Reputation-based Collaborative Approach to Rectify Selfish Vehicles in IEEE802.11p for VANET

Betelhem Yimer Nigatu

A Thesis Submitted to

The Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

February, 2022

Adama, Ethiopia

Reputation-based Collaborative Approach to Rectify Selfish Vehicles in IEEE802.11p for VANET

Betelhem Yimer Nigatu

Advisor: Ketema Adere (Ph.D.)

A Thesis Submitted to
The Department of Computer Science and Engineering
School of Electrical Engineering and Computing

Presented in Partial Fulfillment for the Degree of Masters of Science in
Computer Science and Engineering

Office of Graduate Studies
Adama Science and Technology University

February, 2022
Adama, Ethiopia

Declaration

I hereby declare that this Master Thesis entitled **"Reputation-based Collaborative Approach to Rectify Selfish Vehicles in IEEE802.11p of VANET"** is my original work. That is, it has not been submitted for the award of any academic degree, diploma, or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

Betelhem Yimer Nigatu	____________________	______________
Name of the student	Signature	Date

Recommendation

I, the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled **"Reputation-based Collaborative Approach to Rectify Selfish Vehicles in IEEE802.11p of VANET,"** prepared under my guidance by **Betelhem Yimer Nigatu** submitted in partial fulfillment of the requirements for the degree of Mater's of Science in Computer Science and Engineering**.**

Therefore, I recommend the submission of a revised version of the thesis to the department following the applicable procedures.

Ketema Adere (Ph.D)		
Major Advisor	Signature	Date

Approval Page

I, the advisor of the thesis entitled **“Reputation-based Collaborative Approach to Rectify Selfish Vehicles in IEEE802.11p of VANET,”** hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Ketema Adere (Ph.D)	______________________	_______________
Major Advisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis by **Betelhem Yimer Nigatu,** have read and evaluated the thesis entitled **“Reputation-based Collaborative Approach to Rectify Selfish Vehicles in IEEE802.11p of VANET”** and examined the candidate during the open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Computer Science Engineering.

______________________________	__________________	_______________
Chairperson	Signature	Date
______________________________	__________________	_______________
Internal Examiner	Signature	Date
______________________________	__________________	_______________
External Examiner	Signature	Date

Finally, approval and acceptance of the thesis are contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

________________________	___________________	_______________
Department Head	Signature	Date
________________________	___________________	_______________
School Dean	Signature	Date
________________________	___________________	_______________
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGEMENT

Before anything else, I would like to Thank and Praise God, the Almighty, and Holy Mother Virgin Merry for the things they offer to me, the blessings and grace that I get throughout my life, and the passion for carrying out this thesis work.

The Compilation of this thesis work would be hard to fulfill without my Advisor's assistant, Dr. Ketema Adere. I would love to reveal My enormous and deepest gratitude to him for his encouragement, guidance, and understanding. It was a huge honor for me to study under his advice. I would also like to appreciate and acknowledge My colleagues and classmates for their motivation and counsel. In addition, the researchers I met in the online Discord group for their comments and support in the fulfillment of the VANET simulation.

Moreover, I am supremely delightful to My Family, especially to My Mother, for her attention, prayers, and sacrifice that she pays for my life. It might be tough to imagine this success without her favor. As well, My number one cousin Mery, for her heartwarming support and advice. Lastly, the last appreciation goes for those who gave me any contribution to the realization of this thesis work.

TABLE OF CONTENT

ACKNOWLEDGEMENT iii
TABLE OF CONTENT ii
LIST OF FIGURES v
LIST OF TABLES vi
LIST OF ALGORITHMS vii
LIST OF ACRONYMS AND ABBREVIATIONS viii
ABSTRACT x
CHAPTER ONE 1
1. INTRODUCTION 1
1.1 Background of The Study 1
1.2 Motivation of The Study 3
1.3 Statement of the Problem 4
1.4 Research Questions 5
1.5 Objectives of the Study 5
1.5.1 General Objective 5
1.5.2 Specific Objectives 6
1.6 Methodology of the Study 6
1.7 Scope And Limitation of the Study 8
1.7.1 Scope of the Study 8
1.7.2 Limitation of the Study 9
1.8 Application of the Study 9
1.9 Organization of the Study 9
CHAPTER TWO 10
2. LITERATURE REVIEW AND RELATED WORK 10
2.1 Overview 10
2.2 Comparison of MANET and VANET 10
2.3 Architecture of VANET 12
2.4 Communication in VANET 13
2.4.1 DSRC/WAVE 13
2.4.2 IEEE802.11p MAC Protocol 14
2.5 Mobility Models for VANET 17

2.6 Behavior of Vehicles ... 18
2.6.1 MAC layer Selfish Behavior of Vehicles ... 19
2.7 Schemes for Solving Nodes' Selfishness Misbehavior ... 20
2.8 Related Work ... 21
2.8.1 Related Work Summary ... 25
CHAPTER THREE ... 26
3. PROPOSED SOLUTION ... 26
3.1 Overview of WC-CSMA/CA ... 26
3.2 General Research Assumption ... 28
3.3 Description of the proposed WC-CSMA/CA ... 28
3.3.1 Verifier Vehicle Selection ... 28
3.3.2 Identification Phase ... 31
3.3.3 Motivation Phase ... 35
3.4 Summary ... 39
CHAPTER FOUR ... 40
4. SIMULATION AND PERFORMANCE EVALUATION ... 40
4.1 Overview ... 40
4.2 VANET Simulation Tools ... 40
4.2.1 Mobility Generators ... 41
4.2.2 Network Simulators ... 42
4.2.3 VANET Simulators and Frameworks ... 43
4.3 Performance Evaluation Metrics and Simulation Setup ... 46
4.3.1 Performance Evaluation Metrics ... 46
4.3.2 Simulation Setup ... 46
4.4 Simulation Experiment ... 48
4.5 Simulation Result ... 52
4.5.1 The Effect of SV in terms of Average Throughput ... 53
4.5.2 The Effect of SV In terms of Average Waiting Time ... 54
4.5.3 The Effect of SV In terms of Average Queuing Delay ... 55
4.6 Simulation Result Analysis ... 55
4.6.1 Result Comparison In terms of Average Throughput ... 56
4.6.2 Result Comparison In terms of Waiting time ... 57

4.6.3 Result Comparison In terms of Average Queuing Delay 58
4.7 Evaluation Result 59
4.8 Result Summary 62
4.9 Discussion 64
CONCLUSIONS AND FUTURE WORK 66
Conclusion 66
Study Output 66
Contribution 67
Future Work 67
REFERENCE 68
APPENDIXES 74
Appendix A: 74
Appendix B: 75
Appendix C: 75
Appendix D: 76
Appendix E: 77
Appendix F: 78
Appendix G: 79
Appendix H: 80

