

Cost-effective Video streaming for Internet of Connected Vehicles using Heterogeneous Networks

A dissertation submitted in partial fulfillment of the requirements
for the award of the degree of

Doctor of Philosophy

Submitted by

Debanjan Roy Chowdhury

Under the supervision of

Prof. Sukumar Nandi, Prof. Diganta Goswami



Department of Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
Guwahati 781039, Assam, India

17th August 2023





Dedicated to

*My parents & my family
for their unconditional love and support*



Acknowledgements

I would like to express my sincere gratitude to all those who have contributed to the successful completion of this thesis.

First and foremost, I am deeply grateful to my thesis supervisors, Prof. Sukumar Nandi and Prof. Diganta Goswami, for their invaluable guidance, support, and encouragement throughout the research process. Without their expertise and patience, this thesis would not have been possible. I am very fortunate to be associated with them since last few years, which transformed me in many positive ways. Their immense knowledge in their expertise areas is the primary driving force behind this thesis. Despite their extremely busy schedule, they always accommodated me with warm gestures to listen to all my naive queries and have discussions with me. Each and every discussion with them has enriched me as a researcher and increased my ability of critical thinking. They guided me in all the difficult situations of my research, taught me how to handle them, and actively helped me to get out of them. Most importantly, throughout my entire tenure in this institute, they always took care of my personal well-being and health. They taught me how to always remain positive towards life.

I extend my heartfelt gratitude to my doctoral committee members, Prof. Sajith Gopalan (the chairperson of the committee), Dr. Moumita Patra, and Prof. Ashok Singh Sairam, for their insightful comments and constructive feedback, which helped me to refine my research works. I am grateful to all the respected faculty members of the Department of CSE and the Head of the department for providing a healthy learning environment in the classes and labs. I am thankful to my institute, the Indian Institute of Technology Guwahati, for providing excellent campus infrastructure for comfortable staying and studying. My sincere gratitude towards the Ministry of Education, Govt. of India, for providing monthly assistanceship. I wish to thank all the staff members of the CSE department for giving all the required technical, logistic, and official support during my tenure in this institute.

I am forever grateful to my labmates (NeSec lab) Subhrendu Chattopadhyay, Pranav Kumar Singh, Madhurima Buragohain, Pradeep Kumar Bhale, Sukanta De, Saurav Gupta,

Manoj Das, and Saurav Kumar. I have spent countless good times with them, and it is because of them, the lab has become my second home. A special mention to my senior Subhrendu Chattopadhyay, who mentored me in my initial struggle days. I express my love and gratitude towards all my friends at this institute, without whom my journey of Ph.D. wouldn't have been completed. Nilotpal Biswas, Manju R, Ujjwal Biswas, Menaxi Bagchi, Yashdeep Singh, Hemraj Raikwar, Nidhi Ahlawat, Maithilee Patawar, Sumita Majhi, Surja Sanyal, Hema Yarnagula, Suraj Pandey, Karan Kumar, Rohith Sangineni, Brij Nandan Tripathi, S. Raveesh, and Sunil Kumar are only a few of the many names to mention. I am thankful to all my beloved juniors of the CSE department who always made me feel good with their warm gestures towards me.

Finally, I want to thank the people who are the closest to my heart and are my core strength. I want to express my deepest love and regards to my parents, Mrs. Bula Roy Chowdhury and Mr. Soumen Roy Chowdhury, and my elder sister Mrs. Debaraty Bhattacharya, for their blessings and unconditional love for me. The core values taught by them are the most valuable treasures I will carry throughout my life. I am grateful to my beloved wife, Mrs. Anasua Mitra, for always being on my side and inspiring me in various ways. I thank all my relatives for their love and for supporting me in various ways in my journey of life.

Declaration

I certify that,

- The work contained in this thesis is original and has been done by myself and under the general supervision of my supervisor(s).
- The work reported herein has not been submitted to any other Institute for any degree or diploma.
- Whenever I have used materials (concepts, ideas, text, expressions, data, graphs, diagrams, theoretical analysis, results, etc.) from other sources, I have given due credit by citing them in the text of the thesis and giving their details in the references. Elaborate sentences used verbatim from published work have been clearly identified and quoted.
- I also affirm that no part of this thesis can be considered plagiarism to the best of my knowledge and understanding and take complete responsibility if any complaint arises.
- I am fully aware that my thesis supervisor(s) are not in a position to check for any possible instance of plagiarism within this submitted work.

Debanjan Roy Chowdhury

17th August 2023



Department of Computer Science and Engineering
Indian Institute of Technology Guwahati
Guwahati - 781039 Assam India

Certificate

This is to certify that this thesis entitled “**Cost-effective Video streaming for Internet of Connected Vehicles using Heterogeneous Networks**” submitted by **Debanjan Roy Chowdhury**, in partial fulfillment of the requirements for the award of the degree of Doctor of Philosophy, to the Indian Institute of Technology Guwahati, Assam, India, is a record of the bonafide research work carried out by him under our guidance and supervision, at the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, Assam, India. To the best of our knowledge, no part of the work reported in this thesis has been presented for the award of any degree at any other institution.

Date: 17th August 2023

Place: Guwahati

Prof. Sukumar Nandi
(Main-supervisor)
Professor
Dept. of C.S.E
IIT Guwahati

Prof. Diganta Goswami
(Co-supervisor)
Professor
Dept. of C.S.E
IIT Guwahati

Abstract

Internet of Connected Vehicles (IoCV) comprises smart vehicles which communicate among themselves and are connected to the Internet through static infrastructure nodes. Infrastructure nodes may use heterogeneous network technologies like cellular networks, Wifi networks, or *Dedicated Short Range Communication (DSRC)* networks. Among these networks, cellular networks have limited resources and impose access costs. Therefore, reducing the number of simultaneous cellular connections in an IoCV is a requirement. Smart vehicles of IoCV need persistent Internet connections for various safety messages and infotainment services. Among the infotainment services, video type infotainment services are prevalent. As the major portion of the traffic carried by the Internet core is of video type, reducing video traffic is the need of the hour. To meet the high-quality and low-latency demands for video services, content originators use the services of *Content Distribution Networks (CDN)*. While providing video infotainment services over IoCVs, the objectives of CDN providers are to reduce the traffic volume of the Internet core, reduce service costs, and increase service profitability. To reduce the traffic volume, CDN providers deploy replica servers to serve the demands locally. However, if several vehicles demand the same video content simultaneously, like in the case of a live video streaming, a CDN replica server may get overwhelmed by the number of concurrent and redundant flow requests. As the content demand is homogeneous, the number of one-to-one flows to the CDN replica server can be reduced by bringing the content further closer to an IoCV using edge servers. Using infrastructure nodes as edges incurs deployment costs or carrier partnership costs, whereas using vehicles as edges needs *Vehicle-to-Vehicle (V2V)* collaborations. To reduce the service cost, the CDN provider needs to minimize the usage of simultaneous cellular connections and maximize V2V collaborations while ensuring service quality and client satisfaction. To generate additional revenues, CDN providers offer multi-tier video services where higher-tier clients pay more for enhanced video quality. However, the dynamic connectivity among vehicles and the intermittent availability of different networks (Wifi, cellular, DSRC) make the above-mentioned tasks extremely challenging. Accordingly, the objective of this dissertation is to find cost-effective solutions for CDN providers to run video infotainment services over IoCVs. This dissertation

has four contributions toward the objective. The first contribution is focused on devising a centralized solution for reducing Internet bandwidth usage and the number of simultaneous cellular connections by minimizing the number of edge vehicles. The second contribution has proposed a distributed version of the first contribution, which helps CDN providers to reduce capital expenditure by avoiding setting up expensive servers of high-computing facilities. In the third contribution, a solution is provided for efficient *Vehicle-to-Infrastructure* (V2I) mode selection to increase CDN providers' profit in heterogeneous network scenarios. The fourth contribution of this dissertation devises an edge selection solution for CDN providers to provide multi-tier streaming services. The experiment results show that in comparison to existing solutions, the proposed solutions are the most cost-effective for CDN providers.



Table of Contents

List of Figures	xvii
List of Tables	xxi
1 Introduction	3
1.1 Internet of connected vehicles	3
1.2 IoCV Applications	4
1.2.1 Human safety	4
1.2.2 Transport efficiency	5
1.2.3 Infotainment	5
1.3 Video streaming over IoCVs	6
1.4 Content Distribution Networks	7
1.5 Challenges and Motivations	8
1.5.1 Challenges from IoCVs	9
1.5.1.1 V2V communications	9
1.5.1.2 V2I communications	10
1.5.2 Challenges from live video streaming	10
1.5.3 Challenges from edge vehicle selection	11
1.5.4 Challenges from multi-tier services	12

1.5.5	Challenges from security perspective	12
1.6	Thesis Overview and Contributions	13
1.6.1	Centralized optimum edge vehicle selection for live video streaming over IoCV	13
1.6.2	Cost-effective distributed edge vehicle selection with V2V multicasting	14
1.6.3	Distributed V2I network selection to maximize cellular traffic offloading	15
1.6.4	Distributed edge vehicle selection for multi-tier live video streaming	16
1.7	Thesis Organization	16
2	Background and Literature Survey	19
2.1	Internet of connected vehicles	19
2.1.1	Communication Standards	20
2.1.1.1	IEEE WAVE	20
2.1.1.2	3GPP C-V2X	22
2.1.2	IoCV Routing	24
2.1.2.1	Unicast	24
2.1.2.2	Multicast	25
2.1.2.3	Broadcast	25
2.2	Video distribution	26
2.2.1	Video data	26
2.2.1.1	Video compression	26
2.2.1.2	H.264 video structure	27
2.2.2	H.264 video streaming	28
2.2.3	Video streaming over IoCV	30

2.3	Content distribution networks	30
2.3.1	Replica server placement	32
2.3.2	Content placement	33
2.3.2.1	Outsourcing policies	33
2.3.2.2	Placement policies	34
2.3.2.3	Consistency maintaining policies	35
2.3.3	Request routing	35
2.3.3.1	DNS based redirection	36
2.3.3.2	HTTP redirection	36
2.3.3.3	URL rewriting	36
2.3.3.4	Anycasting	37
2.3.4	CDNs for IoCVs	37
2.4	Summary	39
3	Centralized optimum edge vehicle selection for live video streaming over IoCV	41
3.1	Related Work	43
3.2	Proposed Solution	43
3.2.1	System model	44
3.2.2	Protocol overview	45
3.2.3	Edge selection	48
3.2.4	Client-edge association phase	50
3.2.5	V2V data phase	52
3.3	Protocol Analysis	52
3.3.1	Protocol correctness	52
3.3.2	Protocol complexity	53
3.3.3	Performance analysis	53

3.4	Performance Evaluation	55
3.4.1	Simulation environment	56
3.4.2	Performance metrics	58
3.4.3	Simulation results	61
3.5	Summary	66
4	Cost-effective distributed edge vehicle selection with V2V multicasting	67
4.1	Proposed Solution	68
4.1.1	System model	69
4.1.2	Protocol overview	70
4.1.3	Problem Formulation	71
4.1.4	CNTRL Phase	72
4.1.4.1	DSS-CAST	73
4.1.4.2	States of c-nodes	73
4.1.4.3	PDIS subphase	74
4.1.4.4	ESEL subphase	74
4.1.4.5	ASCN subphase	79
4.1.5	DATA Phase	80
4.1.6	Feasibility Analysis	81
4.2	Performance Evaluation	83
4.2.1	Performance metrics	83
4.2.2	Simulation environment and parameters	84
4.2.3	Simulation results	85
4.3	Summary	90
5	Distributed V2I network selection to maximize cellular traffic offloading	91

5.1	Related Work	93
5.2	System model	94
5.3	Problem formulation	96
5.4	Proposed protocol	100
5.4.1	Network selection (NWSEL)	102
5.4.2	Network selection maintenance (NSMNTN)	104
5.5	Protocol analysis	110
5.5.1	Offloading performance	110
5.5.2	Computation complexity	111
5.5.3	Control overhead	111
5.6	Performance Evaluation	112
5.6.1	Simulation environment and parameters	112
5.6.2	Performance metrics	114
5.6.3	Simulation results	116
5.7	Summary	125
6	Distributed edge vehicle selection for multi-tier live video streaming	127
6.1	Related Work	128
6.2	Proposed work	129
6.2.1	System model	129
6.2.2	Problem formulation	130
6.2.3	Proposed Solution	133
6.2.3.1	CNTRL phase	133
6.2.3.2	DATA phase	134
6.3	Performance evaluation	135
6.3.1	Simulation environment and parameters	135

6.3.2	Performance metrics	135
6.3.3	Simulation results	138
6.4	Summary	141
7	Conclusions and Future research directions	143
7.1	Conclusions	143
7.2	Future research directions	144
7.2.1	Exploring V2V communication of IoCVs with 5G C-V2X	144
7.2.2	Exploring ICN paradigm for video streaming over IoCVs	145
7.2.3	Ensuring security in video services over IoCVs	146
	References	147

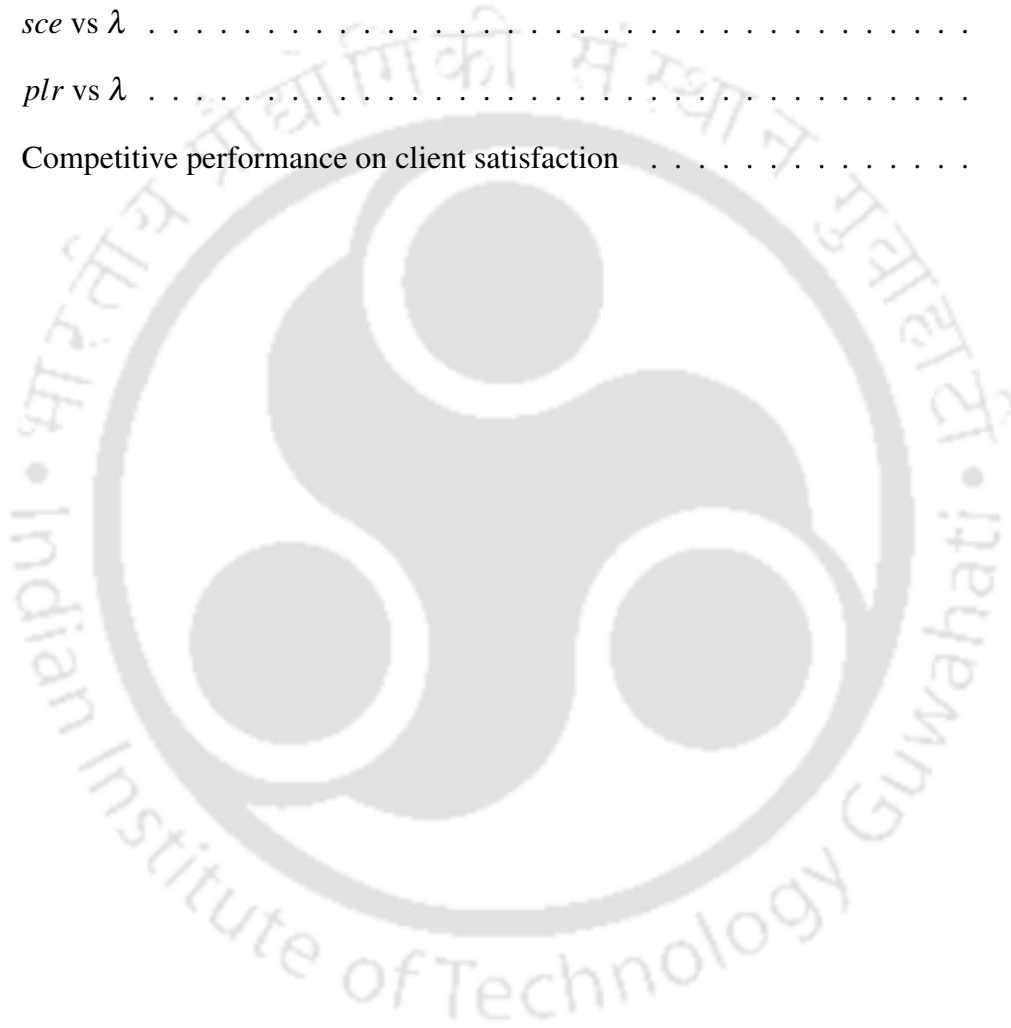
List of Figures

1.1	An example of IoCV	6
1.2	Vehicles fetching the same content directly and simultaneously from the content originator is causing wastage of Internet bandwidth and putting the burden on the content originator server.	7
1.3	To reduce the number of flows to the content originator, the content is distributed among Vehicles from replica CDN servers	8
1.4	To reduce the number of Internet flows and bandwidth requirements, a few edge vehicles selected by CDN are fetching content from replica CDN servers and distributing it among peer vehicles. For timely delivery, the multi-hop V2V collaboration is restricted to 2 hops.	9
1.5	An optimal edge vehicle selection considering live streaming latency constraints and relative velocity constraints	11
2.1	WAVE Protocol stack (<i>this image is reproduced from [1]</i>)	20
2.2	1609.4 multi-channel access (<i>this image is reproduced from [1]</i>)	21
2.3	V2X functionality support for 5G (<i>this image is reproduced from [2]</i>)	22
2.4	V2X communication support (<i>this image is reproduced from [3]</i>)	23
2.5	Hierarchical B frame prediction (<i>this image is reproduced from [4]</i>)	27
2.6	H.264 enhancement layers (<i>this image is reproduced from [5]</i>)	28
2.7	CDN architecture (<i>this image is reproduced from [6]</i>)	31
3.1	scenario of video streaming over IoV	44

3.2	Timeline sequence of phases and packet transmissions for a simple topology consisting of two c-nodes connected through a r-node. Without maintaining any application state, a r-node can help two c-nodes to share content . . .	46
3.3	LTE cost effectiveness vs Supported relative velocity	59
3.4	Packet delivery ratio vs Supported relative velocity	60
3.5	Edge selection ratio (esr) vs c-node ratio	60
3.6	Packet delivery ratio (pdr) vs c-node ratio	61
3.7	LTE cost effectiveness vs Vehicle density	62
3.8	Packet delivery ratio vs Vehicle density	63
3.9	number of video interruptions vs Vehicle density	64
3.10	Peak signal to noise ratio vs Vehicle density	64
4.1	An example scenario of video distribution in VANET by edges of 3-hop domination	69
4.2	Timeline sequence of phases and packet transmissions for a simple topology consisting of two c-nodes connected through a r-node. Without maintaining any application state, a r-node can help two c-nodes to share content . . .	71
4.3	DSS-CAST control headers	73
4.4	DSS-CAST forest of multicast trees	80
4.5	LTE cost effectiveness vs Relative velocity threshold	85
4.6	Packet delivery ratio vs Relative velocity threshold	86
4.7	LTE cost effectiveness vs Vehicle density	87
4.8	Packet delivery ratio vs Vehicle density	87
4.9	No. of video interruptions vs Vehicle density	89
4.10	Peak signal to noise ratio vs Vehicle density	89
5.1	A scenario of an IoCV with max 3-hop V2V collaborations	95
5.2	A small scenario of max 2-hop V2V collaborations	97

5.3	Protocol timeline	100
5.4	Protocol Flowchart	101
5.5	Vehicle state transitions and system instability because of Event1 experienced by C2	105
5.6	Vehicle state transitions and system instability because of Event2 experienced by C0	106
5.7	Vehicle state transitions and system instability because of Event3 experienced by C4	107
5.8	Vehicle state transitions and system instability because of Event4 experienced by C2	109
5.9	Urban irregular road topology of Bucks County, Pennsylvania, 19056, United States (40.1521, -74.9297, 40.1821,-74.8655)	113
5.10	Sensitivity of <i>octr</i> metric w.r.t ϑ	117
5.11	Sensitivity of <i>cpo</i> metric w.r.t ϑ	117
5.12	Sensitivity of <i>plr</i> metric w.r.t ϑ	118
5.13	Sensitivity of <i>bur</i> metric w.r.t ϑ	119
5.14	Sensitivity of <i>PSNR</i> metric w.r.t ϑ	119
5.15	Performance of <i>octr</i> metric w.r.t λ	121
5.16	Performance of <i>cpo</i> metric w.r.t λ	121
5.17	Performance of <i>plr</i> metric w.r.t λ	122
5.18	Performance of <i>bur</i> metric w.r.t λ	123
5.19	Performance of <i>PSNR</i> metric w.r.t λ	124
5.20	Performance of <i>sse</i> metric w.r.t λ	124
6.1	An example scenario with two service tiers with max four hop collaborations	130
6.2	Enhancement layer insertion schedule for CEMV	131

6.3	Protocol timeline, message sequences, and state transitions for a sample three node topology consisting of two ϕ_1 clients and one ϕ_2 client	132
6.4	<i>sce</i> vs $R\vartheta$	136
6.5	<i>plr</i> vs $R\vartheta$	137
6.6	<i>ssr</i> vs $R\vartheta$	137
6.7	<i>sce</i> vs λ	139
6.8	<i>plr</i> vs λ	139
6.9	Competitive performance on client satisfaction	140



List of Tables

3.1	Simulation static parameters	56
5.1	Utility vector \mathcal{U} for Fig. 5.2 scenario	98
5.2	Selection availability matrix \mathcal{F} for Fig. 5.2 scenario	98
5.3	Decision matrix \mathcal{D} for Fig. 5.2 scenario	99
5.4	Simulation parameters	114
6.1	Simulation parameters	135



List of Abbreviations

5G 5th Generation Mobile Network	DNS Domain Name Service
ACK Acknowledgment	DSRC Dedicated Short Range Communication
ANDSF Access Network Discovery and Selection Function	EDCA Enhanced Distributed Channel Access
AoI Area of Interest	ETSI European Telecommunications Standards Institute
AVC Advanced Video Coding	GoP Group of Picture
BS Base Station	GSC Greedy Set Cover
BSM Basic Safety Message	HDS HTTP Dynamic Streaming
CA Central Authority	HEVC High Efficiency Video Coding
CCH Control Channel	HLS HTTP Live Streaming
CDN Content Distribution Networks	IaS Infrastructure as Service
CSMA-CA Carrier Sense Multiple Access	ICN Information-centric Networks
CTS Clear To Send	IDM-IM Intelligent Driver Model with Intersection Management
C-V2X Cellular Vehicle to Everything	ILPP Integer Linear Programming Problem
DASH Dynamic Adaptive Streaming over HTTP	IoCV Internet of connected vehicles
DCF Distributed Coordination Function	
DCT Discrete Cosine Transform	

ISM Industrial, Scientific, and Medical	PSID Provider Service Identifier
ISP Internet Service Provider	PSNR Peak Signal to Noise Ratio
ITS Intelligent Transport System	QoE Quality of Experience
LLC Link Logical Control	QoS Quality of Service
LRU Least Recently Used	RB Resource Blocks
LTE-A Long Term Evolution Advanced	RSS Received Signal Strength
MANET Mobile Ad-hoc Network	RSU Road Side Unit
MCU Multipoint Control Unit	RTS Request To Send
MKDS Minimum K -hop Dominating Set	SCH Service Channel
MSC Minimum Set Covering	SDN Software Defined Networking
MSS Microsoft Smooth Streaming	SEAL Service Enabler Architecture Layer
multi-RAT multiple Radio Access Technology	SNR Signal to Noise Ratio
NAL Network Abstraction Layer	SPS Semi-Persistent Scheduling
NALU Network Abstraction Layer Unit	SUMO Simulation of Urban Mobility
NAV Network Allocation Vector	SVC Scalable Video Coding
NSMNTN Network selection Maintenance	UE User Equipment
NWSEL Network Selection	V2I Vehicle to Infrastructure
OCB Outside the Context of a Basic service set	V2V Vehicle to Vehicle
OFDM Orthogonal Frequency-Division Multiplexing	VAE V2X Application Enabler
OR Opportunistic Routing	VANET Vehicular Ad-hoc Network
PLMN Public Land Mobile Network	VM Virtual Machine
	WAVE Wireless Access in Vehicular Environment
	WSMP WAVE Short Message Protocol

Chapter 1

Introduction

This dissertation proposes cost-effective solutions for *Content Distribution Networks* (CDN) providers to run video streaming applications over *Internet of Connected Vehicles* (IoCVs). The objectives of the solutions are to minimize Internet bandwidth usage, minimize simultaneous cellular network connections, and increase service profitability. The introduction of the thesis has been organized into following sections. Section 1.1 gives a brief introduction to the IoCVs. Section 1.2 discusses various applications and usages of IoCVs. Section 1.3 focuses on various kinds of video streaming applications and the application requirements. The roles of CDN providers for video streaming services over IoCVs are discussed in Section 1.4. The challenges and the motivations of this thesis is discussed in Section 1.5. Thesis overview and contribution summary is given in Section 1.6. Finally, Section 1.7 gives the thesis organization.

1.1 Internet of connected vehicles

An IoCV is a network of communication-capable vehicles with persistent Internet connections [7]. A vehicle of IoCV is equipped with *On-Board Unit* (OBU) which enables wireless communication. Vehicles are connected among themselves through *Vehicle-to-Vehicle* (V2V) communications by forming a *Vehicular Ad-hoc Network* (VANET). The Internet connectivity is achieved through static infrastructure nodes using *Vehicle-to-Infrastructure* (V2I) communications. Various technologies available for V2V and V2I communications are discussed below.

For V2V communications, IEEE has amended its existing Wifi standard 802.11a as 802.11p for *Dedicated Short Range Communication* (DSRC) frequency band and further prescribed the standard series 1609.x for the upper layers, namely, *Wireless Access in Vehicular Environment* (WAVE) [1]. Another standard for V2V communication has been developed by *European Telecommunications Standards Institute* (ETSI) based on ITS-G5 band, namely, *Intelligent Transport System* (ITS) [8]. Recently, 3GPP has proposed another standard with the name of *Cellular Vehicle to Everything* (C-V2X) [9], which extends V2V to V2X, indicating that a vehicle can communicate with any device, for e.g., *User Equipments* (UE), sensors, IoT devices, etc., that are connected to the 5G eco-system. WAVE and ETSI ITS use shared wireless mediums with contention-based access control. They use the 5 GHz range of the *Industrial, Scientific, and Medical* (ISM) frequency bands that can be accessed without any cost. On the other hand, C-V2X needs resource (channel) allocation/reservation before communication and can be used only based on subscriptions.

Infrastructure nodes act as anchor nodes for communications in dynamic topologies. In IoCV, they also act as Internet access points for vehicles. Infrastructure nodes can use different network technologies. For example, *Road Side Units* (RSUs) use DSRC, cellular *Base Stations* (BS) use *Long Term Evolution Advanced* (LTE-A) or *5th Generation Mobile Network* (5G), Wifi access points [10] use 802.11.b/a/g/n/ac/ax, etc. V2I communications may use any of the aforementioned infrastructure nodes and network access technologies. An IoCV accommodates the coexistence of different network access technologies making the network a heterogeneous one [11]. Accordingly, vehicles of IoCV are equipped with *multiple Radio Access Technology* (multi-RAT) interfaces to connect to infrastructure nodes of heterogeneous network technologies.

1.2 IoCV Applications

Applications of IoCV can be broadly classified into three classes as human safety, traffic efficiency, and infotainment, which are described below.

1.2.1 Human safety

The primary motivation of VANETs and IoCVs is to increase on-road human safety. Safety applications of IoCV enhance the safety of drivers, passengers, and pedestrians by taking proactive measures to avoid adverse situations, quickly disseminating critical information,

and advising precautionary measurements. Vehicles of IoCV are equipped with various sensors to interact with surrounding environments. All the sensor data are periodically sent to servers/clouds for processing and mining useful information. This information is disseminated among vehicles by safety applications. A few examples of safety applications are *emergency brake warning*, *intersection collision warning*, *lane change warning*, *post crash warning*, *emergency vehicle (e.g., ambulance) warning*, etc. [12].

1.2.2 Transport efficiency

These applications increase transport efficiency for economic benefits like reducing commute time and saving fuel. Hazardous road warnings, road congestion warnings, economic path planning, etc., are a few examples of transport efficiency applications for IoCV.

1.2.3 Infotainment

This type of IoCV application provides value-added services to onboard passengers. Infotainment applications have become popular as they increase convenience, provide entertainment, and bring opportunities for various service providers to earn profits. A few examples of infotainment services are audio/video advertisements, movie/audio file downloading, online multi-party gaming, video streaming, etc.

An example of an IoCV is shown in Fig. 1.1. Information about an adverse situation is quickly disseminated among vehicles using V2V communications. The message is also reported to the safety application server (run by the government) through the Internet using V2I communication. The server immediately informs the nearest ambulance about the incident, and vehicles collaborate among themselves to clear the lane for the ambulance. Wifi access points are located in important junctions or busy areas, like near shopping malls. Vehicles connect to infotainment servers for value-added services for onboard passengers. Infotainment servers may also push location-specific audio/video advertisements like nearby hotel offers, etc.

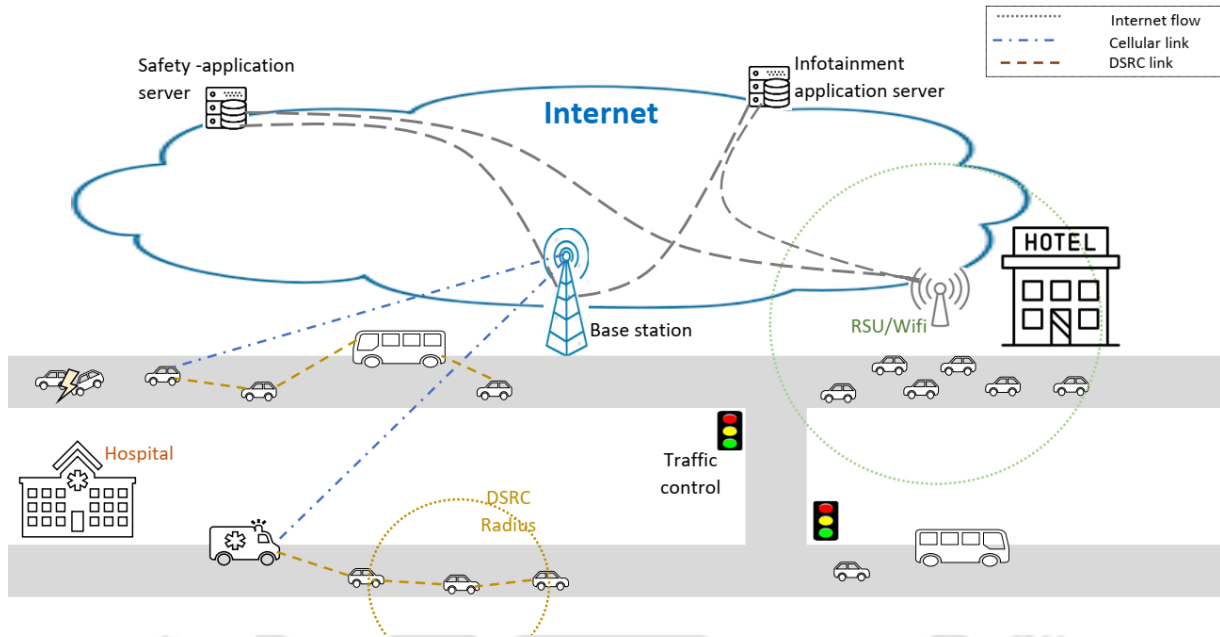


Fig. 1.1 An example of IoCV

1.3 Video streaming over IoCVs

Among the infotainment applications, video streaming applications are the most popular ones. As Cisco predicted, among all Internet traffic, a whopping 82% of traffic is video type [13]. Video streaming applications can be categorized into three types depending on the application requirements as given below.

- **On-demand streaming:** Users demand heterogeneous content asynchronously. The application can tolerate some latency. A playback buffer is used to tolerate jitters.
- **Live streaming:** A large group of users demands homogeneous content simultaneously. Latency should be as low as possible for users to watch the content before it becomes stale. Playback buffer sizes are smaller than that of on-demand streaming and can tolerate jitters of low magnitudes.
- **Real-time conferencing:** Multi-way video traffic is generated by a small group of users which needs to be delivered to all participants in real time. These applications are very sensitive to latency and jitters.

This thesis explores the challenges of video streaming applications over IoCVs through content distribution networks, which is introduced next.

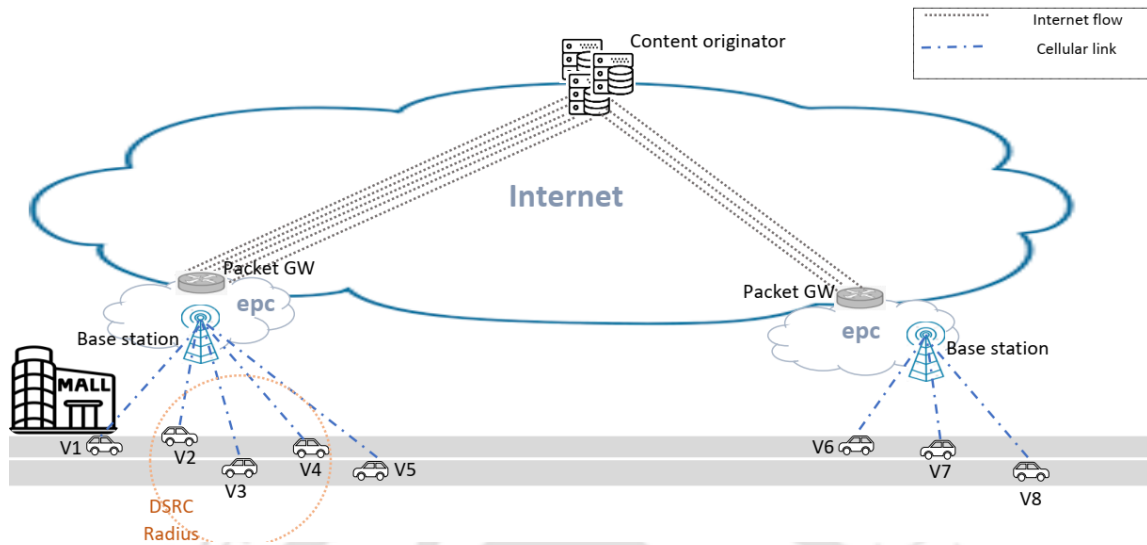


Fig. 1.2 Vehicles fetching the same content directly and simultaneously from the content originator is causing wastage of Internet bandwidth and putting the burden on the content originator server.

1.4 Content Distribution Networks

In case several users (vehicles) try to fetch content directly from a content originator server simultaneously, the server may get overwhelmed to handle so many concurrent flows. For homogeneous content demands, these one-to-one client-server flows are duplication of the same set of packets. As the server is far located from users across the globe, delivery latency is also increased. The situation is shown in Fig. 1.2. To meet users' demands for low latency and high-quality video streaming, content originators take the services of CDN [14]. A CDN provider deploys replica servers near the high-demand areas and keeps copies of original contents to serve the requests locally from the replica servers. This reduces the latency to a great extent compared to the latency experienced in fetching contents from the remote content-originator server. It also greatly reduces the bandwidth loads of the core Internet. The benefit of using CDN service for a live video streaming application is shown in Fig. 1.3. However, as the figure shows, the number of required Internet connections has not been reduced, and the number of concurrent flows (and bandwidth requirements) to the replica servers is high. Further improvements can be achieved if contents are placed further closer to IoCVs using edge servers [15].

A CDN provider may get the responsibility of efficiently distributing contents to an IoCV within an *Area of Interest* (AOI) like a city, stretches of important highways, etc. The CDN brings the contents further closer to the IoCV by caching the contents to the edges of IoCVs.

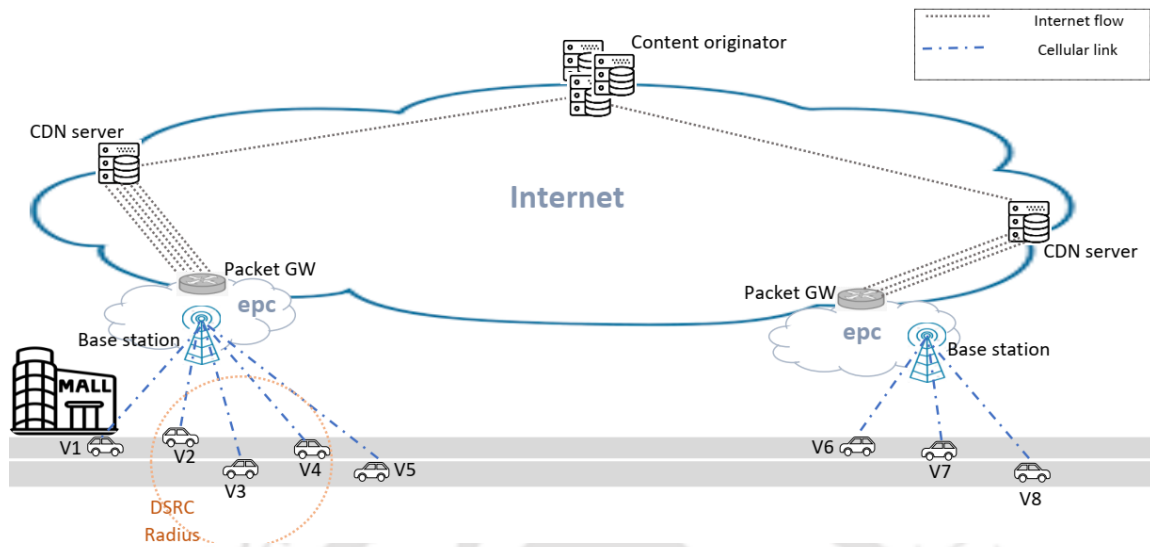


Fig. 1.3 To reduce the number of flows to the content originator, the content is distributed among Vehicles from replica CDN servers

A CDN provider can use static infrastructure nodes as well as a few suitable vehicles as edge servers/nodes. Edge nodes are responsible for fetching contents from the CDN's replica (placed near AOI) servers through the Internet and distributing the contents among other client vehicles. An IoCV scenario for live video distribution with edge vehicles is shown in Fig. 1.4 where the number of simultaneous Internet connections and bandwidth requirements has greatly reduced. To ensure the timely delivery of live video streaming, the multi-hop V2V collaboration must be restricted within a few hops. Fig. 1.4 shows the example of 2 hops V2V collaboration.

1.5 Challenges and Motivations

This dissertation explores the challenges of live video streaming applications over IoCVs from a CDN provider's perspective. The challenges faced by CDN providers can be due to the dynamic topology of IoCVs, constraints of streaming applications, optimum edge vehicle selections, or multi-tier service requirements. Each of the challenges has been discussed subsequently.

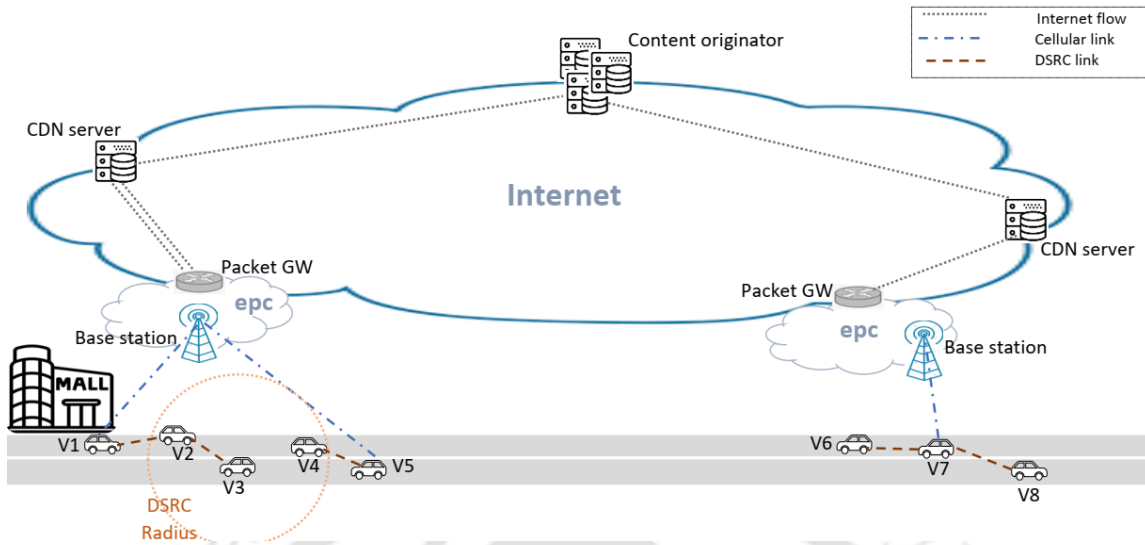


Fig. 1.4 To reduce the number of Internet flows and bandwidth requirements, a few edge vehicles selected by CDN are fetching content from replica CDN servers and distributing it among peer vehicles. For timely delivery, the multi-hop V2V collaboration is restricted to 2 hops.

1.5.1 Challenges from IoCVs

The challenges faced by CDN providers from IoCVs can be further sub-categorized as the challenges from V2V communications and V2I communications.

1.5.1.1 V2V communications

Contention-based shared wireless medium of DSRC makes V2V communication unreliable, whereas, C-V2X technology has the overhead of resource reservations and incurs access costs. Due to the high mobility of vehicles, the mean connectivity duration between two vehicles is usually short to very short (in the case of opposite-direction vehicles), making the topology highly dynamic [16, 17]. Moreover, topologies of IoCVs are very much fragmented [18, 17, 19] into disjoint clusters hindering V2V message propagation. Within a cluster, V2V collaborations can be achieved through single-hop [20, 21] or multi-hop [22–24] collaborations. Single-hop collaborations are easy to maintain, but with the cost of an increased number of edge vehicles. Though multi-hop collaborations solve the issue, it incurs the overhead of multi-hop path creation and maintenance in dynamic ad-hoc environments.

In an IoCV, all vehicles act as routers and are capable of routing/forwarding IP packets. However, only a few vehicles at a time become interested (become clients) for an infotainment

application and run the corresponding client application. All other vehicles are termed ordinary vehicles in this work. Client vehicles may need to exchange application-specific messages for V2V collaborations and content sharing. However, the client vehicles may be connected among themselves through ordinary vehicles that cannot interpret any application-specific message, making V2V collaboration challenging. V2V content distribution can be done through unicast, broadcast, or multicast. Broadcasts may result in network congestion, whereas unicast results in high redundancy in case the content is homogeneous, like live video streaming. Though multicast techniques are appropriate for live video streaming, they have overheads of multi-hop path establishment and maintenance in highly dynamic environments.

Motivated by the above challenges, this dissertation devises suitable V2V communication solutions to efficiently distribute live video streaming content over IoCVs.

1.5.1.2 V2I communications

Based on vehicle mobility and application requirements, one network access technology may be preferred over others for Internet connectivity. The small communication radius of DSRC and Wifi access points results in frequent connectivity disruptions for high-velocity vehicles. On the other hand, the communication radius of cellular base stations is of few kilometers [11]. However, unlike the free network access provisions for DSRC and Wifi, a vehicle needs to bear the cellular network access costs. Slow-moving vehicles prefer Wifi access points over cellular base stations when they are within Wifi coverage. On the other hand, for fast-moving vehicles, persistent Internet connectivity can be provided by cellular base stations. Due to these reasons, selecting the appropriate mode of V2I communication is a challenging task for CDN providers. This dissertation proposes a V2I solution for network selection which reduces service cost and increases the profitability of CDN providers to run live video streaming services over IoCVs.

1.5.2 Challenges from live video streaming

The popular existing video distribution frameworks like Apple's *HTTP Live Streaming* (HLS), *Microsoft Smooth Streaming* (MSS), Adobe's *HTTP Dynamic Streaming* (HDS), and *Dynamic Adaptive Streaming over HTTP* (DASH) use HTTP protocol which requires one-to-one TCP connections between a client and the streaming server. As the content demand is homogeneous for live video streaming, one-to-one flows result in huge redundancy and

Internet bandwidth wastage (refer Fig. 1.3). As shown in Fig. 1.4, redundancy can be reduced if the CDN provider selects a few of the vehicles as edges, which fetch the content from the replica server and distribute the content among peer client vehicles. Accordingly, the CDN provider needs to create and maintain groups of vehicles and schedule V2V content delivery. However, V2V content distribution may experience delay, jitter, and packet loss, making live video streaming over IoCVs extremely challenging. Motivated by these challenges, this dissertation finds solutions for CDN providers to provide live video streaming over IoCVs with the objective of bandwidth saving while providing good *Quality of Experience* (QoE) to the end-users.

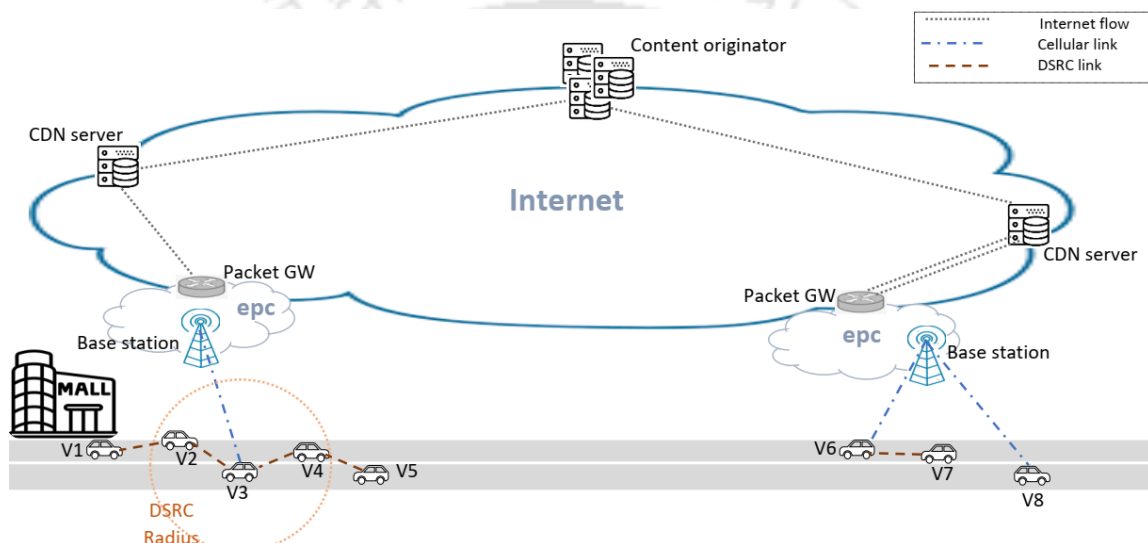


Fig. 1.5 An optimal edge vehicle selection considering live streaming latency constraints and relative velocity constraints

1.5.3 Challenges from edge vehicle selection

Unlike on-demand video streaming, a CDN cannot pre-fetch live video contents in edges to reduce latency as the contents are not available beforehand. As a result, a CDN provider needs to select edge nodes strategically such that the content is distributed among all the client vehicles within the stringent latency constraints of live streaming. CDN providers can use static infrastructure nodes [25–27], or a few of the vehicles as edge nodes [28, 29]. To use RSUs or Wifi access points as edge nodes, a CDN provider needs to bear the cost of dense deployment. To use cellular BSs as edge nodes, a CDN provider needs to collaborate with various carrier enterprises, which increases the operating costs of CDN providers. On the other hand, no extra cost is involved in using a few suitable vehicles as edge nodes.

However, as the connectivity among vehicles is highly dynamic, selecting suitable vehicles as edges is extremely challenging for CDN providers. For example, in Fig. 1.4, vehicle v7 has been selected as an edge to serve vehicle v8 despite the opposite moving directions of the two vehicles making the collaboration unstable. Accordingly, a CDN provider needs to select edge vehicles periodically and allow V2V collaborations only among vehicles of low relative velocities to mitigate dynamic connectivity. Selecting a suitable interval and relative velocity threshold is also a difficult task. A CDN provider also needs to select an optimum number of edge vehicles to minimize simultaneous cellular connections and total bandwidth requirements. In Fig. 1.4, vehicles v1 and v5 are selected as edge vehicles which are non-optimum, as selecting only vehicle v3 as an edge is sufficient to serve v1, v2, v4, and v5 in case of two-hop collaboration. The optimum edge selection for the considered scenario is shown in Fig. 1.5 considering all the constraints. However, for a large IoCV network, the optimum selection of edges with all the constraints of live streaming is a non-trivial task. Motivated by the challenges, this dissertation proposes solutions for optimum edge vehicle selection for CDN providers.

1.5.4 Challenges from multi-tier services

To generate additional revenues (increase profits), a CDN provider usually offers multi-tier services where clients pay more for higher-tier services to get enhanced video quality and other extra privileges. To provide a multi-tier service, a CDN provider must ensure the on-time delivery of the additional packets required for the higher-tier vehicles. As a result, the multi-tier service using V2V collaborations makes all the vehicles receive and forward additional packets to be consumed by only a few of the client vehicles. This additional packet delivery task may have adverse effects on the delay-constrained live video streaming service. Accordingly, CDN providers need to select edge vehicles judiciously such that the higher-tier packet delivery burden is minimized. The above challenges are the motivation for this dissertation to devise a solution for CDN providers to provide multi-tier services over IoCVs.

