is set to 0.25 sec. The packet deadlines are randomly assigned based on an uniform distribution between 2 and 20 SPs. The number of vehicles is incremented from 10 to 30 in increments of 5. Other

Table 6.2: Simulation parameters of routing and scheduling study.

Parameter	Value
Simulation time	200 seconds per run with 25 runs per each point
Packet sizes	Packet In = 32 bytes, Flow Mod = 56+ bytes
Transmission	Type = UDP, Range = 200 m (DSRC)
Propagation model	Friis Propagation Loss Model
Wireless network	DSRC (IEEE 802.11p) = 6 Mbps
Wired network (RSU-RSU)	Ethernet (IEEE 802.3) = 100 Mbps, Propagation delay = 1 ms

simulation configurations are tabulated in Table 6.2. The performance is captured in terms of PDR and shown in Fig. 6.11.

6.5.2 Results Analysis

According to the performance comparison of different attributes in both application scenarios (Fig. 6.7 and 6.8), it is evident that the consideration given to link connectivity and packet dwell time via a global perspective in the proposed scheme has had a significant positive effect, especially in terms of packet reception rate. In both Fig. 6.7a and 6.8a, all the protocols that ignore the link stability/connectivity, have suffered considerable packet losses, whereas, the proposed protocol along with LSP and CDD have no any packet losses due to link breakages. Similarly, in Fig. 6.7b and 6.8b, all the schemes with the exceptions of CDRS and CDD, have packets reaching the corresponding destinations after their respective deadlines, noticeably lowering their ESRs in Fig. 6.7c and 6.8c. Even though,

Figure 6.7: Performance comparison of different attributes in routing schemes (scenario 1).

Figure 6.8: Performance comparison of different attributes in data dissemination schemes (scenario 2).

CDD, which can be considered as the state-of-the-art in cooperative data dissemination, scrutinizes both link and packet dwell times, it has displayed an inferior packet reception rate to the CDRS, as it only entertains single hop transmissions. Meanwhile, the proposed CDRS protocol outperforms all the compared protocols in both scenarios in terms of ESR as depicted in Fig. 6.7c and 6.8c.

In terms of service delay (Fig. 6.7d and 6.8d), all the protocols display competitive performances, with CDRS and CDD slightly outperforming others. More importantly, CDRS has minimized the latency of real-time traffic over non-real-time traffic in both scenarios as shown in Fig. 6.9a and 6.9b, since it classifies and prioritizes packets based on their traffic types. In contrary, other schemes have not distinguished the traffic type, and thus, no clear differentiation can be seen in real-time and non-real-time packet latencies in Fig. 6.9a and 6.9b. Moreover, it can be observed that in other protocols, the latency of real-time packets (shown in red), which require a lower latency, generally suffers more than non-real-time

Figure 6.9: Comparison of service delay based on traffic type.

Figure 6.10: Comparison of solution time and accuracy of results (*Proposed* refers to the models with solution approaches and *Global* refers to the global optimal solution achieved via Gurobi).

packets, which can in fact tolerate a reasonable latency. (For examples, refer to LSP and SPA when number of nodes is 25, and CDD and MRF when number of nodes is 20). Note: Please note that the general patterns in Fig. 6.7 and 6.8 are slightly disturbed due to randomness. Observe that, these results are taken for multiple random instances of vehicular networks using MATLAB. However, all the protocols are tested for the same network instances, and thus, there is no issue to the comparative analysis.

Fig. 6.10 depicts the complexity and the accuracy of the solution approaches in both scenarios. According to both sub figures, the proposed incremental and MWIS based solution approaches drastically reduce solution times, while main-

Figure 6.11: PDR with realistic mobility profiles.

taining a very close gap with the global optimality, in both general routing and data dissemination application scenarios. Even though the solution time increases when the number of nodes increases, yet, it is still in millisecond domain, and proves their applicability in real-time applications.

In simulation study II, we have tested the performance of the primary routing and scheduling model with realistic vehicular mobility. According to its results (Fig. 6.11), the proposed protocol still outruns both LSP and SPA, thus, aligns and verifies the performance results yielded from controlled environments.

6.6 Summary

In this chapter, we introduced a novel cooperative data routing and scheduling framework for general packet transmission in multi-hop scenarios with the support of SDVN. The model cooperatively schedules diverse channel modes in vehicular communication such as V2V, V2I, I2V and I2I with a global perspective, for a better dissemination of data, while scrutinizing the packet deadlines, and link dwell times, and moreover, acknowledging the different packet types with their unique latency requirements. The primary problem is formulated as an ILP with the objectives of maximizing the extent of served packets while minimizing their respective latencies, and all possible packet conflicts as constraints. Furthermore, the aforementioned primary model is easily extended to suit for the data dissemination application scenario under RSU domains with minimum modifications. Moreover, two solution approaches, incremental and MWIS, are proposed to achieve a realizable computational complexity while maintaining a close gap with

the global optimality. Extensive simulation studies based on controlled vehicular environments as well as with realistic mobility patterns, verified the superiority in the proposed models in both application scenarios.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we have studied the effect, a software defined approach has on information dissemination in vehicular networks.

First, in order to have a functional SDVN system model that comply with the unique characteristics of vehicular networks, especially stringent latency requirement, we introduced a novel hierarchical distributed SDVN architecture with a detailed controller placement model, optimizing the placement of the controllers with the consideration on several factors such as the number of controllers, latency, significance of the RSU's geographical location, workload distribution and the vehicular statistics around the RSU location. The proposed architecture splits the control plane into two levels and the top tier is placed on the Internet while the bottom tier is taken down to the RSU level, cutting down the latency experience by data plane elements.