LIST OF FIGURES

Figure 1-1: VANET framework 1
Figure 1-2:Methodology diagram..... 8
Figure 2-1: VANET WAVE architecture 14
Figure 2-2: Working principle of CSMA/CA 17
Figure 2-3:Vehicle behavior in VANET 19
Figure 3-1: Overall flow of the proposed solution 27
Figure 3-2: General network assumption for WC-CSMA/CA 28
Figure 3-3:Flow chart for verifier selection 30
Figure 3-4: Network scenario with verifier vehicles 31
Figure 3-5: SV identification phase 33
Figure 3-6:Identification phase flowchart diagram 34
Figure 3-7: Neighbor vehicle status 35
Figure 3-8: Flow chart for the motivation phase 38
Figure 3-9: SV motivation phase 38
Figure 4-1: Architecture of the Veins platform 44
Figure 4-2: Architecture of the VENTOS platform 44
Figure 4-3: Architecture of the EstiNet platform 45
Figure 4-4 Addis-to-Bishoftu sub-expressway (tested area) 48
Figure 4-5 Expressway tested area 49
Figure 4-6 Edited map.josm file of the tested area 49
Figure 4-7: Zoom out image of the generated traffic motion 50
Figure 4-8: Linear bidirectional road scenario 50
Figure 4-9: Simple GUI simulation for 50 nodes including the SV 51
Figure 4-10: SV GUI simulation 52
Figure 4-11: Throughput Vs. SV effect 53
Figure 4-12: Average waiting time Vs. SV effect 54
Figure 4-13: Average queuing delay Vs. SV effect 55
Figure 4-14: Average throughput comparison 56
Figure 4-15: Average waiting time Comparison 58
Figure 4-16: Average queuing delay comparison 59
Figure 4-17: CSMA/CA VS. WC-CSMA/CA 63

LIST OF TABLES

Table 2-1: VANET and MANET comparison 12
Table 2-2:WAVE protocol description(Hamdi et al., 2020)[3] 15
Table 2-3: Category access for MAC 16
Table 2-4: Node-Oriented schemes 21
Table 2-5 Related work summary 24
Table 4-1: Comparison of VANET mobility generators 42
Table 4-2: Comparison of VANET network simulator 43
Table 4-3: Comparison of VANET simulator 45
Table 4-4 Simulation setup parameters 47
Table 4-5: Simulation report 60
Table 4-6:Total results of CSMA/CA and WC-CSMA/CA 61
Table 4-7: CSMA/CA VS. WC-CSMA/CA 62
Table 4-8 Performance assessment of WC-CSMA/CA 65

LIST OF ALGORITHMS

Algorithm 1: Overall pseudocode for the proposed solution 27
Algorithm 2: Pseudocode for verifier vehicle selection 30
Algorithm 3: Pseudocode for the identification phase 33
Algorithm 4: Pseudocode for motivation phase 37
Algorithm 5: Code for channel access priorities 53

LIST OF ACRONYMS AND ABBREVIATIONS

A

AC: Access Class

AIFS: Arbitration Inter-Frame Interval

AU: Application Unit

B

BoM: Back-off Mechanism

BK: Background Traffic

BE: Best-Effort Traffic

C

CSMA/CA: Carrier Sense Multiple Access with Collision Avoidance

CW: Contention-Window

CWupper: Contention -Window Upper

CWlower: Contention -Window lower

CCH: Control Channel

CT: Connection Trial

C: Credit

D

DSRC: Dedicated-Short-Range - Communication

DCF: Distributed Coordination Function

Ds: Distance

DM: Decision Metrics

E

EDCA: Enhanced Distributed Channel Access

F

FMM: Freeway Mobility Model

I

IEEE802.11p

IM: Identification Metrics

ITS: Intelligence Transportation System

J

JOSM: Javascript Object Notation

L

LLC: Logical Link Control

Lo: Load

M

MMM: Manhattan Mobility Model

MANET: Mobile Ad Hoc Network

MAC: Media Access control

MOVE: Mobility Module Generator For Vehicular Environment

N

NT: Noteworthy

Net-edit: Network Editor

NS2: Network Simulator Two

NS3: Network Simulator Three

O

OMNET++: Objective Modular Network Testbed in C++

OBU: Onboard Unit

P

PDA: Personal Digital Assistants

Pt: Picking Threshold

PHY: Physical

Q

QoS: Quality of Service

R

*RCP**:** Resource Command Processor*

RWPM: Random-Way point Mobility Model

RSU: Road Side Unit

S

S: Score

SCH: Service Channel

SN: Selfish Node

SV: Selfish Vehicle

SCH: Service Channel

T

T: Throughput

TFT: Tit for Tat

TDMA: Time Division Multiple Accesses

TXOP: Transmission Opportunity

TTL: Time-To-Live

W

WAVE: Wireless-Access-In -Vehicular Environment

WC-CSMA/CA: Watchdog and Credit-based CSMA/CA

WLAN: Wireless Area Network

WIBSS: Wave Independent Basic Service Set

Wt: Waiting Time

V

VANET: Vehicular Ad Hoc Network

VO: Voice Traffic

VI: Video Traffic

VV: Verifier Vehicle

V-to-I: Vehicle to Infrastructure

V-to-V: Vehicle to Vehicle

VEINS: Vehicles in Network Simulator

VENTOS: Vehicular Network Open Simulator

ABSTRACT

Network resources are limited and finite, particularly for several users sharing the same medium. In other words, when the same users gain throughput and bandwidth advantage, others can lose their transmission opportunity. Particularly infrastructure-less ad hoc networks, like VANET(Vehicular Ad hoc Network), rely on the cooperation of nodes. Nodes(Vehicles) in VANET may behave selfishly and deviate from the regular and cooperative functionality, wishing to conserve their resource. The implementation of IEEE802.11p is to provide safe communication among vehicles, and it employs CSMA/CA(Carrier Sense Multiple Acess with Collision Avoidance) as a default medium access control. Concurring to CSMA/CA, all the vehicles with the packet to send should follow the channel first. The vehicle takes after the Backoff Mechanism(BoM) when the channel is busy. However, certain vehicles intentionally manipulate the BoM to increase their access probability and gain more bandwidth share than well-behaved vehicles. This problem leads to throughput degradation and channel crowdedness. This study developed a cooperative watchdog and credit-based CSMA/CA(WC-CSMA/CA) mechanism to overcome these issues by motivating the Selfish vehicles(SVs) to cooperate again in the network. It has two main phases. Identification phase: Assigned verifier vehicles(VVs) as a watchdog to monitor the vehicles' abnormal behavior and identify the SVs that manipulate the BoM. Motivation phase: Motivates the identified SVs to be well-behaved again by giving credits, considering their waiting time and number of connection trials, while accessing the channel. The simulation is conducted on OMNET++, using 25,50,100 vehicles for different vehicle densities in a 1km road. From the simulation result for 50 vehicles. The average throughput is increased from 12800.5mbit/sec to 15752.3mbit/sec, a 23.06% enhancement. Average waiting time and average queuing delay are reduced from 3.04ms to 2.5ms and 0.53ms to 0.45ms, which is a 17.94% and 15.5% improvement. Attaining the result, we have concluded that the proposed WC-CSMA/CA in the presence of SVs acquired a better performance in all metrics and different vehicle densities compared with the original CSMA/CA because of its cooperative mechanism.