1.5.5 Challenges from security perspective

Malicious vehicles may attempt to disrupt any service over an IoCV in terms of availability, confidentiality, and integrity. Availability of service can be disrupted by various *denial of service* (DoS) attacks. In case of a centralized service, malicious vehicles try to exhaust the

resources of the centralized server. If the service is running over UDP, malicious vehicles can easily launch UDP flood DoS attacks. In case a service is running over TCP, malicious vehicles can launch *distributed DoS* (DDoS) attack by infecting several other vehicles with Trojan and using them as bots. For example, in a TCP SYN attack, the connection state table of a server quickly gets exhausted, which prevents legitimate vehicles from establishing new connections. If a service is using HTTP, then HTTP GET attack can be launched. In case of a distributed service, malicious vehicles can transmit jamming signals in designated service channels to make the service unavailable to other vehicles. Confidentiality is an integral part of any service. A CDN provider wants to ensure that no unsubscribed vehicle can access the provided content. Accordingly, vehicles and the server must be authenticated before any data exchange takes place. The integrity of a service ensures that when vehicles receive the content indirectly from a server through edge vehicles and multi-hop transmissions, the content is not altered by any of the malicious vehicles in the path. There are a few security concerns that are specific to IoCVs. The anonymity and privacy of a vehicle must be preserved. A vehicle wants to conceal its real identity from any service provider and does not want its location/route to be tracked. There are various routing attacks that may disrupt the multi-hop routing path of any service. For example, in black-hole attacks and grey-hole attacks, a malicious vehicle drops all or a few packets for its downstream vehicles. Though the above-mentioned security vulnerabilities are of serious concern and must be taken care of while designing any service over IoCV, these design aspects are out of the scope of this thesis work and can be addressed in future works.

1.6 Thesis Overview and Contributions

Based on the challenges and motivations mentioned in the previous section, the objective of this dissertation is to provide bandwidth-efficient and cost-effective solutions for CDN providers to offer live video streaming applications over IoCVs. The primary contributions of the thesis are summarized next.

1.6.1 Centralized optimum edge vehicle selection for live video streaming over IoCV

For a live video streaming application, content demands from all vehicles are homogeneous. To reduce the required bandwidth and number of cellular connections, a CDN provider selects

only a few client vehicles as edges that fetch the content from the replica CDN server and distribute the content among peer client vehicles. To maximize the cost-effectiveness of the application, a CDN provider wants to select a minimum number of edge vehicles. However, the CDN provider must ensure that the total distribution time including content fetching from the replica CDN server and multi-hop path propagation, must be within the delivery latency constraint of the live video streaming. As a result, multi-hop V2V collaborations must be restricted within a few hops, which in turn increases the number of edges. Moreover, multi-hop paths may get broken due to the dynamic connectivity of vehicles. Accordingly, a CDN provider needs to make edge selections frequently and optimally. In this dissertation, the above task has been formulated as the *Minimum Set Covering* (MSC) problem [30] and is solved by an existing centralized approximation algorithm. Vehicles periodically send their neighbor topology to the server, where the global topology is built, and an optimum number of edge vehicles are selected by solving the MSC problem on the global topology graph. As fresh edge selections are made after short intervals periodically, the problem of dynamic connectivity is mitigated. As the edge vehicles are selected using the MSC problem, the solution is optimum, which maximizes the cost-effectiveness of the live video streaming applications.

1.6.2 Cost-effective distributed edge vehicle selection with V2V multicasting

Centralized solutions for edge selections have some drawbacks. For a large network, the edge selection task with the global topology graph imposes a significant computational burden on the server. To finish the huge computation task within the delay constraints, a CDN provider needs to make massive investments in procuring high-computing facilities. Moreover, all vehicles need to send their local neighborhood to the server, which may get overwhelmed by numerous simultaneous connection requests. Alternatively, if vehicles are able to select edge vehicles themselves distributively, CDN providers are relieved from expensive server setups. This dissertation proposes a distributed edge selection protocol for CDN providers, which has an approximate optimal solution with time-bounded convergence.

As mentioned in Section 1.5.1.1, multicast techniques are required for network layer V2V collaborations for edge selection. Multicast techniques are well established for wired networks, and few of the techniques have been adapted [31–33] for ad-hoc networks. However, these adaptations are not adequate for the highly dynamic environments of IoCVs. They are also inappropriate for delay-constrained applications. A multicast protocol is proposed

in this dissertation, which is hop-constrained and relative velocity constrained to provide delay-bounded services and mitigate dynamic connectivity. The selected edge vehicles act as multicast sources to distribute contents among peer client vehicles. In the case of live streaming applications, all the multicast sources simultaneously distribute the same content. Therefore, it is sufficient for a client vehicle to be associated with only one of the multicast sources. However, existing multicast protocols do not have any provision for multicast source selection. The proposed multicast protocol has the provision of multicast source selection based on the delay and relative velocity constraints.

1.6.3 Distributed V2I network selection to maximize cellular traffic offloading

IoCV vehicles are connected to the Internet through V2I communications with infrastructure nodes. Multiple options of infrastructure nodes with different network access technologies may be available at a given time for vehicles of IoCvs. A vehicle needs to choose a network for V2I communication judiciously based on application objectives. In dense urban scenarios, a cellular network operator may deploy Wifi access points for cellular traffic offloading to increase the cellular network capacity. The cellular network operator may tie up with CDN providers to encourage its clients to offload from the cellular network to Wifi networks. In return, the cellular network provider agrees to pay some incentives to the CDN provider. As the content demand is homogeneous for live streaming, a CDN provider tries to increase the offloading (and incentives) by extending Wifi coverages through DSRC V2V collaborations wherever possible. However, intermittent availability of Wifi access points with small communication radius may cause frequent vertical handovers (handovers between different network technologies), disrupting the stability of the service. Moreover, the CDN provider needs to ensure the timely delivery of live video content to all the client vehicles. The above-mentioned challenges make the network selection task very challenging in case vehicles make network selections independently. In this dissertation, the task of network selection is shown to be NP-hard by formulating the task as an *Integer Linear Programming Problem (ILPP)*. Accordingly, a distributed algorithm is proposed as the solution which greedily maximizes the cellular traffic offloading while ensuring delay-constrained content delivery.

1.6.4 Distributed edge vehicle selection for multi-tier live video streaming

CDN providers usually offer their clients multi-tier services where the higher-tier clients pay more to enjoy better quality videos and other privileges. An agreement is made for the maximum achievable bit rate for each tier. Multi-tier services are different from adaptable bit-rate services [34, 35]. In an adaptable bit-rate service, all clients are identical from the service provider's perspective, and the bit rates are decided based on current channel conditions and network congestion. In contrast, in the case of multi-tier services, clients are not identical, and the bit-rates are decided as per the agreements based on clients' subscription tiers. This dissertation proposes a solution for CDN providers to offer multi-tier services for live video streaming to client vehicles. The objective is to minimize edge vehicles to save bandwidth with the constraints of delivery latency and tier-specific bit-rate agreements. The edge selection task with all these constraints is formulated as an MSC problem, and a distributive solution is provided accordingly.

1.7 Thesis Organization

The rest of the thesis document is organized into the following chapters:

- **Background and Literature survey:** This chapter lays the foundation for the dissertation. It covers all the required technical aspects and the standardizations for network access technologies, H.264 video communication over networks, and content distribution networks. Literature surveys are done for each of the above-mentioned areas.
- **Centralized optimum edge vehicle selection for live video streaming over IoCV:** The chapter gives a detailed description of the first thesis contribution, which gives a cost-effective edge selection solution to CDN providers to minimize the number of cellular connections and Internet bandwidth usages for live video streaming services over IoCVs.
- **Cost-effective distributed edge vehicle selection with V2V multicasting:** This chapter discusses in detail the second thesis contribution which devises a distributed edge vehicle selection for CDN providers to alleviate the requirements of centralized computation facilities to reduce the service cost.

This chapter also describes the proposed multicast protocol DSS-CAST as a V2V communication solution for distributing live video content among peer client vehicles.

DSS-CAST provides multicast source selection and association mechanism, and is designed to work in dynamic ad-hoc environments like IoCVs.

- **Distributed V2I network selection to maximize cellular traffic offloading:** This chapter explains the third thesis contribution which provides a V2I communication solution for live video streaming over IoCV. The solution provides a distributed algorithm for network access technology selection for individual vehicles, which maximizes the incentive of a CDN provider by greedily maximizing cellular traffic offloading.
- **Distributed edge vehicle selection for multi-tier live video streaming:** The chapter narrates the fourth thesis contribution. The solution is devoted for providing multi-tier live video streaming to increase the CDN provider's revenue.
- **Conclusions and Future research directions:** The final chapter summarizes the contributions of this thesis and discusses the scopes of future research.





Chapter 2

Background and Literature Survey

This chapter lays the technical foundation of this dissertation and is organized in the following ways. Section 2.1 discusses in detail the Internet of connected vehicles, Section 2.2 gives a brief overview of video distribution, and Section 2.3 discusses the Content Distribution Networks.

2.1 Internet of connected vehicles

An IoCV is a network of communication-capable vehicles with persistent Internet connections. An IoCV comprises vehicles and static infrastructure nodes. Vehicles form an ad-hoc network, namely VANET, which is a special subclass of *mobile ad-hoc network* (MANET). Unlike MANETs, the velocity of VANET nodes (vehicles) is very high, and the movements are road topology constrained. Unlike MANETs, static infrastructure nodes are present in VANETs, which usually act as anchor nodes for communications and provide Internet connectivity to vehicles. VANET nodes (vehicles) significantly differ from MANET nodes in terms of energy and computation power constraints, as vehicles have ample power supply and high computation units.

A smart vehicle of an IoCV has a number of sensors installed in it to constantly sense its environment. The sensed data are either processed locally or sent to a remote server for central processing and extracting useful information for various applications. As mentioned in Section 1.2, the primary objective of VANET and IoCV is to increase human safety. IoCV applications are also devised for transport efficiency and infotainment. IoCV applications require V2V/V2I communications for which one-to-one or one-to-many messages need to be

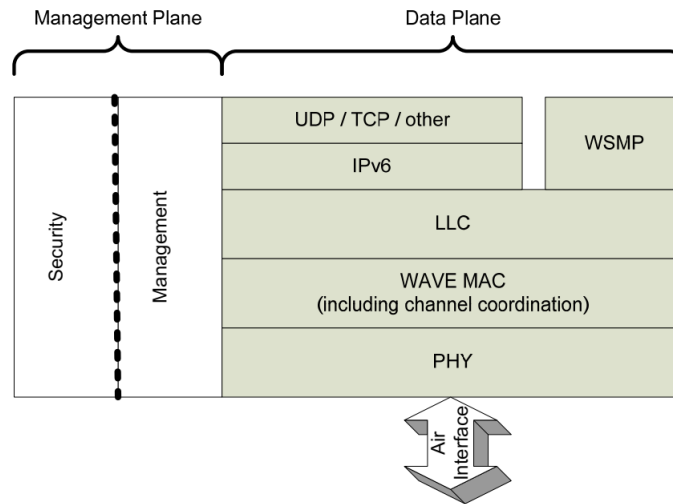


Fig. 2.1 WAVE Protocol stack (this image is reproduced from [1])

delivered correctly and efficiently, which is challenging in the highly dynamic scenarios of VANETs and IoCVs. Accordingly, a series of standards and a number of routing protocols have been defined to meet the distinguished requirements of VANETs which are discussed in subsequent sections.

2.1.1 Communication Standards

Vehicles of IoCVs are equipped with multiple communication interfaces to communicate with vehicles and infrastructure nodes using various technologies. As a result, a vehicle of IoCV needs to support multiple communication protocol stacks to accommodate heterogeneous technologies. Various standardization bodies like IEEE, ETSI, Japan ARIB STD-T109, and 3GPP have proposed different standards and specifications for vehicle communication stacks. Among these, IEEE and 3GPP standards are briefly described below.

2.1.1.1 IEEE WAVE

WAVE is a series of IEEE 1609.x standards defined for different layers of the communication stack as illustrated in Fig. 2.1. The physical layer consists of the DSRC radio channels of range 5.85 - 5.925 GHz band. The band is divided into seven channels of 10 MHz width, though two adjacent channels can be merged to get a 20 MHz channel. One channel (5.89 GHz) is used explicitly for control information and is named as *control channel* (CCH). The

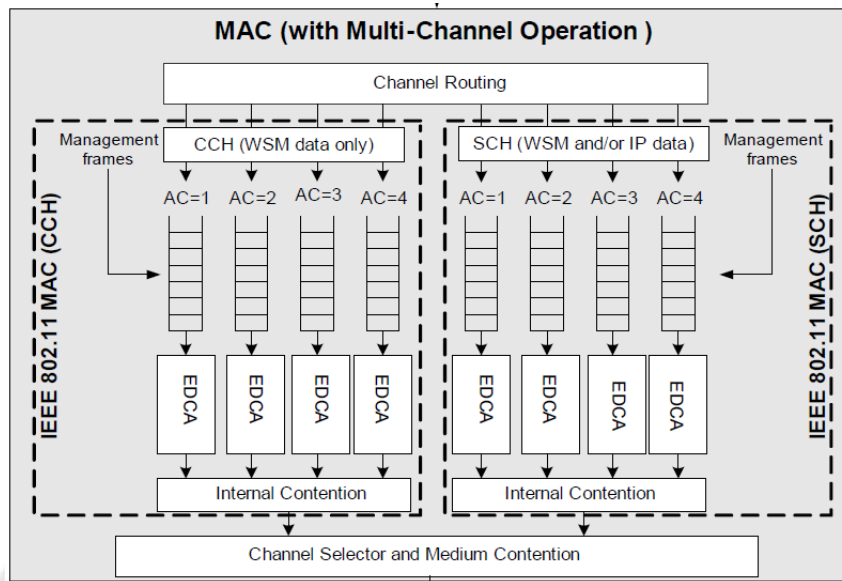


Fig. 2.2 1609.4 multi-channel access (this image is reproduced from [1])

other six channels are named service channels (SCH), out of which channel 5.86 GHz and channel 5.92 GHz are designated for safety applications.

The MAC layer of WAVE is based on 802.11p, which is an amendment of 802.11a. To cope up with highly dynamic environment of VANETs, the amendment is done to allow transmission *Outside the Context of a Basic service set (OCB)* with *dot11OCBActivated* flag on. As a result, the BSSID field can have a wildcard (*) value, and no association or authentication is required with access points or other nodes. IEEE 802.11 provides *Distributed Coordination Function (DCF)* for medium access and error control for a single channel. To make VANET work in multichannel environments, WAVE standard 1609.4 is proposed by IEEE, which creates an instance of 802.11 for each available channel with *Enhanced Distributed Channel Access (EDCA)* coordination function as depicted in Fig. 2.2. The DCF uses *Carrier Sense Multiple Access (CSMA-CA)* along with *Request To Send (RTS)*, *Clear To Send (CTS)*, *Network Allocation Vector (NAV)*, and *Acknowledgment (ACK)* for distributed channel access and collision avoidance. However, RTS/CTS and ACK mechanisms are available only for unicast transmissions. For broadcast/multicast frames, medium reservation and acknowledgments through RTS, CTS, and ACK are not possible. This is because, if a RTS is transmitted, all the neighbors simultaneously reply with CTS which collides with each other in the medium. This phenomenon is called the CTS explosion, and the RTS sender cannot receive all the required CTS before it can start transmitting the original data frame. On the other hand, if somehow the data frame is transmitted, if all the receivers simultaneously send acknowledge (ACK) for successful reception, these ACKs

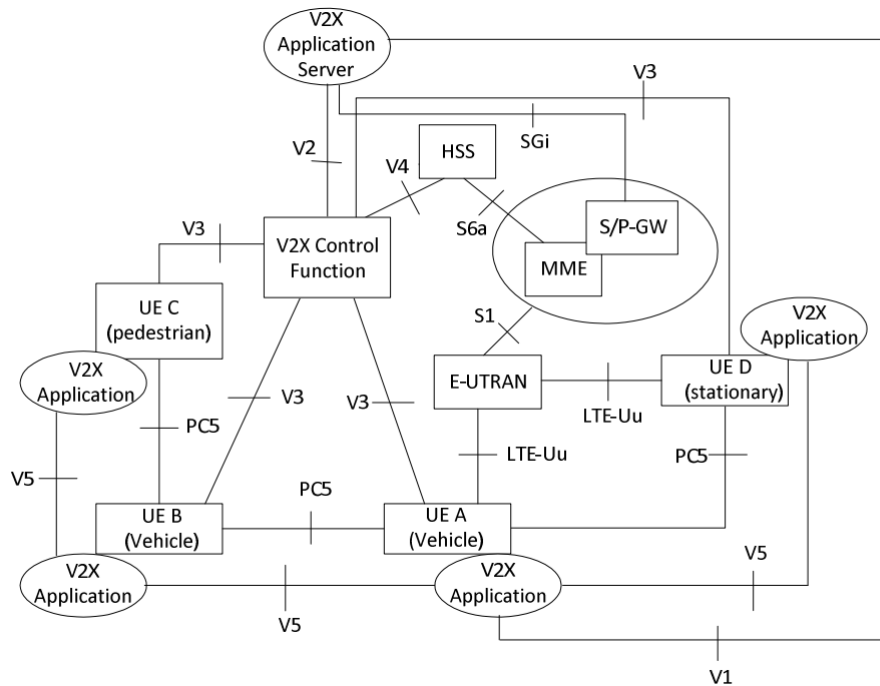


Fig. 2.3 V2X functionality support for 5G (this image is reproduced from [2])

collide with each other (known as ACK explosion), and the sender cannot know whether all intended recipients have actually received the frame.

WAVE proposes a dual-stack above the *Link Logical Control* (LLC) as shown in Fig. 2.1. The IPv6 and TCP/UDP stack is used to connect to the Internet, whereas the *WAVE Short Message Protocol* (WSMP) stack is used to send/receive *WAVE Short Messages* (WSM). The LLC layer distinguishes between the two upper stacks by the *Ethertype* field. WSMP can be sent to any channel, whereas IPv6 traffic can be sent only to SCHs. In the transport layer, while IPv6 uses standard TCP/UDP port numbers, the WSMP uses *Provider Service Identifier* (PSID) to distinguish among the services or applications. WAVE has defined security mechanisms and management protocols across the stack.

2.1.1.2 3GPP C-V2X

3GPP, in its release 16, has added a provision for vehicle-to-everything direct communication. Enhancements are proposed in the 5G ecosystem to accommodate this new provision. The functional architecture of C-V2X is shown in Fig. 2.3. As the diagram shows, the component V2X Control Function is added, which provides *User Equipments* (UEs) and application servers with the necessary parameters for V2X communications. Usually, only one V2X

Control Function provides service for all UEs and V2X application servers within a *Public Land Mobile Network (PLMN)*.

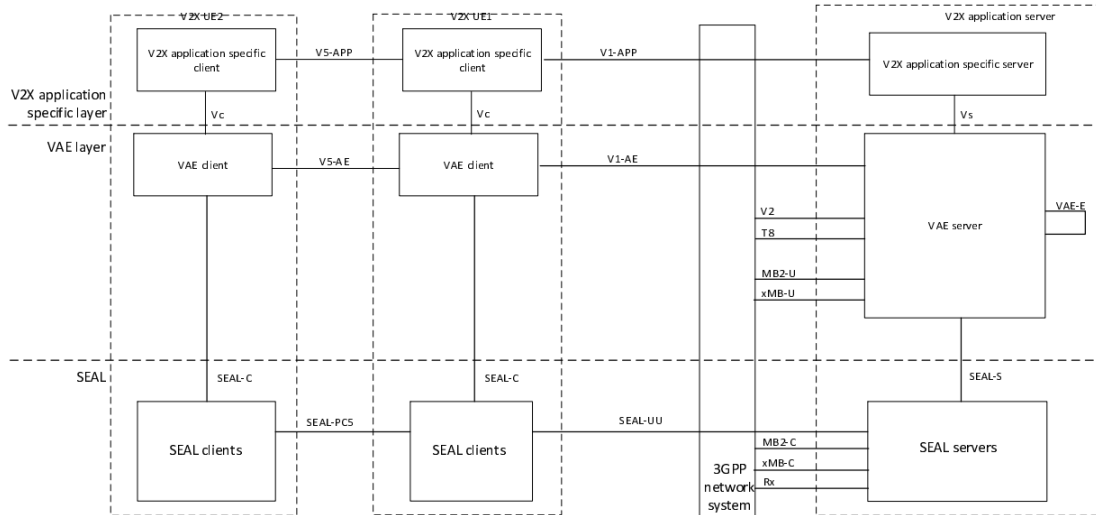


Fig. 2.4 V2X communication support (this image is reproduced from [3])

The communication model for UEs and a V2X application server is shown in Fig. 2.4. As can be seen from the diagram, one instance of *V2X Application Enabler (VAE)* and *Service Enabler Architecture Layer (SEAL)* is present in all UEs and servers. VAE components provide service registration, group management, monitoring reports, mode switching, 3GPP system configuration, etc. The SEAL components are responsible for location management, identity management, key management, network resource management, etc. A C-V2X device uses the standard 5G protocol stack for communication.

In the physical layer, V2X uses the newly introduced PC5 interface for the 5.9 GHz band as side-link channels. A channel can be 10 to 40 MHz in width and is divided into 180 KHz *Resource Blocks (RB)*. A RB is divided into 12 sub-carriers of 15 KHz in the frequency domain and is divided into sub-frames of 1 ms in the time domain. A sub-frame has 14 *Orthogonal Frequency-Division Multiplexing (OFDM)* symbols.

Medium access can be done in either Mode 3 or Mode 4. In Mode 3, the allocation is done centrally by the cellular infrastructure nodes, whereas in Mode 4, the vehicles distributively select RBs. As a result, Mode 3 operation needs vehicles to be present inside cellular network coverage, whereas Mode 4 can operate out of cellular network coverage. In Mode 3, a cellular infrastructure node allocates RBs to vehicles based on requests. The request can be made as *Semi-Persistent Scheduling (SPS)*, where a vehicle places a requirement for periodic RB allocation. In mode 4, vehicles distributively and autonomously reserve required RBs

using a sensing-based SPS scheme as specified in 3GPP Rel. 14/15. The vehicles announce their selection to their neighbors to prevent them from using the same RBs. 3GPP defines a mechanism to detect congestion while reserving RBs, which is used by the vehicles to adapt dynamically with the current channel condition.

2.1.2 IoCV Routing

To realize any IoCV application, application layer messages need to be routed among vehicles in one-to-one or one-to-many mode in a single or multi-hop manner. Routing in the VANET environment is a challenging task due to its highly dynamic network and short association time among vehicles. As per the application's requirement, routing can be of unicast, multicast, or broadcast type. Each type of routing is discussed subsequently.

2.1.2.1 Unicast

Traditional *Mobile Ad-hoc Networks* (MANET) routing protocols are of uni-path type, which is found to be inappropriate for highly dynamic ad-hoc networks. Accordingly, cluster-based routing is designed for VANET to reduce the overhead of end-to-end path maintenance. Cluster-based routing divides the entire network into clusters where one cluster head is elected for each cluster. For inter-cluster communication, paths have to be created via cluster heads. As a result, only cluster heads are burdened with path discovery and maintenance. A few examples of cluster-based unicast routing are COIN [36], APROVE [37], and CBD RP [38].

Another type of routing, namely *Opportunistic routing* (OR), is widely used in dynamic ad-hoc scenarios like IoCVs. In OR, relay selections are made for each individual packet in each hop, which makes OR robust and suitable for dynamic ad-hoc environments. Geo-routing is one kind of OR where the relay selection is based on vehicle position. The assumption is that each vehicle is aware of its own and its one-hop neighborhood GPS locations. Based on this information and the destination vehicle's location, the source node and each intermediate node takes routing and forwarding decision independently. A vehicle tries to forward a message to the neighbor vehicle, which is the closest to the destination vehicle. This approach is called greedy forwarding. However, it may get stuck to a local minima where an intermediate node finds its own location as the closest to the destination among its neighbors. A few examples of geo-routing are GPSR [39], AGF [40], CBF [41], and PBR-DV [42].

2.1.2.2 Multicast

Multicast routing is useful in situations where a subset of nodes becomes interested in a specific service. Interested receivers form groups and cooperate among themselves to deliver packets of interest. Participation or membership in a group may be implicit or explicit. Implicit memberships are usually based on time and a vehicle's location. Dissemination of safety/warning message to the vehicles approaching a road or location is an example of this kind. The works IVG [43], and Mobicast [44] are examples of implicit multicast routing. In explicit membership, individual vehicles get interested in different services and join appropriate groups. However, group creation and maintenance are extremely difficult in highly dynamic scenarios like VANETs. There are few traditional MANET multicast protocols like MOLSr [31], MAODV [32], and ODMRP [33], which are difficult to apply on VANETs.

2.1.2.3 Broadcast

Most safety applications and few transport efficiency and infotainment applications use multi-hop broadcasts to disseminate messages/data among vehicles within an AOI. Flooding is the most trivial way to achieve this goal, where the message originator transmits the message to all its immediate neighbors, which is further re-transmitted by all neighbors in a cascading manner. As a result, flooding results in exponential growth of the number of copies of the message, which makes the shared wireless medium so congested that it becomes unusable. This phenomenon is known as the *Broadcast Storm* problem. Accordingly, various kind of controlled flooding mechanism is used in practice, which aims to reduce the redundancy of message transmission. A controlled flooding mechanism strategically selects only a few of the neighbors for further re-transmitting of a message. To increase the reliability, some application-level acknowledgment mechanisms are used in a few works.

The sender of a message can select the neighbor subset based on various fitness parameters. The fitness parameters depend on the objectives of the broadcast scheme like reducing broadcast time [45], increasing broadcast coverage [46], increasing reliability [47] [48], reducing redundancy [49] [50], message relevance [51] [52], etc. Alternatively, the receivers can contend among themselves distributively to re-transmit a message. The contention can be based on re-transmission probability or re-transmission wait time. In probability-based methods, the receiving vehicles re-transmit the message with some probability. The probability can be predefined [53] [54] or decided dynamically [55] [56] [57]. On the other hand, in re-transmission wait time schemes, each vehicle calculates a wait time based on

parameters like distance from sender [51] [52] [58], location [59] [49], channel congestion [60] [61], etc. When one vehicle re-transmits the message, hearing that, other vehicles cancel their re-transmission schedule.

The above-discussed routing protocols distribute contents of various safety and non-safety IoCV applications. However, routing/distributing video content needs special attention due to its distinguished characteristics, which are discussed in the next section.

2.2 Video distribution

Video type data has its own distinguished properties making its distribution over a network challenging. Moreover, there are different types of video distribution services that vary among themselves based on service requirements. This section discusses the structure of video data, H.264 video streaming, and video streaming over IoCVs subsequently.

2.2.1 Video data

A digital video comprises a series of digital images or frames which are played one after another to produce an illusion of moving objects in human eyes. The quality of a video depends on the temporal sampling or the frame change rate, and the spatial sampling or the quality of the individual frames. Temporal sampling is to take snapshots of the input analog video signal in periodic time intervals. Spatial sampling denotes the two-dimensional discrete sample values (pixels) taken from each snapshot. Each pixel represents the color information of the sample area either in RGB format or YCrCb format. The total number of pixels in a frame is termed as resolution.

2.2.1.1 Video compression

The amount of data required to represent an uncompressed digital video is too large to be efficiently stored or transferred. Therefore, a video is encoded and compressed to a smaller representation removing the huge redundancy of information presented in the original video. The similarity between two subsequent frames is exploited by storing the reference frame, and only the differences of the next frame (residual image) compared to the reference frame. The motion estimation is done with respect to the reference frame to further reduce the residual information. Within a frame, not all pixel information is stored as many of them can

be generated by an extrapolation method by the neighboring pixels. These residual images then go through *Discrete Cosine Transformation* (DCT), or a variant of DCT, to reduce the statistical redundancy. The transform coefficients are then quantized so that non-significant coefficients can be discarded. Finally, the coefficient values are passed through an entropy coder to get the compressed video. Video de-compression follows the steps in a reverse way to generate the video for playback. Latest standards like H.264 or *Advanced Video Coding* (AVC), H.265 or *High Efficiency Video Coding* (HEVC) are widely used nowadays for video encoding and decoding.

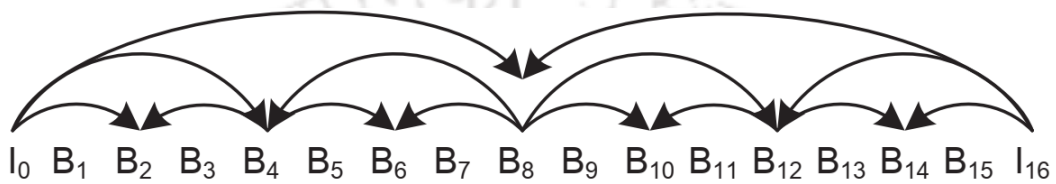


Fig. 2.5 Hierarchical B frame prediction (this image is reproduced from [4])

2.2.1.2 H.264 video structure

Depending on the encoding process, the H.264 standard categorizes a video frame into *intra-coded* (I), *forward predictive coded* (P), and *bi-directionally predictive coded* (B). P frames can be predicted (motion compensated) only from preceding I or P frames, whereas B frames are predicted using both preceding and succeeding frames. Two I frames with few P and B frames in between form a *Group of Picture* (GoP). H.264 standard has prescribed B frame hierarchical prediction as shown in Fig. 2.5. In this example, one I frame (I_0) and 15 B frames (B_1 to B_{15}) are consisting a GoP of size 16. As shown in the diagram, the middle B frame (B_8) is predicted first with I_0 and I_{16} (the first frame of the next GoP). B_4 is predicted using I_0 and B_8 , while B_{12} is predicted using B_8 and I_{16} . This hierarchical prediction continues till all frames of GoP are predicted. Though this technique increases the prediction quality, it increases the decoding time at the receiver as it needs to wait to receive all the frames of a GoP to proceed for prediction.

H.264 supports *Scalable Video Coding* where the lowest layer (base layer or layer 0) contains a self-sufficient video with lowest quality that can be played by a media player. There are one or more enhancement layers that need to be applied subsequently on top of the base layer to gradually increase the video quality. Each enhancement layer has its own identification number. An enhancement layer can increase the video quality in one of the three dimensions of scalability as mentioned below.

- *Temporal scalability*: It represents the increment of video frame rate. A hierarchical GoP B frame prediction structure is used to define enhancement layers. For the example of Fig. 2.5, I_0 and I_{16} consist of layer 0 (base layer) with only 2 frames/sec. Temporal scalability layer 1 consists of one level prediction frame B_8 . Together with layer 0 frames, a layer 1 video frame rate becomes 3 frames/sec. Temporal scalability layer 2 consists of B_4 and B_{12} , and together with layers 0 and 1, the frame rate becomes 5 frames/sec. This continues till the highest level of B frame prediction is reached.
- *Special scalability*: The base layer or layer 0 of the special scalability dimension represents the lowest resolution frames. The upper layers in this dimension increase the frame resolution gradually. H.264 has prescribed a macroblock coding mode for upsampling lower layer frames.
- *Quality scalability*: In this dimension, the enhancement layers successively increase the *Signal to Noise Ratio* (SNR) and decrease the encoding noise of the frames without increasing the resolution. This is achieved through re-quantization of the residual images with successively smaller quantization step sizes.

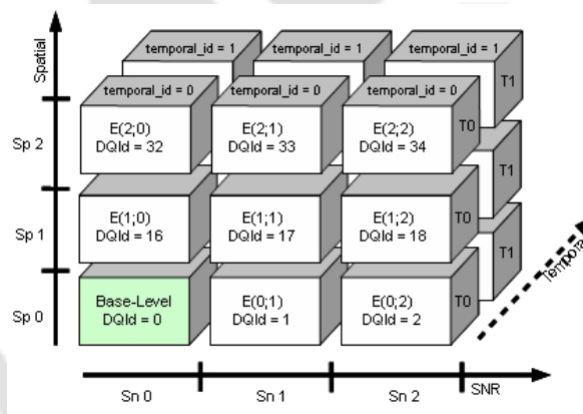


Fig. 2.6 H.264 enhancement layers (this image is reproduced from [5])

Fig. 2.6 shows an example of a H.264 video with two temporal, three spatial, and three quality layers. An enhancement layer can be uniquely identified by the tuple of $\langle \text{DQId}, \text{temporal id} \rangle$. The DQId is calculated as $((\text{spatial layer id} \ll 4) + \text{quality layer id})$. SVC is designed to make video transmission more agile and adaptive according to storage and bandwidth availability. H.264 video transmission over a network is discussed next.

2.2.2 H.264 video streaming

In addition to SVC, the H.264 standard defines the *Network Abstraction Layer* (NAL) specifically for efficient video transmission purposes over networks. In NAL, the encoded

video data is organized into chunks or NAL units (NALU) [62], which can be transferred independently. Depending on its size, a NALU is transmitted in one or more network packets. A NALU consists of a 1-byte header followed by control information or encoded video data representing an integer number of macroblock data [63]. A NALU header contains various parameters indicating the coding dependencies and the priority of the NALU. A NALU containing lower layer data is more important compared to a NALU containing upper layer data.

As mentioned in Section 1.3, video streaming services are sensitive to latency and jitter. As latency and jitter depend on network conditions, video streaming applications need to be agile to dynamically change the streaming bit rate such that users are able to experience a smooth playback. In case the network is congested, or the receiver is resource constrained, the sender may avoid forwarding NALUs of lower priorities without requiring to re-encode the video into lower quality. Unlike H.264 standard, earlier standards compel the sender to encode a single video source to multiple quality versions and store them separately. For example, *Hypertext Transfer Protocol* (HTTP) based video streaming frameworks like DASH [64], HLS, and HDS are popular for on-demand/live streaming of videos. These frameworks is used for streaming single layer (non-scalable) video formats where the entire video file is divided into small segments, and multiple bitrate versions are stored for each segment. A metadata file is sent to clients containing the list of all segments and the bitrate options available for each segment. Depending on current network conditions or buffer occupancy, an appropriate bitrate version of the next segment is chosen [65]. However, the problem with this approach is that once a segment is downloaded into a lower bitrate, the segment quality cannot be improved even if the currently available network bandwidth is increased. With H.264 video format, the issue can be solved as only a few enhancement layers needs to be downloaded opportunistically to improve the quality of a previously fetched segment [66]. In the case of real-time video conferencing, a dedicated server called *Multipoint Control Unit* (MCU) is needed to decode the incoming video source and re-encode it in different versions depending on the needs of multiple end devices. In the case of H.264 videos, a MCU only needs to choose which NALUs to forward and which NALUs to drop for each of the end nodes. However, HTTP-based streaming needs one-to-one TCP connections to the remote server for each client node. As mentioned in Section 1.4, this is a huge waste of bandwidth in the case of live video streaming. Moreover, it puts a load on cellular operators due to the requirement of a large number of simultaneous cellular channel access.

2.2.3 Video streaming over IoCV

Though OR is mostly used for IoCV for its robustness in a highly dynamic environment, they may cause delay and jitter for receiving nodes, because of which OR cannot be readily applied for streaming data. Few works have proposed modifications to existing OR protocols to make them suitable for streaming data. The works [67][68] improve throughput by retaining a relay selection decision for a few subsequent packets instead of treating each packet individually. Increasing redundancy [69] and using network coding [70] are a few of the proposed ways to increase reliability for streaming content for OR protocols. However, this imposes overhead which may lead to network congestion. Acknowledge and re-transmission based schemes are not used for streaming content as they may cause delay and jitter. The works of [67] [71] are examples of unicast, whereas the works [68][69][70] are examples of broadcast routing for V2V video delivery. Multicast techniques in the existing literature are proposed for traditional MANETs. The survey [72] topology-wise classified existing multicast protocols for MANET as source tree-based, shared tree-based, and mesh-based. In the case of source tree-based multicast like MOLSR[31], all source nodes maintain separate trees to reach member nodes. For shared tree-based multicast like MAODV[32], STAMP[73], a predefined/elected core node is responsible for maintaining a shared tree that connects all of the member nodes. Instead of a shared tree, mesh-based multicasts like ODMRP[33] and E-ODMRP[74] maintain a mesh-structured forwarding group to increase reliability at the cost of network overhead.

In the case of live streaming applications, all the multicast sources simultaneously distribute the same content. Therefore, it is sufficient for a client vehicle to be associated with only one of the multicast sources. However, existing multicast protocols do not have any provision for multicast source selection. The above protocols are efficiently used to deliver/disseminate in-network generated video data. However, to efficiently distribute a large volume of externally/remotely generated video data over an IoCV, CDN services are usually opted for, which is described next.

2.3 Content distribution networks

A CDN is a special-purpose network that offers *Quality of Service* (QoS) constrained content delivery/distribution service to large-scale content providers to decrease end users' perceived latency and increase the content provider's scalability. A CDN provider installs surrogate/replica servers in geographically dispersed strategic locations and creates an overlaid

network out of them. Based on content demands, contents are replicated in replica servers close to end users. A user's request for content is redirected by the request routing method to a suitable replica server. The performances of CDNs heavily depend on many design aspects like replica server placement, content selection, outsourcing, cache updation method, request routing mechanism, etc.

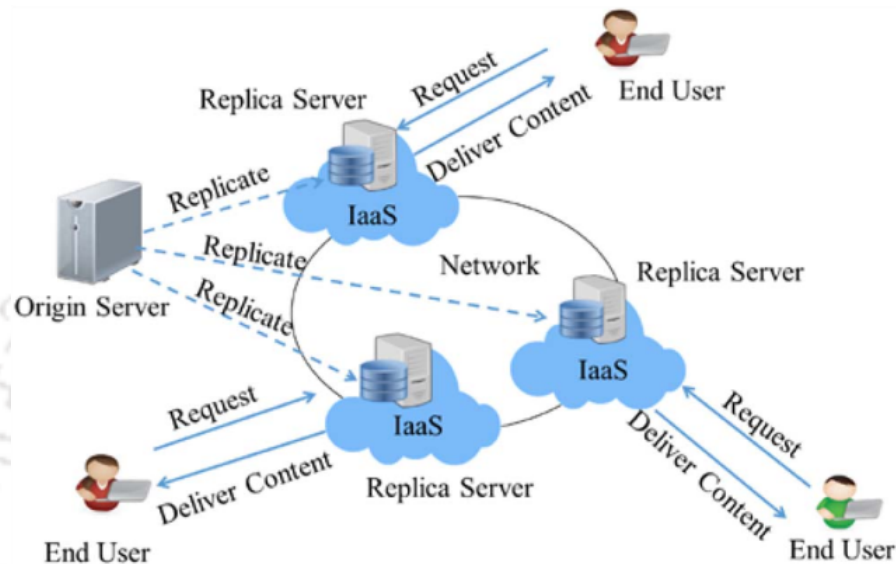


Fig. 2.7 CDN architecture (this image is reproduced from [6])

In the traditional client-server paradigm, a server plays the role of the content provider, where all the clients contact the server and request content. As the number of internet users exploded, a single server became inadequate to serve a large number of clients simultaneously across the globe. Moreover, the centralized server approach has the risk of single-point failure due to hardware failure, link failure, DOS attacks, flash crowds, etc. Nowadays, almost all the major content providers have migrated from the client-server paradigm to the CDN paradigm. CDN takes the responsibility of content delivery/distribution to the end users while maintaining the QoS, minimizing latency, and increasing content availability. As CDN replicates the content in replica servers near the end user's location and serves a local copy of the original content from the replica, the origin server and the core network load are reduced. In return, CDN charges from the content provider for this service. Content originators have the goals of increasing scalability, maximizing QoS, and minimizing bandwidth cost, whereas CDN providers' goals include minimizing infrastructure cost, maximizing replica server utilization, and minimizing content distribution and cache updation cost. To achieve the above-mentioned goals, content providers and CDN providers jointly decide on the design aspects of CDN functionalities which are mentioned below.

2.3.1 Replica server placement

Traditional CDN providers deploy replica servers in strategic locations across the globe and create overlaid networks among them. Nowadays, new-age CDN providers deploy replica servers on clouds. CDN provider's replica placement decision is based on some parameters as enlisted below:

1. *Deployment cost*: The cost to install/deploy a replica server [75]. It is part of the capital expenditure or CAPEX of a CDN provider.
2. *Bandwidth cost*: The cost of transferring data through a network for which a CDN provider has to pay to the *Internet Service Provider* (ISP) [76]. It is part of the operational expenditure or OPEX of a CDN provider.
3. *User perceived latency*: The latency experienced by end users while receiving the content [77].
4. *Available bandwidth*: The available bandwidth provided by an ISP for different links.
5. *Server capacity*: The storage and computation limitations of a given replica server.

CDN providers solve the server placement challenges usually by formulating an optimization problem, putting either the deployment cost or the bandwidth cost, or both of them, as cost functions, and putting other parameters as constraints. A few examples of such optimization formulations are given below:

1. *Facility location problem*: A subset of all the available server locations are chosen to place the replica servers, and each end user are connected to one of the servers so that the total cost of installation of servers together with the delivery of data to each end users can be minimized [78] [79]. It has two variants which are capacitated and un-capacitated. In the capacitated version, the capacity of a given server is put into the constraint so that the assigned load doesn't exceed the server capacity. This problem can be formulated as an ILPP and solved accordingly.
2. *Minimum K-Center*: K servers are placed such that the maximum delivery cost to a user from its nearest server is minimized [80]. This problem can also be represented as an ILPP.
3. *K-Cache location*: It is the optimization problem of placing K servers so that the total delivery cost is minimized [81]. This is also an ILPP problem.

As the above-mentioned problem formulations turn out to be NP-Hard problems, different heuristics and approximation algorithms have been proposed by various researchers. The proposed design algorithms are influenced by the network creation approaches, as discussed next.

- *Traditional approach:* In traditional approaches, overlaid networks are static, where any subsequent changes are time-consuming and costly. Historical/statistical data is used to guess the user content demand patterns while choosing a location for replica installation. In traditional approaches, the installation costs or deployment costs are the prevalent ones.
- *Cloud based approach:* *Cloud-based CDN (CCDN)* providers [82] avoid installing physical servers by taking *Virtual Machines (VM)* in lease from cloud service providers. CDN applications run on these VMs where cloud service providers earn revenues from *Infrastructure as Service (IaaS)* model. This makes CDN deployment dynamic and responsive to end-user content demand changes. VMs can be added or removed from a CDN on the go making the deployment cost model dynamic. The cost of leasing VMs is usually very less compared to fixed proprietary infrastructures. The deployment time also gets reduced drastically. In this scenario, the OPEX cost and the bandwidth cost parameters are dominant in the server placement problem formulation.
- *NFV based approach:* The *network function virtualization (NFV)* is the new age networking paradigm where an instance of a network device along with all functionalities can be created in software, and can be deployed on a general purpose hardware. In a NFV based CDN [83], a CDN provider ties up with ISPs so that an instance of a replica server can be installed/uninstalled dynamically inside any NFV enabled router/switch across the ISP. This approach is more agile than CCDN in the sense that installing/uninstalling NFV can be done faster than deploying/releasing VMs. Also, in the case of NFV CDN, the location of the replicas can be distributed across ISP instead of getting clustered among data centers. CDN providers can run an online optimization problem depending on the end users' dynamic traffic demands.

2.3.2 Content placement

Given a set of replica servers, content placement deals which content to be placed in which replica server. Firstly, it is to be decided which content to be outsourced to the CDN by the content provider. A policy needs to be designed by the CDN provider on how its replica servers may cooperate among themselves while fetching contents and updating their respective cache to maintain consistency. The above-mentioned aspects of CDN design are discussed subsequently.

2.3.2.1 Outsourcing policies

Content outsourcing is the design aspect where the content owner decides which portions of contents are to be outsourced to the CDN and which portions are to be served by itself. Depending on the nature of contents, the design choice falls into two categories as follows.

- *Full site replication*: If the content is static, the entire content can be replicated to replica servers of CDN as the updation cost and consistency issue is negligible. For example, on-demand audios, on-demand videos, images, etc.
- *Partial site replication*: If the content is dynamic or is customized depending on the user profile, having copies of these contents in multiple replica servers is a costly affair in terms of updation and consistency maintenance. A content provider may prefer to serve them from its own data center or from a central server. Examples of this kind of contents are user account information, personalized third-party advertisements, customized content suggestions based on user activity history, etc. If a content provider hosts a website, the static embedded objects like static texts, images, and videos of the page are outsourced to CDN, whereas the customized or dynamic portion of the page is maintained by the content provider. Partial site replication outsourcing policies can be further sub-categorized into the following ways.
 1. *Empirical based*: Content owner empirically or heuristically decides which contents to outsource.
 2. *popularity based*: Based on content popularity, the content owner may decide that highly popular content to be outsourced, whereas infrequently requested contents to be remained with the content owner [84].
 3. *Correlation based*: Based on user activity history, geographical trend, URL connections, etc., contents are correlated and are clustered into groups, and a decision is made for each group as a unit whether to outsource it or not [85].

2.3.2.2 Placement policies

Content placement policies can be broadly classified into two categories as pull based (reactive) and push based (proactive), as described below.

- *Pull based or Reactive*: In this policy, an end user's request is first redirected to a replica server. If the requested content is available in the cache, it is served to the end user. In the case of a cache miss, the replica server needs to fetch a copy of the content before serving it to the end user. Initially, as the cache of the replica server is empty, all user requests result in a cache miss, and the replica continues to fetch the contents till its cache is full. At this time, it needs to have a cache replacement strategy. While the *least recently used* (LRU) is the most simple replacement policy, some use utility-based replacement policies where a replica server tries to keep the contents with more utilities depending on the demand or popularity of the content. In case of a cache miss, content fetching policies can also be of two types as non-cooperative and cooperative.
 - *Non-cooperative*: In this case, if a cache miss happens, a replica fetches the content directly from the content provider's origin server [86]. The pros of this scheme are its simplicity and no content catalog needs to be maintained anywhere.

Con is that fetching large content from the origin server is costly, while a nearby replica may have a copy of the same content.

- *Cooperative*: In this scheme, in case of a cache miss, a replica first tries to fetch the content from any nearby replica server if possible [87]. If none of the nearby replica servers has the required content, it is eventually fetched from the content provider. Pros of this scheme are that a huge amount of bandwidth is saved as the content is fetched locally. Con is that a content catalog has to be maintained either locally or distributively (incurs update traffic overhead) to be aware of the content locations among replicas. Alternatively, query traffic about the content needs to be flooded and wait for a reply from any replica.
- *Push based or Proactive*: In this case, contents are pushed into replica servers strategically and proactively before the end user request is redirected to replica servers. Usually, one central entity, based on statistical data, predicts near-future content demands for each area served by replicas and pushes contents to replicas accordingly [88].

2.3.2.3 Consistency maintaining policies

As multiple copies of a single content are distributed among many replica servers, a change in content in the origin server needs to be reflected in each of the copies maintained by replica servers to provide a consistent content service to end users. The consistency can be maintained in one of the following ways.

1. *Periodic update*: Replicas periodically query for the updates to the content originator and fetch the content if the cache has become stale [89]. This policy can suffer from unnecessary periodic traffic when the content is not changed.
2. *Update propagation*: It is triggered with a change in the content in the origin server. It then pushes the content to every replica server of the CDN having that content.
3. *Cache expiration*: If a content's freshness is expired after a predefined duration, replicas stop providing the content to the end users and try to get a fresh copy either from origin or peer replica servers [90].

2.3.3 Request routing

End users are initially unaware of the existence of any underlying CDN and attempt to contact the prior known origin server for content. It is the request routing mechanism by which the end user's initial request is redirected to a suitable replica server from which the

end user receives the requested content. Request routing techniques heavily depend on the outsourcing techniques of the content provider (described in Section 2.3.2.1) and the content placement policies of the CDN provider (described in Section 2.3.2.2). A few of the request routing techniques is described below.

2.3.3.1 DNS based redirection

This method is used in the full site replication scheme (described in Section 2.3.2.1). When the local *Domain Name Service* (DNS) server of the end-user queries the IP of the domain name of the content provider, the DNS server of the content provider redirects the request to the CDN provider's DNS server [91]. Once the request reaches the CDN provider's DNS server, it can do one of the following two things.

- It can give a list of IPs to the end user's local DNS server. Upon receiving the list, local DNS sends a probe to each of the IPs and checks the response time, and selects the IP with the least latency.
- Depending on the user's local DNS IP address, the CDN provider's DNS server maps the request to an appropriate serving replica and returns its IP. This mapping is an optimization decision based on either minimizing the end user's response latency or based on the cost function, like minimizing the data traffic. Constraints are basically bandwidth limitations, server storage capacity, server computation capacity, etc. This mapping can also be hierarchical. For example, in CCDN architecture, CDN primary DNS server may redirect a request to a data center DNS of a specific cloud site, and the data center DNS server may further redirect the request to an appropriate replica server.