The proposed HD-SDVN is then used as the underlying system model for multiple studies that depict the enhanced effect SDN has on information dissemination in VANET.

Among them, we have initially focused on packet routing under ideal conditions in SDVN. Here, we have assumed that the SDN control plane collectively generates the necessary global network view to optimize the packet transmission in the data plane. Based on that, we have proposed a novel routing framework for SDVN scrutinizing the stability of routes in addition to the shortest travel time. The routing problem is motivated by the minimum cost capacitated flow prob-

lem, and modeled as an ILP problem with the support of mixed logic constraints. The resulting framework routes the packets via multiple routes along the shortest possible paths with enough stability. Additionally, a source routing based FI operation was introduced to efficiently deliver the routing information to the respective nodes. The proposed FI scheme encapsulates all the routing information in a modified FLOW MOD message and sends only to the route requested node, minimizing the extent of communication between the control and data planes.

Later, we have dropped the assumption of the deterministically known network, and acknowledged the uncertainties exist in collection of network information, and diverse network conditions in routing. For that, we introduced a tribrid routing protocol for a general SDVN instance incorporating unicast, broadcast, and SCF routing concepts. The routing problem is modeled following an incremental algorithm where the successive shortest paths are validated for the stability and latency based QoS for a given set of packets. Appropriate distributed routing techniques as in broadcast and SCF are applied to handle the uncertainties and disturbances in the network information on a case by case basis. The proposed protocol makes use of limited broadcasting cycles in uncertain network conditions and optimal SCF in sparse network conditions.

The thesis has further extended the investigation on the SDN based information dissemination in vehicular networks by incorporating the time dimension in addition to the optimal selection of paths. To this end, we have introduced a novel cooperative data routing and scheduling framework for general packet transmission in multi-hop scenarios. The model routes and schedules packets in the multi-hop scenario acknowledging diverse channel modes as in V2V, V2I, I2V and I2I, and different packet types with their unique latency requirements under delay tolerance, link connectivity and packet conflict regions in an ILP problem. Furthermore, the aforementioned primary model is easily extended for the data dissemination application scenario under RSU domains with minimum modifications. Moreover, two solution approaches, incremental and MWIS, are proposed to lower the computational complexity while maintaining a close gap with the global optimality.

Extensive simulation studies incorporating realistic mobility patterns have supported the superiority of each proposal in terms of packet reception ratios while displaying competitive yet closer to the lower bounds, end to end delays and packet overheads.

In summary, the thesis has highlighted the improvement that can be brought upon

vehicular networks through a novel HD-SDVN system model and multiple routing and scheduling frameworks scrutinizing diverse attributes and network conditions for an enhanced performance in information dissemination.

7.2 Future Work

As repeatedly emphasized in the thesis, SDVN positively disrupts the conventional distributed attitude on vehicular networks and opens up a new perspective through the enabling of the logically centralized network view. This empowers the network operators to easily and dynamically manage diverse aspects and attributes of vehicular networks through intelligent applications as distinctively highlighted by several studies in this thesis.

Many foreseeable improvements, directions and applications are possible with the novel SDVN paradigm. Here, we discuss a few of the directives with respect to the content of this thesis.

- Dynamic controller placement

The proposed hierarchical distributed controller architecture and the controller placement method, do not consider the load or the amount of requests that each individual controller handle. The controller placement model is static and the placement of the controllers is fixed. It does not further acknowledge the dynamic environmental parameters such as load, vehicular mobility that fluctuate with the time. But, at some occasions such as National day celebration, the flow of vehicles and average vehicle speed at several locations might change significantly. As the load factor will be more critical in these scenarios, it is better to have a higher number of local controllers temporarily near those areas to handle the massive flow of incoming requests. Thus, a probable future directive is to propose a dynamic controller placement model considering aforementioned parameters. As most of the existing controller placement models even in wired networks are static, it will be a completely new approach to the control plane.

- Minimization of beaconing overhead using trajectory prediction

In SDVN systems, the SDN controllers need to gather network information periodically to make accurate decisions. The accuracy in decisions increases with the frequency of the status update, as it enables the controllers to regularly receive accurate and updated information. However, this takes a toll on the network, even on the proposed HD-SDVN, in which the RSUs collect network information and

pass into the control plane, due to the extra layer of periodical transmissions. In order to balance this trade-off between the accuracy of decisions and periodic beaconing, one possible directive is to use trajectory prediction techniques. For this, we can train and test models using the repository of network information at the control plane via machine learning, and predict future instances of the network.

- Multiple factors in route selection

A major improvement in terms of routing is to incorporate multiple attributes in the path selection, as we have depicted in chapter 4, and 5. One obvious attribute that can be combined with delay and link stability is the trust. In fact, SDVN provides the much needed centralized authority for trust based mechanisms as a logically centralized control plane. The control plane can evaluate the trustworthiness of vehicles based on their past records. Thus, a trust metric can be utilized in the objective function of the proposed ILP model in chapter 4 for a reliable selection of nodes in the routing paths. In addition to trust, many other attributes such as security and load balancing, can also be incorporated for an even better performance.

- Routing with predictive FI

Another possible research that can be carried out is to use predictive FI instead of proactive or reactive. This can be aided by machine learning. Here, the control plane anticipates future events based on the recorded events of the nodes and provides flow rules before the events occurs. Machine learning aided trajectory prediction will again play a crucial role in the predictive FI. The most interesting research aspect here is that, we can reduce the route setup time almost to zero because the nodes already have received the required flow entries when the events occur and thus, no need to employ any route finding mechanism or consult the controller.