Keywords: *BoM, Cooperative, CSMA/CA, IEEE802.11p, Motivation, WC-CSMA/CA, SVs*

CHAPTER ONE

1. INTRODUCTION

1.1 Background of The Study

Vehicular Ad Hoc Network(VANET) is a subpart of Mobile Ad Hoc Network (MANET), which is categorized under wireless network and organizes its communication system by itself without the need of any infrastructure(Ghori et al., 2018). Since discovering an Intelligent Transportation System (ITS), VANET has been getting more acceptance and has caught many researchers' eyes. ITS is one of the communication and information technology, which mainly target the transportation system. It aims to enhance transportation safety, reliability, productivity, and security. Collaborating with advances such as VANET, which creates wireless data communication between vehicles, the implementation of ITS applications is elevated (Malik et al., 2010). In VANET, modes of vehicular communication are labeled into two: vehicle within a vehicle (V-to-V) and vehicle within the infrastructure (V-to-I). Hereof, vehicles achieved all networking functions, they are assumed to be mobile nodes in the roadway, and the performances provided with short-range wireless communication. Figure 1.1 explains the VANET framework(Khan & Lee, 2019). The primary aims of this technology are to forecast warning messages to neighboring vehicles in case of car accidents, support emergency vehicles to pass other vehicles quickly, provide the latest actual-time traffic notification to the drivers(Jafari et al., 2012a).

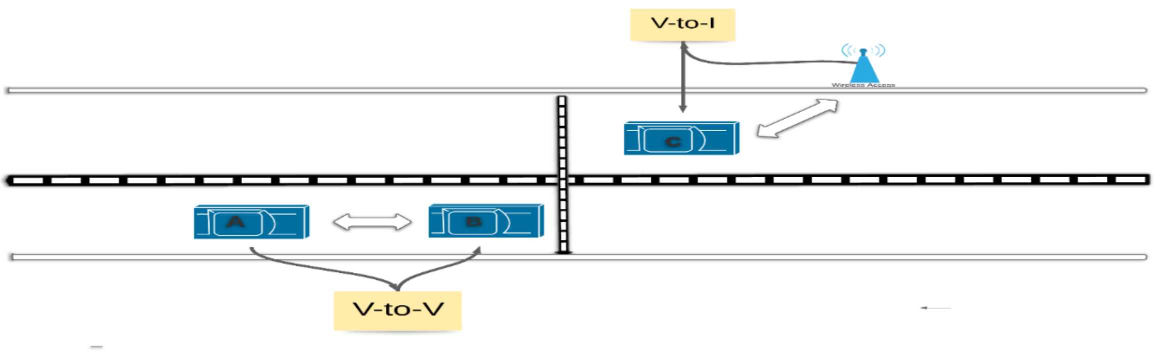


Figure 1-1: VANET framework[1]

Due to a few particular nature, VANET networks are divine from other kinds of ad hoc networks. Its basic features are a short-range communication period, high mobility, dynamic

[1]https://images.app.goo.gl/KdyjVGgBYhLqc7EZ7

topology, restricted bandwidth, node behavior, movement pattern, and node density, which can be mentioned as challenges(Hamdi et al., 2020).

The primary common communication standard for V-to-V and V-to-I is Dedicated-Short-Range -Communication (DSRC), which operates at a 5.9 GHz band (U.S Standard)(Lamssaggad et al., 2021). The media access control (MAC) and physical(PHY) layers are represented by the IEEE802.11p. In contrast, the IEEE1609 family of standards characterizes the Wireless-Access-in-Vehicular-Environment (WAVE), a suite of communication and security benchmarks with VANETs(Khan & Lee, 2019). IEEE802.11p is a standard supporting vehicular safety communication through the wireless network. It uses Enhanced Distributed Channel Access (EDCA) MAC sub-layer protocol, which is one of the mechanisms that bolster Quality of Service(QoS) in WAVE(Chang et al., 2016). In addition, it is a contention-aware medium access protocol that depends on CSMA/CA(Carrier-Sense Multiple-Access with Collision Avoidance)(Song, 2017).

Following the physical division of spectrum belongings, the MAC protocols are classified into MAC protocols with single-channel and MAC protocols with multi-channel. MAC protocols with multi-channel can simultaneously bolster packet transmissions on multiple channels and efficiently use spectrum resources through multiple transceivers and channel switching mechanisms. All multi-channel protocols adopt CSMA/CA as a core channel access mechanism(Ma et al., 2020). The working rule of this mechanism is that any node(vehicle) prepared to transmit will sense the channel first. On the off chance that it is unoccupied, the node might defer the transmission by selecting an arbitrary back-off time. Otherwise, the node follows a Back-off- mechanism (BoM) when a channel is in service. In the BoM strategy, when a node finds a busy channel, it selects a random back-off value in the interval of [0, Contention-WindowUpper (CWub)]. At any time collision happens, the size of the contention window is extended by multiplying its value until it reaches CWub. The CW is reset to its most negligible value upon an effective transmission. When a back-off value comes to zero, the node transmits immediately(Miao et al., 2011)(Hussein et al., 2016)(Gopinath et al., 2020).

IEEE802.11p requires the node to compute for accessing the channel by following the restriction rule of CSMA/CA, which means to hold up for a back-off interval sometime recently before any transmission. As mentioned earlier, limited bandwidth is one of the specific nature of VANET, which constrains vehicles; therefore, some vehicles may behave selfishly and refuse to cooperate with the network to conserve their resources(Pandey, 2015). Vehicle misbehavior is considered abnormal behavior that's the deviation from the expected behavior of other vehicular nodes(vehicles) within the VANET. For example, certain vehicles willfully abuse the MAC protocol to gather more network resources than well-behaved vehicles. These vehicles may select to hold up for a smaller back-off interval, subsequently increasing their chance of getting to the channel and thus diminishing the network throughput share received by the well-behaved station(Kyasanur & Vaidya, 2005)(Cárdenas et al., 2004). This kind of selfish behavior is called BoM manipulation misbehavior.