2.3.3.2 HTTP redirection

Here the content provider sends a redirection message in the HTTP header, and the client initiates a fresh connection with the intended replica server.

2.3.3.3 URL rewriting

This method is used in a partial site replication scheme (described in subsection 2.3.2.1). The embedded objects's url is dynamically rewritten in the origin server to the appropriate replica server.

2.3.3.4 Anycasting

In this method, the same IP is assigned to a set of hosts of different locations. The gateway routers of ISPs get multiple path advertisements from other ISPs' gateways for the same IP. The routing algorithm of the gateway simply thinks of them as multiple paths to reach that IP, and depending on its own metrics, it automatically chooses and stores the shortest one [92].

2.3.4 CDNs for IoCVs

Though CDN techniques are quite mature for wired static networks, various design aspects of CDN need to be redesigned prior to use in dynamic environments of IoCVs. A CDN provider can choose to select static infrastructure nodes or vehicles as replica servers. Vehicles are equipped with enough storage and processing power making them fit to be used as replicas. As these replicas are part of IoCVs, they are often termed as edges. Replica/edge selections for IoCVs hugely differ from replica/edge selections for wired scenarios. As the vehicular topology is highly dynamic, replica placement algorithms need to consider vehicles' direction, speed, neighbor density, contact duration with infrastructure nodes, current load, etc. In IoCVs, user demands change much more rapidly compared to the wired scenario. As the local content demand changes very rapidly with the change of vehicular network topology, strategic algorithms need to be designed for caching policies to meet the variable demand. Due to the small dwelling time, a vehicle may not acquire entire content from a single infrastructure node requiring collaborations among replica servers. Unlike wired CDNs, IoCVs have a shared wireless medium for communications. Therefore, efficient scheduling and cooperation among vehicles are needed to fetch content. IoCVs use heterogeneous network access technologies which differ in costs, data rate, and availability, requiring algorithms to be devised to optimize the network access. A few examples of replica/edge vehicle selection are discussed here. The work of Hassan et al. [93] is on static edge placement, which aims to minimize deployment cost. Zhuang et al. [94] discussed various aspects of edge caching and edge computing in IoV using *Software Defined Networking* (SDN) and NFV.

Replica selection methods in IoCVs can be broadly classified into centralized and distributed methods. In the centralized approach, a single centralized entity (e.g., RSU, access point, or server) is responsible for deciding which infrastructure nodes and vehicles serve as replicas. The centralized entity must have high memory, processing capabilities, and a high power source. For example, [95] has solved K-hop dominating set to choose a minimum

number of vehicle replicas for each content chunk. The work [96] selects vehicles and access points as replicas to cache chunks of content with the objective of maximizing the data transfer amount. The work [95] has solved K-hop dominating set to choose a minimum number of vehicle replicas for each content chunk. The work [97] places contents into RSUs based on popularity and demand to maximize content availability. Sara et al.[98] proposed a centralized edge assignment algorithm with the objective of load balancing of edges for generic Internet access. They formulated the problem as multi-objective optimization, where one objective tries to maximize the number of connected vehicles while the other objective tries to minimize the total traffic handled by individual edges. Ghayet et al.[99] proposed a centralized fuzzy-based QoS balancing method to reduce LTE network access cost for the purpose of providing generic Internet access. Received Signal Strength (RSS), current cluster head load, candidate edges, and V2V link duration are selected as fuzzy criteria. In distributed replica selections, replicas are selected distributively by RSUs/vehicles using local information. For example, in the work [100], depending on its capacity and workload, a replica independently decides whether to retain a content or to delegate to a peer node. In the work [101], the replicas are distributively selected based on high coverage area as more peer nodes can be served. In the work [102], vehicle velocity, number of replicas in the vicinity, and communication range are taken into consideration for replica selection.

The replica/edge selections for video streaming applications need special attention as continuous and latency-constraint delivery of content is required from replicas to every client vehicle. The algorithm design heavily depends on the objective of the streaming service, like minimizing service cost, maximizing user experience, etc. The algorithms also depend on the type of streaming services. For example, pre-fetching and caching in replicas can be done for on-demand streaming, whereas these options are not available for live streaming. A comprehensive survey on edge selection for video streaming is done by Jiang et al. [103]. The works have used either static infrastructure nodes or mobile vehicles as edges. A few of the works are described here. The work [25] proposes caching strategy using static BSs as edges and jointly optimizes the objectives of user-experienced delay and service cost minimization. The work [29] uses both BSs and vehicles as edge cache for maximum possible cellular traffic offloading. In [26], static RSUs are used as edges to maximize revenue for RSUs and relay vehicles for content caching. The work [27] uses RSUs as edge caches to maximize cellular traffic offloading while minimizing the number of video stalls.

2.4 Summary

In this chapter, the technical aspects of the relevant research areas of this dissertation have been discussed. The discussion started with an overview of IOCV, followed by IoCV communication stacks as defined by various standards and various routing protocols of IoCVs to distribute safety and non-safety application messages. As video content is prevalent in most infotainment applications, this chapter has introduced the distinguished features of video content, video transfer methodologies over networks, and the required adaptations for video distribution over IoCVs. As most video content originators take CDN services, this chapter has introduced CDN services in detail along with their design aspects. The challenges and applicability of CDN services over IoCVs have been discussed along with a few state of art solutions.





Chapter 3

Centralized optimum edge vehicle selection for live video streaming over IoCV

Video services are prevalent among all infotainment services over IoCVs. To meet the demands of high-quality and low-latency videos, content originators use the services of CDN providers. The objectives of CDN providers include reducing the load of content originators, reducing core Internet bandwidth requirements, and reducing service costs. CDN providers deploy replica servers near IoCVs to serve the demands locally. Contents are cached into replica servers based on popularity. However, for live video streaming, contents are not available beforehand and the demands are made simultaneously by several vehicles. As a result, a replica server may get overwhelmed to handle numerous simultaneous flows. Moreover, Internet bandwidth is wasted carrying identical content. To avoid the above-mentioned issues, CDN providers use edge nodes to reduce the number of simultaneous flows to replica servers. Edges fetch the content from the replica server and distribute it among vehicles. In the context of IoCV, either static infrastructure nodes or a few of the vehicles can be used as edge nodes. For fast-moving vehicles, cellular BS as infrastructure nodes can provide uninterrupted Internet connectivity because of their large coverage areas. However, to use BSs as edge nodes, a CDN provider needs to involve in partnerships with various carrier enterprises, which increases the operating costs for CDN providers. Moreover, accessing cellular networks is not free for vehicles which in turn increases the per-vehicle cost of availing the infotainment service. On the other hand, no partnership cost is involved in selecting vehicles as edge nodes. The objective of a CDN provider is to select as few edge

vehicles as possible to reduce Internet bandwidth usage and the number of simultaneous cellular connections. However, a less number of edge vehicles increases the average hop-counts of multi-hop V2V paths resulting in increased delivery delay and more packet loss due to dynamic connectivity among vehicles and lossy DSRC medium. Accordingly, the CDN providers need to select edge vehicles optimally such that the service cost is reduced without compromising service quality.

Though there are some existing works on using static infrastructure nodes as edges, limited works are found on edge vehicle selection, and very few among them are designed for streaming services. Some of the existing works minimize the number of edges, while some works focus on enhancing end users' QoE. However, no work is found to have the objective of delay-constrained cost-effective live streaming service over IoCV with vehicles being used as edges. Motivated by the above facts, this work proposes a solution for CDN providers to reduce service costs for live video streaming over IoCV with an optimum number of edge vehicle selection.

The contributions of this work are given as follows.

- This work is a pioneer which has the objective of a minimum number of edge vehicle selection with constrained delivery delay for live video streaming over IoCVs.
- Unlike existing works, this work considers the scenario where client vehicles (i.e., running a streaming application) may be connected by multi-hop paths that have ordinary vehicles (i.e., are not running any streaming application) as intermediate nodes.
- The problem of minimum edge vehicle selection with constrained delivery delay is formulated as the well-known MSC problem and is solved by the centralized greedy approximation method. To the best of knowledge, this is the first time the edge vehicle minimization problem is solved as an MSC problem.

The rest of this chapter is organized as follows. In Section 3.1, a brief literature survey on edge selection is presented. In Section 3.2, a detailed description of the proposed solution and its methodology is given. Section 3.3 gives the theoretical analysis of the proposed solution. The performance analysis and simulation results comparison are made in Section 3.4. Section 3.5 summarizes the contributions of this chapter.

3.1 Related Work

Edge selection in vehicular networks is an active research area with widely varied objectives. For an IoCV, edges can be static infrastructure nodes or mobile vehicles. Edge vehicles are also termed as gateway vehicles in literature as peer vehicles are connected to the Internet through them. Few existing works on edge/gateway minimization and LTE cost minimization are mentioned here. Abderrahim et al.[23] proposed a distributed method to minimize the number of edges for generic Internet access. Aggregation of UE signaling is the objective for which Ahmed et al.[104] formulated the problem as centralized multi-objective binary programming to select minimal edge nodes. Xiaoyu et al. [24] proposed a SDN assisted method to reduce LTE network cost to send in-car generated data to the remote sink. Seyhan et al.[22] proposed a distributed algorithm for efficient safety message dissemination application with a minimum number of edges. For route planning and driving assistance applications in heterogeneous vehicular networks, Iftikhar et al.[20] proposed a centralized game theoretic algorithm to minimize the number of edges. For live video streaming applications, Elias et al.[21] proposed a centralized scheme that maximizes the QoE for onboard users. Hamza Khan et al. [105] proposed an algorithm where network-wide QoE maximization is solved using the Lyapunov drift plus penalty method. In both of the above-mentioned works, network cost is not taken into consideration. Another example of live media streaming application is the work of Nadine et al.[106], where the objective is to minimize the number of edges by solving a binary programming problem. However, this work is for handheld UE devices and is not targeted for vehicular networks.

In the existing literature, no work is found with the objective of minimizing the edge vehicles with constrained delivery latency for live video streaming.

3.2 Proposed Solution

This section describes the proposed solution in detail. Section 3.2.1 describes the system model. Protocol overview is described in Section 3.2.2. The edge selection, client-edge association, and theoretical performance analysis are done in Section 3.2.3, Section 3.2.4, and Section 3.3, respectively.

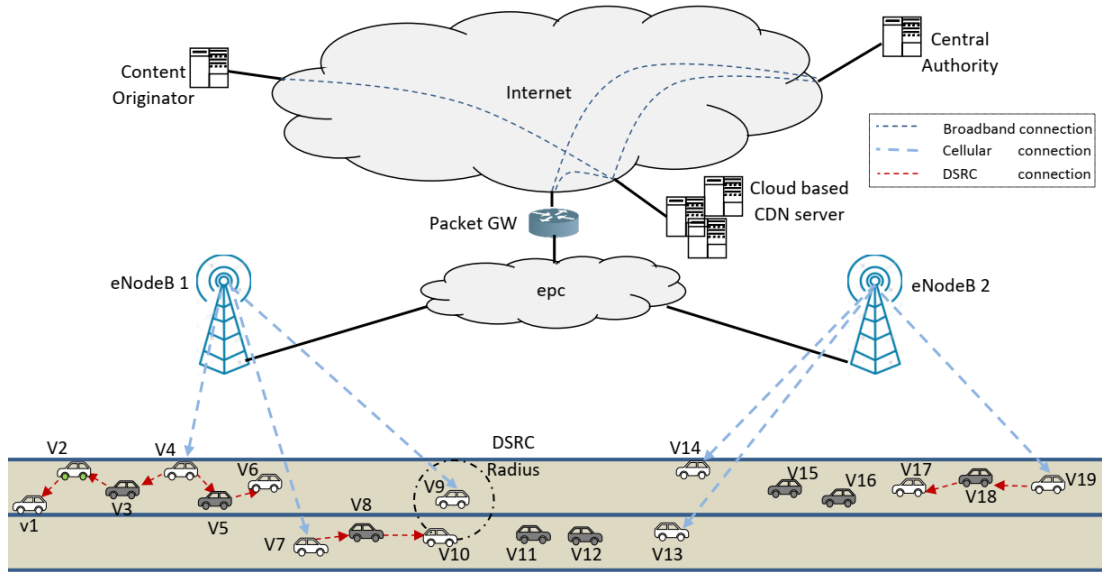


Fig. 3.1 scenario of video streaming over IoV

3.2.1 System model

In the proposed system model, a CDN provider is responsible for providing infotainment services over an IoCV within an AOI. The goal of the CDN provider is to minimize streaming service costs while providing good quality service. CDN provider deploys a replica server using a cloud service that is geographically near the AOI. The CDN replica server fetches the content from the content originator. All vehicles are multi-RAT interface equipped where the DSRC network is used for V2V communication, and the cellular network is used for V2I communication. Few vehicles become clients of the CDN provider to avail of a live video streaming service and run a streaming client application. These vehicles are termed as c-nodes (colored in white in Figure 3.2.1) in this work. Other vehicles are ordinary vehicles not interested in streaming content. The ordinary vehicles are termed as r-nodes (colored in black in Figure 3.2.1) in this work. All vehicles are capable of providing basic routing services. Clearly, c-nodes have all the capabilities of r-nodes, whereas the converse is not true.

All nodes run a beaconing application that exchanges *Basic Safety Messages* (BSM) to gather knowledge about the neighborhood. The BSM application runs on a dedicated safety channel (for example, channel number 172) [1]. In a beacon message, a vehicle includes its information which includes its ID, current position, direction, and velocity. This work assumes that a receiver vehicle appends the experienced SNR value in a beacon message once receiving it. Vehicles create a list of neighbor information (NBR_TABLE) from the

received BSMs and send it to the *Central Authority* (CA) (refer Figure 3.1) periodically, using cellular network [107]. The CA builds the global network topology graph G (bi-directional) from all the received NBR_TABLE by assigning SNR values as edge weights. The CA uses G for various tracking and data mining purposes of IoCV, and may export G to licensed applications when requested. In this work's system model, the CDN server application is one such application. In this work, the words 'vehicle' and 'node' are used interchangeably. Also, without any ambiguity, the phrase 'CDN server' implies 'CDN replica server'.

3.2.2 Protocol overview

In the proposed solution, edge selection is done periodically to mitigate the dynamic connectivity of vehicles. The interval between two edge selections is termed an "epoch" in this work. The epoch duration t_{epoch} is chosen suitably by the CDN server. At the beginning of every epoch, the CDN server selects an optimum number of edge vehicles among the c-nodes considering the delay constraints of the live streaming. Other c-nodes associate themselves with one of the selected edge vehicles by establishing multi-hop paths. As a result, the vehicular network topology is logically divided into number of disjoint groups or clusters. Within a cluster, an edge node acts as the cluster head and the c-nodes associated with that edge becomes members of that cluster. In a given network connectivity graph of vehicles, the problem is to select the optimum number of edge vehicles such that all c-nodes receive streaming data within a constrained delay.

In a multi-hop path from an edge to a c-node, if the relative velocity between any two consecutive hop vehicles is high, that path is likely to be broken before an epoch ends, causing disruption in the streaming service. When vehicles are moving freely on highways, only the vehicles in the same lane (same direction) are expected to collaborate. In case of a congested vehicular scenario, vehicles move slowly allowing V2V collaboration among different lanes (opposite directions). The CDN server needs to decide the value of relative velocity threshold ϑ such that no multi-hop path collaboration happens where two consecutive hop vehicles have relative velocity more than ϑ . The upper bound of relative velocity threshold ϑ can be determined either empirically or with the statistical information of vehicle velocities within an AOI [108]. Smaller ϑ decreases the degree of V2V collaboration resulting in an increased number of edge vehicles (implies increased streaming cost) with the benefit of increased link stability. With the decided value of ϑ , the upper bound of epoch duration t_{epoch} is calculated such that relative movement between any two consecutive hop vehicles is less than the DSRC communication radius R during an epoch. Based on the performance,

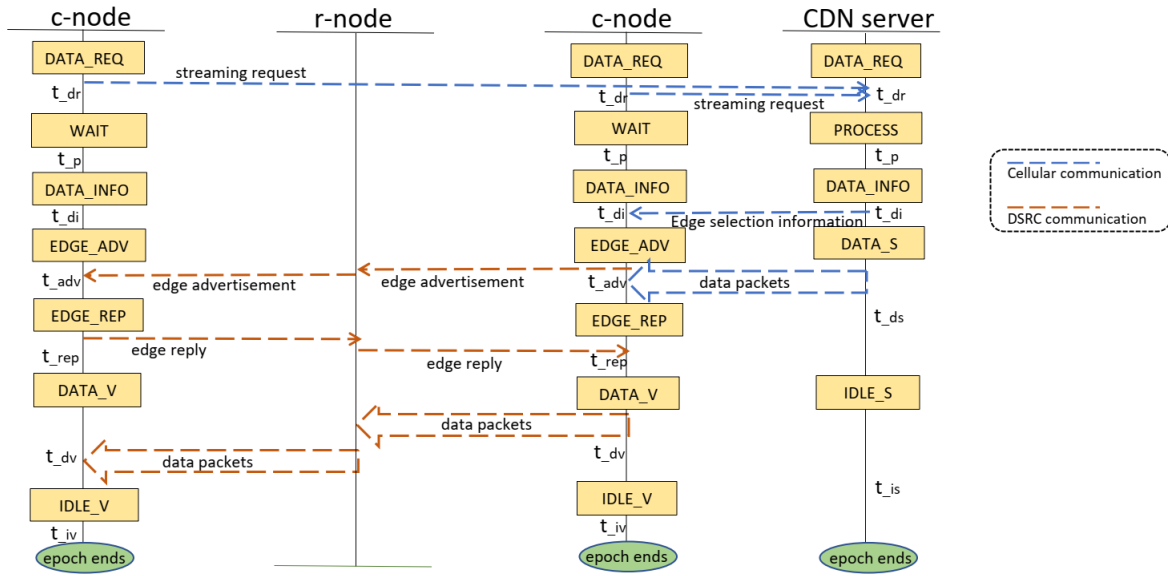


Fig. 3.2 Timeline sequence of phases and packet transmissions for a simple topology consisting of two c-nodes connected through a r-node. Without maintaining any application state, a r-node can help two c-nodes to share content

the CDN server can decide different values of ϑ and t_{epoch} dynamically. The video stream is divided into segments of size β , which need to be distributed within one epoch. If each video data packet carries a payload of size P , then the number of data packets N which needs to be distributed within one epoch is given by $N = \beta/P$.

The CDN server application instance (running on the server), and each of the CDN client application instances (running on individual c-nodes) go through multiple synchronized phases in each epoch. When a client application starts, it gets synchronized with the server application by querying for the remaining time to begin the next epoch, the duration of the phases. The timeline sequence of the proposed solution is given in Figure 3.2. At the start of an epoch, all clients and the server are in *DATA_REQ* phase, and stay in that phase for a duration of t_{dr} . In this phase, clients send a streaming request message to the CDN server, in which it includes its node ID and the current cellular link SNR value (derived from RSS). This message also works as a keep-alive message from clients by which the CDN server assumes that a vehicle is interested in receiving data during the current epoch. With these request messages, the CDN server creates a data structure that associates a vehicle's ID with its IP address (received from the message IP header) and its cellular link SNR value. During this phase, along with gathering information about c-nodes, the server also fetches G from the CA over the Internet. The CDN server sends a relative velocity threshold ϑ along with the request message to the CA. Receiving the request from the CDN server, the CA creates a customized

G where two neighbor vehicles with relative velocity more than ϑ are disconnected by the CA by deleting the edge between them. After the *DATA_REQ* phase, clients enter into the *WAIT* phase, and the server switches to the *PROCESS* phase. The duration of both *WAIT* and *PROCESS* phases is t_p . In *PROCESS* phase, with the help of customized G and the information received from c-nodes, the CDN server selects an optimum number of c-nodes as edge vehicles. The CDN server needs to finish all the required computations for edge vehicle selection within a predefined t_p duration. For this purpose, computation resources of the cloud-based CDN server may be dynamically added/removed by cloud service depending on the size of G . The detailed methodology for edge selection is given in Section 3.2.3. Once the *WAIT/PROCESS* phase is over, all clients and the CDN server switch to *DATA_INFO* phase for a small duration t_{di} when the server informs selected edge vehicles about their selection as edges. The next phase for the server is *DATA_S*, and clients enter into *EDGE_ADV* phase. The duration of *DATA_S* phase is t_{ds} and the duration of *EDGE_ADV* is $t_{adv} = t_{ds}/2$. In *DATA_S* phase, the server pushes one segment of video data to all selected edge vehicles via the cellular interface. At the same time, being in *EDGE_ADV* phase, each edge node advertises its presence to other peer vehicles. Once *EDGE_ADV* phase is over, clients enter into *EDGE_REP* phase of duration $t_{rep} = t_{ds}/2$. In *EDGE_REP* phase, non-edge c-nodes initiate reply messages towards edges. Intermediate r-nodes may help in forwarding both the advertise and reply messages, as shown in Figure 3.2. By the end of *EDGE_REP* phase, all c-nodes are associated with one of the selected edge vehicles. By the end of *DATA_S* phase, the server switches to *IDLE_S* phase and remains idle till the beginning of the next epoch. Meanwhile, *DATA_V* phase starts for all c-nodes and continues for t_{dv} duration. In this phase, edge vehicles distribute the received video segment to other c-nodes as described in Section 3.2.5. Few r-nodes may help in distributing data packets. At the end of *DATA_V* phase, all c-nodes go to *IDLE_V* phase. In *IDLE_V* phase c-nodes sort and assemble the received data packets with sequence numbers before pushing the data to the media player buffer. Duration of *IDLE_V* phase is t_{idle_v} . Section 3.4.1 describes how to decide the phase duration values.

The media player is active throughout all the time a vehicle runs the client application. In t_{epoch} duration, a c-node fetches one segment of video data, which is made available to its media player. As a result, the startup latency of playback is t_{epoch} for any c-node. The fetched video segment is played for the entire next epoch duration t_{epoch} , by which time the c-node fetches the next segment of video data.

3.2.3 Edge selection

The CDN server acquires G from the CA in the form of an adjacency list representation, where an edge weight represents the SNR value of the wireless link between two nodes. Let us assume, at the beginning of an epoch, vehicles present in the scenario are represented by the set V . There are $|V|$ vehicles, and they are indexed by their ID starting from 0 to $|V| - 1$. Let's assume, within these $|V|$ vehicles, only a few vehicles are c-nodes which form the set $C = \{c_i \mid c_i \in V, c_i \text{ is a c-node}\}$. Let us assume a function F such that when a vehicle instance is given as input, it returns the ID of that vehicle. Its inverse function F' returns the vehicle from a valid ID. The CDN server does some pre-processing on G , which are mentioned in the following pre-processing steps:

Pre-processing steps:

1. The server converts G from adjacency list representation to adjacency matrix representation, where the missing edges are assigned SNR value 0, and diagonal elements (representing self-loop edge weight) are assigned by a very high SNR value defined as INF .
2. For each edge e , the wireless link capacity Υ_e is estimated by Shanon's formula: $\Upsilon_e = B \times \log_2(1 + SNR)$ where B is the channel bandwidth.
3. Let the overhead of the header fields of all layers for each frame be H Bytes. H (88 Bytes) is composed of UDP header (8 Bytes), IPv6 header (40 Bytes), 802.11p MAC header (36 Bytes), and frame check sequence (4 Bytes). As one segment of video data is transmitted into N data packets, the total header overhead for the entire segment is $N \times H$. For each edge e , the delay W_e to transmit one segment of video data of size β is calculated as:

$$W_e = \begin{cases} (\beta + (N \times H)) / \Upsilon_e, & \text{when } \Upsilon_e \neq 0 \\ INF, & \text{when } \Upsilon_e \approx 0 \end{cases} \quad (3.1)$$

For diagonal elements, $W_e \approx 0$, which implies data can be transferred from a node to itself in no time. All edge weights are replaced by their respective W_e .

4. On the resulting graph, Floyd-Warshall [30] algorithm is run to get all-pairs shortest path graph. The output graph is represented as matrix format referred as delay matrix D' , where the element $d'_{i,j}$ denotes the minimum multi-hop path delay from the vehicle with ID i to the vehicle with ID j to transmit a video segment of size β . A cost matrix D is created from D' by initializing all elements $d_{i,j} = d'_{i,j}$.
5. For all $c_i \in C$, the CDN server checks if the stored cellular SNR value for c_i is less than a predefined threshold value SNR_t . If yes, the server considers c_i as unfit to become an edge. This is because if the node receives partial data from the server due to poor

cellular connection, it has a cascading effect on all nodes in that cluster. So, the server replaces all elements of the $F(c_i)^{th}$ row of D except $d_{F(c_i),F(c_i)}$ with INF . SNR_t value is chosen such that the data rate is sufficient to deliver one segment of video data within the duration of $DATA_S$ phase.

The problem of minimum edge selection and assignment with constrained delay can be formulated as ILPP, as described next. A set of binary elements representing the edge set Δ of size $|C|$ is defined such that $\Delta_i = 1$, if $c_i \in C$ is selected as an edge. Otherwise, $\Delta_i = 0$. An assignment matrix A of size $|C| \times |C|$ is constructed where the element $a_{i,j} = 1$, if c_i is assigned as edge for c_j . Otherwise, $a_{i,j} = 0$. The objective of the ILPP is given below.

Minimize $\sum_{i=1}^{|C|} \Delta_i$ subject to the following constraints.

1. All c-nodes must be served, and should be served by only one edge.
 $\forall j, \sum_{i=1}^{|C|} a_{i,j} = 1$
2. Edges are always assigned to themselves for streaming.
 $\forall i, a_{i,i} = \Delta_i$
3. Data delivery delay from an edge to any c-node must be within $DATA_V$ phase duration t_{dv} , which is termed as threshold delay.
 $\forall c_j \in C, \sum_{i=1}^{|C|} (d_{F(c_i),F(c_j)} \times a_{i,j}) \leq t_{dv}$
4. All the decision variables are binary.
 $\forall i, j, a_{i,j} \in \{0, 1\}$ and $\forall i, \Delta_i \in \{0, 1\}$

Constraints 1 and 2 are for edge assignment, whereas constraint 3 is for edge selection. ILPP is a known NP-hard problem. It can be seen that with $|C|$ requesting vehicles, the number of decision variables is $\mathcal{O}(|C|^2)$. Heuristic algorithms have non-deterministic convergence time making it unfit for delay constrained services. Moreover, the solution may largely deviate from the optimal solution which defies the objective. On the other hand, approximation algorithms have deterministic convergence time with an upper limit of deviation from optimal solution. Accordingly, the ILPP is divided into two subproblems of edge selection and edge assignment. The edge selection subproblem is solved centrally as an MSC problem in the CDN server, and the edge assignment subproblem is handled distributively as described in Section 3.2.4.

Minimum set-covering problem: An instance of minimum set covering or MSC problem consists of a finite set X , and a given set of predefined subsets of X as: $S = \{S_i | S_i \in X\}$ such that the elements of S cover X , i.e., $X = \cup_i S_i$. The problem is to find the minimum size subset of S , which covers X .

Algorithm 1 Reduction to set covering problem**Input:** set of c-nodes C , delay matrix D , threshold delay t_{dv} , cellular SNR threshold SNR_t **Output:** set X , Set S

```

1:  $X \leftarrow C, S = \{\}$ 
2: for all  $x_i$  in  $X$  do
3:    $s_{x_i} = \{\}$ 
4:    $D_{F(x_i)} \leftarrow$  row from matrix  $D$  corresponding to vehicle  $x_i$ 
5:   for all  $d_{F(x_i),j} \in D_{F(x_i)}$  s.t.  $F'(j) \in X$  do
6:     if  $d_{F(x_i),j} < t_{dv}$  then
7:       Insert  $F'(j)$  in  $s_{x_i}$ 
8:     end if
9:   end for
10:  Insert  $s_{x_i}$  in  $S$ 
11: end for

```

The edge selection subproblem is converted to an MSC problem by the method mentioned in Algorithm 1. The algorithm runs in the CDN server which initializes X with C and creates elements s_{x_i} (subsets of X) for S corresponding to each $x_i \in X$ by the following way. In the $F(x_i)^{th}$ row of D , the considered column indexes are the IDs of those vehicles in set X that have delay less than t_{dv} . The vehicles corresponding to those column indexes are inserted in the subset s_{x_i} . An element of S , i.e., s_{x_i} denotes that if the vehicle x_i is selected as edge, all the c-nodes in the subset s_{x_i} can receive streaming video data from x_i within duration t_{dv} . In other words, vehicle x_i covers the vehicles in the subset s_{x_i} . The goal is to select a minimum number of s_{x_i} such that all elements of X are covered. Initially, all elements of set Δ are assigned to 0. As the MSC problem is also a known NP-hard problem, a well-known approximation algorithm named *Greedy Set Cover*(GSC) [30] is used to solve it. GSC has an approximation ratio $(\ln|X| + 1)$ where $|X|$ (i.e., $|C|$) is the cardinality of the set to be covered. The algorithm follows a simple greedy approach. In each iteration, it selects a subset that covers the maximum remaining uncovered elements. The algorithm is modified in this work if a tie occurs among a few candidate subsets. In this case, a candidate subset is selected, which has a minimum cumulative delay for all newly covered elements. From the solution of the MSC problem, all Δ_i are found for which the corresponding subset s_{x_i} is included in the MSC solution. The value of those Δ_i elements is set to 1 to obtain the solution of the edge selection subproblem.

3.2.4 Client-edge association phase

The edge assignment subproblem defined in Section 3.2.3 is solved distributively in the client-edge association phase. Once the CDN server selects edge vehicles, it informs those

edges about their selection during the *DATA_INFO* phase. The server also includes relative velocity threshold ϑ , video segment size β , and threshold delay t_{dv} in the message.

The algorithm is divided into two phases which are *EDGE_ADV* phase and *EDGE_REP* phase. The *EDGE_ADV* phase continues for t_{adv} where all edges initiate the *ADV* message through the DSRC interface. An *ADV* contains the source vehicle ID I_{v_i} , relative velocity threshold ϑ , video segment size β , accumulated delay d_a (0 for edges), threshold delay t_{dv} , and current hop-count k to the source (0 for edges). Each node (c-node or r-node) that receives a *ADV* message, consults its *NBR_TABLE* and does the following. If the node finds the relative velocity between itself and the sending node greater than ϑ , it discards the message. Otherwise, it finds the SNR value for this sender stored in *NBR_TABLE*, calculates expected transmission rate and delay d_e for data size β (refer Section 3.2.3), and discards the message if $(d_e + d_a) > t_{dv}$. In case $(d_e + d_a) \leq t_{dv}$, it stores locally the values of m_{id} , I_{v_i} , increased hop-count k' as $k + 1$, and the address of sender as immediate upstream node. It changes the packet fields d_a and k to $(d_e + d_a)$ and $(k + 1)$ respectively, and re-transmits the *ADV*. Duplicate messages are ignored by checking the tuple $\langle m_{id}, I_{v_i} \rangle$. If a non-duplicate message is not discarded by the above-mentioned conditions, the receiving node checks if the locally stored hop count k' is greater than $(k + 1)$ value from the current message. If true, it indicates that there exists a shorter path to that edge node. In that case, the node replaces all the previously stored values with the new values of the current message and re-transmits *ADV*. Otherwise, the message is ignored. This hop-count comparison is made to restrict flooding of *ADV* in the network.

Following the *EDGE_ADV* phase, the *EDGE_REP* phase continues for the duration of t_{rep} . In the reply phase, a non-edge c-node initiates a unicast message *REP* to its upstream node through the DSRC interface. On receiving a *REP* message, a node (c-node or r-node) makes an entry for the *REP* sender as a downstream node. If the receiving node is a r-node and has received *REP* for the first time, it forwards *REP* to its upstream node. As c-nodes initiate one *REP* message for themselves, they do not need to forward a received *REP*. On receiving any subsequent *REP* messages, a node only creates an entry for the sender in its downstream node list. It does not forward the message further to its upstream node. In this way, request aggregation takes place, which significantly reduces the total bandwidth requirement.

3.2.5 V2V data phase

A video segment is divided into N packets and transmitted with sequence numbers from 1 to N . In the client-edge association phase, edge vehicles receive one segment of video data from the CDN server through the cellular interface. All c-nodes (including edges) enter into *DATA_V* phase. Edge vehicles that receive at least one *REP* message, start distribution of the video data to their immediate downstream nodes through the DSRC interface. The first packet sent is a metadata packet containing the segment index, the number of packets to be sent, and the expected duration of the transmission. On receiving this metadata packet, downstream nodes start a data-receive timer with the expected duration. Subsequent packets are data packets, which contain packet index along with data. Once the data-receive timer is expired, a receiving node waits for a short random time and initiates the distribution of the received packets to its downstream nodes by the same above-mentioned method. Once all the received data packets are delegated, a r-node becomes idle, whereas a c-node check whether it has missed any packet index while receiving from its upstream node. In case a few indexes are missing, it listens for any ongoing transmissions within the communication range in the hope of recovering them. Once the V2V data phase ends, c-nodes enter into *IDLE_V* phase. In this phase, the c-nodes assemble the received data packets by their indexes and push the data to the media player buffer for playback.

3.3 Protocol Analysis

In this section, a thorough analysis of the proposed solution is done. Discussion on protocol correctness is done in Section 3.3.1, Protocol complexity analysis is done in Section 3.3.2, and the performance analysis of the proposed solution is done in Section 3.3.3.

3.3.1 Protocol correctness

The edge selection phase and client-edge association phase of the proposed solution together solve the ILPP formulated in Section 3.2.3. The MSC solution ensures that each c-node is covered by at least one edge. In the client-edge association phase, each c-node gets associated with only one edge. Combining the above two facts, constraint 1 of ILPP is satisfied. In the solution, an edge becomes its own data provider, which satisfies constraint 2. The solution selects a path from a c-node to an edge with accumulated delivery latency less than t_{dv} , and the MSC solution ensures the existence of at least one such path. With the above facts and

having constraint 1 already satisfied, constraint 3 is also satisfied. Pre-processing step 5 avoids the ill-effect of short-term fading experienced for cellular connection. While receiving data from the CDN server, if a vehicle moves from one cellular base station to another, we assume that an application transparent cellular handover mechanism takes place. Because of the above two facts, the chance of packet loss experienced by an edge through the cellular interface is negligible and can be ignored.

3.3.2 Protocol complexity

Steps 1, 2, 3, and 5 of server pre-processing (refer Section 3.2.3) on the topology graph G have a time complexity of $O(|V|^2)$. Step 4 executes the Floyd-Warshall algorithm, which runs in $O(|V|^3)$. Therefore, running time of server pre-processing is $O(|V|^2) + O(|V|^3)$ or $O(|V|^3)$. Both the outer loop and inner loop of the MSC reduction algorithm (Algorithm 1) go through $|C|$ iterations. All other steps of Algorithm 1 are of $O(1)$, which makes the complexity of MSC reduction algorithm $O(|C|^2)$. For the edge selection problem, the well-known GSC approximation algorithm is used, which runs in $O(|C|^3)$ [30] time. Therefore, the overall running complexity of the centralized edge selection algorithm in the CDN server is given by $O(|V|^3) + O(|C|^2) + O(|C|^3)$. As $|C| < |V|$, the running complexity is simplified to $O(|V|^3)$. In a real scenario, the neighbor set of any given vehicle is significantly small subset of total vehicles present in topology. As a result, the vehicle network connectivity graph is represented by a sparse matrix. Therefore, more efficient algorithms like Johnson's algorithm [109] can be used instead of Floyd-Warshall. Moreover, in the study [19], authors found that vehicular networks are very much fragmented into smaller networks. The probability of a large connected network is extremely low. Therefore, the edge selection algorithm can be run in parallel for individual small connected networks to reduce the processing time.

3.3.3 Performance analysis

In the V2V data phase, a downstream node starts transmission only when its upstream node has finished transmitting all data packets. As a result, in a one-dimensional highway scenario, data packets propagate hop by hop in one direction as a batch, reducing the packet loss probability. However, during transmission, if an upstream and downstream node moves out of each other's communication range, packet loss occurs. Also, while an upstream node is transmitting, a node can experience packet collision by a neighbor cluster's ongoing transmission.

First, the probability is calculated for the event E_1 denoting a node experiences packet loss due to mobility while receiving data from its upstream node. The one-hop DSRC transmission duration t_β of a segment is $t_\beta = \beta/\Upsilon_w = t_{epoch} \cdot \delta/\Upsilon_w$, where $\Upsilon_w \gg \delta$ is current wireless link rate. In the worst case, a node has the maximum relative velocity ϑ with respect to its upstream node. So in duration t_β , the node moves away $\vartheta \cdot t_\beta$ distance from its upstream node. Let the range of DSRC communication be R , and the downstream node experiences packet loss only if its distance with an upstream node is more than $R - \vartheta \cdot t_\beta$ at the beginning of receiving data. As a downstream node is equally likely to be present anywhere between 0 and R distance from its upstream node when it starts receiving data, the probability $P(E_1)$ that it experiences packet loss due to moving out of communication range is

$$P(E_1) = \vartheta \cdot t_\beta / R = (\vartheta \cdot t_{epoch} \cdot \delta) / (R \cdot \Upsilon_w) \quad (3.2)$$

Now the probability is calculated for the event E_2 denoting a downstream node experiences packet collision from a neighbor cluster's transmitting node, given that it is receiving from its upstream node. Let there be l lanes in a highway topology. Let us assume the average cluster hop length is k , which means a data packet traverses an average k hops from an edge in both directions of a one-dimensional highway scenario. Among k hops towards a single direction within a cluster, the solution allows only one vehicle to transmit at a time. This is because each node transmits at most once, and transmits all the packets in a batch. The phases of all vehicles are synchronous, resulting in all edge nodes entering in DATA_V phase simultaneously and starting data packet sharing. As data segment size is fixed, the number of hops traversed by a batch of data packets within each cluster is ideally the same in a given duration. As a result, the transmission duration of $(i-1)^{th}$ hop upstream nodes of all clusters overlap with each other. The conditions of packet collision in a given time instance on i^{th} hop downstream node of a cluster cl_1 while its upstream node is transmitting are

1. cluster cl_1 has at least one neighbor cluster cl_2 , and
2. $(i-1)^{th}$ hop upstream node of cl_2 is within communication range of i^{th} hop node of cl_1 .

Now, the probability that a receiving node is i^{th} hop node in a cluster is $1/k$. The probability that the node has $(i-1)^{th}$ hop upstream node of a given neighbor cluster in its communication range is also $1/k$. As these two events are independent, the probability of a given node experiencing packet collision from a neighbor cluster is $1/k^2$. Now, as the highway contains l lanes, a vehicle can have a max $l-1$ neighbor clusters. So, the probability that a node does not experience packet loss is given by $(1 - 1/k^2)^{l-1}$. So, when a

node's upstream node is transmitting, the probability $P(E_2)$ that the node experiences packet collision is given by the following equation.

$$P(E_2) = 1 - (1 - 1/k^2)^{l-1} \quad (3.3)$$

It is to be noted that E_1 and E_2 are mutually exclusive events. When event E_1 occurs on a c-node, it rules out the possibility of any collision event E_2 on that node. If E_2 occurs, it implies that the c-node is receiving packets from its upstream node, and therefore, event E_1 cannot occur. So we get the total probability $P(E)$ that a given vehicle experiences packet loss is the sum of two mutually exclusive events ($P(E_1) + P(E_2)$), and the equation is given below:

$$P(E) = \{(\vartheta \cdot t_{epoch} \cdot \delta) / (R \cdot \Upsilon_w)\} + \{1 - (1 - 1/k^2)^{l-1}\} \quad (3.4)$$

R is assumed to be constant, and all DSRC links are experiencing the same Υ_w . In that case, the probability of packet loss increases with an increase in ϑ , t_{epoch} , and δ . It should be noted that $P(E)$ depends on relative velocity ϑ , not on the actual velocity of vehicles. As long as relative mobility is less among vehicles (which is usually true in highway lanes), the proposed solution can tolerate high mobility. The average number of hops k in a cluster has a weak positive correlation with the vehicle density λ of the topology. The reason is, with higher λ , the network is well connected, and k can be increased until the data delivery delay exceeds t_{dv} . It can be observed that the term $P(E_2)$ of equation (3.4) reduces with increasing k . Therefore, $P(E)$ experiences a weakly negative correlation with λ . This may seem counter-intuitive in a wireless scenario. However, it is to be noted that the number of transmitting vehicles does not increase with λ . It is dependent only on k , and the vehicles have a hop-by-hop synchronized sequential transmission schedule.

3.4 Performance Evaluation

This section is divided into the following parts. Section 3.4.1 describes the simulation environment, Section 3.4.2 defines performance metrics, and Section 3.4.3 presents the comparative simulation results.

Table 3.1 Simulation static parameters

Parameter	Description	Value
P	data packet payload size	1024 Bytes
Υ_I	Internet download speed	100 Mbps
Υ_L	LTE download speed	25 Mbps
Υ_D	DSRC channel data rate	6 Mbps
R	communication range of DSRC	50 m
t_{dr}	duration of <i>DATA_REQ</i> phase	50 ms
t_p	duration of <i>PROCESS</i> phase	50 ms
t_{di}	duration of <i>DATA_INFO</i> phase	5 ms
t_{idle_v}	duration of <i>IDLE_V</i> phase	5 ms

3.4.1 Simulation environment

The proposed solution is simulated in NS3 version 3.30.1 [110] where a highway topology is created which is 5 km long and has four lanes, namely l_1 , l_2 , l_3 , and l_4 . Out of the four lanes, vehicles on l_1 and l_2 move from west to east, and vehicles on l_3 and l_4 move from east to west. The velocities of the vehicles are not fixed and depend on topology density. The simulation parameters are given in Table 3.1.

With DRSC communication radius R of 50 meters (100 meters diameter), in 200 vehicles/km density, the estimated average neighbor count of a vehicle in one-dimensional highway scenarios is given by $200/1000 \times 100 = 20$. Let the weighted graph $G = \{V, E, W\}$, where V is the set of vertices, E is the set of edges, and $W : E \rightarrow \mathbb{R}$ is a function which assigns a weight $w_i \in \mathbb{R}$ to any edge $e_i \in E$. The data structure used by adjacency list representation of a weighted graph G is an array $A[|V|]$. $A[v_i]$ represents the vertex $v_i \in V$ by its ID I_{v_i} , along with the list of IDs of the neighbor vertices of v_i as $L(v_i)$. Let v_i have neighbor vertex set NB_i . Each element $l \in L(v_i)$ contains a tuple of two values: the ID I_{v_n} of v_n where $v_n \in NB_i$, and the weight w of the connecting edge from v_i to v_n . When sending this data structure as an offline file, instead of connecting each $l \in L(v_i)$ by the next pointer values, they are put into comma-separated csv format. Therefore, the size of one list element l becomes: $sizeof(l) = sizeof(I_{v_n}) + sizeof(w)$. Accordingly, the size of $L(v_i)$ is $sizeof(L(v_i)) = length(L(v_i)) \times sizeof(l)$. Therefore, the size of one array element $A[i]$ is $sizeof(A[i]) = sizeof(I_{v_i}) + sizeof(L(v_i))$. Accordingly, the size S_G of adjacency list representation of graph G is $S_G = |V| \times sizeof(A[i])$. On substituting all the equations, the following equation is derived.

$$S_G = |V| \times (sizeof(I_{v_i}) + (length(L(v_i)) \times (sizeof(I_{v_n}) + sizeof(w)))) \quad (3.5)$$

Vehicle ID I_{v_i} is represented by 16 bit short integer, and edge metric w is represented by 32 bit real number. In case $|V| = 5000$, $sizeof(I_{v_i}) = sizeof(I_{v_n}) = 16$, $sizeof(w) = 32$, and $length(L(v_i)) = 20$ (average neighbor size as discussed above) into equation (3.5), $S_G = 5000 \times (16 + (20 \times (16 + 32))) \text{ bits} \approx 5 \text{ Mb}$. If Υ_I is 100 Mbps, the download time of G is $5/100 \text{ sec} = 50 \text{ ms}$. Accordingly, the *DATA_REQ* phase duration t_{dr} is set as 50 ms.

The CDN server needs to finish all the required computations within the *PROCESS* phase of t_p duration. Computational resources of the cloud-based CDN server may be dynamically added/removed by cloud service if required depending on the size of G . As mentioned in Section 3.3.2, vehicular networks are highly fragmented with a number of small networks. Therefore, parallel processing for each smaller network can be done for faster convergence. For simulation, the duration t_p of *PROCESS* phase is set as 50 ms. In *DATA_INFO* phase, the CDN server sends one small notification message to clients and phase duration t_{di} is set as 5 ms. The *IDLE_V* phase duration t_{idle_v} is set as 5 ms to allow vehicles to sort and assemble the received data packets, and make them available to the media player.

The λ is varied from 50 vehicles/km to 175 vehicles/km. As given in [108], the differences in average velocity among same-direction vehicles in a free-speed model usually do not exceed 50 km/hr. Therefore, ϑ is varied from 8m/s (29 km/hr) to 15 m/s (54 km/hr) to include all possible V2V collaborations. When ϑ is large (small), the topology graph G is stable for a small (large) duration. Vehicle movements are slow in dense scenarios allowing opposite-direction V2V collaborations. With 15 meters/s relative velocity, the maximum duration for which two vehicles can remain within each other's communication range is given by $\lfloor 50/15 \rfloor = 3$. Therefore, we varied t_{epoch} from 1 sec to 3 secs.

The *DATA_S* duration t_{ds} is divided into two parts. In the first part, data reaches to LTE packet gateway from the CDN server, and in the second part, data is transferred from the LTE packet gateway to edge vehicles. So, the value of t_{ds} is obtained as $(\beta/\Upsilon_I + \beta/\Upsilon_L)$. Finally, the value of t_{dv} (*DATA_V* phase duration) is obtained by:

$$t_{dv} = t_{epoch} - (t_{dr} + t_p + t_{di} + t_{ds} + t_{idle_v}) \quad (3.6)$$

The simulation setup is done in the following steps. Nodes are created for vehicles, the central authority (CA) server, and the streaming server. Two NICs (net-devices of NS3) are installed in each vehicle where one NIC is to access DSRC and the other one is to access LTE network. An LTE instance is created with the "LteHelper" class. Along with many nodes, the helper utility creates one packet gateway node. The vehicle nodes are attached to the enb node with the "LteHelper::Attach()" utility function. The server is connected to the packet

gateway node and the CA node by point-to-point links. The latency and data rate of the link is configured to mimic the Internet data rate and latency as given in the simulation parameter. IPv6 addresses belonging to one prefix are assigned to the LTE interfaces of all vehicles, while IPv6 addresses belonging to another prefix are assigned to the DSRC interfaces. A client application class and server application class is created by deriving the NS3's default "Application" class. The client app is installed in a few of the vehicle nodes to make them c-nodes. The server app is installed on the server. A CA server app is installed in the CA. Installation is done using "Node::AddApplication()" utility function. Both the client and server app maintain the vehicle states and use a number of timers to trigger phase changes and message creations. Message handlers are developed accordingly.

3.4.2 Performance metrics

To measure the performance of the proposed scheme, the following performance metrics are defined.