- Multi-item, multi-RSU and continuous time scheduling

The proposed routing and scheduling framework in chapter 6 is a discrete time model and thus, there is a certain error associate with the size of a SP. This can be minimized by converting the problem formulation into continuous time. However, the complexity associated with such continuous time model is very high, and hence, a trade-off has to be made to gain a practical realization, especially for real-time communication.

Moreover, the proposed routing and scheduling problem can be customized to

be applicable in diverse application domains such as multi-item and multi-RSU content dissemination.

- Uncertainty in scheduling

In the proposed routing and scheduling scheme, it is assumed that the taken decisions are executed at the corresponding time steps without any deviations. However, due to the dynamicity in vehicular networks and inherent channel access mechanisms, it might not always realize, generating a sense of uncertainty. Thus, a potential research directive is to cater for the aforementioned uncertainty in decision making via a stochastic routing and scheduling framework.

- Traffic engineering in SDVN

The centralized content management of SDN provides room for deployment of traffic engineering concepts. A lot of research can be found in the wired SDN related to traffic engineering. Similarly, we can develop adaptive traffic engineering applications for vehicular networks adhering to diverse vehicular environments and requirements. However, the distributed controller architecture of the proposed SDVN system makes it challenging to gather data to a centralized data store to implement traffic engineering algorithms. Moreover, collected network data quickly lose their value (information age fast) due to the dynamics of vehicular networks. The age of information is in fact a trending topic and would be very useful as a metric in the SDVN domain.

Author's Publications

Journal Papers

1. K. L. K. Sudheera, M. Ma, and P. H. J. Chong, "Controller placement optimization in hierarchical distributed software defined vehicular networks," *Computer Networks*, vol. 135, pp. 226 - 239, 2018.
2. K. L. K. Sudheera, M. Ma, and P. H. J. Chong, "Link Stability Based Optimized Routing Framework for Software Defined Vehicular Networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2934-2945, March 2019.
3. K. L. K. Sudheera, M. Ma, and P. H. J. Chong, "Connectivity aware tribrid routing framework for a generalized software defined vehicular network," *Computer Networks*, vol. 152, pp. 167 - 177, 2019.
4. K. L. K. Sudheera, M. Ma, and P. H. J. Chong, "Real Time Cooperative Data Routing and Scheduling in Software Defined Vehicular Networks," 2020 (Under Review).

Conference Papers

1. K. L. K. Sudheera, M. Ma, G. G. M. N. Ali, and P. H. J. Chong, "Delay efficient software defined networking based architecture for vehicular networks," in *2016 IEEE International Conference on Communication Systems (ICCS)*, Dec 2016, pp. 1-6.
2. K. L. K. Sudheera, M. Ma, and P. H. J. Chong, "Efficient Flow Instantiation via Source Routing in Software Defined Vehicular Networks," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, Sep. 2017, pp. 1-5.
3. K. L. K. Sudheera, M. Ma, and P. H. J. Chong, "Link Dynamics Based

- Packet Routing Framework for Software Defined Vehicular Networks,” in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1-6.
4. K. L. K. Sudheera, M. Ma, and P. H. J. Chong, “Link Stability Based Hybrid Routing Protocol for Software Defined Vehicular Networks,” in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1-6.
 5. K. L. Kushan Sudheera, M. Ma, and P. H. J. Chong, “Cooperative Data Routing & Scheduling In Software Defined Vehicular Networks,” in *2018 IEEE Vehicular Networking Conference (VNC)*, Dec 2018, pp. 1-8.

Bibliography

- [1] “IEEE Draft Standard for Information Technology – Telecommunications and information exchange between systems – Local and metropolitan area networks –Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 7: Wireless Access in Vehicular Environments,” *IEEE Unapproved Draft Std P802.11p/D5.0*, Nov 2008, p. 1, 2009.
- [2] C. Gomes, “Auto Industry Drives Canada’s Industrial Revival– Auto Parts Employment Outpaces Growth In The United States,” Scotiabank, Tech. Rep., 2016.
- [3] J.-P. Rodrigue, C. Comtois, and B. Slack, *The geography of transport systems*. Routledge, 2013.
- [4] F. Dressler, F. Kargl, J. Ott, O. K. Tonguz, and L. Wischhof, “Research challenges in intervehicular communication: lessons of the 2010 Dagstuhl Seminar,” *IEEE Communications Magazine*, vol. 49, no. 5, pp. 158–164, 2011.
- [5] J. B. Kenney, “Dedicated Short-Range Communications (DSRC) Standards in the United States,” *Proceedings of the IEEE*, vol. 99, pp. 1162–1182, 2011.
- [6] F. Bai, D. D. Stancil, and H. Krishnan, “Toward understanding characteristics of dedicated short range communications (DSRC) from a perspective of vehicular network engineers,” in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*. ACM, 2010, pp. 329–340.
- [7] I. Ku, L. You, M. Gerla, F. Ongaro, R. L. Gomes, and E. Cerqueira, “Towards software-defined VANET: Architecture and services,” in *13th Annual*

- Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, 2014, pp. 103–110.
- [8] X. Ge, Z. Li, and S. Li, “5G Software Defined Vehicular Networks,” *IEEE Communications Magazine*, vol. 55, no. 7, pp. 87–93, July 2017.
- [9] G. G. M. Nawaz Ali, P. H. J. Chong, S. K. Samantha, and E. Chan, “Efficient data dissemination in cooperative multi-RSU Vehicular Ad Hoc Networks (VANETs),” *Journal of Systems and Software*, vol. 117, pp. 508–527, 2016.
- [10] V. Kumar, S. Mishra, and N. Chand, “Applications of VANETs: Present & Future,” *Communications and Network*, vol. 5, pp. 12–15, 2013.
- [11] H. Hartenstein and L. P. Laberteaux, “A tutorial survey on vehicular ad hoc networks,” *IEEE Communications Magazine*, vol. 46, no. 6, pp. 164–171, June 2008.
- [12] Z. Hameed Mir and F. Filali, “LTE and IEEE 802.11p for vehicular networking: a performance evaluation,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, pp. 1–15, 2014.
- [13] M. Jerbi, S.-M. Senouci, Y. Ghamri-Doudane, and M. O. Cherif, “Vehicular Communications Networks: Current Trends and Challenges,” in *Next Generation Mobile Networks and Ubiquitous Computing*, S. Pierre, Ed. IGI - Global, Aug. 2010, pp. 251–262. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-00849110>
- [14] H. Zongjian, C. Jiannong, and L. Xuefeng, “SDVN: Enabling Rapid Network Innovation for Heterogeneous Vehicular Communication,” *IEEE Network Magazine Special Issue on Software Defined Wireless Networks*, 2015.
- [15] ONF, “Software-Defined Networking: The New Norm for Networks, ONF White Paper,” Open Networking Foundation, Tech. Rep., April 2012. [Online]. Available: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf>
- [16] C. Chaudet and Y. Haddad, “Wireless Software Defined Networks: Challenges and opportunities,” in *Microwaves, Communications, Antennas and Electronics Systems (COMCAS), 2013 IEEE International Conference on*, 2013, pp. 1–5.

- [17] M. Abolhasan, J. Lipman, W. Ni, and B. Hagelstein, "Software-defined wireless networking: centralized, distributed, or hybrid?" *IEEE Network*, vol. 29, no. 4, pp. 32–38, July 2015.
- [18] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A Survey on Software-Defined Networking," *IEEE Communications Surveys & Tutorials*, vol. 17, pp. 27–51, 2015.
- [19] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao, "Are we ready for SDN? Implementation challenges for software-defined networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 36–43, July 2013.
- [20] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, and S. Shenker, "Onix: A Distributed Control Platform for Large-scale Production Networks," in *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation*, ser. OSDI'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 351–364. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1924943.1924968>
- [21] K. Phemius, M. Bouet, and J. Leguay, "DISCO: Distributed multi-domain SDN controllers," in *2014 IEEE Network Operations and Management Symposium (NOMS)*, 2014, pp. 1–4.
- [22] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008. [Online]. Available: <http://doi.acm.org/10.1145/1355734.1355746>
- [23] D. Kreutz, F. M. V. Ramos, P. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.
- [24] N. B. Truong, L. Gyu Myoung, and Y. Ghamri-Doudane, "Software defined networking-based vehicular Adhoc Network with Fog Computing," in *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, 2015, pp. 1202–1207.

- [25] J. J. Blum, A. Eskandarian, and L. J. Hoffman, "Challenges of intervehicle ad hoc networks," *Trans. Intell. Transport. Sys.*, vol. 5, no. 4, pp. 347–351, 2004.
- [26] W. Su, S.-J. Lee, and M. Gerla, "Mobility prediction and routing in ad hoc wireless networks," *Int. J. Netw. Manag.*, vol. 11, no. 1, pp. 3–30, 2001.
- [27] C. Perkins, E. Belding-Royer, and S. Das, *Ad hoc On-Demand Distance Vector (AODV) Routing*. RFC Editor, 2003.
- [28] D. A. Maltz and D. B. Johnson, "The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4," 2007.
- [29] H. Li, M. Dong, and K. Ota, "Control Plane Optimization in Software-Defined Vehicular Ad Hoc Networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 7895–7904, Oct 2016.
- [30] M. Yu, J. Rexford, M. J. Freedman, and J. Wang, "Scalable flow-based networking with DIFANE," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, Aug. 2010.
- [31] A. Hari, T. V. Lakshman, and G. Wilfong, "Path Switching: Reduced-state Flow Handling in SDN Using Path Information," in *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '15. ACM, 2015, pp. 36:1–36:7.
- [32] C. Han, M. Dianati, R. Tafazolli, R. Kernchen, and X. Shen, "Analytical Study of the IEEE 802.11p MAC Sublayer in Vehicular Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 873–886, June 2012.
- [33] D. Eppstein, "Finding the k Shortest Paths," *SIAM J. Comput.*, vol. 28, no. 2, pp. 652–673, 1999.
- [34] J. Y. Yen, "Finding the k shortest loopless paths in a network," *management Science*, vol. 17, no. 11, pp. 712–716, 1971.
- [35] Y.-C. Chang, J.-L. Chen, Y.-W. Ma, and P.-S. Chiu, "Vehicular cloud serving systems with software-defined networking," in *Proceedings of the Second International Conference on Internet of Vehicles - Safe and Intelligent Mobility - Volume 9502*, ser. IOV 2015. Berlin, Heidelberg: Springer-Verlag, 2015, pp. 58–67. [Online]. Available: https://doi.org/10.1007/978-3-319-27293-1_6