Consequently, these selfish vehicles (SVs) significantly impact the general network operation. Many kinds of research and studies that had been done based on IEEE802.11p of VANET have considered vehicles as well-behaved. Still, when we come to, in reality, not all vehicles that participate in channel accessing are willing to cooperate. This thesis aims to enrich the performance of the CSMA/CA protocol by resolving the BoM manipulation problems that occur due to the misbehavior of SVs during channel access by utilizing a cooperative mechanism. Since distinct features of VANET, many solutions proposed for MANET might not be suitable for VANET(Uzma Khan & Abstract, 2015). Also, except for a few articles, non-cooperative node-based studies only included SVs identification in their research. Still, their solutions do not involve motivating and reactivating methods for SVs to make them cooperate again. In this case, we proposed a solution to improve the CSMA/CA protocol to overcome the aforementioned issue. The proposed WC-CSMA/CA solution utilizes a cooperative mechanism, which first selects verifier vehicles for monitoring purposes. Then, it employs a watchdog mechanism to identify the SVs and motivates them by giving them credits if they are willing to revive again. The main improvement of this thesis was to propose a solution for BoM manipulation caused by the SV in IEEE802.11p. Using the proposed WC-CSMA/CA solution, we aim to rectify the SV to cooperate and be well-behaved in the network.

1.2 Motivation of The Study

Network resources are limited and finite, particularly for several users sharing the same medium. In other words, when the same users gain throughput and bandwidth advantage, others can lose their transmission opportunity, coming in zero user throughput. Subsequently, one user(node) possesses the channel for the whole period, in case of transmitting valuable data or not, which turns to total selfish misbehavior. When this occurs, the complete network appears to be dysfunctional and not inexactly available to the usual node(Lu et al., 2010). Particularly infrastructure-less ad hoc networks, like VANET, rely on the cooperation of vehicles. To achieve and take care of essential functions like data packet forwarding, routing, and network administrations, however, due to its different features, such as short-range communication period, high mobility, restricted bandwidth. As a result, vehicles may behave selfishly and deviate from the regular functionality, wishing to conserve their resource(Mantas et al., 2018). The implementation of IEEE802.11p is to provide safe communication among vehicles, and it employs CSMA/CA as a default medium access control. However, certain vehicles intentionally abuse this MAC protocol by misusing the channel accessing rule and manipulating the BoM to gain a higher probability of accessing the channel than well-behaved vehicles. This problem leads to throughput degradation, network partitioning, channel crowdedness, total or partial loss of information(Pandey, 2015). The above issues are the primary motivation of this study to propose a cooperative mechanism based on VANET's selfish behavior of vehicles when accessing the channel, which identifies vehicles that behave selfishly and motivates them to revive again in the network.

1.3 Statement of the Problem

Vehicular communication aims to provide better safety and infotainment benefit to the drivers through successful communication. The IEEE802.11p standard is created with EDCA contention-based priority-oriented medium access strategy. For attaining this, it uses CSMA/CA, the medium access control component used as a default in VANET. Concurring to CSMA/CA, all the nodes with the packet to send will follow in the process called external contention for gaining channel access. The node takes after the BoM when the channel is busy. The BoM uses an algorithm that employs input to generatively decrement the weight of some processes in organizing to discover an acceptable rate gradually. Despite VANET's high mobility nature, limited bandwidth, and short-range communication, regularly manipulating the BoM has been considered the fundamental issue based on the misbehavior of vehicles. Vehicles; are called greedy or selfish if they utilize some techniques to decrease the back-off value, which increases their access probability and allows them to gain more bandwidth share. Some common strategies adopted by SV are CW cheating Strategy, fixing CW Strategy, back-off cheating Strategy(Al-Terri et al., 2017).

By applying those strategies, SV is concerned about reducing their waiting time for faster channel access and penalizing the other host vehicles. At that point, it does not respect the ground rule of the channel access strategy. It tries continuously to connect to the medium and mention its use, which dishonor the network process and diminish the QoS that occurs when a networked vehicle denies respecting the restriction of the channel access rule and continuously tries to connect to the channel and keep up their use(M. Li et al., 2015),(Ismail et al., 2021),(van der Heijden et al., 2013).

The typical effects that are caused by the SV which manipuplate the BoM in MAC protocol of IEEE802.11p are throughput degradation, service interruptions, total or partial loss of information, network partitioning and channel crowdedness. We proposed a cooperative solution to resolve this selfish misbehavior problem and improve the CSMA/CA protocol. The proposed solution is setting the verifier vehicles to monitor and identify the SVs by employing a watchdog mechanism and motivating the identified SVs using a credit-based approach to revive these vehicles to participate cooperatively in the network. We aim to make the SV corporate and well-behaved in the network by developing the proposed WC-CSMA/CA mechanism. The network simulation was finalized using VANET standard simulation tools; SUMO to generate a traffic motion and OMNET++ to simulate the network using the vines framework for bidirectional linear road traffic.

1.4 Research Questions

By considering the above critical issues, the following questions are going to be answered by this thesis work:

RQ1: How to resolve the SVs effect in CSMA/CA IEEE802.11p by collaborating a watchdog approach with a credit-based approach?

RQ2: How to design a new mechanism for Selfish Vehicles to identify and motivate them?

RQ3: How can we assess the performance of the newly designed WC-CSMA/CA mechanism with the existing CSMA/CA IEEE802.11p of VANET?

1.5 Objectives of the Study

1.5.1 General Objective

The general objective of this study is to rectify the Selfish Vehicles in CSMA/CA IEEE802.11p of VANET using a Reputation-based collaborative approach.

1.5.2 Specific Objectives

The following listed objectives are needed to meet the general goal and be performed during the study.

- To resolve the SVs effect in CSMA/CA IEEE802.11p by collaborating a watchdog approach with a credit-based approach.
- To design a new mechanism for Selfish Vehicles to identify and motivate them.
- To assess the performance of the newly designed WC-CSMA/CA mechanism with existing CSMA/CA.

1.6 Methodology of the Study

The following methods are used to accomplish the above objectives and answer the research questions.

Stage 1: Identifying the research area

The first and fundamental step is identifying the research area for better understanding to minimize and specify the scope of finding the problem. Generally, this phase focuses on surveying CSMA/CA defined by IEEE802.11p. The tasks that have been fulfilled are identifying different channel access mechanisms used under the MAC layer figuring out and analyzing challenges in this area. Also, suggested solutions, training, and required tools for integration and simulation were assessed. This survey helps to come up with identifying the central gap which exists in CSMA/CA in VANET and finding out the proposed and possible solution for the identified problem. Different articles and papers will be used, investigated, and listed out from standard libraries such as IEEE explore, science direct, springer.