- **LTE cost effectiveness (*lce*):** In an epoch, it is the ratio of vehicles that can access data without requiring a LTE connection with respect to the total vehicle count. The average ratio is taken over all epochs. As only selected edge nodes require LTE connections, the number of required LTE connections in a given epoch is the same as the selected edge count in that epoch. If the total epoch count is ξ_t , selected edge count for the epoch i is $|S|_i$, and the total vehicle count is $|V|$, the metric *lce* is calculated as given below:

$$lce = \frac{\sum_{i=1}^{\xi_t} \left(\frac{|V| - |S|_i}{|V|} \right) \times 100}{\xi_t} \quad (3.7)$$

- **Packet delivery ratio (*pdr*):** For a vehicle, this is the ratio of the received data packet count in an epoch with respect to the expected data packet count in that epoch. The average of all vehicles in all epochs is taken to get the value of *pdr*. If total packet count is N , total vehicle count is $|V|$, received packet count by vehicle v_i is N_{v_i} , the *pdr* of an epoch (*pdr_e*) is calculated as given as below.

$$pdr_e = \frac{\sum_{i=1}^{|V|} \left(\frac{N_{v_i}}{N} \right) \times 100}{|V|} \quad (3.8)$$

When epoch count is ξ , the metric *pdr* is defined as: $pdr = \frac{\sum pdr_e}{\xi}$

- **Number of video interruptions (*nvi*):** This is the ratio of the number of epochs when a client vehicle is unable to fetch video data with respect to total epoch count. When a vehicle fails to fetch data in an epoch, the media player of that vehicle is interrupted

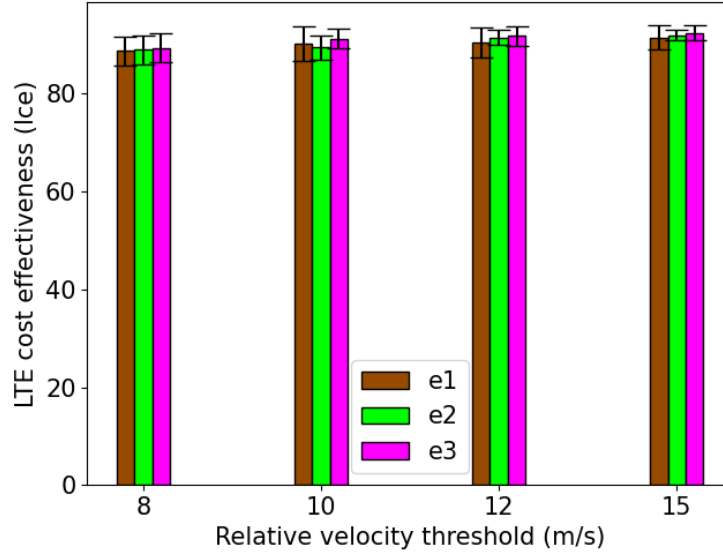


Fig. 3.3 LTE cost effectiveness vs Supported relative velocity

for the next t_{epoch} duration due to buffer underflow. The average of all vehicles is taken to get the value of nvi . If the number of c-nodes is $|C|$, the number of epoch loss faced by i^{th} c-node is ξ_i , and the total epoch count is ξ_t , the metric nvi is calculated as:

$$nvi = \frac{\sum_{i=1}^{|C|} \{(\xi_i / \xi_t) \times 100\}}{|C|} \quad (3.9)$$

- **Peak signal to noise ratio (PSNR):** This is one of the standard techniques for video quality assessment that captures users' satisfaction. The method of calculating $PSNR$ is well-known and can be found in [111]. In the proposed solution, if a frame is lost, it is replaced by the correctly received previous frame. However, this process creates some distortion which is captured by the standard $PSNR$ metric. The average is taken for each frame of each vehicle in all epochs.

In the proposed solution, jitter is consumed by the playback buffer and, therefore, not included as a performance metric. If a packet delay is less than t_{dv} , it is stored in the buffer and sorted to its right place before being consumed by the media player. With a packet delay greater than t_{dv} , the packet is considered lost. So, the packet delay metric is also not included. The startup latency is t_{epoch} as mentioned in Section 3.2.2, and is not included in performance metrics.

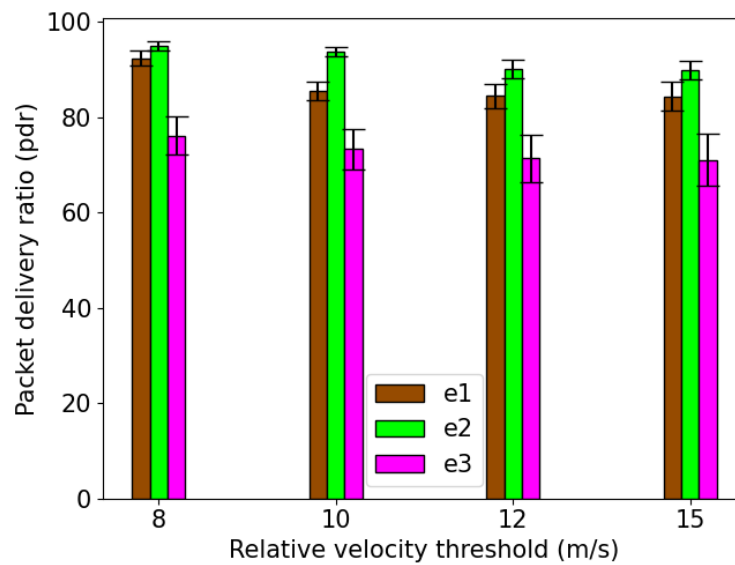


Fig. 3.4 Packet delivery ratio vs Supported relative velocity

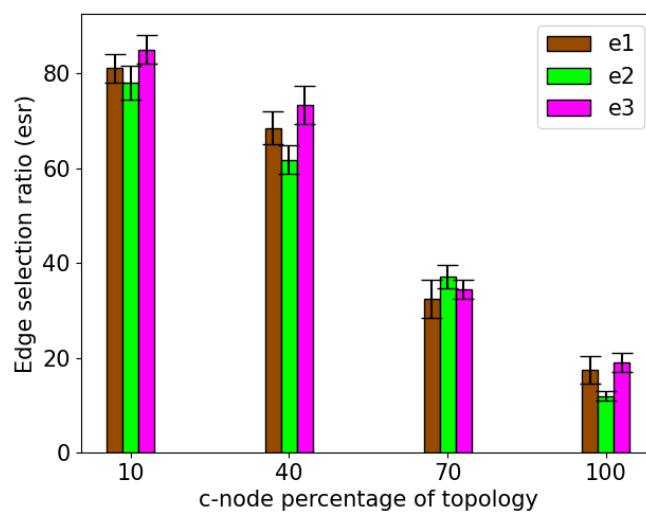


Fig. 3.5 Edge selection ratio (esr) vs c-node ratio

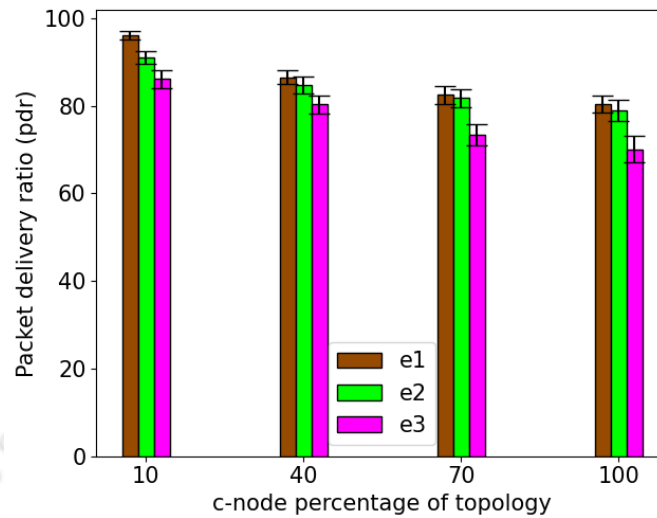


Fig. 3.6 Packet delivery ratio (pdr) vs c-node ratio

3.4.3 Simulation results

Simulation results based on the above-defined metrics are presented here. The performance of the proposed protocol depends on the stability of the vehicle connectivity which in turn depends on the relative movements among vehicles. Epoch duration and relative velocity threshold are the two factors that decide relative movements among vehicles. Therefore, experiments are done to realize the protocol performance sensitivity with respect to varying epoch duration and relative velocity thresholds. The sensitivity is measured for the metrics *lce* and *pdr*, while keeping λ fixed at 100 vehicles/km. The ratio of c-node to r-node is kept as 1:3.

Figure 3.4 and Figure 3.3 show the *pdr* and *lce* metric performances of epoch duration of 1, 2, and 3 secs with varying ϑ . Epoch duration t_{epoch} of 1, 2, and 3 s are abbreviated by e1, e2, and e3 in the legends. Figure 3.3 shows the common trend of increasing *lce* with increasing ϑ . This is expected because, with ϑ increased, less number of edges are deleted by the CA from the topology graph as described in Section 3.2.2. This makes the resultant customized graph denser, increasing the edge selection efficiency and *lce*. It is observed that *lce* metric performance improves with increasing epoch duration. When t_{epoch} increases, t_{dv} also gets increased resulting in an increased average hop count of clusters. This happens because, with the increased time budget t_{dv} , a video segment can traverse more hop distance from an edge. Larger cluster sizes reduce the number of clusters in a network resulting lesser number of edges which in turn improves *lce*. Figure 3.4 shows the effect on *pdr* of different

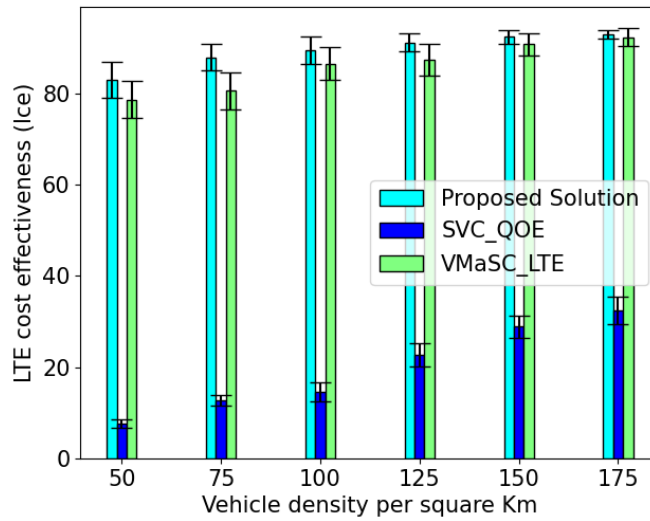


Fig. 3.7 LTE cost effectiveness vs Vehicle density

t_{epoch} values with varying ϑ . In general, pdr decreases with increasing ϑ . This is because, the probability of packet loss increases with an increase in ϑ as can be seen in equation (3.2). It is observed that the performance of $t_{epoch} = 3$ secs is the worst. The reason is, the probability of packet loss increases with t_{epoch} , as shown in equation (3.2). The performance of $t_{epoch} = 1$ sec is less than the performance of $t_{epoch} = 2$ secs. This is because, $t_{epoch} = 1$ suffers from insufficient data phase duration.

Based on the above experiments, it is observed that 2 secs epoch duration gives the best performance. It is found that pdr performs best when ϑ is 10 m/s, whereas lce is best when ϑ is 12 m/s. The solution suggests keeping ϑ as 12 m/s to achieve a good balance between lce and pdr .

The performance of the proposed scheme also depends on the ratio of c-nodes to r-nodes in the topology. Experiments are performed varying the c-node ratio from 10 % to 100 % keeping $\lambda = 100$, $\vartheta = 10m/s$, and $t_{epoch} = 2sec$. To measure the edge selection efficiency, we define a new metric "Edge selection ratio" (esr), which is the ratio (percentage) of c-nodes that are selected as edges by the CDN server. The average value of esr is taken for all simulation runs. Clearly, the CDN server wants to keep esr as low as possible. Fig 3.5 shows the performance of esr with varying c-node ratio. It is observed that, in general, esr gets decreased when the c-node ratio in topology increases. As the ratio of c-nodes w.r.t r-nodes goes down, multi-hop path delays among c-nodes increase. In that case, the CDN server tends to select more c-nodes as edges to satisfy the delay constraint resulting in increased

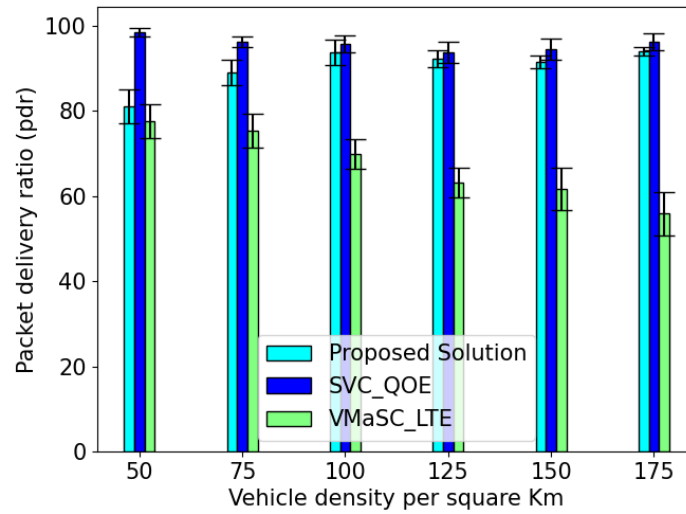


Fig. 3.8 Packet delivery ratio vs Vehicle density

gsr. In an extreme case, when all the c-nodes are sufficiently far from each other, the CDN server may select all the c-nodes as edges. Therefore, *esr* metric may be misleading while assessing protocol performance in a topology with a low c-node ratio. It is observed that *esr* is not influenced by epoch duration as it depends only on the topology. The performance of *pdr* metrics is also measured with varying c-node ratios. From Figure 3.6, it is observed that *pdr* tends to decrease with an increase in the c-node ratio in the topology. As a lesser number of edges are selected when the c-node ratio increases, V2V collaboration also increases, resulting in more DSRC transmissions. Data packets tend to collide due to the lossy nature of DSRC and high velocities among vehicles which reduce *pdr*. On the other hand, as the c-node ratio decreases, most of the c-nodes are selected as edges by the CDN server. As edges receive data packets through dedicated cellular channels, *pdr* gets improved. The performance also decreases as epoch duration increases. The reason is, any client-edge multi-hop path tends to break as epoch duration is increased.

Next, the performance of the proposed solution is compared with two existing solutions. Comparisons are made with one centralized edge selection of single-hop collaboration [21] and one distributed edge selection of multi-hop collaboration [22], namely, VMaSC_LTE. As the work [21] does not have a name, this work assigns the name SVC_QOE to refer to the work [21] for convenience. Though VMaSC_LTE was originally designed to disseminate safety messages, it is modified to distribute video data to compare the edge selection and V2V data delivery performances. The centralized edge selection algorithm of SVC_QOE runs in epochs. Initially, all vehicles use LTE networks and is sorted in ascending order with

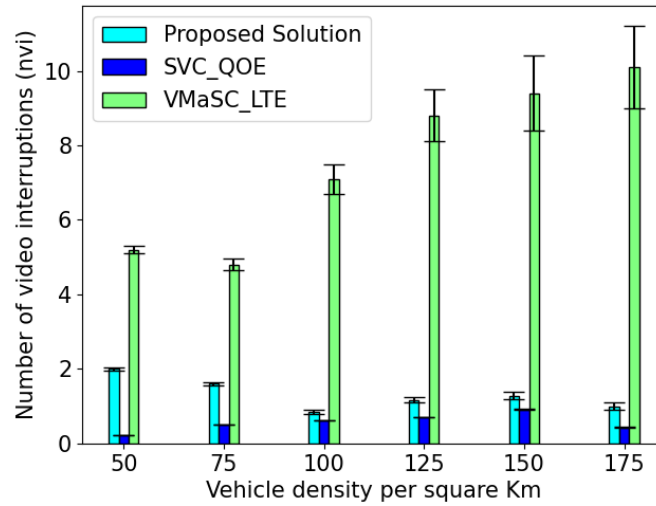


Fig. 3.9 number of video interruptions vs Vehicle density

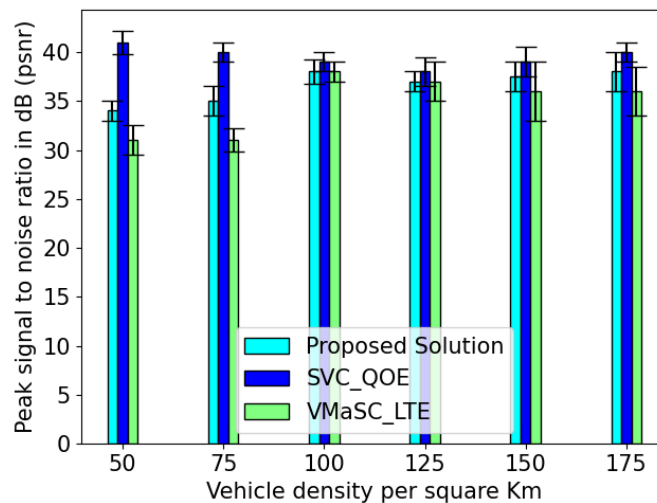


Fig. 3.10 Peak signal to noise ratio vs Vehicle density

LTE data rate. Starting from the first vehicle (lowest LTE data rate), if a vehicle has a DSRC link with a neighbor with more data rate than its LTE data rate, it makes that neighbor vehicle its upstream node. If a vehicle has one or more dependent vehicles, it can make another vehicle as its parent node only if all its dependent node is also in direct communication of that target node. In the distributed edge selection algorithm of VMaSC_LTE, vehicles exchange periodic messages to maintain multi-hop neighborhood information. A vehicle with the lowest average relative velocity within its multi-hop neighbors is eligible to become a cluster head and use the LTE network. Otherwise, it tries to connect to an existing cluster head if available, in a single-hop or multi-hop manner. Otherwise, if no neighbor is available, it becomes a cluster head. To reduce the number of clusters hops in the system, whenever two cluster heads become neighbors, two clusters are merged. If a vehicle's connectivity with its parent vehicle of the cluster is lost, it immediately searches for a new cluster head or cluster member to connect with.

Figure 3.7, Figure 3.8, Figure 3.9, and Figure 3.10 shows the comparative performance of the proposed solution with SVC_QOE and VMaSC_LTE on the defined metrics with varying λ . For all these experiments, t_{epoch} is kept as 2 secs and v as 12 m/s as concluded from the previous experiments. The ratio of c-node and r-node is kept as 1:3.

Figure 3.7 shows lce values against varying λ . As can be seen, the proposed protocol has the trend of increasing LTE cost-effectiveness with the increasing λ . This can be easily understood because in higher λ , a segment of video from an edge can reach more vehicles. SVC_QOE performs poorly in this metric, which is justified because its objective is not to provide cost-effective streaming. In a sparse scenario, the average distance between two vehicles increases which decreases the SNR value resulting lesser channel data rate. As a result, SVC_QOE always prefers using a cellular channel with a good data rate over DSRC, resulting in almost no V2V collaboration. The degree of V2V collaboration increases a little bit as topology becomes denser with vehicles having neighbors of good SNR. Though VMaSC_LTE performs far better than SVC_QOE, it is less efficient than the proposed protocol as VMaSC_LTE has hop-limited V2V collaboration.

Figure 3.8 shows the pdr values with varying λ . It can be seen that the proposed protocol has an overall trend of increasing pdr with increasing λ . In sparse scenarios, the average distance between two vehicles increases, which results in more connection disruptions. With an increase in vehicle density, network connectivity increases, which in turn decreases the number of clusters and increases the average number of hops in clusters. It is shown in 3.3 that the probability of inter-cluster collision decreases with an increase in the average hop-count of clusters (refer equation (3.3)), which increases pdr . SVC_QOE performs best

as most of the vehicles receive data packets directly through cellular channels. When the topology is moderately dense, and vehicle cooperation exists, pdr of SVC_QOE reduces to some extent. VMaSC_LTE performs poorly as the control messages are aperiodic (depends on asynchronous state switching) and are transmitted at any time, which may collide with ongoing transmissions of large data packets. As a result, VMaSC_LTE shows a steady decrease in pdr with increasing λ , as control packet transmission also increases with λ . The proposed protocol and SVC_QOE are able to avoid this problem as the control packets are periodic and do not overlap with data packets.

Figure 3.9 and Figure 3.10 shows the nvi and $PSNR$ metrics respectively, with varying λ . As can be seen, these two metrics are direct consequences of the metric pdr . SVC_QOE receives most of the packets correctly, which results in low nvi and increased $PSNR$. SVC_QOE is the clear winner in these two matrices, while the proposed protocol performs significantly better than VMaSC_LTE. The performance of VMaSC_LTE in metrics nvi and $PSNR$ steadily decreases with increasing λ .

Based on the above experiments, it is found that though SVC_QOE is able to achieve its goal of providing the best streaming experience to users, it fails to meet the primary goal of service cost reduction, i.e., to minimize the number of simultaneous cellular connections. On the other hand, though the edge selection of VMaSC_LTE is efficient, it is very inefficient for data packet distribution. The proposed protocol is able to reduce the number of simultaneous cellular connections significantly while providing a compatible streaming service.

3.5 Summary

This work proposes a cost-effective edge vehicle selection solution for live video streaming over IoCV. It reduces the streaming service cost of the CDN provider by selecting a minimum number of edges while ensuring constrained delivery latency to all client vehicles. The minimum edge selection problem is solved centrally by the CDN server using the greedy approximation technique for the minimum set-covering problem. For performance evaluation, the metrics are defined to assess both the cost-effectiveness of the streaming service and user satisfaction. Comparative result analysis demonstrates the proposed solution as the most cost-effective edge selection protocol. However, the centralized solution may impose the cost of high-performance computation setup on the CDN providers. A distributed edge selection solution may avoid this cost which is described in the next chapter.

Chapter 4

Cost-effective distributed edge vehicle selection with V2V multicasting

The previous chapter proposes a centralized solution for optimum edge vehicle selection for CDN providers to provide live video streaming services over IoCV. However, centralized solutions for edge selections have some disadvantages. Firstly, the CDN server needs to build the global vehicle connectivity graph for the centralized edge vehicle selection. Therefore, all vehicles need to send their local neighborhood information to the CDN server through the Internet, which may make the CDN server overwhelmed to handle numerous simultaneous connection requests. Moreover, to finish the huge computation task of optimum edge selection within the delay constraint of live streaming, the CDN provider needs to make massive investments in procuring high-computing facilities or availing of paid cloud services. Alternatively, if vehicles themselves select edge vehicles distributively, CDN providers are relieved from expensive server setups. However, the distributed algorithm must keep the number of edge vehicles as minimum as possible to increase the cost-effectiveness of the live video streaming service. The distributed solution should possess the time-bounded convergence property to become applicable for the live streaming service. In the existing literature, no distributed solution for edge vehicle selection is found which has the objective of edge vehicle minimization for live streaming with constrained delivery latency. The above facts are the motivations for this work which proposes a solution for cost-effective distributed edge vehicle selection for live video streaming over IoCVs.

Distributed edge vehicle selection requires V2V collaborations where all vehicles need to exchange a number of control messages. However, only a few vehicles of an IoCV are clients of a CDN provider and run a CDN client application. Other vehicles are ordinary vehicles that

are only capable of network layer collaborations. V2V collaborations in the network layer can be achieved through unicast, broadcast, or multicast techniques. For distributing identical content like live video streaming to a subset of vehicles, unicast techniques result in high redundancy whereas broadcast techniques result in network congestion. Multicast techniques are most suitable for distributing identical content to a subset of vehicles. Multicast techniques are well-established for wired networks, and few of the techniques have been adapted for ad-hoc networks. However, these adaptations are not adequate for the highly dynamic environments of IoCVs. They are also inappropriate for delay-constrained applications. The selected edge vehicles act as multicast sources to distribute contents among peer client vehicles. In the case of live streaming applications, all the multicast sources simultaneously distribute identical content. Therefore, it is sufficient for a client vehicle to be associated with only one of the multicast sources. However, existing multicast protocols do not have any provision for multicast source selection. The above facts are the motivations for a V2V multicast solution for live video streaming over IoCVs. Below are the contributions of this work.

- This work formulates the problem of delivery delay constrained minimum edge vehicle selection for live video streaming over IoCV as the distributed MSC problem. To the best of knowledge, this is the first time the edge vehicle minimization problem is solved as distributed MSC problem.
- A multicast protocol, namely DSS-CAST, is proposed for V2V collaboration that is specifically designed for delay-constrained live streaming applications over IoCV. This is the first time a multicast protocol has a source selection provision.

The rest of this chapter is organized as follows. Discussions on existing multicast protocols and V2V collaborations for video streaming over IoCV have been done in Section 2.1.2.2 and Section 2.2.3 of Chapter 2, respectively. A discussion on cost-effective edge vehicle selection is done in Section 3.1 of Chapter 3. Therefore, the related work section is not presented in this chapter. In Section 4.1, a detailed description of the proposed solution and its methodology is given. The performance analysis and simulation results comparison are made in Section 4.2. Section 4.3 summarizes the contributions of this chapter.

4.1 Proposed Solution

This section describes the proposed solution in detail and is organized as follows. The system model with assumptions are given in Section 4.1.1. A brief overview of the proposed solution

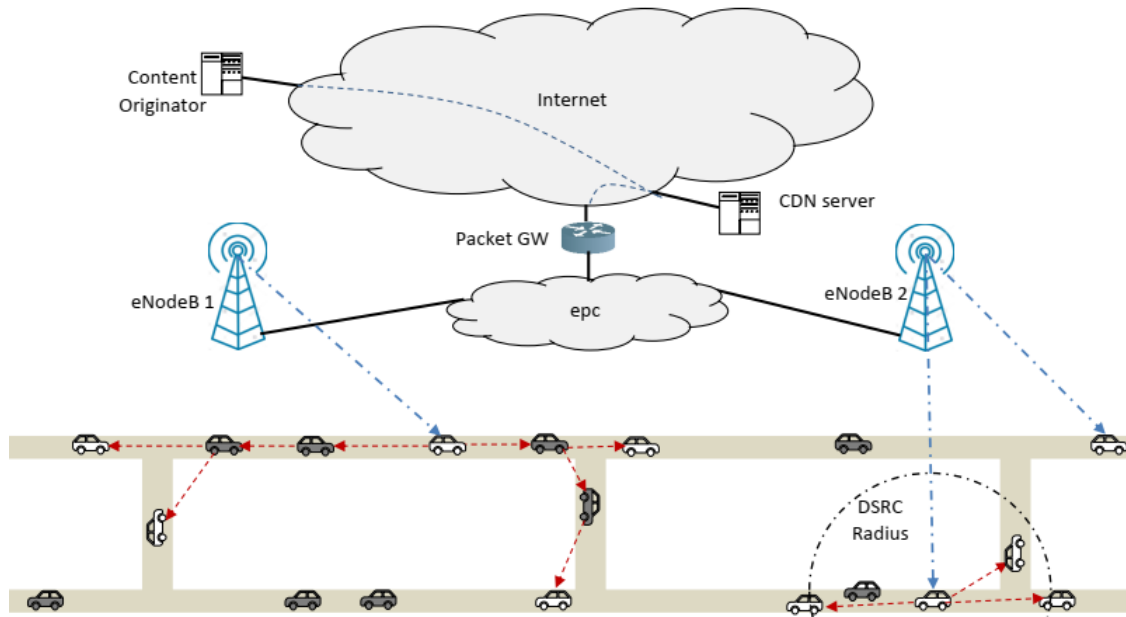


Fig. 4.1 An example scenario of video distribution in VANET by edges of 3-hop domination

is given is Section 4.1.2. The problem formulation is done in Section 4.1.3. The detailed methodologies are described in Section 4.1.4 and Section 4.1.5. Feasibility analysis of the proposed methodologies is done in Section 4.1.6.

4.1.1 System model

In the proposed system, an IoCV scenario is considered where all vehicles have two radio interfaces. The LTE interface is used for V2I communication, and the DSRC interface is used for V2V communication. Fig. 4.1 illustrates a scenario of streaming video distribution over IoCV. A CDN service provider is responsible to provide live video streaming services within an AOI. The CDN provider deploys a CDN replica server in the vicinity of AOI. The goal of the CDN provider is to minimize its service cost by selecting a minimum number of edges (minimum number of simultaneous LTE connections and minimized Internet bandwidth usage) while ensuring uninterrupted streaming service to its clients. The vehicles which run client application instances on them to avail streaming service are termed as c-nodes (white vehicles in Fig. 4.1). A few of the c-nodes become edges which are termed e-nodes. All other vehicles are network layer vehicles, and they are termed as r-nodes (black vehicles in Fig. 4.1) in this work. All vehicles (c/e/r-nodes) are capable of providing basic routing services.

4.1.2 Protocol overview

The vehicle connectivity in an IoCV is highly dynamic due to high velocities of vehicles, due to which V2V collaboration may fail. However, the connectivity among vehicles that have relative velocities less than a chosen threshold R_ϑ , remains stable for a sufficiently short duration termed as “epoch” in this work. The CDN replica server divides the streaming duration into a number of epochs. Each epoch is divided into two phases: CNTRL (control) phase of duration t_{ctrl} and DATA (data) phase of duration t_{data} such that the epoch duration $t_{epoch} = t_{ctrl} + t_{data}$, where $t_{ctrl} \ll t_{data}$. In the CNTRL phase of each epoch, a fresh e-node selection is made, and client-edge associations are built depending on current vehicle connectivity. To maximize the cost-effectiveness, the proposed solution formulates the edge selection problem as an MSC problem that is solved distributively. The video stream is divided into packets of size P . Let’s assume the minimum achievable data rate of DSRC within AOI is observed as Υ_w by the CDN server (reported by vehicles). If the average video data rate is δ , the required number of data packets N , which is distributed in one epoch, is calculated as $N = (t_{epoch} \times \delta)/P$. Each c-node is required to receive total data of size β given by $N \times P = t_{epoch} \times \delta$ within the t_{data} duration (DATA phase of an epoch). The time t_β to transmit β amount data through the DSRC interface of any vehicle is β/Υ_w . Let’s assume the average download rate of LTE in that AOI is observed as Υ_L . So the time T_β to procure β amount data through the LTE interface is β/Υ_L . As all the c-nodes should receive β amount of data within the DATA phase of t_{data} duration, the maximum hop K , the β amount of data can travel is given as $K = \lfloor (t_{data})/t_\beta \rfloor$. On substituting the value of t_β , the following equation is given.

$$K = \left\lfloor \frac{t_{data} \times \Upsilon_w}{\beta} \right\rfloor = \left\lfloor \frac{t_{data} \times \Upsilon_w}{t_{epoch} \times \delta} \right\rfloor = \left\lfloor \frac{(t_{epoch} - t_{ctrl})}{t_{epoch}} \times \frac{\Upsilon_w}{\delta} \right\rfloor \quad (4.1)$$

All the control and data packets exchanged among c-nodes use the DSS-CAST multicast protocol described in Section 4.1.4.1. Data distribution through a multi-hop path from an e-node to any c-node may involve r-nodes as intermediate nodes. In the proposed solution, all c-nodes go through synchronized phases. When a vehicle becomes a c-node, it sends a sync request message S-SYNREQ to the CDN server. The CDN server replies with a sync reply message S-SYNREP which includes the all phase duration, the time remaining to begin the next epoch, max hop K , and supported relative velocity threshold R_ϑ . Fig. 4.2 shows the timeline of phases and packet transmission sequence. When a vehicle becomes a c-node, its media player also gets started. The media player experiences an initial startup delay of t_{epoch}

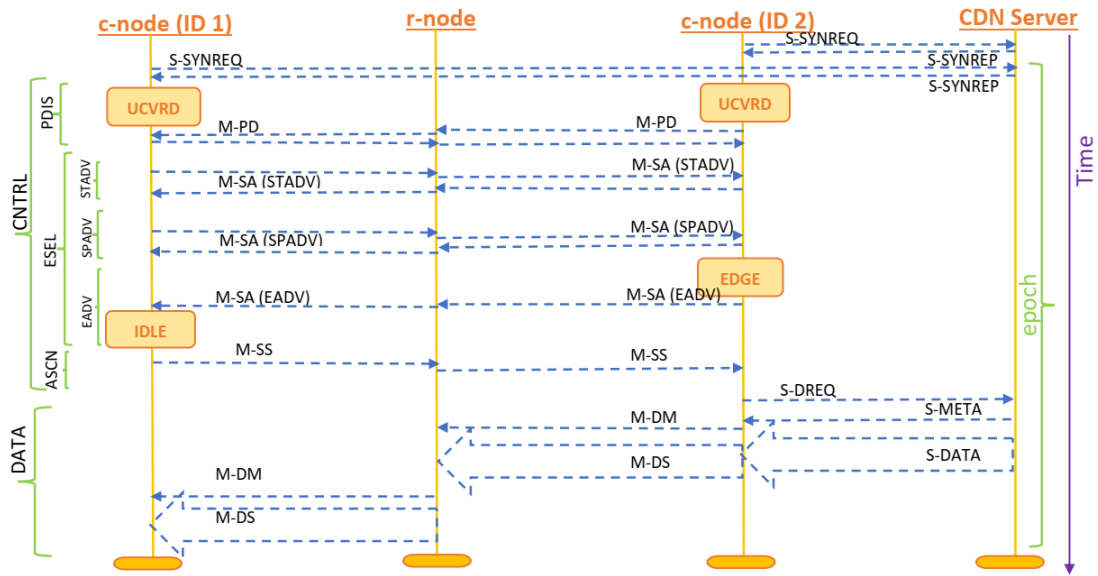


Fig. 4.2 Timeline sequence of phases and packet transmissions for a simple topology consisting of two c-nodes connected through a r-node. Without maintaining any application state, a r-node can help two c-nodes to share content

in the worst case. Once the content is fetched, it is played for the next t_{epoch} duration while the client continues to fetch subsequent contents.

4.1.3 Problem Formulation

The set of vehicles V consists of two subsets. One subset is of c-nodes C , and the other subset is of r-nodes R such that $C \cup R = V$, and $C \cap R = \emptyset$. Let wireless links among vehicles form the set L . The vehicle topology graph is represented as graph G , with V as vertices and L as edges. A subset $E \subseteq V$ represents the set of selected edges or e-nodes. As r-nodes are not running client applications, they cannot become an e-node. So, if E is selected as the solution of *Minimum K-hop Dominating Set* (MKDS) over V , E may be infeasible if E contains one more or r-nodes. Accordingly, we have the constraint of $E \subseteq C$ for E to be a feasible solution. Also, E requires to minimally K -hop dominate only on C instead of V as V includes r-nodes that are not interested in receiving streaming service. Otherwise, $|E|$ may become unnecessarily large to dominate R , making the solution inefficient. R cannot be simply deleted from G and select E as MKDS solution on the resulting graph, as c-nodes may be connected among themselves through r-nodes. Deleting R from G may remove some of the paths of K -hop connectivity among c-nodes resulting in a larger $|E|$ as an inefficient

MKDS solution. Therefore, the problem is formulated as the MSC problem for an efficient solution which is described next.

The well-known MSC problem is defined in Section 3.2.3. The X is initialized with C . The elements s_{c_i} for S is created from each $c_i \in C$ which implies $|S| = |C|$. The elements of s_{c_i} are the c-nodes which are within K -hop distance from c_i . As each c_i covers at least one element (itself), $X = \cup_i s_{c_i}$. The goal is to find the minimum number of s_{c_i} (c-nodes) which covers the set X (all c-nodes). MSC problem is a known NP-hard problem, and a well-known approximation algorithm, namely GSC [30], exists with an approximation ratio of $\ln|X| + 1$. However, GSC is designed as a centralized algorithm which has several drawbacks as discussed earlier. *Distributed Database Coverage Heuristic* (DDCH) [112] algorithm is a distributed version of GSC where the authors have proved that the distributed version achieves the same approximation ratio as the centralized GSC. However, DDCH cannot be directly applied in the proposed system model due to the following reasons:

- DDCH requires all nodes of the network to run the DDCH application synchronously. However, because of the presence of r-nodes in the network, V2V collaboration cannot happen in the application layer.
- Unlike DDCH, the K -hop path between any e-node to c-node needs to be R_{ϑ} constrained for the path to remain stable for an epoch duration. This adaptation is required to handle the highly dynamic topology of IoCVs.
- As mentioned in [113], for some network topology, the number of iterations in DDCH can be of polynomials to the number of nodes to be covered. In case of a large $|C|$, the control phase duration may increase to the extent of making data distribution infeasible within an epoch duration.

Therefore, an adaptation of DDCH is proposed in this work and is used in the CNTRL phase, as described next.

4.1.4 CNTRL Phase

An epoch starts with the CNTRL phase which is responsible for peer discovery, distributed edge selection, and multi-hop client-edge association. In the CNTRL phase, all the messages are exchanged using DSS-CAST multicast protocol which is described as follows.

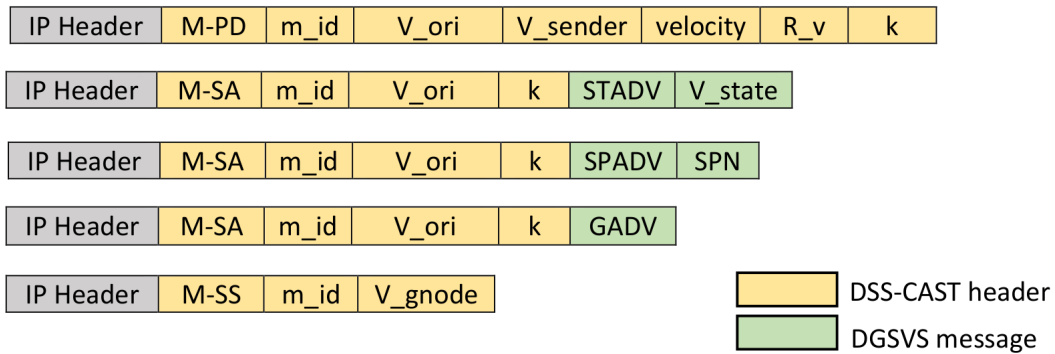


Fig. 4.3 DSS-CAST control headers

4.1.4.1 DSS-CAST

DSS-CAST is proposed as a general-purpose streaming multicast protocol for dynamic ad-hoc networks like VANETs, which can be used by any streaming application. DSS-CAST is an asynchronous reactive protocol unaware of any application-specific phase/state. Edge selection algorithms run in streaming applications which is transparent to the DSS-CAST, and can support and forward only five types of multicast packets which are *peer discovery* (M-PD), *source advertisement* (M-SA), *source selection* (M-SS), *metadata* (M-DM), and *data sharing* (M-DS). DSS-CAST accepts relative velocity threshold R_ϑ , and hop limit K as parameters from a streaming application to build hop-constrained multicast trees. The hop limit of K is required to restrict the delivery delay within the streaming delay threshold. The R_ϑ is used to avoid frequent link disruptions among vehicles. DSS-CAST maintains two data structures, namely the PEER table and the DATA-STAT table, which are generic for any streaming application. PEER table is created from M-PD packets and is consulted to forward M-SA packets. DATA-STAT table is created and used for M-DS type packets. Fig. 4.3 shows the packet formats of DSS-CAST. The nodes wait for a short random time (few μs) before forwarding/re-transmitting any DSS-CAST message to make the transmissions asynchronous within their neighbors. DSS-CAST header length is uniquely determined by the header type. The details of these packets and data structures are described along with the CNTRL phase.

4.1.4.2 States of c-nodes

A c-node can be in any one of the four states, namely UCVRD (uncovered), CNDT (candidate), IDLE (idle), and EDGE (edge). A c-node is in UCVRD state when it is not covered by any e-node within its K -hop neighborhood. In case a c-node is covered, but it has one or more c-node peers of UCVRD state within its K -hop neighborhood, it goes to CNDT state

as it may become an e-node in future to cover those UCVRD c-nodes. When a c-node is covered, and it has no UCVRD c-node peers within its K -hop neighborhood, it goes to the IDLE state. A c-node goes to the EDGE state when it becomes an e-node. As r-nodes are not running client applications, they are unaware of any c-node states. At the start of the CNTRL phase, a new multicast session is started, and all c-nodes reset their state to UCVRD. CNTRL phase is divided into three subphases, namely PDIS (peer discovery), ESEL (edge selection), and ASCN (association), which are described next.

4.1.4.3 PDIS subphase

In the PDIS subphase, vehicles discover their peers within their K -hop neighborhood. The duration of the PDIS subphase is t_{pdis} . In PDIS, each vehicle originates a M-PD multicast packet with the header including multicast session id m_{id} , packet originator's vehicle id v_{ori} , last sender vehicle id v_{sender} , its velocity ϑ , relative velocity threshold R_{ϑ} , and remaining hop count k (refer Fig. 4.3). Initially, the value of k is set as $K-1$, and it is decremented by 1 in each hop. On receiving M-PD, a vehicle calculates relative velocity $r_{\vartheta}^{v_{sender}}$ with v_{sender} . If $r_{\vartheta}^{v_{sender}} > R_{\vartheta}$, the vehicle discards the packet. Otherwise, the vehicle creates a new tuple $\langle m_{id}, v_{ori}, v_{sender}, k \rangle$ in its PEER table. The tuple indicates that v_{ori} can be reached in $(K - k)$ hops using v_{sender} as the next-hop vehicle. If the value of received k is 0, the M-PD is not forwarded by the receiving vehicle. Otherwise, it creates and re-transmits a new M-PD message where it replaces v_{sender} , ϑ with its own values, and decrements k by 1. If a vehicle receives M-PD where v_{ori} is already present in the PEER table, it compares the currently received k with the k stored for the tuple of v_{ori} in PEER table. If current k is smaller than stored k , the received M-PD is discarded. If current k is greater than stored k , it implies that a shorter path exists to reach v_{ori} . In that case, the receiving vehicle replaces the tuple of v_{ori} in the PEER table with the new information and re-transmits the M-PD. By the end of the PDIS sub-phase, each vehicle becomes aware of the presence of other vehicles within its K -hop neighborhood, and a next-hop vehicle for the shortest path to reach each of those peer vehicles. During an entire epoch, PEER table entries remain valid. PEER table is cleared at the beginning of the next epoch when a new multicast session is initiated.

4.1.4.4 ESEL subphase

The next subphase is ESEL, in which a subset of c-nodes is distributively selected as e-nodes. ESEL subphase runs in iterations, with a total duration of is t_{esel} . Initially, all c-nodes are in UCVRD state at the beginning of the ESEL subphase. The number of ESEL

Algorithm 2 ADDCH iteration (runs in each c-node)**Input:** PEER table, state = UCVRD/CNDT, v_{id} , I , H **Output:** Output State: IDLE / EDGE

```

1: if PEER table is empty then
2:   state  $\leftarrow$  EDGE
3:   return
4: end if
5: define iteration count COUNT = 0 //iteration begins
6: while COUNT < ( $I - H$ ) do
7:   //STADV step begins
8:   if state == UCVRD or state == CNDT then
9:     advertise state to  $K$ -hop peer c-nodes
10:    while in STADV step do
11:      if receive STADV message then
12:        create an entry in STATE table
13:      end if
14:    end while
15:   else
16:     while in STADV step do
17:       //No op.
18:     end while
19:   end if
20:   //SPADV step begins
21:   if state == UCVRD or state == CNDT then
22:     calculate SPN based on STATE table
23:     //A c-node covers itself
24:     SPN = SPN + 1
25:     if its LTE SNR >  $\gamma$  then
26:       advertise SPN to  $2K$ -hop peer c-nodes by SPADV message
27:     end if
28:     while in SPADV step do
29:       if receive SPADV message then
30:         create an entry in SPAN table
31:       end if
32:     end while
33:   else
34:     while in STADV step do
35:       //No op.
36:     end while
37:   end if
38:   //EADV step begins
39:   if state == UCVRD or state == CNDT then
40:     if its  $\langle$  SPN,  $v_{id}$   $\rangle$  is highest than entries of SPAN table then
41:       if its LTE SNR >  $\gamma$  then
42:         state  $\leftarrow$  EDGE
43:         advertise state to  $K$ -hop peer c-nodes by EADV message
44:       end if
45:     end if
46:   end if
47:   while in EADV step do
48:     if receive EADV message then
49:       create an entry in EDGE table // Done if in EDGE, IDLE state also. To be used in redundancy elimination step.
50:     if state == UCVRD or state == CNDT then
51:       if has UCVRD peer c-node in STATE table then
52:         state  $\leftarrow$  CNDT
53:       else
54:         state  $\leftarrow$  IDLE
55:       end if
56:     end if
57:   end while
58:   end while
59:   ++COUNT
60: end while

```

iterations is bounded by I so that the ESEL subphase terminates within t_{esel} . Within I ESEL iterations, the first $I - H$ iterations are ADDCH iterations, which is an adaptation of DDCH. If the algorithm fails to converge within $I - H$ iterations, the remaining H iterations run the proposed *Greedy Heuristic Edge Selection* (GHES) algorithm to ensure the convergence within I iterations. An ADDCH/GHES iteration consists of three steps, namely STADV (state advertisement), SPADV (span advertisement), and EADV (edge advertisement), which are described subsequently. The algorithm for ADDCH and GHES are given in Algorithm 2 and Algorithm 3 respectively.

The first iteration step is the STADV (line 7 to 19 of Algorithm 2), in which c-nodes advertise their current state to their K -hop c-node peers. STADV continues for t_{stadv} duration. In this step, c-nodes of UCVRD or CNDT state originate a M-SA multicast packet, and the header includes multicast session id m_{id} , packet originator node id v_{ori} , and remaining hop count k initially set as $K-1$ (refer Fig. 4.3). The payload of M-SA contains application message type STADV and the originator node's state v_{state} . On receiving M-SA, a node (c-node/r-node) checks if it has an entry for the received v_{ori} in its PEER table. If no entry is found, the packet is discarded as the path from v_{ori} violates R_{\emptyset} constraint. Otherwise, it decrements the value of k by 1, and re-transmits M-SA. Duplicate M-SA packets are detected by checking the tuple $\langle m_{id}, v_{ori} \rangle$, and are discarded. When a c-node receives M-SA, it extracts the application message type. If the type is STADV, it stores the state v_{state} for v_{ori} in the STATE table maintained by it. By the end of the STADV step, each c-node knows the states of peer c-nodes within its K -hop neighborhood. For the first iteration, M-SA packets can be piggybacked in M-PD, which reduces the number of packets, and saves t_{stadv} time. For example, in the scenario shown in Fig. 6.3, both of the c-nodes advertise state UCVRD in STADV phase.

Once the STADV step is over, the next iteration step SPADV starts (line 20 to 37 of Algorithm 2). The span (SPN) of a c-node is defined as the number of peer c-nodes, including itself, which are in UCVRD state (i.e., peers which are not covered yet) within its K -hop neighborhood. In SPADV, each c-node (with good LTE SNR) of UCVRD/CNDT state calculates SPN from its STATE table, and advertises SPN within its $2K$ -hop c-node peers. If the measured SNR of the LTE link of a c-node is below a certain threshold γ , it considers itself unfit to be an e-node, and therefore does not advertise its SPN . Otherwise, each c-node originates a M-SA multicast packet which includes the header fields as mentioned in the STADV step, except k is set to $2K-1$. In the payload, it sets the application message type as SPADV, and includes its calculated SPN . Re-transmission of a M-SA packet by intermediate nodes (c/r-nodes) is the same as described in STADV. When the M-SA packet reaches a

c-node, and the payload application message type is found to be SPADV, it creates an entry for v_{ori} in its SPAN table maintained by it. SPADV step continues for t_{spadv} duration. By the end of the SPADV step, all the c-nodes are aware of the spans of their peer c-nodes within the $2K$ -hop neighborhood. If the span of a c-node neighbor is unknown at the end of SPADV, it is assumed to be 0. For example, in the scenario shown in Fig. 6.3, both of the c-nodes advertise SPN as 1 in the SPADV phase.

The next iteration step is EADV (line 38 to 58 of Algorithm 2). From the SPAN table, each c-node in UCVRD or CNDT state determines whether the ordered pair of its own span SPN and v_{id} ($< SPN, v_{id} >$) is lexicographically highest in its SPAN table. If true, and it's LTE SNR $> \gamma$, it becomes an e-node and switches its state to the EDGE. For example, in the scenario shown in Fig. 6.3, c-node with ID 2 becomes the winner and becomes an e-node. Each e-node multicasts a M-SA packet with application message type as EADV. The message format is shown in Fig. 4.3. Re-transmission method of the M-SA packet is the same as described for the STADV step. On receiving a M-SA packet for the first time, if a c-node has UCVRD peer c-node(s) in its STATE table, it changes its state to CNDT. Otherwise, it switches to the IDLE state. The EADV step duration is t_{eadv} . If within t_{eadv} a UCVRD c-node does not receive any EADV message, it remains in the UCVRD state. On receiving each EADV message, a c-node (including e-nodes) adds a tuple $< v_{ori}, (K - k) >$ for the advertised e-node in its EDGE table to keep track of all the e-nodes within its K -hop distance.

At the end of the EADV step, one iteration completes. ADDCH runs for $I - H$ iterations. If few c-nodes are still in UCVRD or CNDT state after $I - H$ iterations of ADDCH, to ensure convergence of ESEL within t_{gsel} , they switch to GHES for the next H iterations (described below). Other c-nodes (of state IDLE/EDGE) do not participate in GHES and remain idle.

GHES iterations run for the remaining H iterations only in the c-nodes that are still in UCVRD or CNDT state. STADV and SPADV steps of GHES are the same as described for ADDCH. Only the EADV step of GHES differs from ADDCH, which is described next. In the EADV step of every GHES iteration, instead of a single winner (e-node), a set of winners are selected distributively to guarantee convergence within H iterations of GHES. Let the current GHES iteration count is h . Let's assume for a c-node c_i , the current tuple count in its SPAN table is T_h . The c-node calculates its current rank $r_h^{c_i}$ in its lexicographically sorted SPAN table. The c-node finds its current advertisement slot as $s_h^{c_i} = \lfloor r_h^{c_i} \times \frac{H}{T_h} \rfloor = \lfloor H \times \frac{r_h^{c_i}}{T_h} \rfloor$. If $s_h^{c_i} \leq h$, the c-node becomes an e-node and advertises its EDGE state by EADV message. EADV advertisement mechanism using M-SA multicast packet is the same as described for ADDCH. The algorithm for the EADV step of GHES iteration is given in Algorithm 3, and an analysis is given in the following *Lemma*.