- [36] M. A. Salahuddin, A. Al-Fuqaha, and M. Guizani, "Software-Defined Networking for RSU Clouds in Support of the Internet of Vehicles," *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 133–144, April 2015.
- [37] M. A. Salahuddin, A. Al-Fuqaha, M. Guizani, and S. Cherkaoui, "RSU cloud and its resource management in support of enhanced vehicular applications," in *2014 IEEE Globecom Workshops (GC Wkshps)*, Dec 2014, pp. 127–132.
- [38] K. Zheng, L. Hou, H. Meng, Q. Zheng, N. Lu, and L. Lei, "Soft-defined heterogeneous vehicular network: architecture and challenges," *arXiv preprint arXiv:1510.06579*, 2015.
- [39] X. Huang, J. Kang, R. Yu, M. Wu, Y. Zhang, and S. Gjessing, "A Hierarchical Pseudonyms Management Approach for Software-Defined Vehicular Networks," in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, May 2016, pp. 1–5.
- [40] M. Zhu, Z. P. Cai, M. Xu, and J. N. Cao, "Software-defined vehicular networks: Opportunities and challenges," in *Energy Science and Applied Technology*. CRC Press, 2015, pp. 247–251.
- [41] P. Duan, C. Peng, Q. Zhu, J. Shi, and H. Cai, "Design and analysis of software defined Vehicular Cyber Physical Systems," in *Parallel and Distributed Systems (ICPADS), 2014 20th IEEE International Conference on*, 2014, pp. 412–417.
- [42] L. Yu-Chun, C. Chien, and S. Chakraborty, "A Software Defined Network architecture for GeoBroadcast in VANETs," in *Communications (ICC), 2015 IEEE International Conference on*, 2015, pp. 6559–6564.
- [43] B. Karp and H.-T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM, 2000, pp. 243–254.
- [44] H. Zongjian, Z. Daqiang, and L. Junbin, "Cost-Efficient Heterogeneous Data Transmission in Software Defined Vehicular Networks," in *High Performance Computing and Communications (HPCC), 2015 IEEE 7th International Symposium on Cyberspace Safety and Security (CSS), 2015 IEEE 12th International Conferen on Embedded Software and Systems (ICESS), 2015 IEEE 17th International Conference on*, 2015, pp. 666–671.

- [45] Z. He, D. Zhang, and J. Liang, “Cost-Efficient Sensory Data Transmission in Heterogeneous Software-Defined Vehicular Networks,” *IEEE Sensors Journal*, vol. 16, no. 20, pp. 7342–7354, 2016.
- [46] A. Kazmi, M. A. Khan, and M. U. Akram, “DeVANET: Decentralized Software-Defined VANET Architecture,” in *2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW)*, April 2016, pp. 42–47.
- [47] S. Hassas Yeganeh and Y. Ganjali, “Kandoo: A Framework for Efficient and Scalable Offloading of Control Applications,” in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, ser. HotSDN ’12. New York, NY, USA: ACM, 2012, pp. 19–24. [Online]. Available: <http://doi.acm.org/10.1145/2342441.2342446>
- [48] T. Zhang, A. Bianco, and P. Giaccone, “The role of inter-controller traffic in SDN controllers placement,” in *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Nov 2016, pp. 87–92.
- [49] A. Bianco, P. Giaccone, S. D. Domenico, and T. Zhang, “The Role of Inter-Controller Traffic for Placement of Distributed SDN Controllers,” *CoRR*, vol. abs/1605.09268, 2016. [Online]. Available: <http://arxiv.org/abs/1605.09268>
- [50] A. Kosta, N. Pappas, and V. Angelakis, “Age of Information: A New Concept, Metric, and Tool,” *Foundations and Trends® in Networking*, vol. 12, no. 3, pp. 162–259, 2017. [Online]. Available: <http://dx.doi.org/10.1561/13000000060>
- [51] S. Kaul, R. Yates, and M. Gruteser, “Real-time status: How often should one update?” in *2012 Proceedings IEEE INFOCOM*, March 2012, pp. 2731–2735.
- [52] Q. Zheng, K. Zheng, H. Zhang, and V. C. M. Leung, “Delay-Optimal Virtualized Radio Resource Scheduling in Software-Defined Vehicular Networks via Stochastic Learning,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 7857–7867, Oct 2016.
- [53] D. Deng, S. Lien, C. Lin, S. Hung, and W. Chen, “Latency Control in Software-Defined Mobile-Edge Vehicular Networking,” *IEEE Communications Magazine*, vol. 55, no. 8, pp. 87–93, Aug 2017.