Stage 2: Formulating research problem and analyzing related work

After identifying the broad research area and marking up the area of interest by reviewing and assessing different materials, the problem that needs to be addressed and resolved will be formulated explicitly. Fundamental research questions that aim to be answered or any assumptions or assertions that the thesis challenges or investigates are drafted out in this stage. Also, this stage involves analyzing related works familiar with the research topic by identifying their limitation and some future results. Finally, it allows identifying the research problem between the existing solutions and the proposed solution.

Stage 3: Designing identification and motivation phase of the proposed solution

This stage is where the framework and workflow of the Proposed WC-CSMA/CA are taken place. Pseudocode, flow chart for the system for identification and motivation phase, the step that shows how the solution works, its overall process, and basic scenarios will be mentioned under these topics. The lucid-chart and draw.io software is used to draw different flowcharts and diagrams to explain the proposed solution in detail.

Stage 4: Creating actual traffic motion using SUMO and Net-edit

Before simulating the proposed solution, the actual traffic model will be created for the linear bidirectional lane. We used open-street-map to generate a map from Google Maps, SUMO for traffic simulation, and Net-edit for editing the vehicular network. The sample map was taken from the Addis-to-Adama expressway, and SUMO generates the traffic movement. Designing this traffic motion aims to show how vehicles are moved on the road by following linear lines and assigning and listing how many vehicles are needed for the proposed solution.

Stage 5: Simulating the proposed system using OMNET++

In this stage, the simulation was conducted using OMNET++ by integrating the traffic motion SUMO file with the veins framework, a vehicular communication simulation tool. We have simulated the proposed WC-CSMA/CA solution to show how it works and how the results make a change, basically to elaborate how the solution improves the performance of CSMA/CA. Then, C++, Python, and XML programming languages modify the existing file for the proposed solution.

Stage 6: Reporting, evaluating, and discussing the result

To evaluate the performance of the proposed solution, in this stage, we are going to compare the WC-CSMA/CA with the existing CSMA/CA channel access mechanism. The performance is assessed based on average throughput, waiting time, and queuing delay. This phase discusses the research outcome based on the simulation result. Charts and diagrams show their result output and tables using MS Excel 2019 to visualize simulation results using the graph and MS office word as a writing tool for this thesis.

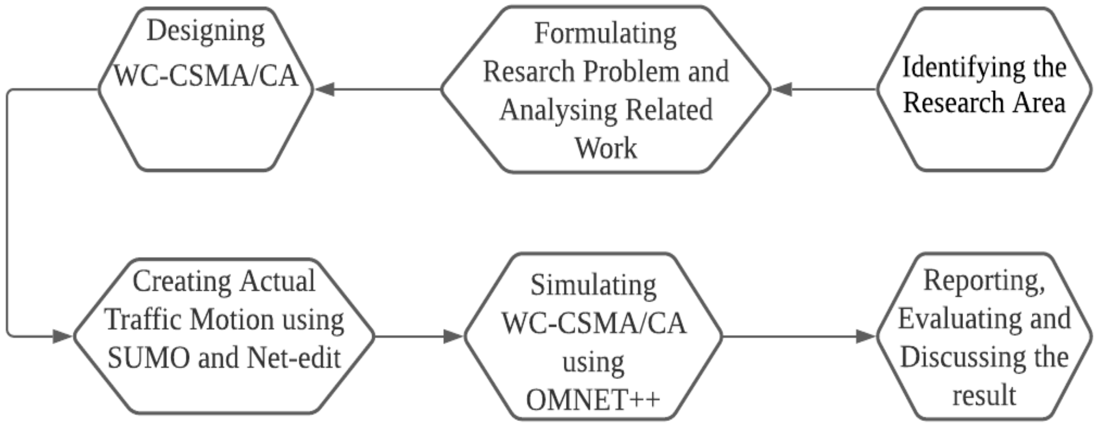


Figure 1-2:Methodology diagram

1.7 Scope And Limitation of the Study

1.7.1 Scope of the Study

This thesis focuses on the CSMA/CA channel accessing protocol in IEEE802.11p of VANET in CSMA/CA. It only focuses on BoM manipulation misbehavior driven by the SV. The proposed solution aims to rectify the misconduct of vehicles, and it has two main phases: the identification and motivation phases. In the identification phase, the watchdog mechanism has been used to identify the SV using verifier vehicles. The motivation phase is rest on the credit-based approach to revive the SV in the network and make them cooperative. The newly WC-CSMA/CA mechanism improves the performance of the existing CSMA/CA mechanism by motivating the selfish vehicles to be cooperative and decrease their selfish behavior. The performance evaluation metrics are average throughput, waiting time, and queuing delay. The simulation is executed by 25,50 and 100 vehicles, including the SVs in the existing CSMA/CA.

Since VANET is a broad research topic, this study focuses on the CSMA/CA of IEEE802.11p. This thesis did not include other types of protocols and standards. And also, different challenges of VANET did not cover in this work, such as collision, node mobility, and contention. As mentioned earlier, only the nodes/vehicles' selfish misbehavior is targeted in this thesis. It aims to improve the CSMA/CA performance by applying a cooperative solution in CSMA/CA. In addition, the thesis will not consider the uncooperative behavior driven by malicious nodes or a desire to reduce energy consumption.

1.7.2 Limitation of the Study

Since time is a crucial measure for thesis compilation, this thesis does not deal with other nodes/vehicles misbehavior, such as packet drop selfish behavior and link breakage selfish behavior except backoff manipulation during channel access. In addition, due to the requirement of additional study and scope constraints, VANET security issues and node density was not included in this thesis. Also, the simulation was conducted only in linear bidirectional road scenarios.

1.8 Application of the Study

This thesis contributes to CSMA/CA of IEEE802.11p performance in the multi-channel protocols of MAC in VANET. VANET technologies are primarily utilized in ITS to enhance transportation quality. Therefore, we aimed to simulate vehicles' selfish misbehavior, which explicitly affects the CSMA/CA of working principle. The simulation was performed under the standard of CSMA/CA by implementing a watchdog and credit-based approach in the newly proposed WC-CSMA/CA using SUMO and OMNET++ simulation tools. From this study, the reduction of SV impact on IEEE802.11p of VANET will be expected. The primary significance of this work is to rectify the selfish behavior of the MAC protocol, specifically in CSMA/CA driven by SV, by identifying the selfish misbehavior of vehicles and motivating them to cooperate again in the network using two basic approaches: mentioned earlier. Finally, the simulation result will be assessed in terms of those performance metrics: Average throughput, Average waiting time, and Average queuing delay.