Algorithm 3 GHES iteration (runs in each c-node)

Input: PEER table, state = UCVRD/CNDT, v_{id} , H

Output: Output State: IDLE / EDGE

```

1: define iteration count  $h = 0$ 
2: while  $h < H$  do
3:   //STADV step performed same as DDCH adaptation
4:   //SPADV step performed same as DDCH adaptation
5:   //EADV step begins
6:   if state == UCVRD or state == CNDT then
7:     insert own span into SPAN table
8:     sort SPAN table lexicographically by  $\langle SPN, v_{id} \rangle$ 
9:     get tuple count  $T_h$  of sorted SPAN table, and find own rank  $r_h^{v_{id}}$  in it
10:    find its slot  $s_h^{v_{id}}$  as  $\lfloor H \times \frac{r_h^{v_{id}}}{T_h} \rfloor$ .
11:    if  $s_h^{v_{id}} \leq h$  then
12:      state  $\leftarrow$  EDGE
13:      advertise state to  $K$ -hop peer c-nodes by EADV message
14:    end if
15:  end if
16:  while in EADV step do
17:    if receive EADV message then
18:      create an entry in EDGE table
19:      if state == UCVRD or state == CNDT then
20:        if has UCVRD peer c-node in STATE table then
21:          state  $\leftarrow$  CNDT
22:        else
23:          state  $\leftarrow$  IDLE
24:        end if
25:      end if
26:    end if
27:  end while
28:  ++  $h$ 
29: end while

```

Lemma 1: No c-node remains uncovered at the end of I iterations of ESEL subphase

Proof: An ESEL iteration (ADDCH/GHES) has the following properties:

1. At the beginning of the ESEL subphase of an epoch, all c-nodes are uncovered. (refer Section 4.1.4.4)
2. In the worst case, at least one e-node is selected in each ESEL iteration. This is because, there must be one or more c-nodes with highest SPN count in the global topology graph, and in case of a tie, the node with highest ID wins the race. In case of all c-nodes are disjoint, line 2 of Algorithm 2 is executed.
3. An e-node covers at least one c-node (refer line 23,24 of Algorithm 2).
4. Once a c-node is selected as an e-node, it retains its EDGE state throughout that epoch. When a c-node becomes e-node, it does not participate in the further ESEL iterations. As shown in Algorithm 2, only the c-nodes with UCVRD or CNDT states participate in the iterations. As a result, the state of an e-node never changes in that epoch.
5. The property 4 implies that once a c-node is covered by an e-node, it never goes to the UCVRD state in subsequent ESEL iterations. This is because, the e-node which covers that c-node retains its EDGE state.
6. Property 2, 3, 4, and 5 implies that the number of UCVRD c-nodes gets reduced in every iteration.
7. Property 2, 3, 4, 5 and 6 implies that the number of tuples in the SPAN table for any c-node can never get increased in future iterations of that epoch.

A c-node goes to the IDLE state only when it is covered and finds no uncovered c-node in its K-hop neighborhood. Consequently, if a c-node does not go to EDGE/IDLE state within $I - H$ ADDCH iterations, it continues to participate in GHES iterations until it becomes an e-node, or it finds itself covered along with all its peers. As $r_h^{c_i} \leq T_h$ is always true, the value of $s_h^{c_i}$ is bounded by H . As the value of h strictly increases from 1 to H , there must be a value of h , say h' , where $s_h^{c_i} \leq h'$ which satisfies the condition of c-node c_i to be an e-node. This implies that if a c-node does not go to the IDLE state, it goes to the EDGE state by the end of the ESEL subphase. Thus, no c-node can remain in the UCVRD state on the completion of the ESEL subphase. \square

4.1.4.5 ASCN subphase

In this subphase, a c-node selects its nearest (in terms of hop-count) e-node as its streaming provider. A tie is broken by an e-node id. It consults the PEER table to get the next hop

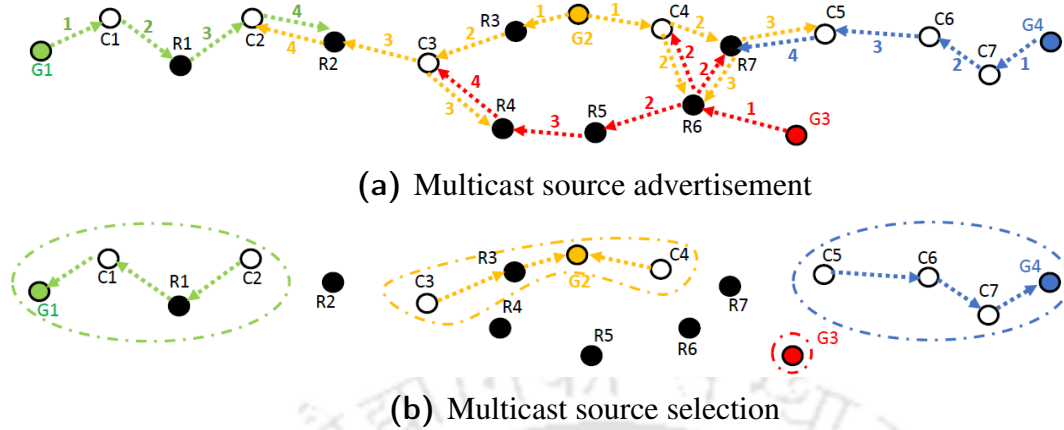


Fig. 4.4 DSS-CAST forest of multicast trees

for the chosen e-node. A c-node sends a unicast packet M-SS to that next hop to join the multicast tree. M-SS header contains multicast session id m_{id} , and selected e-node id v_{enode} . If a r-node receives M-SS, it consults its PEER table to find the next-hop node for v_{enode} and forwards further towards the e-node. If a path from a c-node c_i to its nearest e-node e_i contains another c-node c_j as an intermediate node, by the shortest path property, e_i is also the nearest e-node of c_j . Consequently, request aggregation takes place in c-nodes, where each c-node forwards M-SS only once. On receiving a M-SS packet, each intermediate node (c-node/r-node) sets the DATA-PHASE flag true, which indicates its participation in the multicast tree for data distribution and becomes an upstream node for the M-SS sender. The duration of the ASCN subphase is t_{ascn} . When an e-node receives any M-SS packet, it sets the DATA-SHARE flag to true. Otherwise, it refrains from data sharing. As each e-node has its own multicast tree and a c-node joins only one multicast tree rooted by one e-node, at the end of the ASCN subphase, the entire vehicular network graph is clustered by a forest of multicast trees. The situation is shown in Figure 4.4.

4.1.5 DATA Phase

In the DATA phase, β amount of video data is distributed among all the c-nodes within t_{data} . The DATA phase gets started in all c-nodes synchronously. At the beginning of this phase, e-nodes request the CDN replica server for data, by sending a S-DREQ packet. The server replies to e-nodes with a metadata packet S-META followed by data packets S-DATA. S-META contains the size of data β and the number of packets N . Receiving S-META, e-nodes transmit a multicast meta-data packet M-DM to its downstream nodes, which includes β , N , and transmission duration t_{β} . Any node (r-nodes/c-nodes) which has DATA-PHASE flag

true participates in the DATA phase, while other nodes remain idle. Receiving a M-DM, a node sets a data-receive timer DATA-TIMER with the duration t_β , and creates a hash table DATA-STAT containing tuples $\langle index, RCVD \rangle$. The key $index$ ranges from 1 to N , and the boolean flag RCVD is used to keep track of received data packets. As soon as an e-node receives a S-DATA data packet from the replica server through its LTE interface, it sends out that packet through its DSRC interface as a M-DS multicast data packet which includes $index$ along with data. Downstream nodes start accumulating M-DS packets and set RCVD flags true for the corresponding tuples in the DATA-STAT table. When DATA-TIMER expires in the receiving nodes, they send M-DM to their downstream nodes and multicast the received M-DS packets to them. In each hop, data packets propagate among c-nodes by the *Accumulate and Delegate* (AND) method described above. This AND method incorporates a transmission schedule among nodes which avoids packet loss due to collisions. As the DATA phase duration t_{data} is set as $K \times t_\beta$, by the end of the DATA phase, all c-nodes are expected to receive N number of video data packets.

Once a r-node delegates all the received data packets to its downstream nodes, it becomes idle. Unlike r-nodes, when a c-node finishes delegating the received data packets to its downstream nodes, it looks into its DATA-STAT table to check whether it has received all the expected N data packets. Due to the shared wireless DSRC medium, a few packets may get lost. In that case, a c-node tries to overhear packets of missing indexes from any ongoing transmissions in its 1-hop neighborhood. At the end of the DATA phase, a c-node sorts all the received data packets with packet index, and pushes the video content to the media player buffer for playback.

4.1.6 Feasibility Analysis

To estimate the duration of the CNTRL phase, the number of ADDCH ($I - H$) and GHES (H) iterations needs to be decided. Jia et al. [113] in their work showed that the number of iterations in DDCH protocol can be $\Omega(|V|)$ where $|V|$ is the number of nodes to be covered in a connected graph. However, various studies [18, 17, 19] on the connectivity of VANET topology show that within a major city, a vehicular network is mostly fragmented into a number of disjoint clusters. 90% of these clusters are very small, consisting of very few vehicles, typically less than 15 [17, 19]. With low probability, one or two large/giant clusters can be found non-persistently. Naboulsi et al. [19] found that the probability of a giant cluster size exceeding 2000 vehicles is very low when the vehicle communication radius is 100 m. Being a distributed algorithm, DDCH/ESEL runs independently within each cluster. If 10%

of all vehicles get interested in a particular streaming service at any given time, the number of interested vehicles within a cluster may become 200 at most. ESEL solves MKDS with K values ranging from 1 to 7. The value of K further reduces the number of ESEL iterations by the factor of K . For example, with the K value 5, the number of ESEL iterations needed for the extremely rare form of graphs like the caterpillar graph, or $K_{n/3}$ graph (as shown in [113]) takes approximately 5 or 14 iterations. Though Jia et al. [113] showed the analysis of these graphs from a theoretic perspective, in a real VANET scenario, the probability of these graphs is extremely rare and can be ignored. Accordingly, we decide to run 10 ADDCH iterations and 5 GHES iterations which gives the values of I and H as 15 and 5, respectively.

Next, the duration t_{ctrl} of the CNTRL phase is estimated, and showed that the duration is feasible for a live video streaming service. 802.11p header size is 36 Bytes, and FCS is 4 Bytes. IPv6 header is 40 Bytes. UDP header is of 8 Bytes. All add up to 88 Bytes or 704 bits. Various DSS-CAST header field lengths (refer Fig. 4.3) are as follows. It has a DSS-CAST type field of 8 bits, m_{id} , v_{ori} , and v_{sender} of 16 bits each, 8 bits of R_{θ} , 16 bits of velocity (8 bits each for X and Y dimension), and 8 bits of k . The CNTRL phase message type size is 8 bits, and data (SPN/state) is 16 bits. So, the size of M-PD is given as $8 + (3 \times 16) + 8 + 16 + 8 = 88$ bits. Size of M-SA is given as $8 + (2 \times 16) + 8 + 8 + 16 = 72$ bits. Similarly, the size of a M-SS packet is $8 + (2 \times 16) = 40$ bits. The layer 2 frame size of each packet is obtained by adding 704 bits to the packet size. Accordingly, the layer 2 frame size containing M-PD, M-SA, and M-SS is given by 792, 776, and 744 bits, respectively. This work considers the average DSRC data rate Υ_w ranges from 6 Mbps to 20 Mbps [114]. If Υ_w takes the minimum value of 6 Mbps, the transmission time of one M-PD frame takes $792 / (6 \times 10^6)_{sec} = 132 \mu sec$. The transmission time of one M-SA frame takes $776 / (6 \times 10^6)_{sec} \approx 130 \mu sec$. The transmission time of one M-SS frame takes $744 / (6 \times 10^6)_{sec} = 124 \mu sec$. Before each transmission, a DSS-CAST waits for a random amount of time ranging from 1 to 10 μsec to make the transmission time within its neighbors asynchronous. Considering the above waiting time and the layer 2 DCF medium access time, it is assumed that the average effective transmission duration of all frames with DSS-CAST control packets (M-PD, M-SA, or M-SS) is $250 \mu sec$. The maximum value of k is recommended within 7 as larger values of k make multi-hop path maintenance and data sharing infeasible for dynamic Ad-hoc networks. The propagation time of radio waves is ignored as vehicles are within a distance of a few meters from each other. Queuing delay is ignored as it is assumed that there is no congestion in the network. The processing time is also ignored as vehicles are equipped with high computation power and storage. Considering the above discussion, the duration of each subphase of the CNTRL phase can be calculated. M-PD packets travel a maximum of 7 hops which gives $t_{pdis} = (250 * 7) \mu sec = 1.7 msec$.

STADV and GADV type M-SA packet travel maximum 7 hops, which gives t_{stadv} and t_{gadv} as $(250 * 7)\mu sec = 1.7msec$. SPADV type M-SA packet travels a maximum of 14 hops which makes $t_{spadv} = (250 * 14)\mu sec = 3.4msec$. So, one ESEL iteration takes $2 \times 1.7 + 3.4 = 6.8msec$, and 15 iteration takes $15 \times 6.8msec = 102ms$ time. M-SS packet of ASCN subphase travels up to maximum 7 hops, which gives $t_{ascn} = (250 * 7)\mu sec = 1.7msec$. Consequently, t_{cntrl} becomes $1.7 + 102 + 1.7 = 105.4msec$, and the CNTRL phase duration is set as 110 msec. This estimation is with the minimum DSRC data rate of 6 Mbps [114]. If better data rates (maximum 27 Mbps [114]) are available, t_{cntrl} is further reduced. Increased DSRC channel data rate also increases the value of K (refer equation (4.1)), which makes the solution more cost-effective.

For estimating the duration of the DATA phase, the total epoch duration is decided first. If the relative velocity threshold is allowed up to 15 m/sec (54 km/hr), then in 3 secs, the maximum relative movement between any two vehicles can be 45 meters which is almost half of the transmission range of 100 meters. Therefore, max epoch duration t_{epoch} is decided as 3 secs. Consequently, we get the duration t_{data} as $t_{epoch} - t_{cntrl}$.

4.2 Performance Evaluation

This section is organized as follows. The performance metrics are defined in Section 4.2.1, the simulation environment is described in Section 4.2.2, and the comparative analysis of simulation results is presented in Section 4.2.3.

4.2.1 Performance metrics

The proposed solution is for CDN providers to offer live video streaming as an infotainment application over IoCV. Therefore, the CDN provider needs to assess the performance of the service with respect to QoS as well as client-perceived QoE. As mentioned in [115], the parameters to assess the QoE of a video service over a network can be broadly classified into subjective and objective types. The subjective parameters like *Mean Opinion Score* (MOS) are measured by human perception and are usually difficult to quantify. On the other hand, objective parameters like *PSNR*, *number of video stalling* are easy to quantify and can be controlled in run-time. Among the objective type QoE parameters, PSNR is widely used as a video quality parameter, whereas startup delay, buffering ratio, etc., are used as streaming quality parameters [116]. On the other hand, QoS parameters include average packet delay,

jitter, packet loss, etc. Among the above parameters, startup delay, average packet delay, and jitter are not included in the performance metrics of the proposed solution for the following reasons. As discussed in Section 4.1.2, the startup delay of the proposed solution is fixed at t_{epoch} . Jitter is taken care of by the playback buffer. In case a packet delay is more than t_{data} , that packet is considered lost. Along with assessing QoS and QoE, the CDN provider also needs to evaluate the cost-effectiveness of the proposed solution. Accordingly, the performance metrics lce , pdr , nvi , and $PSNR$ that are defined in Section 3.4.1 of Chapter 3 are reused here for performance evaluation.

The proposed solution is not tied to any specific QoE model. The above-defined metrics can be used as input parameters for existing QoE assessment models. For example, Yin et al. [117] have proposed a QoE model which includes average video quality, average quality variation, rebuffering, and startup delay as input parameters to assess QoE. The video quality assessment function is defined as any non-decreasing function which maps bit rate to client-perceived quality. PSNR is a good candidate function that possesses this property. The defined metric nvi directly resembles rebuffering. Another QoE model is proposed by Yen-Fu et al. [118] where mean opinion score (MOS) is calculated using PSNR and frame rate as input parameters. Kim et al. [119] has proposed the assessment of QoE based on QoS parameters like packet loss, delay, jitter, etc. The CDN provider can choose any of the existing QoE models and use the defined metrics as input parameters for the chosen model.

4.2.2 Simulation environment and parameters

Simulation is performed on NS3 version 3.30.1 [110] with an urban grid topology of 4 square km AOI. All the roads have two lanes where vehicles move in opposite directions. The scenario is created with *Simulation of Urban Mobility* (SUMO) [120], vehicle traces are generated and imported in NS3. Vehicle injection rate is varied to create various vehicle densities λ from 50 vehicles per square km to 175 vehicles per square km. The velocities of vehicles are varied with λ of topology. With high λ , vehicles are nearer to each other and move slowly. In case λ is low, vehicles are apart from each other and move fast. Vehicle traffic flow velocities of different roads and lanes are varied to simulate various V2V collaborations with relative velocity thresholds R_{ϑ} from 29 km per hr (8 m/s) to 54 km per hr (15 m/s). This is because, in the Indian road conditions, the average velocity difference among same-direction vehicles is within 50 km/hr [121]. With low R_{ϑ} , the client-edge association remains stable for a longer duration and vice versa. To observe this effect, we varied the duration of epoch t_{epoch} from 1 sec to 3 sec. LTE download speed Υ_L is set as 25

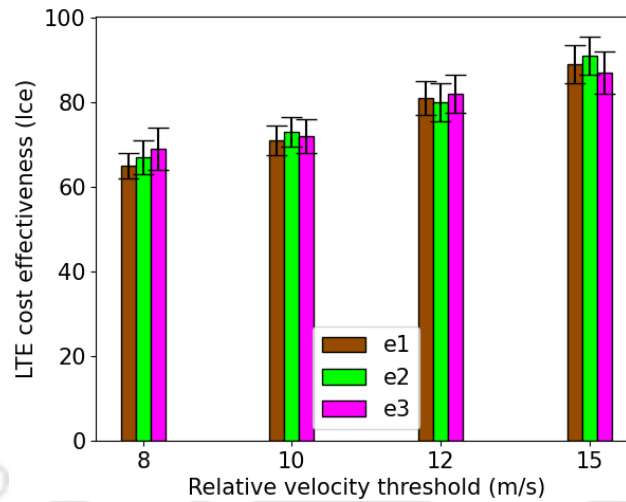


Fig. 4.5 LTE cost effectiveness vs Relative velocity threshold

Mbps. DSRC transmission range is set as 100 meters. A 10 MHz DSRC channel is used with minimum available data rate Υ_w as 6 Mbps [114]. In Section 4.1.6, the value of t_{ctrl} is derived. Using the equation (4.1), the value of max hop limit K can be calculated. The ratio of c-node and r-node is kept at 1:3 in all experiments. Performance analysis based on simulation results on the above-defined metrics is presented next.

4.2.3 Simulation results

First, the sensitivity of the performance metrics is analyzed with respect to relative velocity and epoch duration. For these experiments, vehicle density λ is fixed at 100 vehicles per square km. Epoch duration t_e of 1, 2 and 3 s are abbreviated by e1, e2, and e3 in the legends of Fig. 4.5 and Fig. 4.6. Fig. 4.5 shows the trend of increasing lce with an increase in R_ϑ . When R_ϑ is small, only the vehicles of the same direction with low relative velocity can collaborate. In contrast, when R_ϑ is large, any vehicles of any direction and relative velocity can collaborate. As a result, for high R_ϑ , all the c-nodes can be covered using fewer edges, which improves the lce metric. The duration of an epoch does not play any role for lce metric, as it only depends on the current vehicle connectivity at the beginning of the CNTRL phase. Fig. 4.6 shows that the performance of pdr metric decreases with an increase in R_ϑ for all epoch duration. As vehicles with any directions and relative velocities can collaborate in high R_ϑ , the multi-hop path between an e-node to a c-node may break during the epoch resulting in data packet loss. When epoch duration increases, multi-hop paths become even

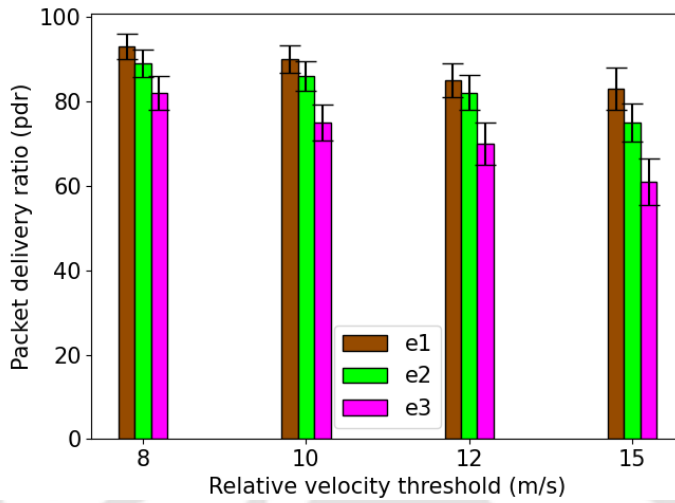


Fig. 4.6 Packet delivery ratio vs Relative velocity threshold

more unstable towards the end of an epoch. This can be observed from the plot that e3 has the worst performance among the three, whereas e1 performs best. It is concluded from the above experiments that the relative velocity threshold could be set as 10 m/s as it is able to achieve a good balance between the LTE cost minimization and client experience aspects for the epoch duration of 1 sec.

Next, the performance of the proposed solution is compared with existing protocols. The performance is compared with one centralized edge selection scheme proposed by Elias et al. [21], one distributed edge selection scheme VMaSC_LTE proposed by Seyhan et al. [22], and with the solution proposed in the previous chapter is abbreviated as ‘Contribution 1’ [Chapter 3]. Elias et al. have not provided any abbreviation for their work, so the name SVC_QOE is used to refer to the work. Though VMaSC_LTE is proposed for safety message dissemination, it is modified to be used VMaSC_LTE for video streaming. A brief description of SVC_QOE and VMaSC_LTE is given in the previous chapter (Section 3.4.3, page 63). The comparative analysis with these protocols are shown in Fig. 4.7, Fig. 4.8, Fig. 4.9, and Fig. 4.10. In all the experiments, the relative velocity threshold R_{θ} is fixed as 10 m/s, and the epoch duration as 1 sec.

Fig. 4.7 illustrates the performance of all protocols for *lce* metric with varying vehicle density λ . It is observed that SVC_QOE has poor *lce* values compared to the other two protocols. In most cases, vehicles in SVC_QOE select the LTE interface for video downloading to maximize QoE. From sparse to moderate vehicle density, VMaSC_LTE and the proposed protocol show better LTE effectiveness with increasing λ . The *lce* increases

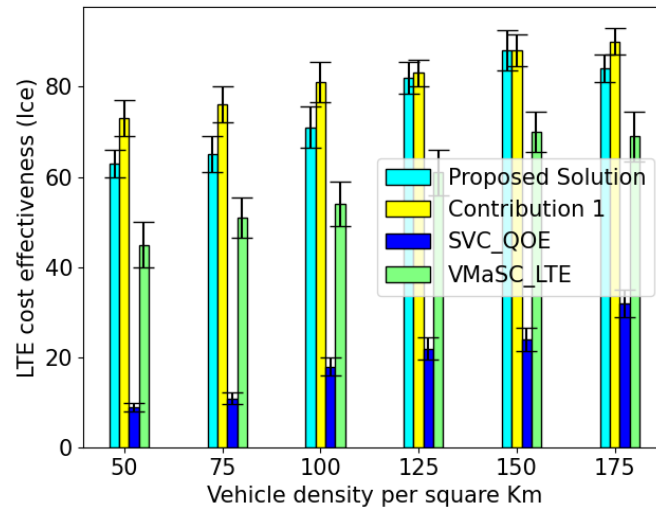


Fig. 4.7 LTE cost effectiveness vs Vehicle density

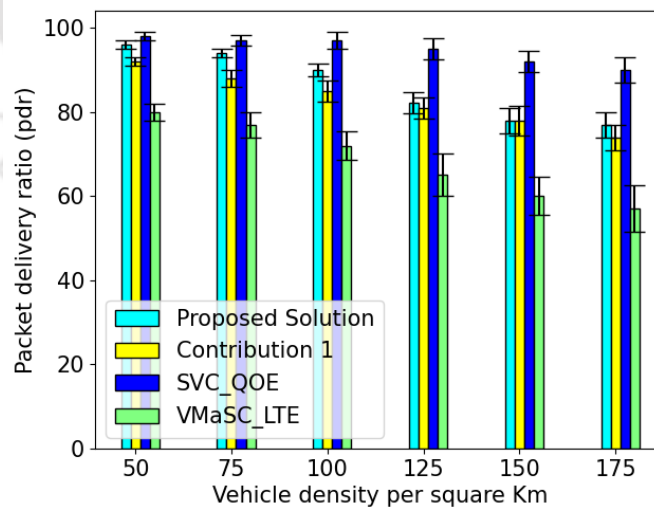


Fig. 4.8 Packet delivery ratio vs Vehicle density

because network connectivity increases with λ , which results in better V2V collaborations. However, for high-density scenarios, the lce tends to decrease because the number of control packets gets increased for distributed protocols. As a result, control packets collide among themselves, making inefficient edge selections. Though the proposed solution outperforms SVC_QOE and VMaSC_LTE, it lags behind the solution of Contribution 1 as the latter selects an optimum number of edges by solving the MSC problem centrally with global topology knowledge.

Fig. 4.8 shows the competitive results of all protocols for the pdr metric. It can be seen that pdr metric value of SVC_QOE and VMaSC_LTE deteriorates with increasing λ . This is expected because the number of simultaneous transmissions increases with vehicle density, causing collisions among data packets. The same trend is followed by the proposed solution from sparse to moderate vehicle density. However, in a high-density scenario, pdr metric performance tends to stabilize as it selects more edges in high-density scenarios due to the loss of a few control packets. If a vehicle (edge) procures data from LTE interface using a dedicated LTE channel, almost all data packets are received correctly. As a result, with more edges in the scenario, the overall pdr metric of the system improves. SVC_QOE outperforms other protocols in all situations as the vehicles tend to select LTE over DSRC for QoE maximization. VMaSC_LTE fails in the pdr metric and performs worst in all situations. This is because VMaSC_LTE control packets collide with data packets due to the cluster merge process. The pdr performance of the proposed protocol is slightly better than that of the solution posed in Contribution 1. This is due to the fact that the solution of Contribution 1 selects a lesser number of edges resulting in more V2V collaborations through the lossy DSRC medium.

Fig. 4.9 and Fig. 4.10 shows the performances of all protocols with respect to metrics nvi and $PSNR$, respectively. It can be observed that these two metrics are directly related to the performance of pdr metric. For nvi metric, the percentage threshold is set as 50% for the experiments. As observed from Fig. 4.9 for all protocols, when the pdr metric deteriorates, nvi metric value gets increased, resulting in poor client experiences. Similarly, from Fig. 4.10 we observe that all protocols show poor client experiences where they have poor pdr metric performance. As expected, SVC_QOE outperforms others in both metrics, and VMaSC_LTE performs worst in all cases.

The above experiments show that though SVC_QOE provides the best client experience for video streaming, it is not suitable for cost-effective video streaming. On the other hand, though the decentralized edge selection technique of VMaSC_LTE can reduce the number of simultaneous LTE connections, it fails as a video streaming service. Despite having

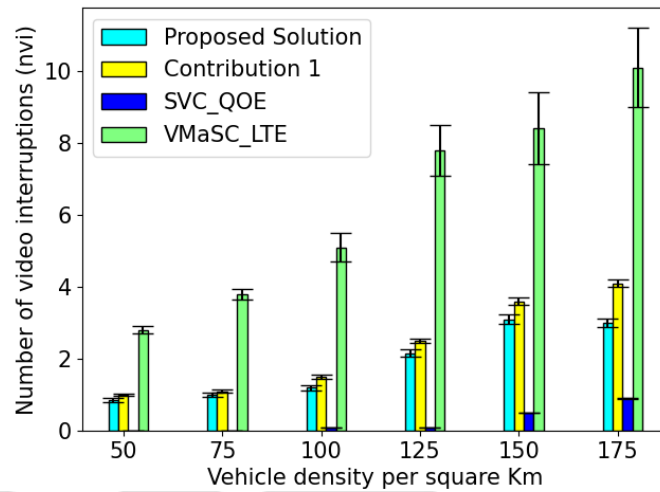


Fig. 4.9 No. of video interruptions vs Vehicle density

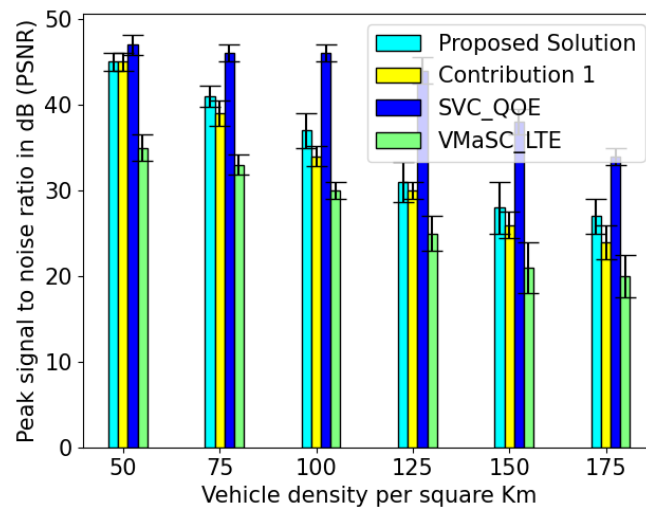


Fig. 4.10 Peak signal to noise ratio vs Vehicle density

a distributed edge selection technique, the proposed solution performs much better than VMaSC_LTE in QoE while lagging behind SVC_QOE by an acceptable margin. Though the solution of Contribution 1 does the most efficient edge selection, its performances on *pdr*, *nvi*, and *PSNR* metrics lag behind the proposed solution of this chapter. Being a distributed solution, the proposed solution relieves the CDN provider from having a high computing setup and handling numerous simultaneous flow requests. The proposed solution clearly outperforms the other protocols from the cost-effectiveness perspective.

4.3 Summary

In this work, a cost-effective solution is proposed for CDN providers to offer video streaming as an infotainment service over IoCVs. The solution has a distributed edge minimization scheme that is able to reduce the number of simultaneous LTE connections while ensuring time-constrained delivery of streaming video. The edge minimization task is formulated as the MSC problem and is solved using an adaption of the distributed version of the GSC algorithm for MSC. Unlike existing protocols, the solution assumes the presence of network layer vehicles that are unaware of any streaming application. Accordingly, a multicast protocol, namely DSS-CAST, is proposed, which specializes in streaming applications for dynamic ad-hoc networks like VANETs. Performance metrics are defined to evaluate the cost-effectiveness as well as client satisfaction for video streaming. The simulation results show that though the proposed solution marginally lags behind the solution of Contribution 1, it is the most cost-effective solution for CDN providers as it relieves the CDN providers from expensive server setups. Compared to other protocols, the proposed solution provides competitive service quality for live video streaming. Though the solution is able to minimize the number of simultaneous cellular network connections, the solution needs to be extended for scenarios where multiple V2I network options are available. The next chapter is devoted to this task.

Chapter 5

Distributed V2I network selection to maximize cellular traffic offloading

The previous chapter proposes a V2V collaboration solution for distributed edge vehicle selection for the scenario where the cellular network is the only V2I option for the vehicles. However, in an IoCV, vehicles may connect to the Internet through infrastructure nodes of heterogeneous network access technologies like RSUs of DSRC, BSs of cellular networks [122], and access points of Wifi networks [10]. The vehicles of IoCV are equipped with multi-RAT interfaces [11]. Infrastructure nodes of different network access technologies have their own pros and cons. Connecting to DSRC and Wifi network is free, whereas using C-V2X or cellular network incurs cost. In urban scenarios, cellular network operators usually deploy Wifi access points in congested areas to offload the cellular traffic and increase capacity [123]. However, intermittent deployment of Wifi access points with a small communication radius causes Internet connectivity disruptions and frequent handover problems for high-velocity vehicles. On the other hand, though the cellular coverage is wide, a few areas may remain out of cellular network coverage as blind spots. In the above-mentioned scenarios, vehicles can collaborate among themselves through the DSRC network to extend the effective coverage areas of Wifi access points or to cover cellular blind spots. In rural areas, Wifi access points are mostly unavailable, making V2V collaborations the only option for cellular traffic offloading. V2V collaborations are not possible where vehicle densities are very less, and cellular networks are the fallback option in those scenarios. So, the coexistence of all three different network options is very much essential for an IoCV to realize uninterrupted Internet connectivity. Subject to availability, a vehicle should be able to choose among the three network options. Though most of the existing works have provided network selection

solutions for two network options like {cellular, Wifi} [99], {cellular, DSRC}, and {Wifi, DSRC} [26, 27], network selection methods for three options of {cellular, Wifi, DSRC} has not been explored. Accordingly, this work proposes a novel solution for network selection for the scenarios where all three network options may coexist. However, the dynamic vehicular topology and intermittent availability of networks make the network selection task challenging for delay-sensitive applications, which is described next.

Based on application requirements, one network access technology may be preferred over others based on its pros and cons. Few delay-tolerant applications [124] may allow a vehicle moving out of Wifi access point coverage to wait for the next encounter with a Wifi access point. In contrast, delay-sensitive applications [125] require persistent Internet connectivity. Safety application messages are small [12], whereas infotainment applications usually need to distribute a large volume of data. Out of the infotainment services, the volume of video traffics is the highest. So, optimizing the volume of video traffic in video-based infotainment services is a requirement. However, streaming quality and timely content delivery must be ensured for live streaming services for client satisfaction. To fetch the content from the server, a vehicle should prefer to select a Wifi network over a cellular network to connect to the Internet. For V2V collaboration, a vehicle needs to select the DSRC network. Due to the highly dynamic vehicle topology, the selection of network options is subject to the availability of networks. As a result, network option selection needs to be dynamic and requires topology-specific computations. As live video streaming is delay-constrained, computations for network access technology selection need fast convergence.

Motivated by the above facts, this work proposes a distributed solution for V2I communication network selection for live video streaming over IoCV to maximize cellular traffic offloading. Without any ambiguity, this work interchangeably uses the phrase “*network selection*” to denote “*network access technology selection*”. The key contributions of this work are enlisted below:

- This work is the pioneer in proposing a distributed solution for network selection to offload cellular traffic for live video streaming services over IoCVs.
- This is the first time the network selection problem for vehicles accommodates three different network options together, which are cellular, Wifi, and DSRC.
- The task of network selection to maximize cellular traffic offloading is shown to be a NP-Hard problem by formulating the task as a centralized *integer linear programming problem* (ILPP).
- A lightweight distributed solution using a greedy method is proposed for cellular traffic offloading.

- Evaluation of the proposed scheme is done based on the aspects of cellular traffic offloading, service efficiency, and user satisfaction. Simulation results show that the proposed solution has the best service effectiveness for live video streaming over IoCVs.

The rest of the chapter is organized as follows. The existing related works are discussed in Section 5.1. The description of the proposed protocol is given in Section 5.4. The performance of the proposed solution is evaluated in Section 5.6, and the simulation results are presented. The work is summarized in Section 5.7.

5.1 Related Work

Network selection has been a topic of interest for the IoCV research community for the past few years. Heterogeneous network selections are also dealt with in vertical handover solutions. However, vertical handovers are controlled from the lower layers based on the metrics like signal strength, mobility, etc. For example, the work [126] has proposed an efficient vertical handover scheme between the cellular network and WLAN by using the metrics *faded signal to noise ratio*, *residual channel capacity*, and *connection lifetime*. Different network systems are under different administrative domains of different assigned IP addresses making seamless vertical handover a challenging task. The work [127] proposed a hybrid solution for seamless IP communication with the coexistence of *Proxy Mobile IPv6*(PMIPv6) and *Host Identity Protocol*(HIP). The work [128] also proposed a seamless vertical handover solution using *Mobile IPv6* (MIPv6) and PMIPv6. The work [129] provides a centralized vertical handover algorithm that jointly optimizes the data rate and network costs for vehicles. The collaboration between DSRC infrastructure nodes and cellular base stations is investigated in work [130]. A double auction game is formulated to decide optimum load-sharing strategies to maximize traffic. However, this work considers the traffic offloading in the opposite direction from overloaded DSRC infrastructure nodes to the cellular network.

Higher-layer network selection algorithms deal with selecting a suitable network access technology from a set of available options for each traffic flow, depending on application requirements or service objectives. The work [99] gives a network selection solution between LTE and Wifi with the objective of QoE maximization per user traffic flow. However, the work has proposed enhancements on LTE *Access Network Discovery and Selection Function* (ANDSF) standard to incorporate application-specific information and mobility prediction. The work [131] proposed a SDN based network selection algorithm for jointly

minimizing network access cost and delay for broadcast or dissemination services over vehicular networks. However, the solution is not suitable for streaming media as it deals with heterogeneous data items with caching provisions. The work [125] has proposed SDN based network selection scheme for delivery delay-constrained temporal data downloads for maximizing service ratio. An offloading strategy of delay-tolerant non-safety vehicular traffic is proposed by [124]. Here the assumption is that a vehicle may be under the coverage of multiple DSRC-based infrastructure nodes, and an SDN controller suggests the best infrastructure nodes for vehicles. The work [27] proposed a network selection algorithm for on-demand video traffic offloading. A storage time aggregated graph is used for video stalling minimization. However, the scheme uses pre-fetching techniques of video chunks, which cannot be used for live video streaming. Though the Contribution 1 [Chapter 3] of this thesis has proposed a network selection solution for live video streaming, it has considered only cellular and DSRC networks without considering the presence of Wifi networks.

All of the aforementioned works have proposed centralized solutions. A few of the existing distributed solutions for network selection is discussed next. The work [132] proposed a distributed network selection solution where individual vehicles solve the optimal stopping problem to decide network switching between cellular and WLAN to maximize their utility which is the weighted combination of cost and delay. However, the solution is not suitable for streaming services. The work [133] gave a distributed network selection solution that includes the possibilities of V2V collaborations as a coalition game among vehicles. The download costs and time of different types of networks are jointly considered to form the game and to decide strategies. However, the assumption is that the data items are heterogeneous in nature and can be cached. As a result, the solution is not applicable to live video streaming. The Contribution 2 [Chapter 4] of this thesis has proposed a distributive solution for live video streaming. However, the solution is based on cellular and DSRC networks without considering Wifi networks.

It is observed that the network selection problem with the coexistence of three networks of cellular, Wifi, and DSRC has not been explored in the literature.

5.2 System model

This work considers IoCVs, where vehicles are equipped with multi-RAT interfaces (i.e., cellular, Wifi, and DSRC). Cellular network operators deploy Wifi access points in some strategic locations where user densities are expected to be high, like near shopping malls,

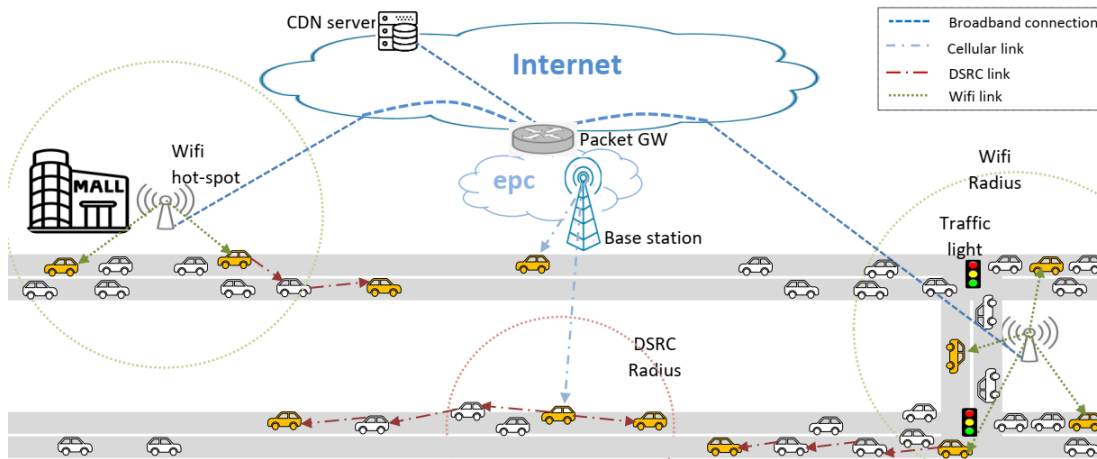


Fig. 5.1 A scenario of an IoCV with max 3-hop V2V collaborations

at important traffic junctions, etc., as shown in Fig. 5.1. Unlike cellular base stations, the coverage of Wifi access points is small and intermittent. However, as urban vehicle mobility patterns usually follow the *Intelligent Driver Model with Intersection Management (IDM-IM)* model, an individual vehicle's average dwelling time inside a Wifi access point is enough to consider Wifi as a stable network switching option. As explained in [18], the IDM-IM mobility pattern fragments a network topology into a number of clusters. In the clusters formed in highly congested areas or at traffic signal points, intra-cluster vehicles have high connectivity and link stability among themselves. The aforementioned properties make V2V collaborations an attractive option for traffic offloading.

In the considered scenario, a CDN provider is responsible for providing live video streaming services within an AOI. A few of the vehicles within AOI wish to avail of the live streaming content and become clients of the CDN provider, while all other vehicles are assumed as ordinary vehicles. The CDN client vehicles run the CDN client application. In Fig. 5.1, the client vehicles and ordinary vehicles are depicted in yellow and white color, respectively. It is assumed that the cellular network provider ties up with the CDN provider to encourage its clients to offload from the cellular network to Wifi networks. In return, the cellular network provider agrees to pay incentives to the CDN provider. To maximize the incentives, the CDN provider tries to increase cellular traffic offloading by extending effective Wifi coverage through DSRC V2V collaborations wherever possible. The task of the CDN provider is to design a network selection algorithm for its CDN client application such that individual client vehicles try their best to avoid using the cellular network without compromising the streaming experience. However, if vehicles far apart from each other avail the service where no Wifi is available, they have to fall back to the cellular network.

In rural areas, some places even fall outside of cellular coverage, creating some blind spots. In that case, if connected vehicles are available in the vicinity, the CDN provider may use V2V collaborations to extend the coverage for the client vehicles in the blind spots. This work assumes that all vehicles (client/ordinary) are well-behaved and they participate in V2V collaboration.

The live video stream is divided into segments, and each segment is divided into a number of packets. Video playback duration is divided into fixed playback intervals of Π . During a playback interval, the CDN server produces one segment of the video, which is distributed to all client vehicles and is played in the following playback interval while a freshly generated video segment is being distributed. As a result, the startup latency is fixed as Π , and the video playback is expected not to get halted in ideal conditions. The distribution latency is strictly constrained by Π as the corresponding segment is considered stale for distribution after one playback interval. Unlike the case of on-demand videos, the segments cannot be pre-fetched or cached by edges beforehand.

5.3 Problem formulation

Let the set of client vehicles is denoted as $C = \{c_0, c_1, c_2, \dots, c_{|C|-1}\}$. A vehicle has three network access technology selection options which are cellular, DSRC, and Wifi. A client vehicle can select only one of the networks at a given time to avail of streaming data. This work defines four network selection states ∇_i for a vehicle c_i which are *Wifi selected* (WS), *DSRC selected* (DS_{c_k} , where c_k denotes the streaming source), *cellular selected* (CS), and *not selected* (NS). A vehicle has pre-defined utilities for being in a state, and the summation of all vehicle utilities represents the utility of the CDN provider. Let the utilities of being in state WS, DS_{c_k} , CS and NS are U_{ws} , U_{ds} , U_{cs} , and U_{ns} . When a vehicle is in the NS state, it cannot receive any streaming content. As the NS state is undesirable for a vehicle, the utility value U_{ns} is the lowest among all. Accessing a Wifi network incurs no cost, and client vehicles can join a multicast group to receive the one-way data traffic from a Wifi access point simultaneously. Therefore, the Wifi network is the most preferred one, and being in WS state has the highest utility value U_{ws} . In case a vehicle chooses to avail streaming data through the DSRC network with multi-hop V2V collaborations, a few data packets may get lost due to the unreliable shared medium of the DSRC network. Accordingly, utility value U_{ds} is lesser than U_{ws} . On the other hand, though cellular network coverage is widespread and uses dedicated channels for communication, it makes the CDN provider lose incentives from the cellular network provider. As the objective of the CDN provider is to avoid cellular traffic as

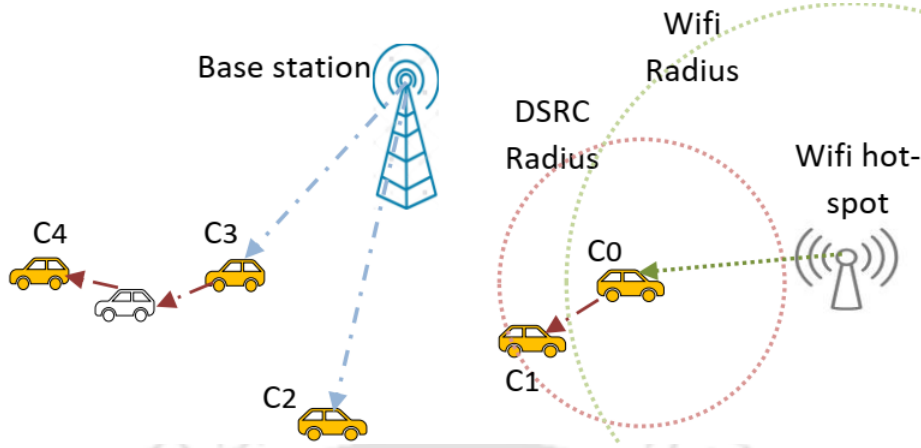


Fig. 5.2 A small scenario of max 2-hop V2V collaborations

much as possible, the utility value U_{cs} is lesser than the utility value U_{ds} . Accordingly, any choice of utility values must adhere to the following relation.

$$U_{ws} > U_{ds} > U_{cs} > U_{ns} \quad (5.1)$$

Based on the above relation, an increased utility of a CDN provider not only denotes the increased volume of offloaded traffic but also indicates better service quality experienced by vehicles. Therefore, a CDN provider tries to achieve better service effectiveness by increasing its utility. If a client vehicle chooses DSRC, it also needs to select a suitable peer client vehicle within its K -hop neighborhood as its streaming source. This work denotes a client vehicle as a streaming source if it avails streaming content directly from Wifi or cellular network and distributes the content to other client vehicles within its K -hop neighborhood using DSRC. Clearly, the selection of a streaming source implies the selection of DSRC access technology. An ordinary vehicle cannot become a streaming source as it is not running a CDN client application. The max hop count K is determined such that the data delivery delay of a K^{th} hop path does not exceed the delay constraints of live video streaming. Intermediate vehicles of a multi-hop path may be client vehicles or ordinary vehicles which participate in V2V collaborations. In case a client vehicle is in the coverage of more than one base station or more than one Wifi access point, a horizontal handover mechanism in the lower layer chooses the best one, which is out of this work's scope.

A utility vector \mathcal{U} of length $|C| + 3$ is defined such that the first $|C|$ elements u_j for $0 \leq j \leq (|C| - 1)$ represent the utility of being in DS state with client vehicles $c_j \in C$ as the streaming source. The next three elements $u_{|C|}$, $u_{|C|+1}$, and $u_{|C|+2}$ represent the utilities of

Table 5.1 Utility vector \mathcal{U} for Fig. 5.2 scenario

DS					WS	CS	NS
c_0	c_1	c_2	c_3	c_4			
U_{ds}	U_{ds}	U_{ds}	U_{ds}	U_{ds}	U_{ws}	U_{cs}	U_{ns}

being in WS, CS, and NS, respectively. Clearly, $\forall \{j | 0 \leq j \leq (|C| - 1)\} u_j = U_{ds}$, $u_{|C|} = U_{ws}$, $u_{|C|+1} = U_{cs}$, and $u_{|C|+2} = U_{ns}$. The utility vector is given in TABLE 5.1. The choice of network is subject to temporal availability due to the dynamic connectivity of IoCVs. An example scenario is given in Fig. 5.2, where Wifi is available in a partial region, whereas the entire region is under cellular network coverage. In the example, c_0 can choose any state, whereas c_2 has options to be in either CS or NS state as it is not in the coverage of Wifi and has not any vehicle within its communication range. In a given time instance T , let \mathcal{F} be the state selection availability matrix (binary) of size $|C| \times (|C| + 3)$. Each row $\mathcal{F}[i]$ of \mathcal{F} represents the available network selection state options for the client vehicle $c_i \in C$ at time T . The first $|C|$ elements of $\mathcal{F}[i]$, i.e. $f_{i,j}$ with $0 \leq j \leq (|C| - 1)$ represents the DS state selection availability of c_i with $c_j \in C$ as streaming source. A vehicle c_i can select a vehicle c_j as its streaming source only if c_j is within the K-hop neighborhood of c_i . The next three elements $f_{i,|C|}$, $f_{i,|C|+1}$, and $f_{i,|C|+2}$ represent the selection availability of WS, CS, and NS states respectively. An element $f_{i,j} \in \mathcal{F}$ is set to 1 if the j^{th} selection option is available for c_i by the aforementioned conditions, or is set to 0 otherwise. The \mathcal{F} table for the scenario of Fig. 5.2 is given in TABLE 5.2.