- [54] X. Xiao and X. Kui, "The Characterizes of Communication Contacts Between Vehicles and Intersections for Software-Defined Vehicular Networks," *Mobile Networks and Applications*, vol. 20, no. 1, pp. 98–104, 2015.
- [55] G. Secinti, B. Canberk, T. Q. Duong, and L. Shu, "Software Defined Architecture for VANET: A Testbed Implementation with Wireless Access Management," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 135–141, July 2017.
- [56] R. Dos Reis Fontes, C. Campolo, C. Esteve Rothenberg, and A. Molinaro, "From Theory to Experimental Evaluation: Resource Management in Software-Defined Vehicular Networks," *IEEE Access*, vol. 5, pp. 3069–3076, 2017.
- [57] R. d. R. Fontes and C. E. Rothenberg, "Mininet-WiFi: A Platform for Hybrid Physical-Virtual Software-Defined Wireless Networking Research," in *Proceedings of the 2016 ACM SIGCOMM Conference*, ser. SIGCOMM '16. New York, NY, USA: ACM, 2016, pp. 607–608. [Online]. Available: <http://doi.acm.org/10.1145/2934872.2959070>
- [58] B. Heller, R. Sherwood, and N. McKeown, "The Controller Placement Problem," in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, ser. HotSDN '12. New York, NY, USA: ACM, 2012, pp. 7–12.
- [59] A. Sallahi and M. St-Hilaire, "Optimal Model for the Controller Placement Problem in Software Defined Networks," *IEEE Communications Letters*, vol. 19, no. 1, pp. 30–33, 2015.
- [60] S. Lange, S. Gebert, T. Zinner, P. Tran-Gia, D. Hock, M. Jarschel, and M. Hoffmann, "Heuristic Approaches to the Controller Placement Problem in Large Scale SDN Networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 1, pp. 4–17, 2015.
- [61] H. T. Hashemi and S. Khorsandi, "Load Balanced VANET Routing in City Environments," in *2012 IEEE 75th Vehicular Technology Conference (VTC Spring)*, May 2012, pp. 1–6.
- [62] N. J. Patel and R. H. Jhaveri, "Trust based approaches for secure routing in vanet: A survey," *Procedia Computer Science*, vol. 45, pp. 592 – 601, 2015, international Conference on Advanced Computing

- Technologies and Applications (ICACTA). [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1877050915003488>
- [63] M. Zhu, J. Cao, D. Pang, Z. He, and M. Xu, "SDN-Based Routing for Efficient Message Propagation in VANET," in *Wireless Algorithms, Systems, and Applications: 10th International Conference, WASA 2015, Qufu, China, August 10-12, 2015, Proceedings*, K. Xu and H. Zhu, Eds. Cham: Springer International Publishing, 2015, pp. 788–797.
- [64] T. Taleb, E. Sakhaee, A. Jamalipour, K. Hashimoto, N. Kato, and Y. Nemoto, "A Stable Routing Protocol to Support ITS Services in VANET Networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3337–3347, Nov 2007.
- [65] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," Tech. Rep. 2070-1721, 2003.
- [66] C. M. Silva, B. M. Masini, G. Ferrari, and I. Thibault, "A Survey on Infrastructure-Based Vehicular Networks," *Mobile Information Systems*, vol. 2017, p. 28, 2017. [Online]. Available: <https://doi.org/10.1155/2017/6123868>
- [67] C. Liu, Y. Shu, O. Yang, Z. Xia, and R. Xia, "SDR: A Stable Direction-Based Routing for Vehicular Ad Hoc Networks," *Wireless Personal Communications*, vol. 73, no. 3, pp. 1289–1308, Dec 2013. [Online]. Available: <https://doi.org/10.1007/s11277-013-1278-6>
- [68] Q. Yang, A. Lim, S. Li, J. Fang, and P. Agrawal, "ACAR: Adaptive Connectivity Aware Routing Protocol for Vehicular Ad Hoc Networks," in *2008 Proceedings of 17th International Conference on Computer Communications and Networks*, Aug 2008, pp. 1–6.
- [69] V. Naumov and T. R. Gross, "Connectivity-Aware Routing (CAR) in Vehicular Ad-hoc Networks," in *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, May 2007, pp. 1919–1927.
- [70] Z. Mo, H. Zhu, K. Makki, and N. Pissinou, "MURU: A Multi-Hop Routing Protocol for Urban Vehicular Ad Hoc Networks," in *2006 Third Annual International Conference on Mobile and Ubiquitous Systems: Networking Services*, July 2006, pp. 1–8.

- [71] D. Abada and A. Massaq, "Article: Improving Relay Selection Scheme for Connecting VANET to Internet over IEEE 802.11p," *International Journal of Computer Applications*, vol. 132, no. 2, pp. 1–8, December 2015, published by Foundation of Computer Science (FCS), NY, USA.
- [72] A. Vahdat and D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks," Tech. Rep., 2000.
- [73] Y. Ohta, T. Ohta, E. Kohno, and Y. Kakuda, "A Store-Carry-Forward-Based Data Transfer Scheme Using Positions and Moving Direction of Vehicles for VANETs," in *2011 Tenth International Symposium on Autonomous Decentralized Systems*, March 2011, pp. 131–138.
- [74] L. MacVittie, "Reactive, Proactive, Predictive: SDN Models," 2013.
- [75] K. L. K. Sudheera, M. Ma, G. G. M. N. Ali, and P. H. J. Chong, "Delay efficient software defined networking based architecture for vehicular networks," in *2016 IEEE International Conference on Communication Systems (ICCS)*, Dec 2016, pp. 1–6.
- [76] Ming Zhu, Zhiping Cai, Jiannong Cao, and Ming Xu, "Efficient multiple-copy routing in software-defined vehicular networks," in *2015 International Conference on Information and Communications Technologies (ICT 2015)*, April 2015, pp. 1–6.
- [77] L. Chen and C. Englund, "Cooperative its - eu standards to accelerate cooperative mobility," in *2014 International Conference on Connected Vehicles and Expo (ICCVE)*, Nov 2014, pp. 681–686.
- [78] B. Dong, W. Wu, Z. Yang, and J. Li, "Software Defined Networking Based On-Demand Routing Protocol in Vehicle Ad Hoc Networks," in *2016 12th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*, Dec 2016, pp. 207–213.
- [79] H. Ghafoor and I. Koo, "CR-SDVN: A Cognitive Routing Protocol for Software-Defined Vehicular Networks," *IEEE Sensors Journal*, vol. 18, no. 4, pp. 1761–1772, Feb 2018.
- [80] C. Huang, M. Chiang, D. Dao, W. Su, S. Xu, and H. Zhou, "V2V Data Offloading for Cellular Network Based on the Software Defined Network (SDN) Inside Mobile Edge Computing (MEC) Architecture," *IEEE Access*, vol. 6, pp. 17 741–17 755, 2018.