1.9 Organization of the Study

This thesis is organized as follows into Four chapters. Each chapter is formulated according to different topics. Chapter one includes the introduction, problem of the statement, motivation, objectives, methodology, scope, limitation, and application of the thesis work. In chapter two, literature reviews and related work are explained briefly. The section gives a general clue for the reader to know more about the research area. It addressed the problem, which is targeted to be resolved by the thesis. The third chapter describes the proposed WC-CSMA/CA mechanism, including its’ general working principle and architecture. In chapter four, simulation analysis and resalt evaluation are involved. Finally, the last part describes the conclusion, recommendations, and future work in this research area.

CHAPTER TWO

2. LITERATURE REVIEW AND RELATED WORK

2.1 Overview

This chapter is set apart into two portions. The first portion is an introductory part, which gives a brief description of the domain area of this thesis work. The primary concept, which will be addressed, is an introduction to MANET and VANET. Since VANET is a subcategory of MANET, it is better to distinguish some of its driven and different characteristics from MANET. For this reason, detailed characteristics and various features are mentioned. Wireless vehicular communication of VANET is also described here. The overall IEEE802.11p wireless standard used in VANET is discussed, including its architectural components, specifically the MAC layer. The CSMA/CA, the default channel access mechanism that supports the multi-channel MAC protocols, is given briefly in this session by covering its working principle. Mobility model, node(vehicles) behavior, particularly vehicles misbehavior in CSMA/CA, which is driven by selfish vehicles, and different techniques and algorithms used for identifying the SVs are also discussed in this part. In the second portion, a survey of recent related work is given, concerning different schemes and algorithms used in IEEE802.11p, for identifying and motivating selfish vehicles.

2.2 Comparison of MANET and VANET

MANET is a stand-alone wireless network established without utilizing infrastructure to function. In the include of MANET, during communication, the nodes (phone, laptop, PDA) can set up and take off the network by auto-configuration. Also, the node moves in random form; these feature of nodes makes the topology uncertain(Goyal et al., 2011). Communication between nodes is accomplished once a sender and receiver nodes link or connect. For example, suppose a receiver node lies on the sender node communication range. In that case, it will communicate directly unless the communication is completed with the assistance of the intermediate node, as it takes part and acts as a router(Raja & Baboo, 2014).

As noted earlier, VANET is a unique class of MANET. Meaning that the wireless network idea of MANET was implemented in vehicular communication in an advanced form(Hamdi et al., 2020),(Ritesh Gupta & Patel, 2016). Employs vehicles as mobile/moving nodes in

MANET to make a short-term network without utilizing any infrastructure(Jain & Chahal, 2016),(Sood & Kanwar, 2014). The opportunity of street forward data exchange among vehicles under an ad-hoc network environment can be defined as VANET(Eze et al., 2016). It has acquired more considerations from the government, car manufacturers, and researchers because of the emerging concept of the ITS. This technology is implemented to monitor and smartly control vehicular traffic by employing different technologies and communication modes and systems, i.e., VANET(Raut & Devane, 2018).

Both MANET and VANET have some common characteristics and techniques, like, self-organization, low bandwidth, no centralized nodes. VANET has some distinct characteristics and features, making such a bit one of a kind out of MANET. Using various terms, significant aspects of VANET, which make it different from MANET, are stated as follow:

In terms of mobility: In MANET, mobile nodes are independent of migrating in any direction, while within VANET, vehicles have to follow a specific pathway. It possibly moves within the same or opposite direction on a pre-defined road (rural, urban), so mobility patterns (Pattnaik & Pattanayak, 2017).

In terms of power constraints: The MANET's nodes' size is smaller; they have limited memory, storage, and power capacity. Moreover, unlike MANET, VANET vehicles have gained an adequate power supply from automobile batteries (Chouhan et al., 2016).

In terms of network topology: Dissimilar to a MANET with the irregular movement of nodes, vehicles in VANET tie and leave out the network many times. These characteristics make revamp in the topology (Jain & Chahal, 2016).

In terms of node density: MANET has a scattered node density, however in VANET, there is probably a high density, and traffic in some roads and other roads may not have yielded withinside the variable density of nodes in the area. This feature makes it a more significant frequent variable, leading to a frequent disconnection (Ritesh Gupta & Patel, 2016).

In terms of intermittent connectivity: The change of network topology permanency in MANET is not that fast compared to VANET. Even so, there is a rapid and very dynamic change in network topology. This results in frequent network disconnection, the link between vehicles' stability can be disconnected quickly. In addition, the density of vehicles varies in different rode scenarios, which makes this problem worse (Chouhan et al., 2016).

In terms of network size: The dense area, such as highways, and downtown, required a massive network. Therefore, the VANET routing strategy must be qualified on a large-scale network. This result showed that the network size is geographically boundless(Ritesh Gupta & Patel, 2016). There are also other differences which are described in table 2.1.

Table 2-1: VANET and MANET comparison

No.	Characteristics	VANET	MANET
1.	Production cost	Expensive	Inexpensive
2.	Topology change	Frequent	Scattered
3.	Node density	Varies frequently	Sparse
4.	Transmission range	Up to 500	Up to 100
5.	Bandwidth	1000 Kps	100 Kps
6.	Mobility pattern	regular	Random

2.3 Architecture of VANET

VANET network is composed of vehicles prepared with sensors, settled infrastructure (access point set within the RSU), and wireless communication (supported by IEEE802.11p) for creating communication and permitting them to exchange data with every other.

The VANET architecture can be divided into ad hoc and infrastructure domains (Ayyappan & Mohan Kumar, 2016).

A. **Ad-hoc Domain:** This domain consists of mobile nodes, such as furnished vehicles provided with intelligent systems. It essentially consists of two components for wireless access, the onboard unit (OBU) and the application unit (AU).

On-Board Unit: An OBU is a vehicle-mounted wireless device that consists of memory, a resource command processor(RCP), user interface, and short-range communication device using IEEE802.11p radio technology for non-safety application(Ritesh Gupta & Patel, 2016)(Jafari et al., 2012b).

Application Unit (AU): An AU is a dedicated or standard device used for safety applications. The difference between OBU and AU is rational. The AU can only communicate through the OBU, which is obligated to all network tasks and mobility (Ritesh Gupta & Patel, 2016)(Jafari et al., 2012b).

B. Infrastructure Domain: Comprises the roadside unit (RSU) and other access networks implemented to maintain vehicular communication.

Roadside Unit: RSU is a static state located next to streets and other designated locations and is suitable for short-range communication with devices that use IEEE802.11p radio protocol technology. It is used to extend the reach of VANET communication and other routing strategies(Ritesh Gupta & Patel, 2016).