Table 5.2 Selection availability matrix \mathcal{F} for Fig. 5.2 scenario

	DS					WS	CS	NS
	c_0	c_1	c_2	c_3	c_4			
c_0	0	1	0	0	0	1	1	1
c_1	1	0	0	0	0	0	1	1
c_2	0	0	0	0	0	0	1	1
c_3	0	0	0	0	1	0	1	1
c_4	0	0	0	1	0	0	1	1

At any given time instance, vehicles may have multiple selection options available. So, the task is to choose network access technology for all vehicles such that the total utility of the CDN provider is maximized. A binary decision matrix \mathcal{D} of size $|C| \times (|C| + 3)$ is defined to represent the network selection decisions of all client vehicles. The row and column representation of \mathcal{D} is the same as \mathcal{F} . However, an element $d_{i,j} \in \mathcal{D}$ is set to 1 only if the j^{th} option is selected for c_i . The problem of network access technology selection can be formulated as an ILPP as given below.

$$\text{Maximize } \sum_{i=0}^{|C|-1} \sum_{j=0}^{|C|+2} (u_j \times d_{i,j}), \quad d_{i,j} \in \mathcal{D}, u_j \in \mathcal{U} \quad (5.2)$$

Subject to the constraints:

1. $\forall i, \sum_{j=0}^{|C|+2} f_{i,j} \times d_{i,j} \leq 1, \quad f_{i,j} \in \mathcal{F}$
2. $\forall i, \forall \{j | 0 \leq j \leq (|C| - 1)\}, \quad d_{i,j} + \sum_{k=0}^{|C|-1} d_{j,k} \leq 1$
3. $\forall i, \quad d_{i,i} = 0$
4. $\forall i, j \quad d_{i,j} \in \{0, 1\}$

The objective function defined in equation (5.2) is to maximize the total utility of the CDN provider by adding up individual vehicle utilities of network selection. Constraint 1 ensures that a vehicle selects at most one network option at a time, and the selection must be subject to availability. Constraint 2 prevents the streaming dependency loop among vehicles when DSRC is used. If a client vehicle c_i selects c_j as the streaming source, c_j should not select another client vehicle as its streaming source, and must select either Wifi or cellular network to avail streaming data. Constraint 3 prevents a client vehicle from choosing itself as its own streaming source. Constraint 4 ensures that all decision variables are binary. The decision matrix \mathcal{D} for the scenario of Fig. 5.2 is given in TABLE 5.3.

Table 5.3 Decision matrix \mathcal{D} for Fig. 5.2 scenario

	DS					WS	CS	NS
	c_0	c_1	c_2	c_3	c_4			
c_0	0	0	0	0	0	1	0	0
c_1	1	0	0	0	0	0	0	0
c_2	0	0	0	0	0	0	1	0
c_3	0	0	0	0	0	0	1	0
c_4	0	0	0	1	0	0	0	0

The formulated problem is a special case of ILPP, where all decision variables are binary. The number of decision variables of the formulated problem is $|C| \times (|C| + 3) = |C|^2 + 3|C|$. As ILPP is known to be NP-hard, the large number of decision variables rules out the possibility of an exact solution. Additionally, due to dynamic vehicular connectivity, the availability matrix \mathcal{F} keeps changing over time. Consequently, the decision matrix \mathcal{D} also needs to be frequently re-calculated based on current \mathcal{F} . Even with a greedy or heuristic

approach, any centralized solution may incur significant computational overhead due to the requirement of time-bound computation for delay-sensitive applications. Therefore, this work proposes a lightweight distributed solution described in the next section.

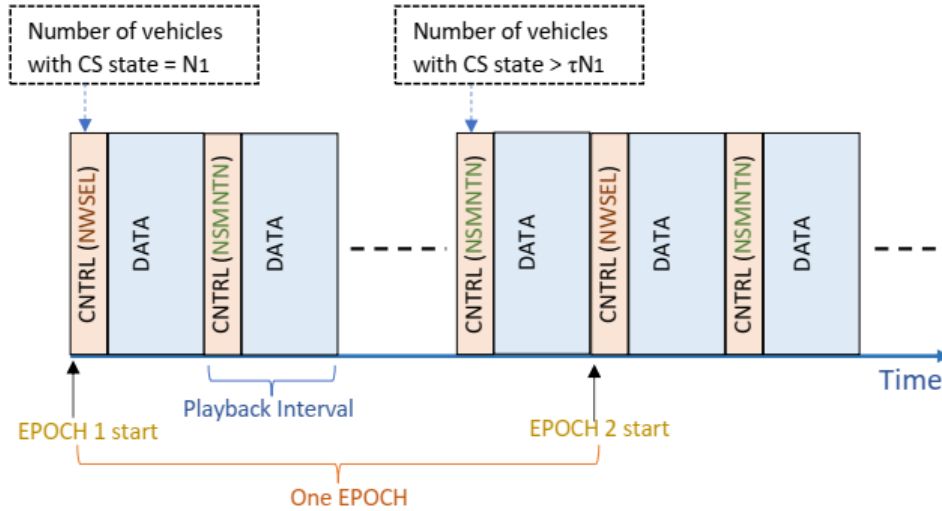


Fig. 5.3 Protocol timeline

5.4 Proposed protocol

This work proposes a distributed algorithm for the heterogeneous network selection problem to greedily maximize the service utility for CDN providers. Due to the dynamic topology, a selected network option for a client vehicle may become stale/unavailable over time, disrupting the streaming service. For example, if two vehicles are moving with 100 km/hr velocity in opposite directions (relative velocity 200 km/hr), the relative movement between them in 1 s is approximately 55 meters. Considering the DSRC communication radius and Wifi radius as 100 m, this movement may disrupt connectivity. On the contrary, if two vehicles move in the same direction with 100 Kms/hr and 120 km/hr, respectively (relative velocity 20 km/hr), the relative movement in 1 s is only 5.5 meters, and the connectivity of the two vehicles can be considered static within 1 sec. Accordingly, the proposed solution allows V2V collaboration between two vehicles only if their relative velocity is below a chosen threshold ϑ , and selects the playback interval duration Π small enough such that the network connectivity remains static within Π .

The playback interval Π is further divided in control (CNTRL) phase of Π_c duration and data (DATA) phase of Π_d duration such that $\Pi = \Pi_c + \Pi_d$, where $\Pi_c \ll \Pi_d$. In the

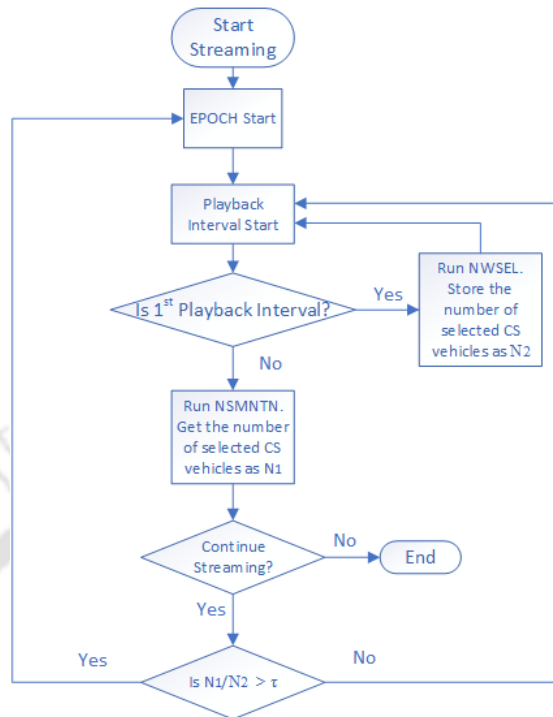


Fig. 5.4 Protocol Flowchart

CNTRL phase, vehicles distributively select network interfaces, and in the DATA phase, a vehicle receives video data through the selected network. Once a vehicle joins the client group to avail of a streaming service, it receives the timing information of the CNTRL and DATA phase from the CDN server. CDN server keeps track of the number of client vehicles availing of the service. The network selection task in the CNTRL phase can be either a fresh *network selection* (NWSEL) or *network selection maintenance* (NSMNTN), which are described subsequently in Section 5.4.1 and Section 5.4.2 respectively. NWSEL makes an optimum network selection that maximizes the utility of the CDN provider. However, the optimum result comes at the cost of significant control message overhead. In a highly dense vehicular topology, this control message overhead may congest the shared DSRC medium. If Π and ϑ are chosen sufficiently small, the network topology change is not significant after one playback interval. In that case, making a fresh optimum network selection is redundant. To reduce the control message overhead, NSMNTN is run in the CNTRL phases of the few subsequent playback intervals to take care of the minor connectivity changes. NSMNTN is an algorithm with significantly less control overhead compared to NWSEL. The NSMNTN greedily tries to select a network only if the current selection has become stale/unavailable or a better network option is available. However, after a few playback intervals, as the network connectivity changes significantly, the NSMNTN makes inefficient network selection, which

can be mitigated in the following way. At the end of each CNTRL phase, the vehicles in the CS state inform the CDN server about their state. The CDN server keeps track of the number of CS vehicles in NWSEL and subsequent NSMNTN runs. In case the number of CS vehicles in a NSMNTN run increases more than a certain ratio τ compared to the number of CS vehicles given by the last NWSEL run, the CDN server instructs all the client vehicles to run NWSEL to compute a fresh optimal network selection. The interval between two successive NWSEL tasks is termed an EPOCH in this work. Fig. 5.3 depicts the timeline of the proposed solution. The EPOCH duration t_e is $r \times \Pi$ where r is selected dynamically depending on the performance of NSMNTN. The flowchart of the proposed solution is illustrated in Fig. 5.4.

In the DATA phase, the vehicles with WS or CS state receive one segment of live video streaming data from the Internet using Wifi or cellular network respectively. Vehicles with DS_{c_k} state receive data from the c_k through multi-hop V2V collaboration. However, the vehicles which are not interested in a streaming service do not run client applications in them and act as ordinary router nodes. Consequently, the client vehicles may have ordinary router vehicles as intermediate hops in the multi-hop path connectivities among themselves. Accordingly, V2V collaborations must happen in the network layer through multicast techniques. This work ignores the presence of selfish and malicious nodes and assumes that all vehicles cooperate in routing and forwarding packets. In the existing MANET multicast protocols, all member nodes get associated with all the multicast source nodes. However, for live video streaming, the content is identical for all multicast sources, making it sufficient for a member node to get associated with any one of the multicast source nodes. Consequently, a multicast node needs to select a suitable source node. In existing multicast protocols, there is no provision for source selection. Moreover, they are not suitable for delay-constrained delivery in highly dynamic networks. This work uses the DSS-CAST multicast protocol proposed in the thesis Contribution 2 [Chapter 4]. Vehicles form groups by getting associated with the respective source vehicle c_k as multicast root.

5.4.1 Network selection (NWSEL)

For the network selection task, a modified version of the protocol proposed in Contribution 2 of this thesis is used. In the CNTRL phase, client vehicles distributively select suitable network options, and in the DATA phase, a video segment is fetched through the selected network. Let the observed DSRC data rate is Υ_D , and the video data rate δ . As shown in Section 4.1.2, to distribute all the required data packets within the time constraint of the

Algorithm 4 NWSEL algorithm (runs in individual client vehicle)

Input: max hop count K **Output:** network selection state ∇_i of the vehicle

```

1:  $\nabla_i \leftarrow NS$ 
2: bool CNDT  $\leftarrow FALSE$ 
3: if  $\mathcal{F}(WS) == TRUE$  then
4:    $\nabla_i \leftarrow WS$ 
5: end if
6: Multicast  $\nabla_i$  to client vehicles within  $K$ -hop neighborhood
7: if receive message from a neighbor  $c_k$  with  $\nabla_k == WS$  then
8:    $\nabla_i \leftarrow DS_{c_k}$ 
9: end if
10: if receive message from a neighbor  $c_m$  with  $\nabla_m == NS$  then
11:   CNDT  $\leftarrow TRUE$ 
12: end if
13: while  $\nabla_i == NS \parallel (\nabla_i == DS_{c_k} \ \&\& \ CNDT == TRUE)$  do
14:   Multicast  $\nabla_i$  to client vehicles within  $K$ -hop neighborhood
15:   if no message from  $K$ -hop neighborhood with  $\nabla_k == NS$  then
16:     CNDT  $\leftarrow FALSE$ 
17:   end if
18:    $\phi_i \leftarrow 0$ 
19:   Calculate  $\phi_i$  as the  $K$ -hop neighbor count with  $\nabla == NS$ 
20:    $++ \phi_i$  // as it can be streaming source of itself
21:   Multicast  $\phi_i$  to client vehicles within  $2K$ -hop neighborhood
22:   if  $\phi_i$  is the highest among all the  $\phi$  from  $2K$ -hop neighbors then
23:      $\nabla_i \leftarrow CS$ 
24:   end if
25:   Multicast  $\nabla_i$  to client vehicles within  $K$ -hop neighborhood
26:   if receive message from neighbor  $c_k$  with  $\nabla_k == CS$  then
27:      $\nabla_i \leftarrow DS_{c_k}$ 
28:   end if
29: end while

```

duration of the DATA phase, client vehicles can do maximum K hops V2V collaboration where K is bounded by the following equation.

$$K = \left\lfloor \frac{(\Pi - \Pi_c)}{\Pi} \times \frac{\Upsilon_D}{\delta} \right\rfloor \quad (5.3)$$

Vehicle connectivities are dynamic due to the high velocities of vehicles. However, if the relative velocity between two vehicles is small, the link between two vehicles remains stable for some duration. A suitable relative velocity threshold ϑ is chosen accordingly such that only the vehicles with relative velocity among them lower than the threshold ϑ are allowed for multi-hop V2V collaborations among themselves.

The NWSEL task is performed by each vehicle distributively as given in Algorithm 4. A client vehicle c_i checks if it is within Wifi coverage currently. If yes, it sets its network selection state ∇_i to WS, and otherwise to NS. All client vehicles send their network selection state ∇_i to their respective K -hop neighbors. If a client vehicle c_i receives a message with $\nabla_k = \text{WS}$ from c_k , it sets its own ∇_i to DS_{c_k} as it can avail streaming data from c_k through DSRC. It also sets a flag named CNDT as TRUE if it receives messages from one or more client vehicles with $\nabla = \text{NS}$. CNDT flag denotes that a client vehicle is a candidate to select CS state to provide streaming data to K -hop neighborhood client vehicles of NS state. The next few steps are run in iterations. Client vehicles with $\nabla = \text{NS}$ or $\nabla = \text{DS}_{c_k}$ with CNDT as TRUE exchange ∇ within K -hop neighborhood. A client vehicle calculates its fitness score ϕ_i , which is denoted by the number of client vehicles within its K -hop neighborhood with $\nabla = \text{NS}$. If a client vehicle finds no neighbor within K -hop with $\nabla = \text{NS}$, it sets CNDT as FALSE. Client vehicles share their calculated ϕ to their respective $2K$ -hop neighborhood. Receiving all the ϕ from $2K$ -hop neighborhood, a client vehicle c_i sets ∇_i to CS if its own ϕ_i is highest among the received all ϕ , and declares ∇_i to its K -hop neighborhood. Receiving a message with $\nabla_k = \text{CS}$ from c_k , a client vehicle c_i sets its state $\nabla_i = \text{DS}_{c_k}$. With the aforementioned steps, one NWSEL iteration ends, and only the remaining client vehicles with $\nabla = \text{NS}$, or $\nabla = \text{DS}_{c_k}$ with flag CNDT = TRUE, participates in the next iteration. Iterations end when no client vehicle is in state $\nabla = \text{NS}$. At the end of the NWSEL algorithm, client vehicles with CS state report their state to the CDN server.

5.4.2 Network selection maintenance (NSMNTN)

Among the four states of a vehicle, the NS state is the only unstable state, as vehicles can not receive streaming data in that state, and therefore try to come out of this state. The other three states, DS_{c_k} , WS, and CS, are stable states. The set of all client vehicle states $\nabla_S = \{\nabla_0, \nabla_1, \dots, \nabla_{|C|-1}\}$ denotes the state of the entire system. Accordingly, the system has $4^{|C|}$ states, out of which $3^{|C|}$ states are stable and $4^{|C|} - 3^{|C|}$ states are unstable. The utility of the system state $\mathcal{U}(\nabla_S)$ is defined as $\sum_{i=0}^{|C|-1} \mathcal{U}(\nabla_i)$. The system becomes unstable when one or more vehicles go to the NS state. System instability may occur due to the mobility of vehicles. The initial state of the system ∇_S^{ini} for the current run of the NSMNTN algorithm depends on the outcome of the last run of the NSMNTN algorithm, and one or more events occurred during the data phase of the last playback interval. All possible events are enlisted below.

- **Event1:** Vehicles moving outside of a Wifi zone.

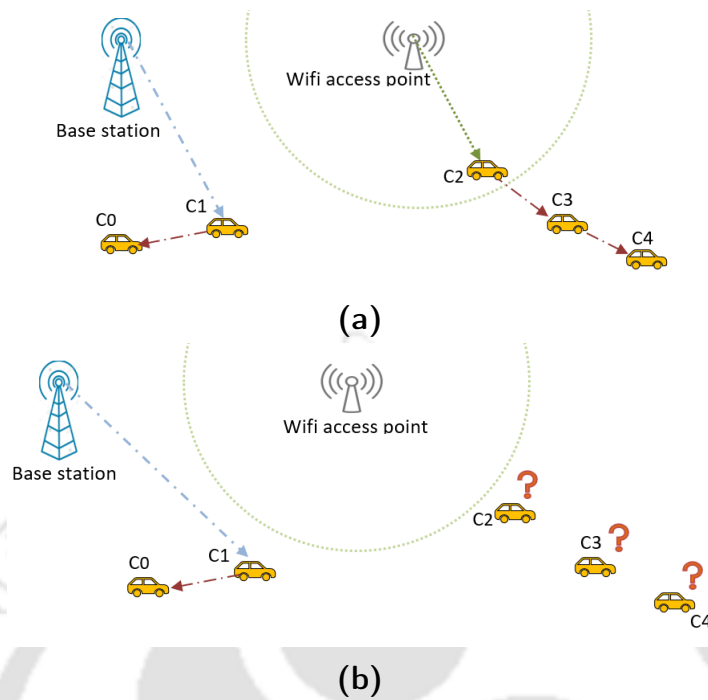


Fig. 5.5 Vehicle state transitions and system instability because of Event1 experienced by C2

- **Event2:** Vehicles moving inside of a Wifi zone.
- **Event3:** Breakage of one or more V2V multi-hop path connectivity due to vehicle mobility.
- **Event4:** Meeting of two or more streaming source vehicles within K -hop distance.

The NSMNTM algorithm (refer Algorithm 5) is run in individual client vehicles in rounds, where the first 3 rounds are mandatory. The algorithm terminates in a client vehicle when the vehicle is not in NS state after 3rd round. Each round comprises one computation round (lines 3-38) and one communication round (line 39). In the computation round, a client vehicle selects a network option based on current availability. In the communication round, a client vehicle makes its K -hop neighborhood aware of its current network section decision. Once the states of the K -hop neighborhood are known, if a vehicle finds a network option with more utility compared to its currently selected network, it switches to that network. However, the changed network selection state of a vehicle may affect other client vehicles' network selection decisions. All possible state transitions for each of the aforementioned events are described below.

- **Transitions for Event1:** If a client vehicle c_i experiences Event1, it goes to NS state (lines 12,13) in the 1st round. As a result, all the client vehicles that selected c_i as their

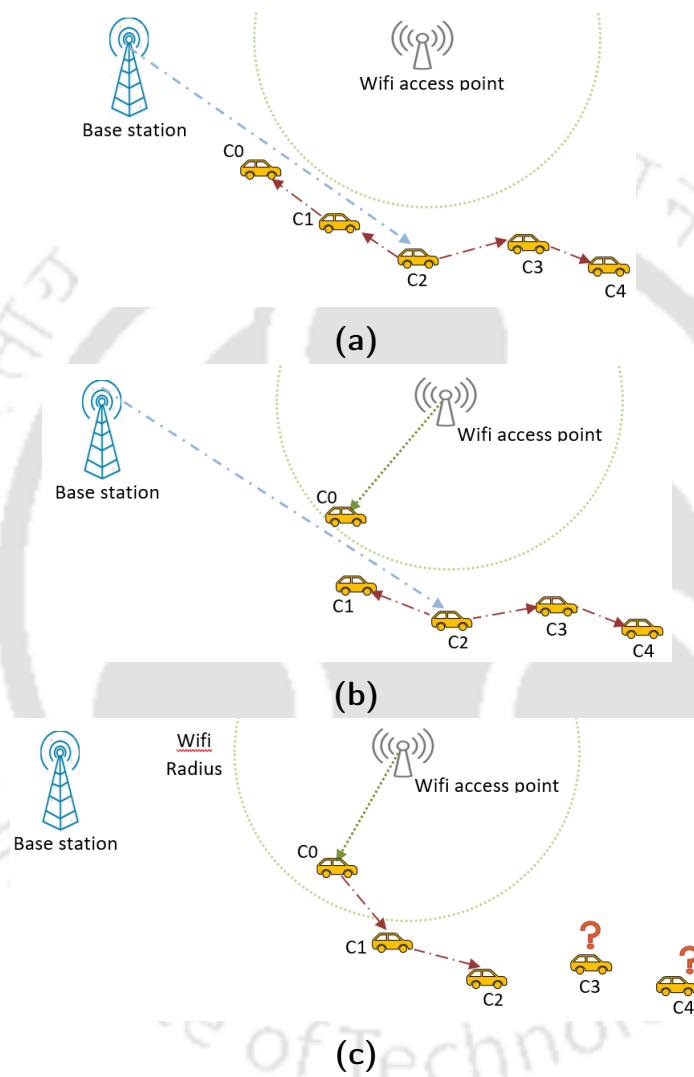


Fig. 5.6 Vehicle state transitions and system instability because of Event2 experienced by C0

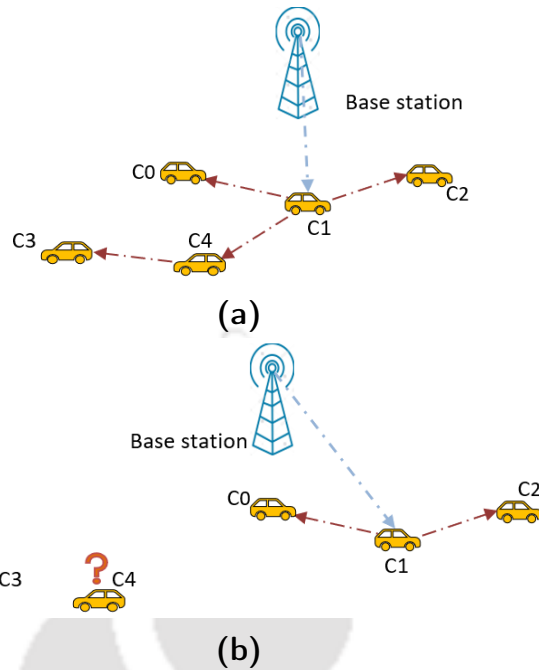


Fig. 5.7 Vehicle state transitions and system instability because of Event3 experienced by C4

streaming source also go to NS state (lines 18,19) in the 2^{nd} round. The situation is shown in Fig. 5.5.

- **Transitions for Event2:** If a client vehicle c_i experiences Event2, it switches to WS state (lines 10,11) to maximize its utility in the 1^{st} round. If the initial state of the c_i is NS or CS, no other client vehicles are affected. Let us assume the client vehicle's initial state is DS_{c_k} , and its streaming source is c_k with state CS. As c_i switches to WS, c_k switches state from CS to DS_{c_i} (lines 15,16) selecting c_i as streaming source to increase its utility in the 2^{nd} round. However, the state transition of c_k affects all the client vehicles with DS_{c_k} state, which earlier selected c_k as their streaming source, forcing them to move to NS state (lines 18,19) in the 3^{rd} round. The scenario is demonstrated in Fig. 5.6.
- **Transitions for Event3:** If a client vehicle c_i experiences Event3, it switches to NS state (lines 18,19) in the 1^{st} round. Fig. 5.7 is an example of the situation.
- **Transitions for Event4:** If two streaming source vehicles c_i and c_k with CS state come within their K -hop neighborhood, one of them, lets say c_i switches state from CS to DS_{c_k} (lines 15,16) selecting c_k as its streaming source in the 1^{st} round. As a result, all vehicles which earlier selected c_i as the streaming source are forced to move to NS state (lines 18,19) in the 2^{nd} round. An example scenario is shown in Fig. 5.8.

Algorithm 5 NSMNTN (runs in client vehicle $c_i \in C$)

Input: current state ∇_i , K -hop client vehicle neighbor list $c_i^{nbr} \in C$, neighbor state list $\nabla_i^{nbr} \subseteq \{\nabla_0, \nabla_1, \dots, \nabla_{|C|-1}\}$, neighbor priority list $\mathcal{P}_i^{nbr} \subseteq \{\mathcal{P}_0, \mathcal{P}_1, \dots, \mathcal{P}_{|C|-1}\}$, network availability function $\mathcal{F}()$,
Output: selected network state ∇_i

```

1: itr  $\leftarrow$  0
2: while itr  $\leq$  3 ||  $\nabla_i ==$  NS do
3:   ++ itr
4:    $\mathcal{P}_i \leftarrow$  0
5:   for all  $\nabla_i \in \nabla_i^{nbr}$  do
6:     if  $\nabla_i ==$  NS then
7:       ++  $\mathcal{P}_i$ 
8:     end if
9:   end for
10:  if  $\nabla_i \neq$  WS &&  $\mathcal{F}(\text{WS}) ==$  TRUE then
11:     $\nabla_i \leftarrow$  WS
12:  else if  $\nabla_i ==$  WS &&  $\mathcal{F}(\text{WS}) ==$  FALSE then
13:     $\nabla_i \leftarrow$  NS
14:  else if  $\nabla_i ==$  CS then
15:    if  $\exists c_k \in c_i^{nbr}, \nabla_k \in \nabla_i^{nbr} : (\nabla_k ==$  WS ||  $(\nabla_k ==$  CS &&  $\mathcal{P}_i <$   $\mathcal{P}_k))$  then
16:       $\nabla_i \leftarrow$  DS $_{c_k}$ 
17:    end if
18:  else if  $\nabla_i ==$  DS $_{c_j}$  &&  $\mathcal{F}(\text{DS}_{c_j}) ==$  FALSE then
19:     $\nabla_i \leftarrow$  NS
20:  else if itr  $\geq$  3 &&  $\nabla_i ==$  NS then
21:    if  $\exists c_k \in c_i^{nbr}, \nabla_k \in \nabla_i^{nbr} : (\nabla_k ==$  WS ||  $\nabla_k ==$  CS) then
22:       $\nabla_i \leftarrow$  DS $_{c_k}$ 
23:    else
24:      bool  $h \leftarrow$  TRUE
25:      for all  $\mathcal{P}_j \in \mathcal{P}_i^{nbr}$  do
26:        if  $\mathcal{P}_i <$   $\mathcal{P}_j$  then
27:           $h \leftarrow$  FALSE
28:          break;
29:        else if  $(\mathcal{P}_i == \mathcal{P}_j)$  &&  $i <$   $j$  then
30:           $h \leftarrow$  FALSE
31:          break;
32:        end if
33:      end for
34:      if  $h ==$  TRUE then
35:         $\nabla_i \leftarrow$  CS
36:      end if
37:    end if
38:  end if
39:  Multicast  $\nabla_i$  and  $\mathcal{P}_i$  to client vehicles within  $K$ -hop neighborhood
40: end while

```

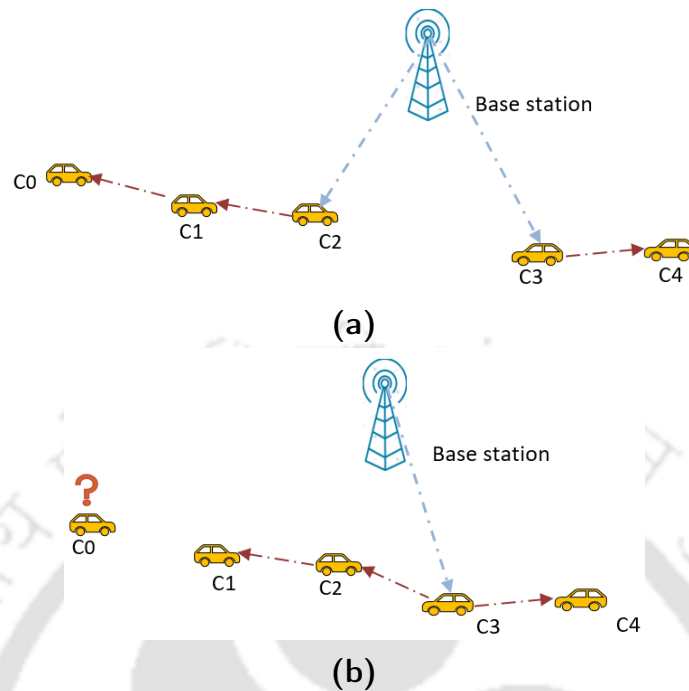


Fig. 5.8 Vehicle state transitions and system instability because of Event4 experienced by C2

As the small duration of a playback interval results in small displacements of vehicles during a playback interval, we assume that the events are consistent. For example, once a vehicle moves inside (outside) of a Wifi zone at the beginning of a playback interval, it remains inside (outside) of that Wifi zone during that interval. Similarly, if a multi-hop path is found working (broken) at the beginning of a playback interval, it continues to work (remains broken) for the rest of that playback interval. It is also assumed that a vehicle has at least one of the three network options available at any given time. Based on the above assumptions, the following two lemmas can be proved.

Lemma 1: Starting from any one of the possible $4^{|C|}$ initial states, the NSMNTM algorithm puts the system to one of the $3^{|C|}$ stable states within finite ε number of rounds where $3 < \varepsilon \leq |C|$.

Proof: From the shown state transitions of Event1, Event2, Event3, and Event4, it is clear that a client vehicle can move to NS state in 1^{st} , 2^{nd} , or 3^{rd} round of the NSMNTM algorithm. After the 3^{rd} round, no client vehicle goes to NS state. As the events are consistent for one playback interval, the conditions of lines 10 and 12 never become true after 1^{st} round. The condition of line 15 never becomes true after 2^{nd} round, and the condition of line 18 never

becomes true after 3^{rd} round. Consequently, after the 3^{rd} round, only lines 21-38 of the computation round are executed till the client vehicle moves out of the NS state. From 4^{th} round onwards, at least one client vehicle moves from NS state to either CS state in line 35 or DS_{c_k} state in line 22. Once a vehicle moves to a stable state, it never moves to the unstable NS state in the algorithm as the events are consistent. Consequently, within $|C|$ rounds, all client vehicles are bound to move to a stable state, putting the entire system into a stable state. Therefore, the required number of iterations ϵ is finite, and $3 < \epsilon \leq |C|$. However, in dense vehicle scenarios, once a vehicle c_k moves to CS state in a round, all of its K-hop neighbors move to DS_{c_k} in the next round. As a result, the upper bound of ϵ is much lower in practical scenarios. Moreover, vehicular topology is mostly fragmented in disjoint clusters, with 90% clusters having a very less number of vehicles [19], which brings down the upper limit of ϵ significantly. \square

Lemma 2: *If the system is in one of the $3^{|C|}$ stable states, it remains in the stable state if no external event occurs.*

Proof: The system goes to an unstable state only if one or more client vehicles move to unstable states. It is shown that a client vehicle can go to an unstable state only by external events Event1, Event2, Event3, or Event4. In the absence of any of these external events, the conditions of lines 10, 12, 15, and 18 do not satisfy and prevent a client vehicle from switching to an unstable state. As a result, the system continues to be in a stable state. \square

5.5 Protocol analysis

The offloading performance, computation complexity, and control packet overhead of the proposed solution are analyzed in this section.

5.5.1 Offloading performance

The proposed solution runs either NWSEL or NSMNTM algorithm in the CNTRL phase of each playback interval. The offloading performance of NWSEL is optimum (with an approximation ratio), and the analysis can be found in [134]. On the other hand, the NSMNTM algorithm greedily maximizes system utility, implying greedily maximized cellular traffic offloading. In each computation round, based on network availability, a client vehicle always greedily selects a stable network state ∇_i with the highest utility. After round

three, once a ∇'_i of utility $\mathcal{U}(\nabla'_i)$ is selected by c_i , it never selects a ∇''_i with utility $\mathcal{U}(\nabla''_i)$ in the future rounds such that $\mathcal{U}(\nabla'_i) < \mathcal{U}(\nabla''_i)$. As the system utility $\mathcal{U}(\nabla_S) = \sum_{i=0}^{|C|-1} \mathcal{U}(\nabla_i)$, NSMNTM algorithm greedily maximizes system utility $\mathcal{U}(\nabla_S)$.

5.5.2 Computation complexity

Being a distributed algorithm, the proposed solution is run in individual client vehicles. The number of computational steps of NWSEL and NSMNTN algorithms depends on the neighbor counts of individual client vehicles. Accordingly, the computational complexity of the proposed solution running in client vehicle c_k is $\mathcal{O}(Nbr(c_k))$ where $Nbr(c_k) \in C$ is the neighbor set of c_k . On the contrary, a centralized approximation solution has the computational complexity of $\mathcal{O}(|C|^3)$ as shown in Section 3.3.2 of Chapter 3.

5.5.3 Control overhead

NSMNTN algorithm is lightweight regarding control packet overhead compared to the NWSEL. There are three communication rounds in each NWSEL iteration (lines 13 to 29 of Algorithm 4). In the first and third communications rounds, messages are forwarded up to K hops, while in the second round, messages are forwarded up to $2K$ hops. On the other hand, each NSMNTN iteration has only one communication round where the messages are forwarded up to K hops. Let vehicle density is λ per km^2 . Let the radius of DSRC communication is Φ_{dsrc} meters which gives the K hop and $2K$ hop communication areas \mathcal{A}_K and \mathcal{A}_{2K} as $3.141 \times (\frac{K \times \Phi_{dsrc}}{1000})^2 \text{ km}^2$ and $3.141 \times (\frac{2K \times \Phi_{dsrc}}{1000})^2 \text{ km}^2$ respectively. Accordingly, we get the number of vehicles present within \mathcal{A}_k and \mathcal{A}_{2K} as $(\lambda \times \mathcal{A}_K)$ and $(\lambda \times \mathcal{A}_{2K})$ respectively. If the ratio of client vehicles and ordinary vehicles is $x : y$, then the average number of K -hop neighbor client vehicles \mathcal{N}_K of a client vehicle can be calculated as $(\lambda \times \mathcal{A}_K \times \frac{x}{x+y})$. Similarly, the average number of $2K$ -hop neighbor client vehicles \mathcal{N}_{2K} is calculated as $(\lambda \times \mathcal{A}_{2K} \times \frac{x}{x+y})$. Clearly, $\mathcal{N}_{2K} = 4 \times \mathcal{N}_K$. For simplicity, all message sizes of NWSEL and NSMNTN are assumed to be the same as φ bits. In a NWSEL iteration, a client vehicle generates 3 messages. Two of them are forwarded by \mathcal{N}_K client vehicles, and the remaining one is forwarded by \mathcal{N}_{2K} client vehicles. As a result, the total number of bits \mathcal{B}_{nwsel} transmitted for the control messages generated by one client vehicle is given by $((2 \times \mathcal{N}_K \times \varphi) + (\mathcal{N}_{2K} \times \varphi))$. Substituting value of \mathcal{N}_{2K} with \mathcal{N}_K gives $\mathcal{B}_{nwsel} = 6 \times \mathcal{N}_K \times \varphi$. On the other hand, in a NSMNTN iteration, a client vehicle generates one control message, which is forwarded by \mathcal{N}_K client vehicles. Accordingly, the total number of bits \mathcal{B}_{nsmntn}

transmitted for the control messages generated by one client vehicle is given by $\mathcal{N}_K \times \varphi$. Clearly, $\mathcal{B}_{nwsel} = 6 \times \mathcal{B}_{nsmntn}$. In other words, one NWSEL iteration imposes six times more control packet overhead compared to one NSMNTN iteration. However, it is mentionable that in practical scenarios, the number of participating vehicles is reduced after each iteration as few vehicles move out from the unstable NS state to any of the stable states. As a result, a lesser number of messages are forwarded, reducing the control traffic volume significantly. Let us assume for both NWSEL and NSMNTN, the average number of iterations is the same as \mathcal{J} . As shown in Fig. 5.4, the first playback interval of an EPOCH runs the NWSEL algorithm. If an EPOCH contains γ playback intervals, the efficiency \mathcal{E} of the proposed solution can be formulated as below.

$$\mathcal{E} = \frac{\gamma \times \mathcal{B}_{nwsel} \times \mathcal{J}}{(\mathcal{B}_{nwsel} + (\gamma - 1) \times \mathcal{B}_{nsmntn}) \times \mathcal{J}} \quad (5.4)$$

Substituting value of \mathcal{B}_{nwsel} with \mathcal{B}_{nsmntn} in the above equation we get

$$\mathcal{E} = \frac{6\gamma}{5 + \gamma} \quad (5.5)$$

In the worst case, \mathcal{E} becomes 1.71 when γ is 2, and in the best case, $\mathcal{E} \rightarrow 6$ for large values of γ .

5.6 Performance Evaluation

This section is organized into three parts. The simulation environment parameters are described in Section 5.6.1. Section 5.6.2 defines the performance metrics. Section 5.6.3 discusses the simulation results based on the defined metrics.

5.6.1 Simulation environment and parameters

The performance of the proposed solution is evaluated on a two-dimensional urban scenario with IDM-IM [18] vehicle mobility pattern. A road topology (refer Fig. 5.9) of $2 \times 3 = 6$ Km² of Bucks County, Pennsylvania, 19056, United States is received using *OpenStreetMap* tool [135]. The vehicle mobility traces on the given road topology are generated using the SUMO tool [120]. Simulation tool NS3 (version 3.36.1) [110] is used to implement the

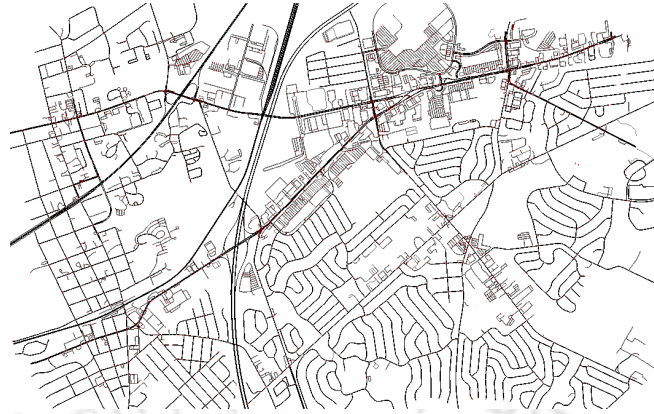


Fig. 5.9 Urban irregular road topology of Bucks County, Pennsylvania, 19056, United States (40.1521, -74.9297, 40.1821,-74.8655)

protocol. The SUMO-generated vehicle mobility traces are imported in the NS3. Wifi access points are added in major traffic junctions with a total coverage area of less than 10% off the total area of the scenario. TABLE 5.4 enlists the simulation parameters used in the experiments. As given in [121], in the Indian road conditions, the average velocity difference among same-direction vehicles in a free-speed model is within 50 km/hr. Accordingly, ϑ is varied from 8m/s (29 km/hr) to 15 m/s (54 km/hr) so that all possible same-lane V2V collaborations can be included. With ϑ 15 m/sec, in 3 secs, the relative movement between two vehicles is 45 meters which is almost half of the DSRC transmission range of 100 meters. Therefore, the upper value of Π is set to 3 secs for the experiments. If Π is set too small (less than 1 sec), the control phase overhead becomes considerable compared to the data phase, and the video segment size needs to be very small. Considering the above facts, simulation experiments are performed with epoch duration t_{epoch} of 1, 2, and 3 secs. For experimentation, the value of K is set as 5. Depending on the various CNTRL message sizes, DSRC data rate, and K , the CNTRL phase duration is calculated as shown in Section 4.1.6 of Chapter 4. H.264 videos are organized into a number of NALUs (refer Section 2.2.2) which can be transmitted independently. In this experimentation, the trace and the offset trace of the video "Silence of the Lambs" are used [4]. The trace file gives the sequences of NALUs, their corresponding sizes, and the corresponding PSNR values for the video sequence. In case the n^{th} frame is not received correctly at the receiver and is replaced by the last $(n - i)^{th}$ correctly received frame, the offset trace gives the PSNR $_n^i$ value. The ratio of client nodes and ordinary nodes is kept at 1:3 in all experiments.

Table 5.4 Simulation parameters

Parameter	Description	Value
AOI	Area of interest	6 km ²
λ	Vehicle density	50-150 vehicles/km ²
ϑ	Relative velocity threshold	8 m/s-15 m/s
Φ_{dsrc}	DSRC communication radius	100 m
Φ_{wifi}	Wifi communication radius	100 m
Υ_{lte}	LTE download speed	25 Mbps
Υ_{dsrc}	DSRC data rate	24 Mbps
Υ_{wifi}	Wifi data rate	24 Mbps
Π	Playback duration	1s -3 s
K	Max DSRC hop collaboration	5
T_s	Simulation Duration	50 s

5.6.2 Performance metrics

The proposed solution is assessed based on traffic offloading performance, protocol efficiency, user satisfaction, and overall service effectiveness. User satisfaction for video services over IP networks can be assessed by subjective or objective parameters [115]. Subjective parameters like MOS are based on individual human perceptions, whereas objective parameters like PSNR, startup latency, and buffering ratio [116] can be easily quantified and controlled. As mentioned in Section 5.2, the startup latency is fixed as Π . Packet delay and jitters are consumed by the playback buffer. Packets delayed more than the DATA phase is considered lost and is included in plr metric. Accordingly, the following performance metrics are defined to evaluate the proposed solution.

- **Offloaded cellular traffic ratio ($octr$):** The metric $octr$ represents the ratio of offloaded cellular traffic with respect to the total traffic demand. As the volume of demanded traffic is the same for all client vehicles, $octr$ is calculated as the ratio of client vehicles that are not receiving data through the cellular network with respect to the total number of client vehicles. Let C represent the set of client vehicles, and C_o represent the set of client vehicles that are receiving offloaded traffic through either Wifi or DSRC. Then the $octr$ metric for one playback interval is given by the following equation.

$$octr = \frac{|C_o|}{|C|} \times 100 \quad (5.6)$$

The average value of $octr$ is taken for all playback duration.

- **Control packet overhead (cpo):** This metric is used to measure the solution's efficiency in reducing control packet overhead. The metric cpo denotes the cost, as well

as the ratio of control traffic volume (Γ_C) with respect to data traffic volume (Γ_D) during one playback interval Π . For distributed protocols, control traffic includes all the control messages transmitted for network selection, streaming source selection, and multicast tree formation. For centralized protocols, control traffic includes the beacon messages, and the accumulated neighbor information sent by all vehicles to the central server. If the control packets are transmitted in the DSRC medium, no cost is associated, whereas control packets sent to the server using cellular networks incur a cost. To compensate the cost, the volume ratio is multiplied with a penalty weight $W_{cpo} > 1$ to get the control packet overhead cpo .

$$cpo = \begin{cases} \frac{\Gamma_C}{\Gamma_D}, & \text{if DSRC network is used} \\ \text{Min}\left(\frac{\Gamma_C}{\Gamma_D} \times W_{cpo}, 1\right), & \text{if cellular network is used} \end{cases} \quad (5.7)$$

W_{cpo} can be the incentive lost by the CDN provider for not being able to offload the control traffic from the cellular network.

- **Packet loss ratio (plr):** The metric evaluates the efficiency of network selection and V2V collaboration. The metric plr is measured as the ratio of received data packets with respect to the total number of expected data packets in a playback interval. If the total data packet count is N for a playback interval, the set of client vehicles is C , client vehicle c_i receives N_{c_i} data packets, then the plr of n^{th} playback interval (plr_n) is calculated as given below.

$$plr_n = \frac{\sum_{i=1}^{|C|} \left(\frac{N_{c_i}}{N}\right) \times 100}{|C|} \quad (5.8)$$

The plr is calculated as the average value for all playback intervals.

- **Buffer underflow ratio (bur):** This metric helps assess users' satisfaction. If a client vehicle selects a stale/unavailable network option or the multi-hop path delay exceeds the delay threshold, it fails to receive sufficient packets during that playback interval to playback the video in the following playback interval. As a result, the vehicle experiences the buffer underflow condition. The metric bur is calculated as the percentage of time a client vehicle faces the buffer underflow condition in the entire streaming duration. The average of bur is taken for all vehicles. If the client vehicle count is $|C|$, the number of playback intervals for which buffer underflow event experienced by i^{th} client vehicle is ξ_i , and the total number of playback intervals is ξ , the metric bur is defined as follows.

$$bur = \frac{\sum_{i=1}^{|C|} \{(\xi_i/\xi) \times 100\}}{|C|} \quad (5.9)$$

- **Peak signal to noise ratio ($PSNR$):** The well-known metric $PSNR$ [111] is used to quantify users' satisfaction. Due to the lossy nature of the wireless DSRC network,

few packets are lost during the multi-hop distribution of the video data. In case a video frame is lost, the proposed solution uses the commonly used error consoling technique to replace the lost frame with the last correctly received frame. *PSNR* metric is used to capture the distortion caused by this error conceal technique. An average is taken for all client vehicles for all playback intervals.

- **Streaming service effectiveness (*sse*):** This metric is defined to assess the overall service effectiveness in terms of traffic offloading (*TO*), protocol efficiency (*PE*), and user satisfaction (*US*). The metric *sse* is defined as the weighted sum of each of the assessment components. Though the weights are equally distributed by default, a CDN provider can vary the weights based on its emphasis on the components. The metric *sse* is calculated as follows.

$$sse = (0.33 \times TO) + (0.33 \times PE) + (0.33 \times US) \quad (5.10)$$

Where,

$$TO = \frac{octr}{100} \quad (5.11)$$

$$PE = (0.5 \times (1 - cpo)) + (0.5 \times (1 - \frac{plr}{100})) \quad (5.12)$$

To quantify *US*, the QoE model provided in equation (5) of the work [117] is used. $\frac{PSNR}{100}$ is used as the video quality function. The metric $\frac{bur}{100}$ is used as re-buffer time with the weight set as 1 because *bur* can be assumed as the time ratio with 0 PSNR value. The proposed solution does not provide adaptive-rate service, so the weight for the bit-rate variation is kept as 0. The startup latency is fixed as Π , so the weight is set as 0. The resulting equation for *US* becomes as follows.

$$US = \frac{psnr}{100} - \frac{bur}{100} \quad (5.13)$$

As the values of the components *TO*, *PE*, and *US* range from 0 to 1, and the summation of all component weights equals to 1, the value of *sse* also ranges from 0 to 1.

5.6.3 Simulation results

This section is organized into two parts. In the first part, the sensitivity of the defined metrics is analyzed with respect to time and relative velocity threshold. In the second part, the competitive performance of the proposed solution is presented.