- [81] S. Singh, S. Negi, S. K. Verma, and N. Panwar, "Comparative Study of Existing Data Scheduling Approaches and Role of Cloud in VANET Environment," *Procedia Computer Science*, vol. 125, pp. 925 – 934, 2018, the 6th International Conference on Smart Computing and Communications.
- [82] J. Zhang, Q. Zhang, and W. Jia, "VC-MAC: A Cooperative MAC Protocol in Vehicular Networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1561–1571, March 2009.
- [83] G. G. M. N. Ali, E. Chan, and W. Li, "On Scheduling Real-Time Multi-item Queries in Multi-RSU Vehicular Ad Hoc Networks (VANETs)," in *2013 27th International Conference on Advanced Information Networking and Applications Workshops*, March 2013, pp. 7–12.
- [84] M. A. Rahman, G. G. M. N. Ali, P. H. J. Chong, S. K. Samantha, M. F. Muntasir, and C. Chen, "On Scheduling Real-Time Multi-item Query with Network Coding in Multi-RSU Vehicular Networks," in *2016 IEEE 22nd International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, Aug 2016, pp. 218–227.
- [85] K. Liu, J. K. Y. Ng, V. C. S. Lee, S. H. Son, and I. Stojmenovic, "Cooperative Data Scheduling in Hybrid Vehicular Ad Hoc Networks: VANET as a Software Defined Network," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1759–1773, June 2016.
- [86] C. Yi, G. Jinhua, and W. Yue, "SDN enabled content distribution in vehicular networks," in *Innovative Computing Technology (INTECH), 2014 Fourth International Conference on*, 2014, pp. 164–169.
- [87] E. Bozkaya and B. Canberk, "QoE-Based Flow Management in Software Defined Vehicular Networks," in *2015 IEEE Globecom Workshops (GC Wkshps)*, Dec 2015, pp. 1–6.
- [88] G. Luo, J. Li, L. Zhang, Q. Yuan, Z. Liu, and F. Yang, "sdnMAC: A Software-Defined Network Inspired MAC Protocol for Cooperative Safety in VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 6, pp. 2011–2024, June 2018.
- [89] I. Yaqoob, I. Ahmad, E. Ahmed, A. Gani, M. Imran, and N. Guizani, "Overcoming the Key Challenges to Establishing Vehicular Communication: Is SDN the Answer?" *IEEE Communications Magazine*, vol. 55, no. 7, pp. 128–134, July 2017.

- [90] Kushan Sudheera Kalupahana Liyanage and Maode Ma and Peter Han Joo Chong, "Controller placement optimization in hierarchical distributed software defined vehicular networks," *Computer Networks*, vol. 135, pp. 226 – 239, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1389128618300951>
- [91] D. J. Deng, S. Y. Lien, C. C. Lin, S. C. Hung, and W. B. Chen, "Latency Control in Software-Defined Mobile-Edge Vehicular Networking," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 87–93, 2017.
- [92] M. Yu, J. Rexford, M. J. Freedman, and J. Wang, "Scalable Flow-based Networking with DIFANE," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, pp. 351–362, Aug. 2010.
- [93] S. Mehar, S. M. Senouci, A. Kies, and M. M. Zoulikha, "An Optimized Roadside Units (RSU) placement for delay-sensitive applications in vehicular networks," in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, Jan 2015, pp. 121–127.
- [94] Y. Fu, J. Bi, K. Gao, Z. Chen, J. Wu, and B. Hao, "Orion: A Hybrid Hierarchical Control Plane of Software-Defined Networking for Large-Scale Networks," in *2014 IEEE 22nd International Conference on Network Protocols*, Oct 2014, pp. 569–576.
- [95] A. Koshibe, A. Baid, and I. Seskar, "Towards distributed hierarchical SDN control plane," in *2014 International Science and Technology Conference (Modern Networking Technologies) (MoNeTeC)*, Oct 2014, pp. 1–5.
- [96] S. Vissicchio, L. Vanbever, and O. Bonaventure, "Opportunities and research challenges of hybrid software defined networks," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 2, pp. 70–75, 2014.
- [97] K. A. Hafeez, A. Anpalagan, and L. Zhao, "Optimizing the Control Channel Interval of the DSRC for Vehicular Safety Applications," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 5, pp. 3377–3388, May 2016.
- [98] CISCO, "The Cisco Application Policy Infrastructure Controller - White Paper," 2014. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/unified-fabric/white-paper-c11-730021.html>