2.4 Communication in VANET

There are two kinds of communication in VANET, which IEEE802.11p supports.

V-to-V: No physical infrastructure is required. The vehicles communicate independently and exchange data. This kind of communication is taken care of by the OBU (installed in cars), and a multi-hop broadcast is used to send and receive data between them.

V-to-I: This kind of communication establishes a transmission linkage between the vehicles and the infrastructure over a high bandwidth. Mainly RSU is used, which can be installed on the side of the road with a certain distance between them. Communication takes place via a WAVE as a wireless medium(Malik et al., 2010).

2.4.1 DSRC/WAVE

VANET's wireless access standards are categorized into three. Short-range, long-range, and medium-range, including cellular systems, WLAN/Wi-fi standards, DSRC/WAVE standards. In VANET, short and medium-range communication services are supported by DSRC/IEEE802.11p for V-to-V and V-to-I communication. The DSRC is fully designed to meet VANET's requirements, including self-organization, self-configuration**,** high mobility, and dynamic topology. In the US, DSRC operates on a 75 MHz spectrum in a 5.9 GHz frequency band, while in Europe and Japan, it works on a 30 MHz spectrum in the 5.8 GHz band(Y. Li, 2015). It is used for various applications, including safety messaging, information dissemination for real-time traffic, collecting toll at multiple points, (Malik et al., 2010). It offers a communication range of 1 km and supports a data rate. With DSRC, the RSU and the OBU communicate via the IEEE802.11p standard WAVE. It is the essence of DSRC. It uses it to exchange information between vehicles. In general, the terms DSRC and WAVE are used interchangeably. IEEE802.11p is a backbone for the physical and MAC layers of the DSRC/WAVE protocol(Kamble & Kounte, 2019).

The IEEE 802.11 standard was originally developed for MANET. It provides information about node mobility. However, due to vehicles' high mobility, it is inherently not suitable for VANET. Therefore, the extended IEEE802.11b standard was modified and developed. The standard also poses some challenges to VANET, such as high mobility, vehicle speed, and traffic patterns. Finally, the researchers developed a brand-new version of IEEE802.11b, namely IEEE802.11p. This standard applies to the MAC and Physical layer and can be used in high-speed vehicles to achieve smooth data exchange.(Karagiannis et al., 2011),(Hartenstein & Laberteaux, 2010),(M.Shahid Anwer, 2017). Fig 2.1 illustrates the WAVE architecture, and table 2.1 describes each standard.

2.4.1 IEEE802.11p MAC Protocol

The MAC protocol can ensure fair channel access between vehicles, reduce access delay and maximize channel usage. However, the VANET MAC protocol must adjust to rapid topology changes, rapid vehicle mobility, high bandwidth requirements, and different QoS requirements(Menouar et al., 2006). Therefore, according to the physical distribution of the spectrum resources, the VANET MAC protocol is classified into MAC protocol with a single-channel and MAC protocol with a multi-channel.

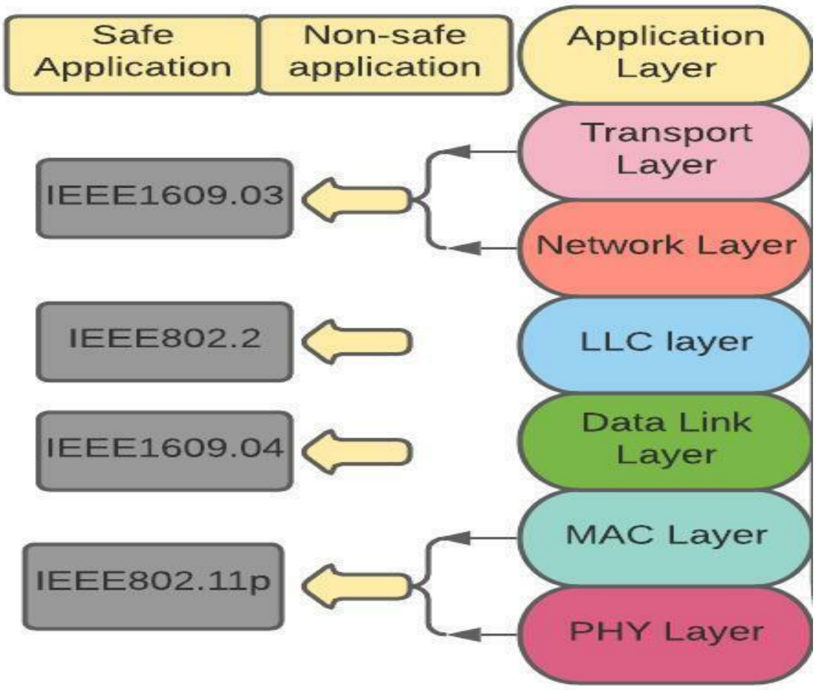


Figure 2-1: VANET WAVE architecture[2]

[2]https://th.bing.com/th/id/OIP.tpD8eX7ySAMSfqCNUe64KAHaF7?pid=ImgDet&rs=1

Table 2-2:WAVE protocol description(Hamdi et al., 2020)[3]

WAVE Layer	STANDARDS	Description
Transport layer	IEEE1609.03	Control WAVE System and addressing Routing service
Network layer		
LLC(Logical Link Control)	IEEE802.2	With the help of the IEEE802.2 Sublayer, the data link layer can send some data over the network.
Data-link layer	IEEE1609.04	Specifies the IEEE802.11p MAC extensions to support multi-channel operation.
MAC layer	IEEE802.11p	Determines the MAC and PHY actions required by an IEEE 802.11device to work on VANET.
PHY layer		

The MAC protocol with a single channel mainly uses carrier identification and multiple access mechanisms to solve the numerous node resource allocation problems. Each communication session ends on a medium to solve transmission conflicts. Transmission of multichannel data packets and effectively using the spectrum's resources, with multiple transceivers or channel frequency hopping mechanisms. According to the hierarchical structure of VANET, multi-channel MAC classification protocols use the sub-layer standard EDCA, which is one of the mechanisms supporting QoS in WAVE. It is CSMA-based channel access and collision avoidance protocol(Reddy & Ramanathan, 2018).

The above architecture and description table show that the MAC WAVE layer is standardized for IEEE802.11p. Therefore, use the MAC EDCA sub-layer protocol with different traffic classes. EDCA is an extended version of the essential Distributed Coordination Function (DCF), initially developed for IEEE802.11e. DCF sticks with the channel access procedure with CSMA/CA and provides collision avoidance for distributed and scattered wireless communications. Therefore, mobile nodes based on IEEE802.11p should pursue the contention-based EDCA pattern as a method for MAC with different Data Access classes (AC), with distinct transmission priorities. Namely AC4, AC3, AC2, and AC1 with four available data traffic categories under these data access levels, which are Background Traffic (BK), Best-Effort Traffic (BE), Voice Traffic (VO), and Video Traffic

[3]https://th.bing.com/th/id/OIP.r5ekhPqerMQLbi-MK8868wHaEx?pid=ImgDet&rs

(VI). Each category has parameters to access, such as Arbitration-Inter-Frame-Interval (AIFS), contention window (CW), and Transmission Opportunity (TXOP)(Chang et al., 2016).