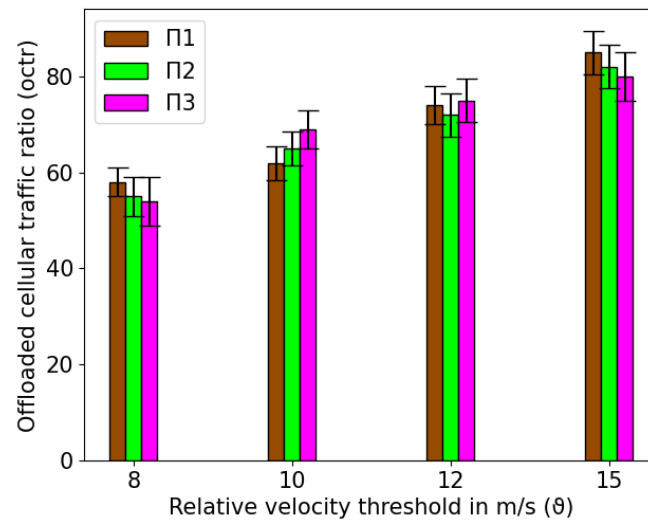


Fig. 5.10 Sensitivity of *octr* metric w.r.t ϑ

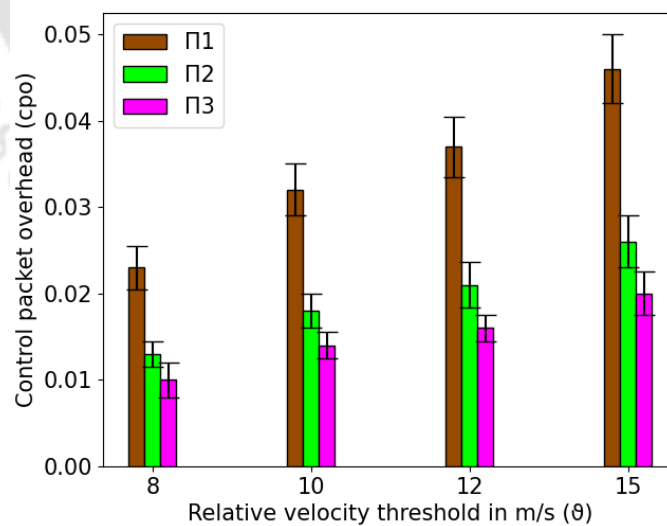


Fig. 5.11 Sensitivity of *cpo* metric w.r.t ϑ

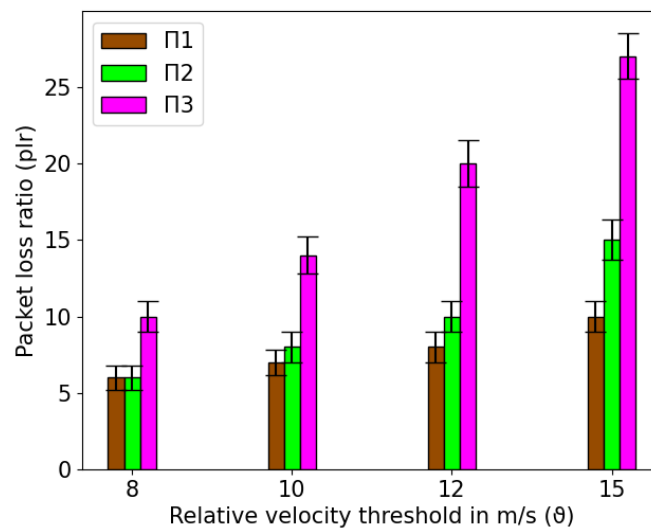


Fig. 5.12 Sensitivity of *plr* metric w.r.t ϑ

The network topology of an IoCV is dynamic due to relative movements among vehicles. The relative movements among vehicles depend on relative velocities and time. Accordingly, the sensitivity of performance metrics is measured with respect to different values of relative velocity threshold ϑ and the playback interval duration Π . In these experiments, vehicle density λ is kept fixed at 100 vehicles per km^2 . The legends $\Pi 1$, $\Pi 2$ and $\Pi 3$ of Fig. 5.10 - Fig. 5.14 denote the performances of playback interval of 1, 2, and 3 s respectively.

Fig. 5.10 shows that the metric *octr* does not depend on the playback interval Π as NWSEL and NSMNTN algorithms only depend on the current vehicular topology at the beginning of a playback interval. However, *octr* gets improved with the higher values of relative velocity threshold ϑ because higher ϑ allows more vehicles to collaborate together, resulting in a reduced number of vehicles with CS state.

The relation of the metric *cpo* with varying Π and ϑ is shown in Fig. 5.11. A message generated by a client vehicle in NWSEL or NSMNTN algorithm is forwarded by neighbor client vehicles within K or $2K$ hops. As ϑ increases, more client vehicles are present within K or $2K$ hops resulting in more message forwarding and increased *cpo*. The NWSEL or NSMNTN algorithm is run in the CNTRL phase at the beginning of a playback interval. As a result, an increment in Π increases the interval between two successive CNTRL phases, which reduces the per-second control packet overhead.

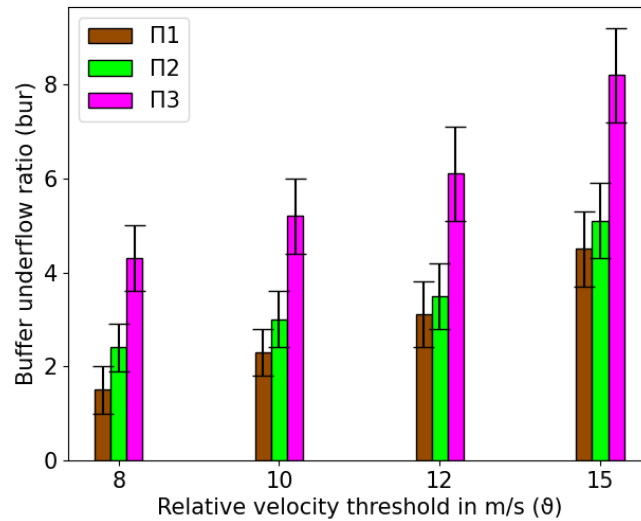


Fig. 5.13 Sensitivity of *bur* metric w.r.t θ

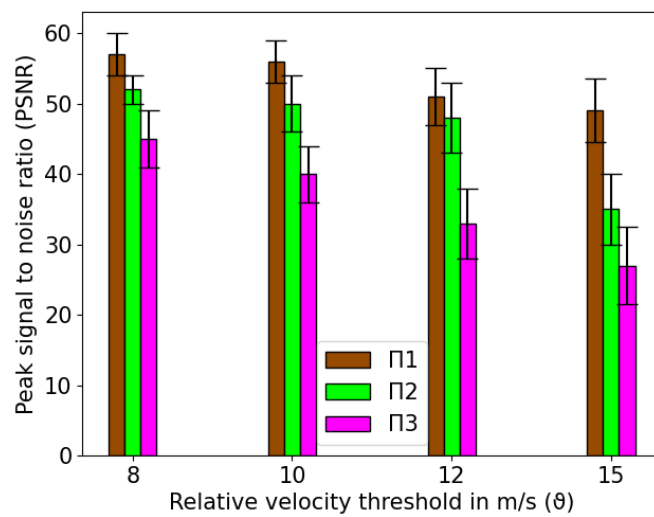


Fig. 5.14 Sensitivity of *PSNR* metric w.r.t θ

Fig. 5.12 illustrates the high sensitivity of plr metric with respect to both the playback duration Π and relative velocity threshold ϑ . As Π and ϑ increase, the relative displacements (upper bounded by $\Pi \times \vartheta$) among vehicles within the DATA phase of a playback interval also get increased, resulting in significant topology changes. As a result, a source vehicle may move out from a Wifi zone, or a multi-hop path between a source vehicle and a client vehicle may get broken. In the above cases, few client vehicles may not receive the data packets in the DATA phase, decreasing the performance of plr metric.

Fig. 5.13 and Fig. 5.14 show the sensitivity of metrics bur and $PSNR$ with respect to Π and ϑ . It is observed that these two metrics are directly related to the performance of the plr metric. As Π and ϑ increase, more packets get lost due to the change in network topology. With large playback duration, a source vehicle with WS state may move out of a Wifi zone within the DATA phase, causing disruption of data packets of all its dependent client vehicles. Moreover, a multi-hop path may get broken due to high relative velocities among vehicles resulting in data disruption for client vehicles with DS_{c_k} state. As a result, performances of both bur and $PSNR$ get decreased with Π and $PSNR$.

It is observed that there is a trade-off between CDN's business profitability ($octr$) versus service efficiency (plr , cpo) and user satisfaction (bur , $PSNR$) objectives while choosing a suitable Π and ϑ . Higher ϑ increases the performance of $octr$, whereas it decreases the performances of plr , cpo , bur , and $PSNR$. On the other hand, though $octr$ is not sensitive with respect to Π , the performance of cpo conflicts with the performances of plr , bur , and $PSNR$. The proposed solution keeps Π as 2 s and ϑ as 12 m/s to achieve a good balance among the conflicting metrics performances.

In the second part of the performance evaluation, CLCH performance is compared with existing solutions for heterogeneous network selection. CLCH cannot be compared with solutions designed for on-demand videos as they use pre-fetching and caching options. Accordingly, CLCH is compared with Contribution 1, Contribution 2, and with the work [136], which is abbreviated as QVHC in this work for convenience. QVHC attempts to maximize the usage of DSRC network for video streaming. Initially, all vehicles attempt to use DSRC network. If a vehicle experiences the DSRC packet loss rate above a certain threshold for a pre-determined time, it makes a vertical handover to LTE network. Once it switches to the LTE network, it waits for a predefined amount of time before it assess the DSRC network. In the assessment, if it is found that the DSRC packet loss rate has come down below the threshold value, the vehicle immediately makes vertical handover to switch back to the DSRC interface.

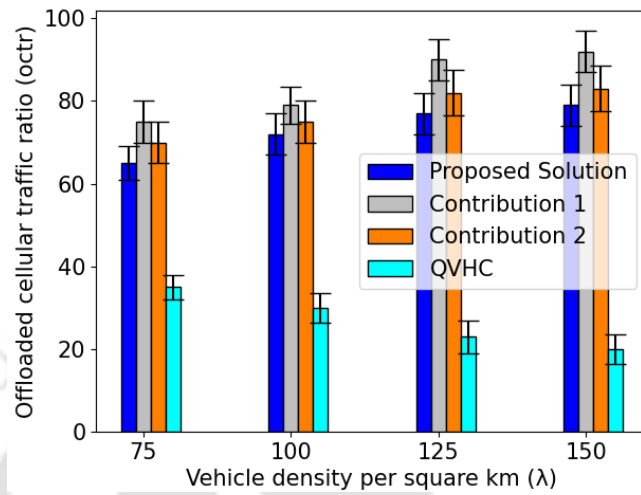


Fig. 5.15 Performance of *octr* metric w.r.t λ

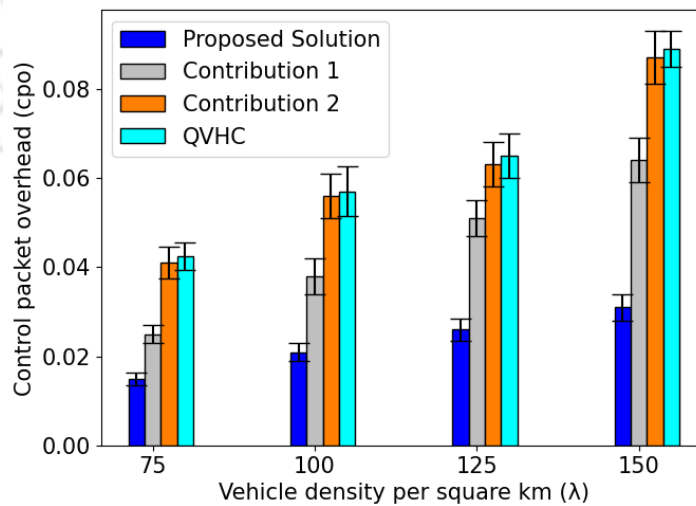


Fig. 5.16 Performance of *cpo* metric w.r.t λ

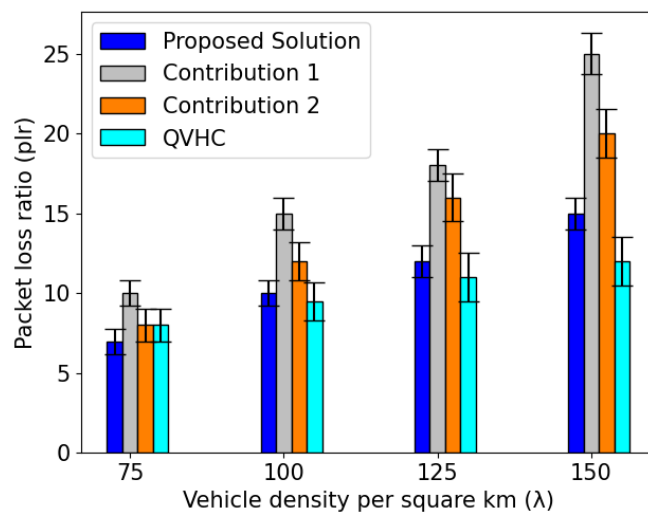


Fig. 5.17 Performance of *plr* metric w.r.t λ

Contribution 1, Contribution 2, and QVHC provide higher layer solutions for heterogeneous network selection for video streaming with to offloading cellular traffic without using pre-fetching or caching options. To compare the performance of CLCH with both centralized and distributed solutions, Contribution 1 is chosen as the representative centralized solution, whereas Contribution 2 and QVHC are chosen to represent distributed solutions. Required changes are made in Contribution 1, Contribution 2, and QVHC to include the Wifi network option. EPOCH is implemented in QVHC, and a multicast tree is formed to select source vehicles at the beginning of each EPOCH. The competitive performances are measured with respect to varying vehicle density λ . For all experiments, the value of Π and ϑ are kept as 2 s and 12 m/s, respectively.

Fig. 5.15 illustrates the effects of varying λ on *octr* metric where all the protocols show a trend of improved *octr* with increased λ . As λ increases, more vehicles get covered within Wifi range and K-hop coverage of source vehicles resulting in improved *octr*. Being a centralized protocol, the thesis Contribution 1 outperforms others by optimum (approximated) source vehicle selection with the knowledge of global vehicular topology. Though the thesis Contribution 2 algorithm is a distributed version of Contribution 1, few control packets get lost due to the lossy nature of the DSRC network, resulting in a deviation from optimality. The proposed solution runs the NWSEL algorithm in the first playback interval of each EPOCH which is a modified version of the thesis Contribution 2. However, in the subsequent playback intervals of an EPOCH, the NSMNTN algorithm is run which is greedy in nature resulting in sub-optimal source vehicle selections.

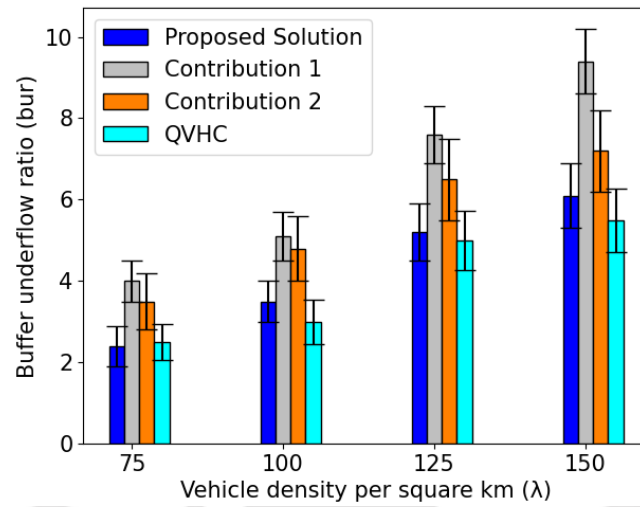


Fig. 5.18 Performance of *bur* metric w.r.t λ

The performances of protocols on *cpo* metric with varying λ are shown in Fig. 5.16. As explained in Section 5.5.3, the proposed solution uses the lightweight NSMNTN algorithm as much as possible, which reduces control packet overhead compared to Contribution 2. The calculations of message sizes are shown in Contribution 2. In Contribution 1, control packets are transmitted in two phases during the CNTRL phase. In the first phase, one small control packet is transmitted for a beacon message by a client vehicle in the DSRC network. As beacon messages are exchanged only within one hop, no message forwarding takes place. In the second phase, due to the centralized nature of the thesis Contribution 1, a client vehicle accumulates all the received beacon messages, creates a neighbor list, and sends it as a large control packet to a central server using the cellular network. As several vehicles connect to the server simultaneously using a limited number of cellular channels, channel blocking and dropping probability increases. Moreover, the central server may get overburdened to handle numerous simultaneous flows from client vehicles. To include the costs of the second phase control packet transmissions into the overhead metric, as discussed in the definition of *cpo*, a penalty factor $W_{cpo} = 5$ is multiplied by the control traffic volume. As a result, the thesis Contribution 1 imposed control packet overhead becomes significant compared to the proposed solution. It is observed that *cpo* increases with vehicle density for all protocols as the increased number of vehicles results in more control packet generation and forwarding.

The competitive performances of *plr* metric with varying λ are shown in Fig. 5.17. The general trend is observed as decreasing performance of *plr* with increasing λ . The chances of packet loss for source vehicles with CS state are very less as the vehicles communicate

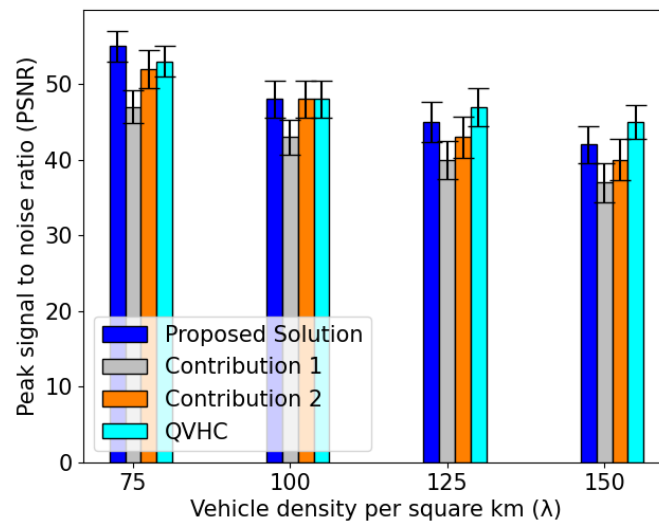


Fig. 5.19 Performance of *PSNR* metric w.r.t λ

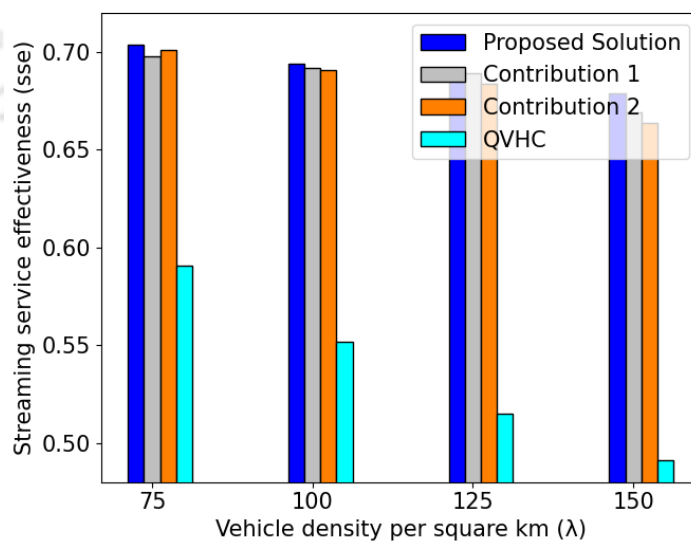


Fig. 5.20 Performance of *sse* metric w.r.t λ

using the dedicated cellular channels allocated to them. The chances of packet loss for source vehicles of WS state using a Wifi network are also very less compared to the client vehicles of DS_{c_k} state using multi-hop DSRC communication. As vehicle density increases, the chances of packet collision also increase due to the shared wireless medium of the DSRC network. Though Contribution 1 performs well for highway scenarios (refer Chapter 3), its performance is the worst in the considered irregular urban scenario. The proposed solution selects more source vehicles with CS state, which reduces the average number of dependent client vehicles (of DS_{c_k} state) for each source vehicle. Consequently, the number of data packet transmissions through the DSRC network is reduced, resulting in reduced packet loss and improved plr performance.

Fig. 5.18 and Fig. 5.19 depict the competitive performances of the protocols on bur and $PSNR$ metrics. The performances of these two metrics are related to the performance of the plr metric. As λ increases, plr performance decreases resulting in decreased performance of bur and $PSNR$ metrics. The proposed solution performs best among the three protocols as it selects more source vehicles that reduce data packet communication through the DSRC network.

The above experiment results demonstrate the pros and cons of the compared protocols. Being a centralized protocol, Contribution 1 offers the best traffic offloading performance. However, centralized algorithms are not scalable and impose a significant computational burden on servers. Moreover, compared to other protocols, Contribution 1 lags in the service efficiency and user satisfaction parameters. The thesis Contribution 2 is the distributed version of the thesis Contribution 1, making it preferable compared to Contribution 1. However, the control packet overhead imposed by Contribution 2 on the DSRC network is significantly high. Therefore, to assess the overall streaming performances of the compared protocols, the metric sse is used. The competitive performances of steaming service effectiveness sse are shown in Fig. 5.20. It is observed that the service effectiveness degrades for all protocols as vehicle density increases. The proposed solution outperforms others in sse implying that it is able to achieve the best balance between service profitability and service efficiency, as well as user satisfaction, making it the most attractive choice for CDN providers.

5.7 Summary

This work has proposed a novel distributed protocol for network access technology selection for live video streaming services over IoCVs. The network selection problem is formulated

as an ILPP, and accordingly, a distributed protocol has been proposed. The proposed solution is designed to greedily maximize the cellular traffic offloading using Wifi access points and DSRC V2V collaborations while ensuring the service constraints for live video streaming. The proposed solution has significantly reduced the control packet overhead compared to existing protocols. Unlike existing works, the proposed solution can provide network selection solutions for scenarios where Wifi, DSRC, and cellular network access technologies coexist. Simulation results show that the proposed solution protocol most effectively balances the goal of traffic offloading, service efficiency, and user satisfaction. The proposed solution is attractive for CDN providers to increase the profitability of live video streaming services over IoCVs. However, to generate additional revenue, CDN providers may wish to offer multi-tier video streaming service, which is discussed in the next chapter.



Chapter 6

Distributed edge vehicle selection for multi-tier live video streaming

The previous three chapters are dedicated to make live video streaming services cost-effective over IoCVs for CDN providers. Along with reducing service costs, a CDN provider may wish to generate additional revenues by offering multi-tier video streaming services to its clients. A CDN provider offers its clients a set of subscription tiers where the CDN provider agrees to provide a pre-defined maximum achievable bitrate for each tier, where higher-tier clients pay more and are entitled to receive higher bitrate (quality) videos. However, enhanced-quality video requires a few additional packets to be delivered to the higher-tier vehicles. As a result, for delay-constrained live video streaming service, all the vehicles receive and forward additional packets that are to be consumed by only a few of the client vehicles belonging to higher tiers. The burden of on-time delivery of additional packets can have adverse effects on the cost-effectiveness and the quality of the service. For example, let there be only two tiers of service: tier1 and tier2. All vehicles need tier1 packets, whereas only the tier2 vehicles need tier2 packets. Let the end-to-end multi-hop path (from an edge to farthest hop vehicle) delay to deliver all tier1 packets be constrained by t_d . If a tier2 vehicle is the farthest vehicle from an edge, all intermediate tier1 vehicles need to take the burden of receiving and propagating additional tier2 packets to the farthest vehicle, though they don't require those packets. Moreover, the transmission time in each hop gets increased to transmit additional packets for tier2. As a result, the end-to-end delivery delay may exceed the delay constraint t_d . However, if the tier2 vehicle is nearer to the edge, these problems can be avoided. Therefore, in addition to keep the number of edges minimum, we also need

to keep higher-tier vehicles closer to the edges. With an increase in the number of tiers, the problem becomes non-trivial, as shown in Section 6.2.2.

In the existing literature, though there are solutions for adaptive bitrate video streaming services over IoCVs, no work is found for multi-tier streaming services. Multi-tier services are different from adaptive bitrate services. In an adaptable bitrate service, all clients are identical from the service provider's perspective, and the bitrates are decided based on current channel conditions and network congestion. In contrast, in the case of multi-tier services, clients are not identical, and the bitrates are decided as per the agreements based on clients' subscription tiers. Motivated by the above facts, this work proposes a cost-effective multi-tier service for CDN providers to increase their profitability. The contribution highlights are enlisted below.

- This work is the first of its kind to propose a cost-effective and additional revenue-generating solution for CDN providers to offer a live video streaming service over IoCVs where the service is constrained by multi-tier bitrate agreements.
- The edge vehicle selection problem with multi-tier bitrate constraints is formulated as the MSC problem to select an optimum number of edge nodes. The problem is solved as an adaptation of the existing distributed approximation algorithm of MSC.

The rest of the work is organized as follows. The existing works are discussed in Section 6.1. The proposed solution is described in Section 6.2. The performance of the proposed solution is evaluated in Section 6.3, and the simulation results are presented. The work is summarized in Section 6.4.

6.1 Related Work

The work [34] proposes a scheme for adaptive bitrate to maximize the total QoE of the system. Guo et al. [35] proposed an adaptive bitrate scheme to maximize the time-averaged network reward. Both of the works assumed homogeneous video quality requirements from client vehicles, whereas, in the context of this work, client vehicles of different tiers have heterogeneous video quality (bitrate) requirements. All the works mentioned above are for on-demand videos and involve static infrastructure nodes as edges. However, as mentioned in Section 1.5.3, using static edges is less cost-effective for CDN providers. Also, the solutions for on-demand videos are not applicable to live streaming videos as the latter cannot be cached due to the unavailability of the content beforehand.

There are only a few works available on live video streaming over IoCV in the existing literature. The work [28] heuristically solves the resource allocation problem for each multicast relay vehicle for live video streaming. Few works have termed edge vehicles as gateway vehicles. The work [21] has the objective of maximizing of quality of experience (QoE) for real-time videos. No work is found for live video distribution using edge vehicles which minimizes service cost while ensuring delivery delay and multi-tier bitrate constraints.

6.2 Proposed work

This section is divided into three parts. Section 6.2.1 describes the system model and assumptions. The problem formulation for edge selection is given in Section 6.2.2. The detailed methodology is described in Section 6.2.3.

6.2.1 System model

This work considers a heterogeneous vehicular network with multi-RAT-equipped vehicles where the DSRC interface is used for V2V communication, and the cellular interface is used for V2I communication. Though the presence of Wifi network is not considered in this system model for simplicity, this work can easily be extended to accommodate Wifi network as described in the previous chapter. A CDN provider wishes to provide a multi-tier video streaming service to vehicles within an AOI and deploys a CDN server near AOI. An example scenario is shown in Fig. 6.1. The distributed video is assumed to be of H.264 type, where a base layer is needed to play a video with a minimum bitrate, and some enhancement layers are needed to be added on top of the base layer for higher quality (bitrate) videos (refer Section 2.2.1.2). Clients of a CDN provider belong to different subscription tiers where higher tier clients pay more and are entitled to receive enhancement layers for better video quality. Similar to Contribution 2 [Chapter 4], the proposed protocol divides the entire video streaming duration into many epochs of duration t_e . An epoch is further divided into a control phase (CNTRL) of duration t_c and a data phase (DATA) of duration t_d such that $t_c + t_d = t_e$. In each epoch's CNTRL phase, vehicles distributively select a few of them as edges. All other client vehicles associate themselves with any one of the selected edges within its K -hop. The edges are selected such that all of its associated client vehicles receive all the required video data packets within the DATA phase satisfying individual clients' subscription tier bitrate agreements. The problem formulation and solution for edge vehicle selection are described

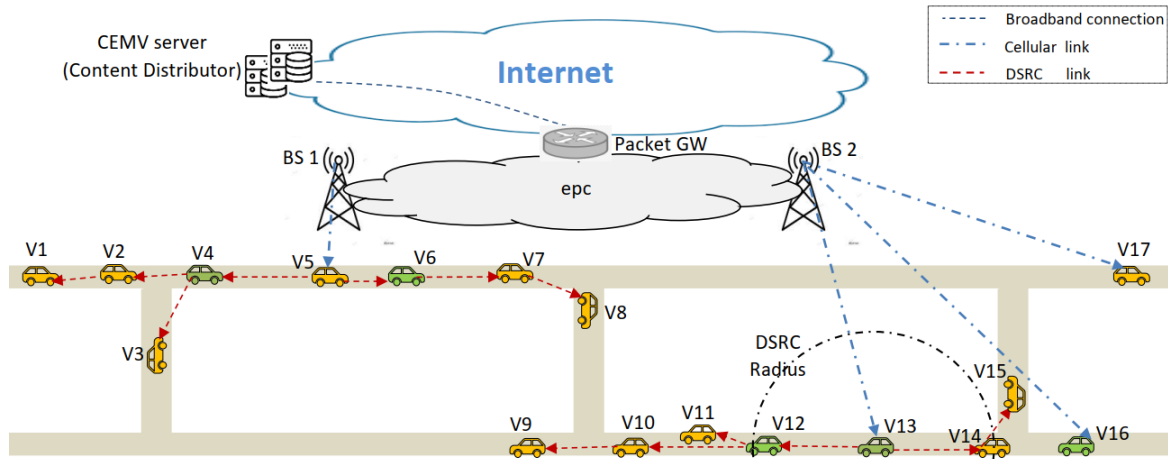


Fig. 6.1 An example scenario with two service tiers with max four hop collaborations

in subsequent sections. In the DATA phase, each selected edge vehicle fetches video content from the CDN replica server through the cellular interface and distributes the content to its associated client vehicles through multi-hop V2V collaborations. Fresh edge selection is made at the beginning of every epoch to mitigate the dynamic network connectivity of VANET.

6.2.2 Problem formulation

Similar to the solution proposed in Contribution 2 [Chapter 4], this solution also restricts multi-hop DSRC collaboration to a maximum of K hops. However, unlike Contribution 2, this solution divides the DATA phase duration t_d into K distinct time slots (T_1 to T_K) of t_d/K duration each. Packets transmitted by an edge in T_1 slot are received by a k^{th} hop ($1 \leq k \leq K$) distant vehicle in time slot T_k . Let's assume the minimum achievable DSRC data rate within the AOI is Υ_{dsrc} . The streaming video content is divided into segments where one segment is distributed during one epoch. The layers of a segment are packetized before distribution. Let's assume a CDN provider has M service tiers consisting the set $\Phi = \{\phi_1, \phi_2, \dots, \phi_M\}$ to offer where the subscription cost for ϕ_i is greater than subscription cost for ϕ_j whenever $i > j$. Let's assume there are total L layers available for a video where l_1 represents the base layer and l_L denotes the highest enhancement layer. The highest quality video can be achieved by adding all the layers from l_1 to l_L on top of one after another. For each service tier ϕ_i , the CDN provider needs to decide on a suitable enhancement layer l_{ϕ_i} where a ϕ_i tier vehicle receives video layers from l_1 up to maximum l_{ϕ_i} with a maximum achievable video bitrate of

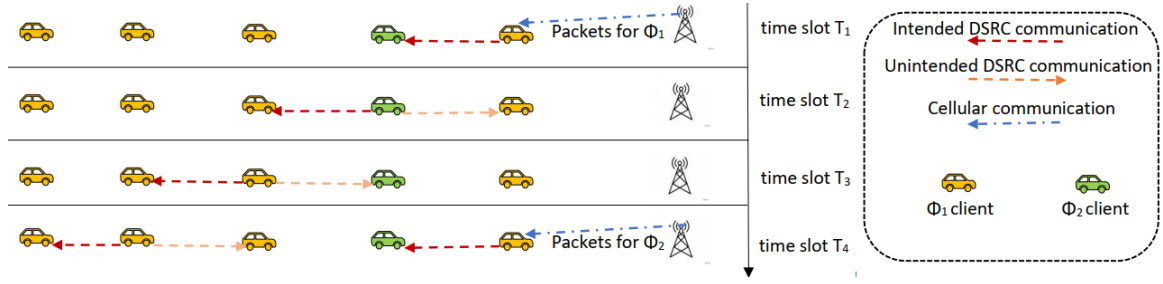


Fig. 6.2 Enhancement layer insertion schedule for CEMV

Υ_{ϕ_i} . Service tier bitrates are chosen such that the following two equations hold.

$$\Upsilon_{\phi_1} \leq \frac{\Upsilon_{dsrc}}{K} \cdot \frac{t_d}{t_e} \quad (6.1)$$

$$\Upsilon_{\phi_{i+1}} \leq 2 \cdot \Upsilon_{\phi_i} \quad \forall \phi_i, \phi_{i+1} \in \Phi. \quad (6.2)$$

The equation (6.1) ensures that all the ϕ_1 layer packets are transmitted within time slot T_1 of t_d/K duration. The equation (6.2) ensures that all the additional packets for an upper tier can be transmitted within a single time slot. The situation is illustrated in Fig. 6.2. To avoid packet collisions in a shared wireless medium, after transmitting tier ϕ_i packets, an edge node must wait for at least 2 time slots before it can further pipeline ϕ_{i+1} tier's enhancement layer packets. Therefore, an edge node transmits ϕ_1 packets in T_1 , ϕ_2 packets in T_4 , and accordingly, ϕ_i packets in $T_{1+3 \cdot (i-1)}$. Consequently, the maximum possible service tier number M is bounded by the following equation.

$$M = \lfloor \arg \max_i ((1 + 3 \cdot (i - 1)) \leq K) \rfloor \quad (6.3)$$

The packets for ϕ_{i+1} transmitted in $T_{1+3 \cdot (i-1)}$ can propagate maximum $K - 3 \cdot (i - 1)$ hops within the DATA phase duration t_d . Clearly, all ϕ_i type client vehicles need to be located within $K - 3 \cdot (i - 1)$ hops from any one of the edge vehicles to receive all l_{ϕ_i} layer packets within t_d . Fig. 6.1 shows an example for $K = 4$ with M calculated as 2 (from equation (6.3)) where green vehicles represent ϕ_2 clients and yellow vehicles represent ϕ_1 clients. The CDN provider wants to keep the number of edge vehicles as low as possible to maximize the cost-effectiveness of the service. The task mentioned above can be formulated as the MSC problem, as explained below.

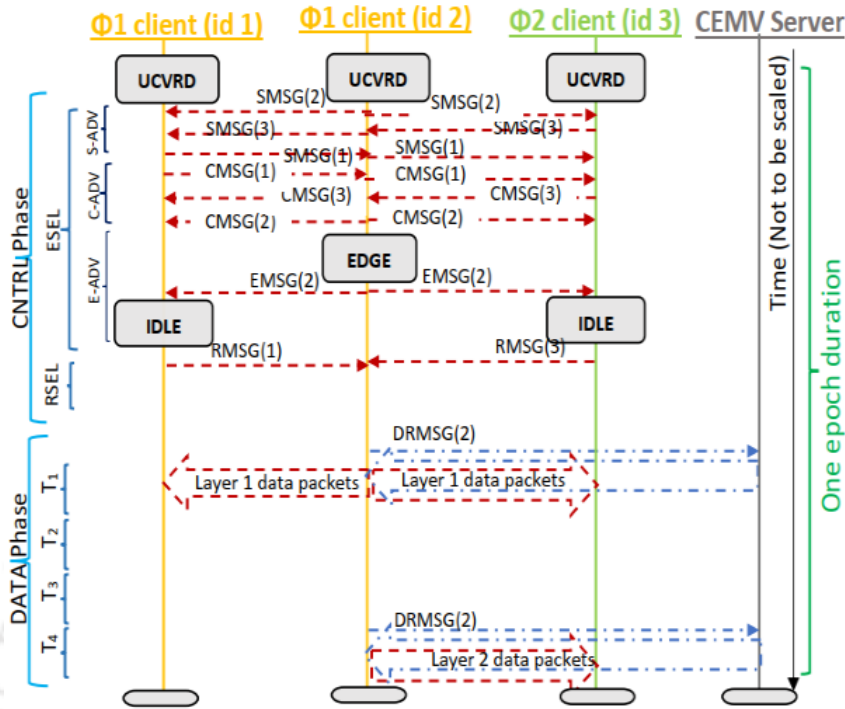


Fig. 6.3 Protocol timeline, message sequences, and state transitions for a sample three node topology consisting of two ϕ_1 clients and one ϕ_2 client

The well-known MSC problem is defined in Section 3.2.3. V is assumed to be the set of client vehicles within AOI in the considered scenario. Each interested vehicle $v_n \in V$ is considered as a potential edge, and a subset of V denoted as s_{v_n} is created for each v_n . The subset s_{v_n} includes all the ϕ_i type vehicles which are within $K - 3 \cdot (i - 1)$ hops from v_n , $\forall \phi_i \in \Phi$. The vehicle v_n is also included in s_{v_n} as it is considered as 0 hop from itself. All such s_{v_n} (n ranges from 0 to $|V|-1$) forms the set S . As each s_{v_n} contains at least one element v_n , collectively, all elements of S definitely cover V . The task is to choose the minimum number of elements from S such that the chosen elements (subsets of V) cover V . Clearly, the task is an instance of the MSC problem, which is a known NP-hard problem. The approximation algorithm GSC for MSC runs in polynomial time [30] with the approximation ratio of GSC is $\ln|V| + 1$. However, GSC is a centralized algorithm that needs knowledge of global topology. DDCH[112] is a distributed version of GSC with the same approximation ratio. Contribution 2 has proposed the algorithm ADDCH (Algorithm 2), which is an adaptation of DDCH. However, ADDCH does not support video streaming with multi-tier bitrate agreements. Accordingly, a modified version of ADDCH is proposed in this work which is described next.

6.2.3 Proposed Solution

All client vehicles run the distributed solution, which is a modified version of the ADDCH algorithm proposed in Contribution 2. Client vehicles can be in any of the four states of UCVRD (uncovered), EDGE (edge), CNDT (candidate), or IDLE (idle). The protocol timeline, message sequences, and state transitions of a simple three-node topology for one epoch are given in Fig. 6.3. In the diagram, the message originator (not forwarder) vehicle id is shown within parenthesis along with the message name. The CNTRL and DATA phases are described below.

6.2.3.1 CNTRL phase

CNTRL phase is responsible for distributed edge vehicle selection and assignment. Within an epoch, the CNTRL phase is further divided into subphases of edge selection (ESEL) and relay selection (RSEL).

- *ESEL*: The ESEL subphase runs in iterations, and comprises three steps which are S-ADV (state advertisement), C-ADV (coverage count advertisement), and E-ADV (edge advertisement). At the beginning of the ESEL subphase, all vehicles are uncovered and reset to the UCVRD state. In the S-ADV step, vehicles advertise their states and service tier ϕ_m to their K -hop neighborhood. In the C-ADV step, each vehicle calculates and advertises its total coverage count ρ_t to its $2K$ -hop neighbors. The ρ_t for a vehicle is calculated as $\sum_{m=1}^M \rho[m]$ where $\rho[m]$ denotes the number of vehicles of type ϕ_m in its K -hop neighbors with UCVRD state, and the corresponding hop distance is less than $K - 3 \cdot (m - 1)$. Depending on the vehicle's own tier type, appropriate ρ_m is incremented by 1 before calculating ρ . In the E-ADV step, a vehicle considers itself as an edge only if its own ρ is highest among its K -hop neighbors and switches its state to EDGE. A tie is broken with the highest vehicle id . The new edge advertises its new state to its K -hop neighbors. After an ESEL iteration, only the vehicles with UCVRD or CNDT states participate in the next ESEL iteration. The ESEL phase terminates when all vehicles move to either the IDLE or EDGE state, and there is no vehicle to participate in the next ESEL iteration. The steps are shown in Algorithm 6.

- *RSEL*: In this subphase, a vehicle of IDLE state looks for the edge with the lowest hop count. The vehicle selects the next hop node to the selected edge node as its relay node and sends a message to it. On receiving the message, the relay node sets a flag, namely RELAY, to TRUE. The selected edge vehicles and relay vehicles are valid for only one epoch (the duration for which topology is considered static), after which a fresh selection takes place in the next epoch.

Algorithm 6 Edge selection (runs in individual client vehicle)**Input:** number of service tier M , max hop count K **Output:** state of a vehicle IDLE/EDGE

```

1: state  $\leftarrow$  UCVRD
   {//start of a new epoch}
2: while state == UCVRD || state == CNDT do
3:   //S-ADV phase begins
4:   advertise state and service tier  $\phi$  to its  $K$ -hop neighborhood.
5:   //C-ADV phase begins
6:   initialize coverage list  $\rho[M]$  by  $\rho[m] \leftarrow 0 \forall m \in \{1 \leq m \leq M\}$ 
7:   initialize  $\rho[\phi] \leftarrow 1$  {//as it covers itself}
8:   total coverage count  $\rho_t \leftarrow 0$ 
9:   for all received  $\phi_m$  from neighbor vehicles do
10:    if hop count  $\leq K - 3 \cdot (m - 1)$  && state of that neighbor == UCVRD then
11:       $\rho[\phi_m] = \rho[\phi_m] + 1$ 
12:    end if
13:  end for
14:   $\rho_t = \sum_{m=1}^M \rho[m]$ 
15:  advertise  $\rho_t$  to its  $2K$ -hop neighborhood.
16:  //E-ADV phase begins
17:  if its own  $\rho_t$  is highest among received  $\rho_t$  from neighbors then
18:    state  $\leftarrow$  EDGE
19:    advertise itself as EDGE to its  $K$ -hop neighborhood.
20:  else if received EDGE advertisement(s) then
21:    if it has neighbor vehicle(s) with UCVRD state then
22:      state  $\leftarrow$  CNDT
23:    else
24:      state  $\leftarrow$  IDLE
25:    end if
26:  end if
27: end while

```

6.2.3.2 DATA phase

At the start of the DATA phase, the selected edge vehicles request the CDN server for data packets. The DATA phase duration t_d is divided into K time slots T_1 to T_K each of duration t_d/K . During T_1 , the CDN server sends the required video packets up to l_{ϕ_1} layer. As soon as an edge vehicle starts receiving data packets through its cellular interface, it immediately transmits them through its DSRC interface. During T_2 , the vehicles within 1-hop from the edge vehicle, with the RELAY flag true, re-transmit the content. The content propagates hop-by-hop in each subsequent time slot. Higher tier enhancement layer packets for ϕ_{i+1} are received and pipelined by edge vehicles in $T_{1+3 \cdot (i-1)}$ slots subsequently. At the end of the DATA phase, vehicles rearrange out-of-order packets according to their index and push the received packets to their media player buffer. The fetched video in epoch e_i is played for

Table 6.1 Simulation parameters

Parameter	Description	Value
AOI	Area of interest	4 km ²
λ	Vehicle density	50-200 vehicles/km ²
R_{ϑ}	Relative velocity threshold	8 m/s-15 m/s
Π_{dsrc}	DSRC communication radius	100 m
Υ_{lte}	LTE download speed	25 Mbps
Υ_{dsrc}	DSRC channel data rate	24 Mbps
t_e	Epoch duration	1s -3 s
K	Max DSRC hop collaboration	5
M	Max number of service tier	2

the entire next epoch e_{i+1} while the client vehicles acquire new data packets to be played in epoch e_{i+2} .

6.3 Performance evaluation

This section is divided into three parts. Section 6.3.1 describes the simulation environment and various simulation parameters. The performance metrics are defined in Section 6.3.2. The simulation results based on defined metrics are discussed in Section 6.3.3.

6.3.1 Simulation environment and parameters

The performance evaluation is done using simulation in NS3 (version 3.30.1) [110]. An urban grid scenario is considered where roads have two lanes in opposite directions. Vehicle traces are generated using the SUMO tool [120]. The simulation parameters are enlisted in the TABLE 6.1. As discussed in Section 2.2.2, a H.264 video is organized into a series of NALUs that can be transmitted independently. In this experimentation, the trace of the video "Elephants Dream" [4] is used. NALUs belonging to layer $\langle 0, 3, 6 \rangle$ or less are delivered to all vehicles (ϕ_1/ϕ_2), and NALUs belonging to layer $\langle 0, 4, 6 \rangle$ is delivered to ϕ_2 vehicles only.

6.3.2 Performance metrics

To evaluate the performance, the following metrics has been defined.

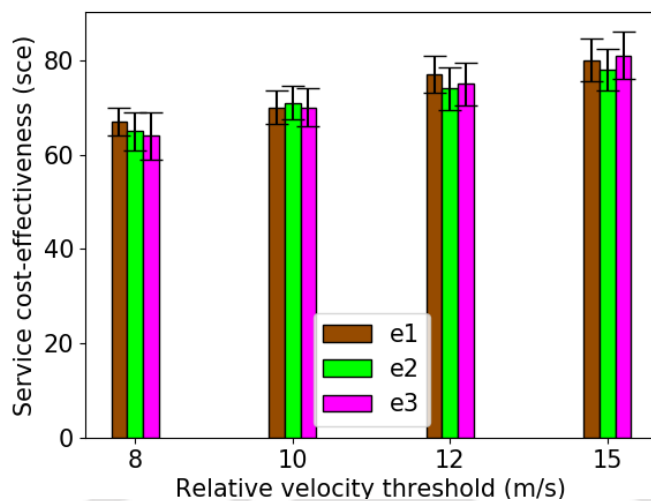
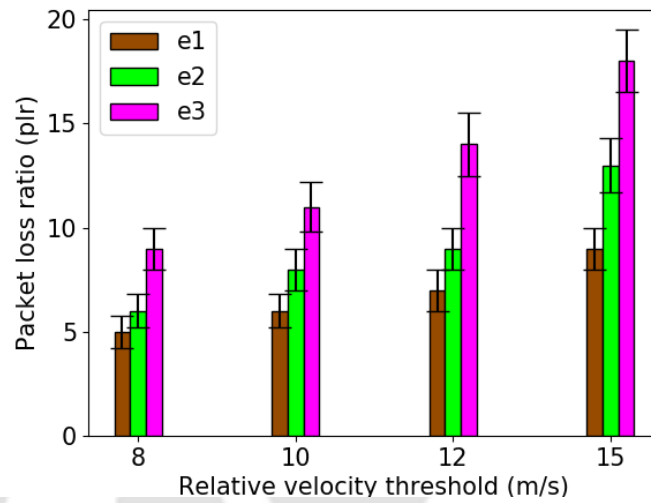
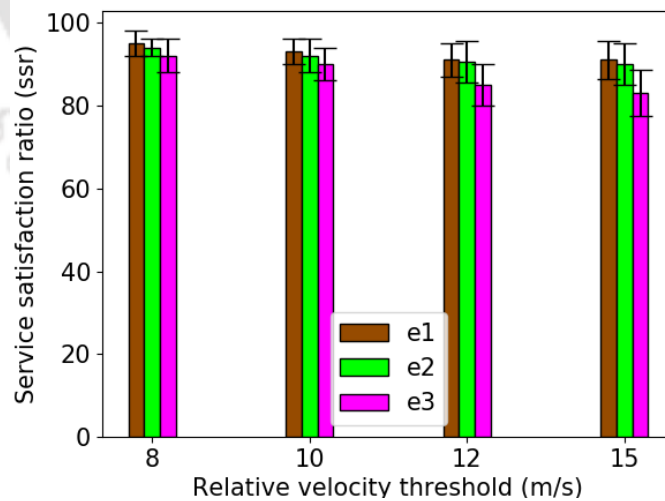


Fig. 6.4 *sce* vs $R\vartheta$

- **Service cost-effectiveness (*sce*)** It is the ratio of offloaded video data compared to the total data demanded by all client vehicles. It is measured as the ratio of client vehicles receiving video from edge vehicles compared to the total number of client vehicles. Let the set V is total client vehicles, and set E is the set of selected edge vehicles where $E \subseteq V$. Then, $sce = (|V| - |E|)/|V|$
- **Packet loss ratio (*plr*)** It is the percentage of the lost data packets compared to the number of total expected data packets for a vehicle. The number of expected data packets is different for each tier. The average is taken for all vehicles (of all tiers) in all epochs.
- **Service satisfaction ratio (*ssr*)** The metric is defined to capture the percentage of client vehicles satisfied with their received bitrate depending on their subscription tiers. The satisfied and unsatisfied states are represented by 1 and 0. An average is taken for all vehicles in all epochs. In an epoch, if a ϕ_i tier client receives packets of all the layers from l_1 to l_{ϕ_i} , the client is satisfied and otherwise not satisfied. The average satisfaction value is taken for all vehicles in all epochs.

Though the delay, jitter, and start-up latency are a few commonly used metrics for streaming quality assessment, they are not considered in this work for the reasons mentioned below. Delay and jitter are consumed by the playback buffer. A packet delay of more than t_d is considered a loss event which is captured by *plr* metric. The start-up latency has a fixed value of t_e as vehicles spend the first epoch to acquire data to start playback from the second epoch. Simulation results based on the defined metrics are presented in the next section.

Fig. 6.5 *plr* vs $R\vartheta$ Fig. 6.6 *ssr* vs $R\vartheta$

6.3.3 Simulation results

The sensitivity of sce , plr , and ssr metrics with respect to t_e and R_ϑ are shown in Fig. 6.4, Fig. 6.5, and Fig. 6.6 respectively. Epoch duration t_e of 1, 2, and 3 s are abbreviated by e1, e2, and e3 in the legends of these figures. The experiments are performed with R_ϑ varied from 8 to 15 m/s while keeping λ fixed at 150 vehicles per square km, and the ratio of ϕ_2 to ϕ_1 vehicles as 1:6.