- [99] S. Singh, P. Kumari, and S. Agrawal, "Comparative Analysis of Various Routing Protocols in VANET," in *2015 Fifth International Conference on Advanced Computing Communication Technologies*, Feb 2015, pp. 315–319.
- [100] C. Sarr, C. Chaudet, G. Chelius, and I. G. Lassous, "Bandwidth Estimation for IEEE 802.11-Based Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 10, pp. 1228–1241, Oct 2008.
- [101] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, "DevoFlow: Scaling flow management for high-performance networks," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 254–265, 2011.
- [102] O. N. Foundation, "OpenFlow Switch Specification - Version 1.3.0," 2012. [Online]. Available: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.1.pdf>
- [103] A. Tamir, "The k-centrum multi-facility location problem," *Discrete Applied Mathematics*, vol. 109, no. 3, pp. 293 – 307, 2001. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0166218X00002535>
- [104] I. ILOG, "Cplex optimization studio," 2014. [Online]. Available: <http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer>
- [105] G. F. Riley and T. R. Henderson, *The ns-3 Network Simulator*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 15–34.
- [106] J. Haerri, F. Filali, and C. Bonnet, "Performance comparison of AODV and OLSR in VANETs urban environments under realistic mobility patterns," in *Proc. of 5th IFIP Mediterranean Ad-Hoc Networking Workshop (Med-Hoc-Net-2006), Lipari, Italy*, vol. 1045, 2006.
- [107] I. Grigorik, "Latency: The New Web Performance Bottleneck," 2012. [Online]. Available: <https://www.igvita.com/2012/07/19/latency-the-new-web-performance-bottleneck/>
- [108] OpenSignal, "State of Mobile Networks: USA (August 2016)," 2016. [Online]. Available: <http://www.opensignal.com/reports/2016/08/state-of-the-mobile-network-usa>
- [109] K. L. K. Sudheera, M. Ma, and P. H. J. Chong, "Link Dynamics Based Packet Routing Framework for Software Defined Vehicular Networks," in

- GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–6.
- [110] K. S. K. L., M. Ma, and P. H. J. Chong, “Efficient Flow Instantiation via Source Routing in Software Defined Vehicular Networks,” in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, Sep. 2017, pp. 1–5.
 - [111] K. L. K. Sudheera, M. Ma, and P. H. J. Chong, “Link Stability Based Optimized Routing Framework for Software Defined Vehicular Networks,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2934–2945, March 2019.
 - [112] G. G. N. Sandamali, R. Su, Y. Zhang, and Q. Li, “Flight routing and scheduling with departure uncertainties in air traffic flow management,” in *2017 13th IEEE International Conference on Control Automation (ICCA)*, July 2017, pp. 301–306.
 - [113] G. G. N. Sandamali, R. Su, and Y. Zhang, “Flight Routing and Scheduling Under Departure and En Route Speed Uncertainty,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2019.
 - [114] D. Abada and A. Massaq, “Improving Relay Selection Scheme for Connecting VANET to Internet over IEEE 802.11p,” 2015.
 - [115] I. Gurobi Optimization, “Gurobi optimizer reference manual,” 2016. [Online]. Available: <http://www.gurobi.com>
 - [116] A. Bemporad and M. Morari, “Control of systems integrating logic, dynamics, and constraints,” *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.
 - [117] A. M. Sharp, “Incremental algorithms: solving problems in a changing world,” Ph.D. dissertation, Cornell University, 2007.
 - [118] Q. Wen, R. Chen, L. Nai, L. Zhou, and Y. Xia, “Finding Top K Shortest Simple Paths with Improved Space Efficiency,” in *GRADES’17*, 2017.
 - [119] M. H. Eiza, Q. Ni, T. Owens, and G. Min, “Investigation of routing reliability of vehicular ad hoc networks,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, no. 1, p. 179, Jul 2013.
 - [120] G. Cardano, *The Rules of Algebra: (Ars Magna)*, 2007.
 - [121] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, “Recent Development and Applications of SUMO - Simulation of Urban MObility,” *Interna-*

- tional Journal On Advances in Systems and Measurements*, vol. 5, no. 3 - 4, pp. 128–138, December 2012.
- [122] K. L. K. Sudheera, M. Ma, and P. H. J. Chong, “Link Stability Based Hybrid Routing Protocol for Software Defined Vehicular Networks,” in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–6.
- [123] K. S. K. Liyanage, M. Ma, and P. H. J. Chong, “Connectivity aware tribrid routing framework for a generalized software defined vehicular network,” *Computer Networks*, vol. 152, pp. 167 – 177, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S138912861830608X>
- [124] G. Vaira and O. Kurasova, “Parallel Bidirectional Dijkstra’s Shortest Path Algorithm,” in *Proceedings of the 2011 Conference on Databases and Information Systems VI: Selected Papers from the Ninth International Baltic Conference, DB&IS 2010*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2011, pp. 422–435.
- [125] Google, “Google Maps,” 2016.
- [126] OpenStreetMap contributors, “Planet dump retrieved from <https://planet.osm.org> 2017,” <https://www.openstreetmap.org>.
- [127] K. L. Kushan Sudheera, M. Ma, and P. H. J. Chong, “Cooperative Data Routing & Scheduling In Software Defined Vehicular Networks,” in *2018 IEEE Vehicular Networking Conference (VNC)*, Dec 2018, pp. 1–8.
- [128] K. L. K. Sudheera, M. Ma, and P. H. J. Chong, “Real Time Cooperative Data Routing and Scheduling in Software Defined Vehicular Networks,” *IEEE Transactions on Vehicular Technology (Under Review)*, 2019.
- [129] G. G. Nadeesha Sandamali, R. Su, and Y. Zhang, “Short-term En route Air Traffic Flow Management Under Departure and Wind Uncertainties with a Heuristic and Greedy Solution Approach,” in *2019 American Control Conference (ACC)*, July 2019, pp. 2121–2126.
- [130] R. Olmi, “Heuristic Algorithm for finding Maximum Independent Set,” 2010. [Online]. Available: <https://www.mathworks.com/matlabcentral/fileexchange/28470-heuristicalgorithm-for-finding-maximum-independent-set>
- [131] J. M. Robson, “Algorithms for Maximum Independent Sets,” *J. Algorithms*, vol. 7, pp. 425–440, 1986.