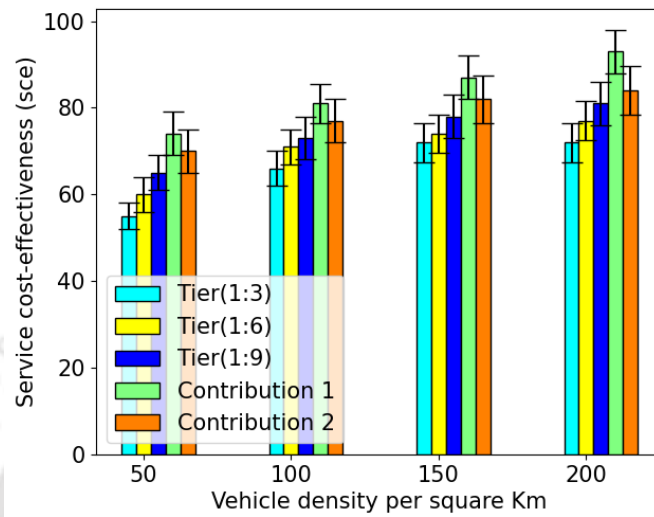
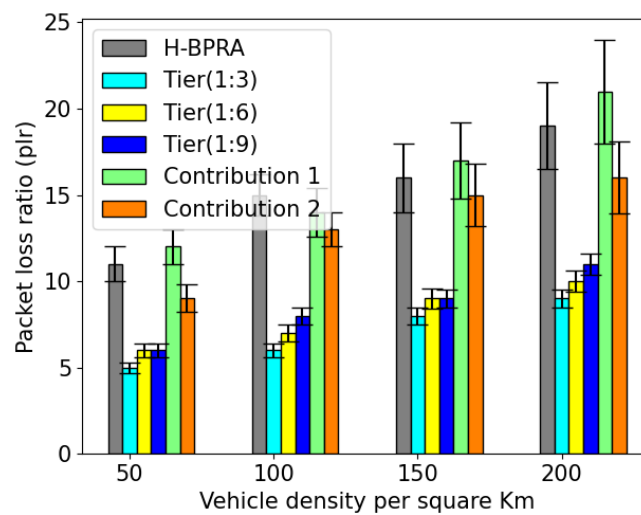
Fig. 6.4 shows the performance of sce w.r.t t_e and R_ϑ . As R_ϑ increases, more vehicles (possibly of opposite directions) can collaborate, resulting in larger groups reducing the number of edge vehicles in the network. It is observed from Fig. 6.4 that epoch duration does not affect sce . This is because edge selection depends only on vehicle connectivity at the beginning of an epoch.

The sensitivity of plr metric against t_e and R_ϑ is shown in Fig. 6.5. The probability of link breakage of a multi-hop path increases with R_ϑ . Moreover, when epoch duration increases, relative displacement among vehicles within an epoch also increases, further increasing multi-hop path breakage probability. This can be observed in Fig. 6.5 where e1 performs best whereas e3 performs worst.

Fig. 6.6 shows the performance of ssr with varying t_e and R_ϑ . During the DATA phase, an edge distributes lower-layer packets before pipelining higher-layer packets in the network. As multi-hop paths become unstable at the end of an epoch, higher layers are more probable for loss. This can be observed from Fig. 6.6 where ssr gets decreased with increasing R_ϑ and t_e .

From the above experiments, it is observed that t_e as 1 s with R_ϑ as 12 m/s is able to achieve a well balanced performance between cost-effectiveness (lce), streaming performance (plr), and client satisfaction (ssr).

The competitive performance of the proposed solution is compared against H-BPRA [28], Contribution 1 (refer Chapter 3), and Contribution 2 of this thesis as they are proposed for live video streaming over IoCVs. However, H-BPRA uses static RSUs as edge nodes, making it incomparable for sce metric. Contribution 1 and Contribution 2 do not consider multi-bit rate video, which makes them incomparable for ssr metric. H-BPRA is run with only one modulation scheme and two video layers. H-BPRA uses multicast techniques for layered video transmissions and runs in two phases. In the first phase, it ensures the delivery of the base layer packets to all the client vehicles. In the second phase, based on the remaining time and vehicle demands, it schedules the delivery of the higher layer video packets. The

Fig. 6.7 *sce* vs λ Fig. 6.8 *plr* vs λ

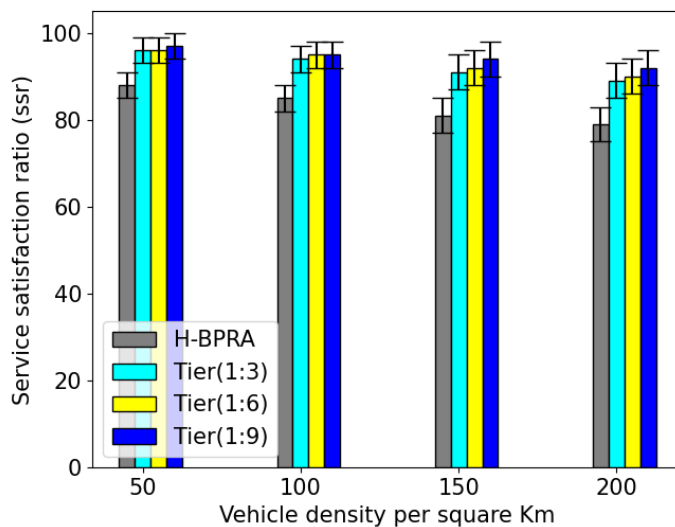


Fig. 6.9 Competitive performance on client satisfaction

legends Tier(1:x) of Fig. 6.7, Fig. 6.8, and Fig. 6.9 represent the ratio of $\phi_2:\phi_1$ as 1:x. These experiments are performed with $R_{\phi} = 12$ m/s and $t_e = 1$ s.

Fig. 6.7 shows the *sce* performance with varying λ . With an increase in λ , more vehicles are covered in each hop resulting in increased *sce*. Results show that with a low ϕ_2 ratio, the proposed solution improves towards Contribution 2 as in the extreme case of no ϕ_2 vehicle, it falls back to Contribution 2. Being a centralized algorithm, Contribution 1 makes optimal edge selection and thus outperforms others. In the proposed solution and Contribution 2, CNTRL messages need to propagate multi-hop through a shared DSRC wireless medium, causing some control packet loss and resulting in sub-optimal edge selection in highly dense scenarios. Contribution 1 and Contribution 2 do not consider the location constraints for multi-tier clients resulting in less edge selection.

Fig. 6.8 analyzes the effects of λ on the performances on *plr*. With higher λ , packet collision probability increases in shared DSRC wireless medium, resulting in higher *plr*. Tier(1:3) of the proposed solution performs best as it has more edges, resulting in fewer multi-hop transmissions. Contribution 1 performs poorly as it has a lesser number of edges resulting in more V2V collaborations. It is observed that H-BPRA performs similarly to Contribution 1.

Fig. 6.9 shows the competitive performances of the protocols for *ssr* metric with varying λ . In the H-BPRA protocol, video layers are distributed in a phase-wise manner. Once a layer is forwarded within a phase, the vehicles near edge RSUs remain idle for the rest of that phase.

As a result, H-BPRA failed to deliver all the required packets within the considered time budget in the considered experiment scenario. On the other hand, the proposed solution does pipelining the upper enhancement layers as soon as possible without waiting for the lower layers to be distributed to the furthest vehicles. Moreover, higher-layer recipient vehicles are nearer to edge vehicles reducing higher-layer distribution time. Tier(1:9) outperforms others as there are fewer ϕ_2 clients needing l_{ϕ_2} layer packets while most of the clients are satisfied with l_{ϕ_1} layer packets.

6.4 Summary

This work has investigated multi-tier bitrate-constrained live video distribution over IoCVs to generate additional revenues for CDN providers. The work has proposed a solution to select a minimum number of edge vehicles to maximize the cost-effectiveness of the service. The challenging task of mobile edge vehicle selection is formulated as the MSC problem, which is solved by a distributed approximation algorithm. Simulation results show that compared to other protocols, the proposed solution most effectively balances the content distributor's goals of cost-effectiveness and streaming performance while providing a way to generate additional revenue through multi-tier service.



Chapter 7

Conclusions and Future research directions

7.1 Conclusions

This dissertation has yielded four contributions towards making video infotainment services over IoCV cost-effective and revenue-generating for CDN providers. The solutions use a few of the vehicles of an IoCV as edge nodes for a CDN provider to avoid the cost of dense infrastructure node deployment or the partnership cost with various carrier enterprises for using base stations as edges.

The first contribution has provided a solution for optimum edge vehicle selection to reduce the number of simultaneous cellular connections and the Internet bandwidth requirements to make live video streaming services cost-effective for CDN providers. The problem of optimum edge vehicle selection for delay-constrained live video streaming has been formulated as the Minimum Set Covering (MSC) problem to demonstrate the NP-Hardness of the problem. An existing centralized approximation solution has been used as the solution. Simulation results show that the proposed solution offers the most efficient edge selection for CDN providers.

To avoid the scalability issues with the centralized solution of the first contribution, and to relieve the CDN providers from high setup costs for high-end servers, the second contribution of this dissertation has provided a distributed solution for optimum edge vehicle selection. A novel multicast protocol also has been proposed for V2V collaboration in the network

layer. Simulation results show that the proposed solution is the most cost-effective for CDN providers.

While the second contribution is devoted to provide a V2V communication solution, the third contribution of this dissertation delves into V2I communication to find a cost-effective network selection solution in the presence of cellular, Wifi, and DSRC networks. The problem of optimum network selection has been shown as NP-hard by formulating the problem as an ILPP. Accordingly, a distributed network selection algorithm has been provided, which greedily tries to maximize the cellular traffic offloading for live video streaming over IoCVs. The simulation results demonstrate that the proposed solution most effectively balances traffic offloading, streaming performances, and client satisfaction.

While the first three contributions look to increase the cost-effectiveness of video infotainment services for CDN providers, the fourth contribution of this dissertation has provided a solution to generate additional revenue for them. The contribution is a pioneer work to enable CDN providers to run multi-tier video streaming services over IoCV. It is shown that the problem of edge vehicle selection for multi-tier service must include an additional constraint of hop count for the higher-tier vehicles. The problem of edge selection for multi-tier service is shown to be NP-Hard by formulating as MSC, and a distributed solution is provided. Simulation results show that the proposed solution most efficiently handles the higher-tier additional packet overheads by including hop constraints for higher-tier vehicles.

The solutions provided in this dissertation help increase the cost-effectiveness and profitability of the services while ensuring service quality and client satisfaction. As a result, CDN providers may find the solutions attractive while offering streaming video infotainment services over IoCVs.

7.2 Future research directions

Future research directions can follow the two below-mentioned paths.

7.2.1 Exploring V2V communication of IoCVs with 5G C-V2X

This dissertation has considered that V2I communication of IoCV can be of heterogeneous network technologies comprising cellular, Wifi, and DSRC. However, for V2V communication, only one network technology has been considered as DSRC, whereas the competing

technology "Cellular Vehicle to Everything (C-V2X)" has not been explored. After the 3GPP release version 16, which allocates a separate 5Gz band (side-links) for direct (without via base station) V2V communications, the C-V2X technology is drawing the attention of the industry. One of the major advantages of using C-V2X is that vehicles can become part of a much greater ecosystem where they can directly communicate with any device connected to the 5G network. For example, the real-time locations of pedestrians are available through direct side-link communications with nearby UEs, which can be utilized by automated/driverless vehicles. However, using C-V2X technology for V2V communication in the context of video streaming has some challenges.

Video streaming over IoCV needs explicit group management among the interested vehicles for V2V collaborations. However, peer vehicle discovery, group management, and multi-hop relay selection in C-V2X are challenging tasks. Unlike the shared medium of DSRC, C-V2X technology needs channel resource reservation prior to V2V communication. Vehicles contend among themselves to make the reservations. The 3GPP specification has prescribed *Semi-Persistent Scheduling* (SPS), which may make new participants starve for resources. The above-mentioned challenges can be the future research directions of this dissertation.

7.2.2 Exploring ICN paradigm for video streaming over IoCVs

In the context of video streaming over IoCVs, the contents are meant to be distributed among a large group of vehicles. It is irrelevant for a vehicle where the content is originated. As long as the content is authenticated and meets data integration criteria, a vehicle can procure the content from any peer vehicle of any location. The id or IP of the content originator is irrelevant. Therefore, the *Information Centric Network* (ICN) can be a better alternative compared to the IP-based routing paradigm where the routing is done based on the content. *Named Data Networking* (NDN) is one of the implementations of ICN, which has the in-built property of request aggregation and in-network caching. If a video is generated by a vehicle within an IoCV, it becomes the publisher of the content. NDN can distribute that content to the interested vehicles without the requirement of explicit group management.

However, though NDN is well-defined for wired networks, its behavior is still not well-structured for Ad-hoc wireless networks. The nodes of wireless networks have a single wireless interface of half duplex nature. As a result, interest packet forwarding, *Pending Interest Table* (PIT), and Forwarding Information Base (FIB) maintaining methods defined for wired networks cannot be applied to wireless networks. The behaviors of the PIT and

FIB of a vehicle need to be designed to handle the dynamic local neighborhood. Moreover, NDN cannot provide delay-constrained service, which is required for streaming services. The future research directions of this dissertation can be to mitigate the above-mentioned challenges of NDNs to provide video streaming over IoCVs.

7.2.3 Ensuring security in video services over IoCVs

In this thesis work, it is assumed that no vehicle in the considered scenario is selfish or malicious. However, as vehicles and OBUs are manufactured by different vendors, some of them may be malicious or contain vulnerabilities. As a result, any service over IoCV must be protected from selfish or malicious behaviors. A selfish vehicle tries to maximize utility at the cost of other vehicles' degraded performance. For example, in a multi-hop V2V data distribution scenario, upon receiving data packets from an upstream vehicle, a selfish vehicle may choose not to delegate them to its downstream vehicles. A selfish node can report false data so that it is always chosen as an edge node to receive data directly from the server. Game theory can be applied to mechanism design to mitigate selfish behaviors so that honest behavior becomes the most attractive strategic decision for all vehicles. Unlike a selfish node that works for its own interest, a malicious node's goal is to disrupt a service either fully or partially. As mentioned in Section 1.5.5, services over IoCVs are subject to various attacks like UDP flood attacks, TCP SYN attacks, HTTP GET attacks, Black-hole attacks, Grey-hole attacks, etc. Accordingly, attack detection and mitigation techniques need to be developed. The anonymity and privacy of the vehicles from any unauthorized entity need to be ensured. However, vehicle activities must be exposed to government law-enforcing agencies. Accordingly, pseudo-identity and certificates must be issued, distributed, and verified in an efficient manner. All the above-mentioned issues can be future directions of this thesis.

References

- [1] “IEEE Guide for Wireless Access in Vehicular Environments (WAVE) - Architecture,” *IEEE Std 1609.0-2013*, pp. 1–78, March 2014.
- [2] “Universal Mobile Telecommunications System (UMTS) LTE Architecture enhancements for V2X services (3GPP TS 23.285 version 16.4.0 Release 16),” *ETSI TS 123 285 V16.4.0 (2020-10)*, 2020.
- [3] “Application layer support for Vehicle-to-Everything (V2X) services,” *ETSI TS 123 286 V16.4.0 (2020-10)*, 2020.
- [4] P. Seeling and M. Reisslein, “Video Transport Evaluation With H.264 Video Traces,” *IEEE Communications Surveys Tutorials*, vol. 14, no. 4, pp. 1142–1165, Fourth Quarter 2012.
- [5] M. Blestel and M. Raulet, “Open SVC Decoder: A Flexible SVC Library,” in *Proceedings of the 18th ACM International Conference on Multimedia*, ser. MM ’10. New York, NY, USA: Association for Computing Machinery, 2010, p. 1463–1466. [Online]. Available: <https://doi.org/10.1145/1873951.1874247>
- [6] J. Sahoo, M. A. Salahuddin, R. Glitho, H. Elbiaze, and W. Ajib, “A Survey on Replica Server Placement Algorithms for Content Delivery Networks,” *IEEE Communications Surveys and Tutorials*, vol. 19, no. 2, pp. 1002–1026, 2017.
- [7] Z. Su, M. Dai, Q. Xu, R. Li, and H. Zhang, “UAV Enabled Content Distribution for Internet of Connected Vehicles in 5G Heterogeneous Networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5091–5102, 2021.
- [8] ETSI EN 302 665, “Intelligent transport systems (ITS); communications architecture,” Version 1.1.1, ETSI ITS WG2, Sophia Antipolis, France, Sep. 2010. [Online]. Available: http://www.etsi.org/deliver/etsi_en/302600_302699/302665/01.01.01_60/en_302665v010101p.pdf
- [9] “Vehicle-to-Everything (V2X) services in 5G System (5GS), Stage 3,” in *3GPP TS 24.587 version 16.1.1 Release 16*, 2020.
- [10] N. Cheng, N. Lu, N. Zhang, X. S. Shen, and J. W. Mark, “Vehicular wifi offloading: Challenges and solutions,” *Vehicular Communications*, vol. 1, no. 1, pp. 13 – 21, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S221420961300003X>

- [11] E. Ndashimye, S. K. Ray, N. I. Sarkar, and J. A. Gutiérrez, “Vehicle-to-infrastructure communication over multi-tier heterogeneous networks: A survey,” *Computer Networks*, vol. 112, pp. 144–166, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1389128616303826>
- [12] P. Papadimitratos, A. D. La Fortelle, K. Evenssen, R. Brignolo, and S. Cosenza, “Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation,” *IEEE Communications Magazine*, vol. 47, no. 11, pp. 84–95, November 2009.
- [13] “Cisco Visual Networking Index (VNI) Complete Forecast Update, 2017–2022,” https://www.cisco.com/c/dam/m/en_us/network-intelligence/service-provider/digital-transformation/knowledge-network-webinars/pdfs/1213-business-services-ckn.pdf, [Accessed:2022-01-06].
- [14] F. A. Silva, A. Boukerche, T. R. M. B. Silva, L. B. Ruiz, E. Cerqueira, and A. A. F. Loureiro, “Vehicular networks: A new challenge for content-delivery-based applications,” *ACM Computing Surveys*, vol. 49, no. 1, jun 2016. [Online]. Available: <https://doi.org/10.1145/2903745>
- [15] C. Cranor, M. Green, C. Kalmanek, D. Shur, S. Sibal, J. Van der Merwe, and C. Sreenan, “Enhanced streaming services in a content distribution network,” *IEEE Internet Computing*, vol. 5, no. 4, pp. 66–75, 2001.
- [16] G. Pallis, D. Katsaros, M. D. Dikaiakos, N. Loulloudes, and L. Tassioulas, “On the structure and evolution of vehicular networks,” in *2009 IEEE International Symposium on Modeling, Analysis Simulation of Computer and Telecommunication Systems*, 2009, pp. 1–10.
- [17] N. Loulloudes, G. Pallis, and M. D. Dikaiakos, “The dynamics of vehicular networks in large-scale urban environments,” in *2015 IEEE Conference on Collaboration and Internet Computing (CIC)*, 2015, pp. 192–199.
- [18] M. Fiore and J. Härri, “The networking shape of vehicular mobility,” in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc ’08. New York, NY, USA: Association for Computing Machinery, 2008, p. 261–272. [Online]. Available: <https://doi.org/10.1145/1374618.1374654>
- [19] D. Naboulsi and M. Fiore, “Characterizing the instantaneous connectivity of large-scale urban vehicular networks,” *IEEE Transactions on Mobile Computing*, vol. 16, no. 5, pp. 1272–1286, 2017.
- [20] I. Ahmad, R. M. Noor, I. Ahmedy, S. A. A. Shah, I. Yaqoob, E. Ahmed, and M. Imran, “VANET–LTE based heterogeneous vehicular clustering for driving assistance and route planning applications,” *Computer Networks*, vol. 145, pp. 128–140, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1389128618308417>
- [21] E. Yaacoub, F. Filali, and A. Abu-Dayya, “QoE Enhancement of SVC Video Streaming Over Vehicular Networks Using Cooperative LTE/802.11p Communications,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 1, pp. 37–49, Feb 2015.

- [22] S. Ucar, S. C. Ergen, and O. Ozkasap, "Multihop-Cluster-Based IEEE 802.11p and LTE Hybrid Architecture for VANET Safety Message Dissemination," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 2621–2636, April 2016.
- [23] A. Benslimane, T. Taleb, and R. Sivaraj, "Dynamic Clustering-Based Adaptive Mobile Gateway Management in Integrated VANET — 3G Heterogeneous Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 559–570, March 2011.
- [24] X. Duan, X. Wang, Y. Liu, and K. Zheng, "SDN Enabled Dual Cluster Head Selection and Adaptive Clustering in 5G-VANET," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, Sep. 2016, pp. 1–5.
- [25] K. Poularakis, G. Iosifidis, A. Argyriou, and L. Tassiulas, "Video delivery over heterogeneous cellular networks: Optimizing cost and performance," in *IEEE INFOCOM, Conference on Computer Communications*, 2014, pp. 1078–1086.
- [26] A. Al-Hilo, D. Ebrahimi, S. Sharafeddine, and C. Assi, "Revenue-driven video delivery in vehicular networks with optimal resource scheduling," *Vehicular Communications*, vol. 23, p. 100215, June 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2214209619302621>
- [27] Y. Sun, L. Xu, Y. Tang, and W. Zhuang, "Traffic Offloading for Online Video Service in Vehicular Networks: A Cooperative Approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 8, pp. 7630–7642, August 2018.
- [28] M. Hu, Z. Zhong, and C.-Y. Chang, "A Multicast Scheduling Approach for Layered Video Service in Vehicular Ad Hoc Networks," in *Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 2013, pp. 387–392.
- [29] L. Vigneri, T. Spyropoulos, and C. Barakat, "Low Cost Video Streaming through Mobile Edge Caching: Modelling and Optimization," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1302–1315, June 2019.
- [30] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Chapter 35.3, Introduction to algorithms (Third Edition)*. MIT press, 2009.
- [31] P. Jacquet, P. Minet, A. Laouiti, L. Viennot, T. Clausen, and C. Adjih, "Multicast optimized link state routing," *IETF manet draft-ietf-manet-olsr-molsr-01.txt*, 2002.
- [32] E. M. Royer and C. E. Perkins, "Multicast operation of the Ad-Hoc on-demand distance vector routing protocol," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, 1999, pp. 207–218.
- [33] S.-J. Lee, W. Su, and M. Gerla, "On-demand multicast routing protocol in multihop wireless mobile networks," *Mobile networks and applications*, vol. 7, no. 6, pp. 441–453, 2002.
- [34] F. Sun, B. Liu, F. Hou, H. Zhou, J. Chen, Y. Rui, and L. Gui, "A QoE centric distributed caching approach for vehicular video streaming in cellular networks," *Wireless Communications and Mobile Computing*, vol. 16, no. 12, pp. 1612–1624, August 2016. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/wcm.2636>

- [35] Y. Guo, Q. Yang, F. R. Yu, and V. C. M. Leung, "Cache-Enabled Adaptive Video Streaming Over Vehicular Networks: A Dynamic Approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 5445–5459, June 2018.
- [36] J. Blum, A. Eskandarian, and L. Hoffman, "Mobility management in IVC networks," in *IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No.03TH8683)*, June 2003, pp. 150–155.
- [37] B. Hassanabadi, C. Shea, L. Zhang, and S. Valaee, "Clustering in Vehicular Ad Hoc Networks using Affinity Propagation," *Ad Hoc Networks*, vol. 13, pp. 535–548, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1570870513002278>
- [38] Tao Song, Weiwei Xia, Tiecheng Song, and Lianfeng Shen, "A cluster-based directional routing protocol in VANET," in *2010 IEEE 12th International Conference on Communication Technology*, Nov 2010, pp. 1172–1175.
- [39] 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*, ser. MobiCom '00. New York, NY, USA: ACM, 2000, pp. 243–254. [Online]. Available: <http://doi.acm.org/10.1145/345910.345953>
- [40] V. Naumov, R. Baumann, and T. Gross, "An Evaluation of Inter-vehicle Ad Hoc Networks Based on Realistic Vehicular Traces," in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '06. New York, NY, USA: ACM, 2006, pp. 108–119. [Online]. Available: <http://doi.acm.org/10.1145/1132905.1132918>
- [41] H. Füßler, H. Hartenstein, M. Mauve, W. Effelsberg, and J. Widmer, "Contention-based forwarding for street scenarios," in *1st International workshop in intelligent transportation (WIT 2004)*, no. CONF, 2004.
- [42] S. Schnauffer and W. Effelsberg, "Position-based unicast routing for city scenarios," in *2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks*, June 2008, pp. 1–8.
- [43] A. Bachir and A. Benslimane, "A multicast protocol in ad hoc networks inter-vehicle geocast," in *The 57th IEEE Semiannual Vehicular Technology Conference, 2003. VTC 2003-Spring.*, vol. 4, April 2003, pp. 2456–2460 vol.4.
- [44] Y. Chen, Y. Lin, and S. Lee, "A Mobicast Routing Protocol in Vehicular Ad-Hoc Networks," in *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*, Nov 2009, pp. 1–6.
- [45] S. Gonzalez and V. Ramos, "Preset delay broadcast: a protocol for fast information dissemination in vehicular ad hoc networks (VANETs)," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, p. 117, 2016.
- [46] B. Tian, K. M. Hou, and H. Zhou, "The Traffic Adaptive Data Dissemination (TrAD) Protocol for both Urban and Highway Scenarios," *Sensors*, vol. vol. 16, no. No. 6, pp. 724–731, 2016.

- [47] G. Korkmaz, E. Ekici, F. Özgüner, and U. Özgüner, “Urban Multi-hop Broadcast Protocol for Inter-vehicle Communication Systems,” in *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, 2004, pp. 76–85. [Online]. Available: <http://doi.acm.org/10.1145/1023875.1023887>
- [48] S. Khakbaz and M. Fathy, “A reliable method for disseminating safety information in vehicular ad hoc networks considering fragmentation problem,” in *2008 The Fourth International Conference on Wireless and Mobile Communications*. IEEE, 2008, pp. 25–30.
- [49] R. Oliveira, C. Montez, A. Boukerche, and M. S. Wangham, “Reliable data dissemination protocol for VANET traffic safety applications,” *Ad Hoc Networks*, vol. 63, pp. 30 – 44, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1570870517300835>
- [50] Nawut Na Nakorn and Kultida Rojviboonchai, “DECA: Density-aware reliable broadcasting in vehicular ad hoc networks,” in *ECTI-CON2010: The 2010 ECTI International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, May 2010, pp. 598–602.
- [51] D. R. Chowdhury, V. K. Jain, and S. Maurya, “Travel Angle Based Fast Data Dissemination to Relevant Vehicles in VANET,” in *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, 2018, pp. 1–5.
- [52] D. R. Chowdhury and V. K. Jain, “VRDD: vehicular relevance-based data dissemination in a vehicular ad hoc network,” *IET Intelligent Transport Systems*, vol. 13, no. 12, pp. 1792–1803, 2019. [Online]. Available: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-its.2019.0294>
- [53] N. Wisitpongphan, O. K. Tonguz, J. S. Parikh, P. Mudalige, F. Bai, and V. Sadekar, “Broadcast storm mitigation techniques in vehicular ad hoc networks,” *IEEE Wireless Communications*, vol. 14, no. 6, pp. 84–94, 2007.
- [54] Z. J. Haas, J. Y. Halpern, and Li Li, “Gossip-based ad hoc routing,” in *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, June 2002, pp. 1707–1716 vol.3.
- [55] O. K. Tonguz, N. Wisitpongphan, and F. Bai, “DV-CAST: A distributed vehicular broadcast protocol for vehicular ad hoc networks,” *IEEE Wireless Communications*, vol. vol. 17, no. no. 2, pp. pages 47–57, April 2010.
- [56] H. Alshaer and E. Horlait, “An optimized adaptive broadcast scheme for inter-vehicle communication,” in *IEEE 61st Vehicular Technology Conference*, vol. 5, May 2005, pp. 2840–2844.
- [57] H. Alshaer and E. Horlait, “An optimized adaptive broadcast scheme for inter-vehicle communication,” in *2005 IEEE 61st Vehicular Technology Conference*, vol. 5, May 2005, pp. 2840–2844 Vol. 5.
- [58] L. A. Villas, A. Boukerche, G. Maia, R. W. Pazzi, and A. A. Loureiro, “DRIVE: An efficient and robust data dissemination protocol for highway and urban vehicular ad hoc networks,” *Computer Networks*, vol. vol. 75, pp. pages 381 – 394, 2014.

- [59] W. Viriyasitavat, O. K. Tonguz, and F. Bai, "UV-CAST: an urban vehicular broadcast protocol," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 116–124, November 2011.
- [60] C. Sommer, O. K. Tonguz, and F. Dressler, "Traffic information systems: efficient message dissemination via adaptive beaconing," *IEEE Communications Magazine*, vol. 49, no. 5, pp. 173–179, May 2011.
- [61] M. M. Taha and Y. M. Hasan, "VANET-DSRC protocol for reliable broadcasting of life safety messages," in *2007 IEEE International Symposium on Signal Processing and Information Technology*. IEEE, 2007, pp. 104–109.
- [62] S. Wenger, Y.-K. Wang, T. Schierl, and A. Eleftheriadis, "RTP payload format for scalable video coding," Tech. Rep., 2011.
- [63] Y.-K. Wang, M. M. Hannuksela, S. Pateux, A. Eleftheriadis, and S. Wenger, "System and Transport Interface of SVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1149–1163, 2007.
- [64] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," *IEEE MultiMedia*, vol. 18, no. 4, pp. 62–67, 2011.
- [65] H. K. Yarnagula, P. Juluri, S. K. Mehr, V. Tamarapalli, and D. Medhi, "QoE for Mobile Clients with Segment-Aware Rate Adaptation Algorithm (SARA) for DASH Video Streaming," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 15, no. 2, jun 2019. [Online]. Available: <https://doi.org/10.1145/3311749>
- [66] S. Kumar, R. Devaraj, A. Sarkar, and A. Sur, "Client-side qoe management for svc video streaming: An fsm supported design approach," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 1113–1126, 2019.
- [67] C. Rezende, A. Boukerche, H. S. Ramos, and A. A. F. Loureiro, "A Reactive and Scalable Unicast Solution for Video Streaming over VANETs," *IEEE Transactions on Computers*, vol. 64, no. 3, pp. 614–626, 2015.
- [68] M. D. Felice, E. Cerqueira, A. Melo, M. Gerla, F. Cuomo, and A. Baiocchi, "A distributed beaconless routing protocol for real-time video dissemination in multimedia VANETs," *Computer Communications*, vol. 58, pp. 40–52, 2015, special Issue on Networking and Communications for Smart Cities. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0140366414002990>
- [69] C. Rezende, A. Mhammeri, A. Boukerche, and A. A. Loureiro, "A receiver-based video dissemination solution for vehicular networks with content transmissions decoupled from relay node selection," *Ad Hoc Networks*, vol. 17, pp. 1–17, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1570870513003028>
- [70] F. Naeimipoor and A. Boukerche, "A Hybrid Video Dissemination Protocol for VANETs," in *2014 IEEE International Conference on Communications (ICC)*, June 2014, pp. 112–117.

- [71] A. Aliyu, A. H. Abdullah, N. Aslam, A. Altameem, R. Z. Radzi, R. Kharel, M. Mahmud, S. Prakash, and U. M. Joda, "Interference-Aware Multipath Video Streaming in Vehicular Environments," *IEEE Access*, vol. 6, pp. 47 610–47 626, 2018.
- [72] L. Junhai, Y. Danxia, X. Liu, and F. Mingyu, "A survey of multicast routing protocols for mobile Ad-Hoc networks," *IEEE Communications Surveys Tutorials*, vol. 11, no. 1, pp. 78–91, 2009.
- [73] L. Canourgues, J. Lephay, L. Soyer, and A.-L. Beylot, "STAMP: shared-tree ad hoc multicast protocol," in *MILCOM 2006-2006 IEEE Military Communications conference*. IEEE, 2006, pp. 1–7.
- [74] S. Y. Oh, J.-S. Park, and M. Gerla, "E-ODMRP: Enhanced ODMRP with motion adaptive refresh," *Journal of Parallel and Distributed Computing*, vol. 68, no. 8, pp. 1044–1053, 2008.
- [75] F. Wang, J. Liu, M. Chen, and H. Wang, "Migration Towards Cloud-Assisted Live Media Streaming," *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 272–282, 2016.
- [76] J. Wu and K. Ravindran, "Optimization algorithms for proxy server placement in content distribution networks," in *2009 IFIP/IEEE International Symposium on Integrated Network Management-Workshops*, 2009, pp. 193–198.
- [77] S. Ahuja and M. Krunz, "Algorithms for Server Placement in Multiple-Description-Based Media Streaming," *IEEE Transactions on Multimedia*, vol. 10, no. 7, pp. 1382–1392, 2008.
- [78] G. Rodolakis, S. Siachalou, and L. Georgiadis, "Replicated Server Placement with QoS Constraints," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 10, pp. 1151–1162, 2006.
- [79] N. Laoutaris, G. Smaragdakis, K. Oikonomou, I. Stavrakakis, and A. Bestavros, "Distributed Placement of Service Facilities in Large-Scale Networks," in *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, 2007, pp. 2144–2152.
- [80] Y. Jin, Y. Wen, and K. Guan, "Toward Cost-Efficient Content Placement in Media Cloud: Modeling and Analysis," *IEEE Transactions on Multimedia*, vol. 18, no. 5, pp. 807–819, 2016.
- [81] P. Krishnan, D. Raz, and Y. Shavitt, "The cache location problem," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 568–582, 2000.
- [82] C. Papagianni, A. Leivadreas, and S. Papavassiliou, "A Cloud-Oriented Content Delivery Network Paradigm: Modeling and Assessment," *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 5, pp. 287–300, 2013.
- [83] N. Bouten, J. Famaey, R. Mijumbi, B. Naudts, J. Serrat, S. Latré, and F. De Turck, "Towards NFV-based multimedia delivery," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015, pp. 738–741.

- [84] H. Ben-Ammar and Y. Ghamri-Doudane, "RevOPT: An LSTM-based Efficient Caching Strategy for CDN," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 1–6.
- [85] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: evidence and implications," in *IEEE INFOCOM '99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No.99CH36320)*, vol. 1, 1999, pp. 126–134 vol.1.
- [86] K. Lim, Y. Bang, J. Sung, and J.-K. K. Rhee, "Joint optimization of cache server deployment and request routing with cooperative content replication," in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 1790–1795.
- [87] M. Yang and Z. Fei, "A model for replica placement in content distribution networks for multimedia applications," in *IEEE International Conference on Communications, 2003. ICC '03.*, vol. 1, 2003, pp. 557–561 vol.1.
- [88] B. Wu and A. Kshemkalyani, "Objective-optimal algorithms for long-term web prefetching," *IEEE Transactions on Computers*, vol. 55, no. 1, pp. 2–17, 2006.
- [89] H. Wang, P. Liu, and J.-j. Wu, "A QoS-Aware Heuristic Algorithm for Replica Placement," in *2006 7th IEEE/ACM International Conference on Grid Computing*, 2006, pp. 96–103.
- [90] X. Tang and S. Chanson, "Analysis of Replica Placement under Expiration-Based Consistency Management," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 11, pp. 1253–1263, 2006.
- [91] A. Shaikh, R. Tewari, and M. Agrawal, "On the effectiveness of DNS-based server selection," in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*, vol. 3, 2001, pp. 1801–1810 vol.3.
- [92] C. Partridge, T. Mendez, and W. Milliken, "RFC1546: Host anycasting service," 1993.
- [93] H. A. Omar, W. Zhuang, and L. Li, "Gateway Placement and Packet Routing for Multihop In-Vehicle Internet Access," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 3, pp. 335–351, 2015.
- [94] W. Zhuang, Q. Ye, F. Lyu, N. Cheng, and J. Ren, "SDN/NFV-Empowered Future IoV With Enhanced Communication, Computing, and Caching," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 274–291, 2020.
- [95] Q. Yuan, H. Zhou, J. Li, Z. Liu, F. Yang, and X. S. Shen, "Toward Efficient Content Delivery for Automated Driving Services: An Edge Computing Solution," *IEEE Network*, vol. 32, no. 1, pp. 80–86, 2018.
- [96] B. B. Chen and M. C. Chan, "MobTorrent: A Framework for Mobile Internet Access from Vehicles," in *IEEE INFOCOM 2009*, 2009, pp. 1404–1412.

- [97] F. Bruno, M. Cesana, M. Gerla, G. Mauri, and G. Verticale, "Optimal content placement in ICN vehicular networks," in *2014 International Conference and Workshop on the Network of the Future (NOF)*. IEEE, 2014, pp. 1–5.
- [98] S. Retal and A. Idrissi, "A multi-objective optimization system for mobile gateways selection in vehicular Ad-Hoc networks," *Computers & Electrical Engineering*, vol. 73, pp. 289–303, 2019.
- [99] G. e. m. Zhioua, N. Tabbane, H. Labiod, and S. Tabbane, "A Fuzzy Multi-Metric QoS-Balancing Gateway Selection Algorithm in a Clustered VANET to LTE Advanced Hybrid Cellular Network," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 2, pp. 804–817, Feb 2015.
- [100] C.-A. La, P. Michiardi, C. Casetti, C.-F. Chiasserini, and M. Fiore, "Content Replication in Mobile Networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 9, pp. 1762–1770, 2012.
- [101] N. Khaitiyakun and T. Sanguankotchakorn, "An analysis of data dissemination on vanet by using content delivery network (cdn) technique," in *Proceedings of the 10th Asian Internet Engineering Conference*, 2014, pp. 37–42.
- [102] N. Kumar, J. J. Rodrigues, J. Lloret, and A. Dua, "Replication-aware data dissemination for vehicular ad hoc networks using location determination," *Mobile Networks and Applications*, vol. 20, pp. 251–267, 2015.
- [103] X. Jiang, F. R. Yu, T. Song, and V. C. M. Leung, "Resource Allocation of Video Streaming Over Vehicular Networks: A Survey, Some Research Issues and Challenges," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–21, 2021.
- [104] A. S. Chekkouri, A. Ezzouhairi, and S. Pierre, "A new integrated VANET-LTE-A architecture for enhanced mobility in small cells HetNet using dynamic gateway and traffic forwarding," *Computer Networks*, vol. 140, pp. 15 – 27, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1389128618301762>
- [105] H. Khan, S. Samarakoon, and M. Bennis, "Enhancing Video Streaming in Vehicular Networks via Resource Slicing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 3513–3522, 2020.
- [106] N. Abbas, Z. Dawy, H. Hajj, S. Sharafeddine, and F. Filali, "Traffic offloading with maximum user capacity in dense D2D cooperative networks," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.
- [107] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibañez, "Internet of Vehicles: Architecture, Protocols, and Security," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3701–3709, 2018.
- [108] "Speedtest global index," <https://www.speedtest.net/global-index>, [Accessed:2021-05-24].
- [109] D. B. Johnson, "Efficient Algorithms for Shortest Paths in Sparse Networks," *J. ACM*, vol. 24, no. 1, p. 1–13, Jan. 1977. [Online]. Available: <https://doi.org/10.1145/321992.321993>

- [110] “ns-3.30,” <https://www.nsnam.org/releases/ns-3-30/>, [Accessed:2020-02-02].
- [111] “Peak signal-to-noise ratio, wikipedia,” https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio, [Accessed:2020-05-04].
- [112] B. Liang and Z. J. Haas, “Virtual Backbone Generation and Maintenance in Ad Hoc Network Mobility Management,” in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, vol. 3, 2000, pp. 1293–1302 vol.3.
- [113] L. Jia, R. Rajaraman, and T. Suel, “An efficient distributed algorithm for constructing small dominating sets,” *Distributed Computing*, vol. 15, no. 4, pp. 193–205, 2002.
- [114] Y. L. Morgan, “Notes on dsrc amp; wave standards suite: Its architecture, design, and characteristics,” *IEEE Communications Surveys Tutorials*, vol. 12, no. 4, pp. 504–518, Fourth 2010.
- [115] F. Kuipers, R. Kooij, D. De Vleeschauwer, and K. Brunnström, “Techniques for Measuring Quality of Experience,” in *Wired/Wireless Internet Communications*, E. Osipov, A. Kessler, T. M. Bohnert, and X. Masip-Bruin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 216–227.
- [116] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang, “Developing a Predictive Model of Quality of Experience for Internet Video,” in *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, ser. SIGCOMM ’13. New York, NY, USA: Association for Computing Machinery, 2013, pp. 339–350. [Online]. Available: <https://doi.org/10.1145/2486001.2486025>
- [117] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, “A control-theoretic approach for dynamic adaptive video streaming over HTTP,” in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, 2015, pp. 325–338.
- [118] Y.-F. Ou, Z. Ma, T. Liu, and Y. Wang, “Perceptual Quality Assessment of Video Considering Both Frame Rate and Quantization Artifacts,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 3, pp. 286–298, 2011.
- [119] H. J. Kim and S. G. Choi, “A study on a QoS/QoE correlation model for QoE evaluation on IPTV service,” in *2010 The 12th International Conference on Advanced Communication Technology (ICACT)*, vol. 2, 2010, pp. 1377–1382.
- [120] “Simulation of Urban MObility,” <https://www.eclipse.org/sumo/>, [Accessed:2021-03-03].
- [121] S. Velmurugan, M. Errampalli, K. Ravinder, K. Sitaramanjaneyulu, and S. Gangopadhyay, “Critical evaluation of roadway capacity of multi-lane high speed corridors under heterogeneous traffic conditions through traditional and microscopic simulation models,” in *Journal of Indian Roads Congress*, vol. 71, no. 3, 2010, pp. 235–264.
- [122] “3gpp release 14,” https://www.etsi.org/deliver/etsi_tr/121900_121999/121914/14.00.00_60/tr_121914v140000p.pdf, [Accessed:2022-06-01].

- [123] Q. Chen, G. Yu, A. Maaref, G. Y. Li, and A. Huang, "Rethinking Mobile Data Offloading for LTE in Unlicensed Spectrum," *IEEE Transactions on Wireless Communications*, vol. 15, no. 7, pp. 4987–5000, 2016.
- [124] G. Raja, A. Ganapathisubramaniyan, S. Anbalagan, S. B. M. Baskaran, K. Raja, and A. K. Bashir, "Intelligent Reward-Based Data Offloading in Next-Generation Vehicular Networks," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3747–3758, 2020.
- [125] P. Dai, K. Liu, X. Wu, Z. Yu, H. Xing, and V. C. S. Lee, "Cooperative Temporal Data Dissemination in SDN-Based Heterogeneous Vehicular Networks," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 72–83, 2019.
- [126] A. S. Sadiq, K. A. Bakar, K. Z. Ghafoor, J. Lloret, and R. Khokhar, "An intelligent vertical handover scheme for audio and video streaming in heterogeneous vehicular networks," *Mobile Networks and Applications*, vol. 18, no. 6, pp. 879–895, 2013.
- [127] S. Céspedes and X. Shen, "On Achieving Seamless IP Communications in Heterogeneous Vehicular Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3223–3237, 2015.
- [128] J. Dias, A. Cardote, F. Neves, S. Sargento, and A. Oliveira, "Seamless horizontal and vertical mobility in VANET," in *2012 IEEE Vehicular Networking Conference (VNC)*, 2012, pp. 226–233.
- [129] S. Bi, C. Chen, R. Du, and X. Guan, "Proper Handover between VANET and Cellular Network Improves Internet Access," in *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, 2014, pp. 1–5.
- [130] Y. Hui, Z. Su, and T. H. Luan, "Collaborative Content Delivery in Software-Defined Heterogeneous Vehicular Networks," *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 575–587, 2020.
- [131] C. Li, Y. Zhang, T. H. Luan, and Y. Fu, "Building Transmission Backbone for Highway Vehicular Networks: Framework and Analysis," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8709–8722, 2018.
- [132] J. Kang, Y. Chen, R. Yu, X. Zhang, H. Chen, and L. Zhang, "Vertical handoff in vehicular heterogeneous networks using optimal stopping approach," in *2013 8th International Conference on Communications and Networking in China (CHINACOM)*, 2013, pp. 534–539.
- [133] Y. Hui, Z. Su, T. H. Luan, and J. Cai, "A Game Theoretic Scheme for Optimal Access Control in Heterogeneous Vehicular Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4590–4603, 2019.
- [134] B. Liang and Z. J. Haas, "Virtual Backbone Generation and Maintenance in Ad Hoc Network Mobility Management," in *Proceedings IEEE INFOCOM 2000, Conference on Computer Communications*, vol. 3, 2000, pp. 1293–1302 vol.3.
- [135] "OpenStreetMap," <https://www.openstreetmap.org>, [Accessed:2022-04-04].

- [136] M. Ben Brahim, Z. Hameed Mir, W. Znaidi, F. Filali, and N. Hamdi, "QoS-Aware Video Transmission Over Hybrid Wireless Network for Connected Vehicles," *IEEE Access*, vol. 5, pp. 8313–8323, 2017.



Publications

Research outputs from thesis:

Following are the research outputs for each of the thesis contributions.

Contribution 1: Debanjan Roy Chowdhury, Sukumar Nandi, Diganta Goswami, *Video streaming over IoV using IP multicast*, **Journal of Network and Computer Applications (JNCA)**, Volume 197, 2022, 103259, ISSN 1084-8045, <https://doi.org/10.1016/j.jnca.2021.103259>. [Q1, Impact Factor: 7.57]

Contribution 2: Debanjan Roy Chowdhury, Sukumar Nandi, and Diganta Goswami. 2022. *Distributed Gateway Selection for Video Streaming in VANET Using IP Multicast*. **ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)** 18, 3, Article 81 (August 2022), 24 pages. <https://doi.org/10.1145/3491388> [Q1, Impact Factor: 4.15]

Contribution 3: Debanjan Roy Chowdhury, Sukumar Nandi, and Diganta Goswami, *A Cost-effective Live video streaming for Internet of Connected Vehicles using Heterogeneous networks*. A revised version (minor) of the submitted manuscript is currently under consideration in the journal **Ad Hoc Networks**.

Contribution 4: Debanjan Roy Chowdhury, Sukumar Nandi and Diganta Goswami, *Cost-effective Distributed Edge vehicle selection for Multi-tier Video streaming over Vehicular network*, 2022 **IEEE 47th Conference on Local Computer Networks (LCN)**, 2022, pp. 227-232, doi: 10.1109/LCN53696.2022.9843472. [rank A]

Research outputs outside thesis:

Following are the research outputs outside of thesis works.

Collaboration 1: Pradeep Kumar Bhale, Debanjan Roy Chowdhury, Santosh Biswas and Sukumar Nandi. *OPTIMIST: Lightweight and Transparent IDS with Optimum Placement Strategy to Mitigate Mixed-rate DDoS Attacks in IoT Networks*, in **IEEE Internet of Things Journal**, vol. 10, no. 10, pp. 8357-8370, 15 May, 2023, doi: 10.1109/JIOT.2023.3234530. [Q1, Impact Factor: 10.23]

Collaboration 2: Roshan Singh, Debanjan Roy Chowdhury, Sunit Nandi, Sukumar Nandi, *ATOM : A Decentralized Task Offloading Framework for Mobile Edge Computing through Blockchain and Smart Contracts*. The manuscript has been **accepted and presented** in the conference **IEEE International Conference on Metaverse Computing, Networking and Applications (MetaCom) June 26-28, 2023, Kyoto, Japan**.

Brief biography of the Author

Debanjan Roy Chowdhury is from Kolkata, West Bengal, India. After completing school, he completed the B.Tech program in the discipline of Information Technology from the University of Kalyani, Nadia, West Bengal, in 2008. He has eight years of work experience while he worked in Azingo, NOKIA, and Ixia. In 2016, he joined the M.Tech program in the department of CSE at the PDPM Indian Institute of Information Technology Design & Manufacturing, Jabalpur (IIITDMJ). After completing the program in 2018 with the best M.Tech thesis of CSE, he joined the Department of CSE at the Indian Institute of Technology Guwahati (IITG) to pursue Ph.D. As a research scholar, he is interested in Vehicular Networks, Wireless Networks, and the Internet of Things. During his tenure as a research scholar from 2018 to 2023, he has published papers in JNCA, TOMM, LCN, and IOTJ. He is interested in vehicle-to-vehicle group management and content distribution using C-V2X side channels. Currently, he is employed in Cisco.