Table 2-3: Category access for MAC

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

2.4.1.1 CSMA/CA

The IEEE 802.11p spectrum consists of six service channels (SCH) and one control channel (CCH). The CCH can be used for broadcast communications compromised with high-priority data and management frames, particularly secure communications. It must be the privileged channel used to send safety application messages. The SCH can be used for unicast communication, transmission, and service and safety applications. The MAC layer in IEEE802.11p is just like the IEEE802.1e QoS extension. They are grouped into one of four unique queues based on their ranking. Each queue uses the classic CSMA/CA mechanism to induce access rights to the medium. Still, the CSMA/CA parameters (back-off, waiting time, size of the contention window) vary from queue to queue to frames with a high priority level that can be selected. With CSMA/CA, a candidate broadcaster recognizes the channel earlier before it transmits. By Counting on the channel status, free or busy, the transmission begins or differs. It means that a node that wants to send sense the medium first, and when it is free for an AIFS, the node postpones the transmission by choosing a random fallback time(Busson, 2014). Figure 2-2 explains its working principle. The backoff procedure in 802.11 works as follows(Miao et al., 2011) :

I. The node chooses a uniformly random back-of time from the interval [0, CW +1], where the initial value of CW is CW_{lower};
II. The slot size is increased (doubled) if the next transmission attempt fails until the value of CW equals to CW_{upper};
III. The back-off value is reduced only, If the channel is free;
IV. If it reaches a back-off value of 0, It sends immediately.

For highly relevant safety messages to be exchanged quickly and reliably, even in a dense environment, the IEEE802.11p MAC protocol considers the priority of messages with different access classes, as shown above. If a node intends to accumulate the channel with a better probability independent from others, the best resolution is to lower its waiting time, called back-off misbehavior. The goal of a back-off misbehaving node is to gain unfair access to the channel by manipulating its back-off time to the detriment of degrading the performance of well-behaved nodes, whether malicious or selfish(Lu et al., 2010). Section 2.6 describes the selfish behavior of vehicles that occurs in the MAC layer.

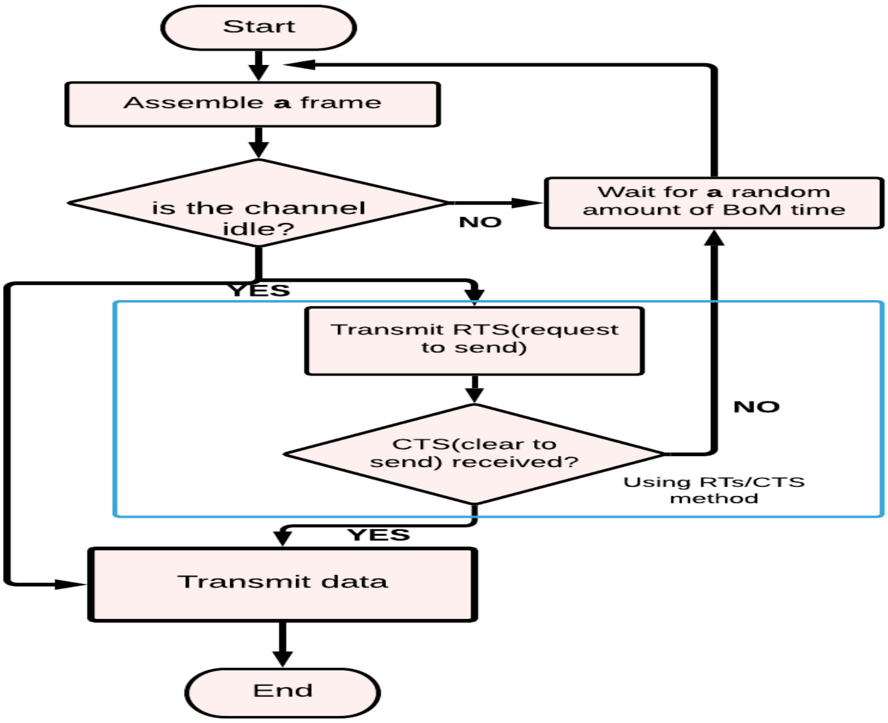


Figure 2-2: Working principle of CSMA/CA[4]

2.5 Mobility Models for VANET

Several mobility models were introduced for simulating VANETs. The typically used mobility models are:

1. Freeway Mobility Model (FMM)

Freeway is a model based on generated maps. The simulation area, represented by a generated map, includes many highways, each side consisting of several lanes. The nodes are placed randomly on the tracks when the simulation begins. Then It moved at speeds based on the story. Lane change is not allowed on this model(Filali & Bonnet, 2006).

[4] https://th.bing.com/th/id/OIP.6TkyvZ1GsPT4Url60_ChFwHaIG?pid=ImgDet&rs

2. **Manhattan Mobility Model (MMM)**

It is also a map-generated model that simulates an urban road environment. A map is generated before a simulation begins, including vertical and horizontal roads. Each has two lanes, allowing movement in both directions (north/south for steep roads and east/west for flat roads). At the start of a simulation, vehicles are placed all over the streets. Then, they move infinitely at the rate of speeds based on history (following a formula similar to that of the passing model). Unlike the previous model, a vehicle will switch lanes at an intersection(Madi & Al-Qamzi, 2013).

3. **Random-Way point Mobility Model (RWPM)**

The node placed withinside the Random Waypoint version is evenly dispersed between nine and sixteen m/s, protecting the speed limits of the different avenue classes in the simulation scenarios. Furthermore, the pause time between subsequent trips is evenly distributed between zero and 10 seconds to simulate fast-forwarding to the destination(Potnis & Mahajan, 2006).

2.6 Behavior of Vehicles

The most typical routing protocols and communication standards suppose that all vehicles communicate cooperatively on the VANET. However, in practice, limited bandwidth and lack of computing resources make it tough to confirm that all vehicles in the network selflessly share their precious resources with others. In consequence, the assumption is not fully applied in the VANET environment. Non-cooperativeness in an ad hoc network is called nodes' unwilling behavior to participate in the network. Nodes may intentionally or not disrupt the regular operation of the network, wishing to save their resources or having limited bandwidth. Non-cooperative vehicles in VANET are classified into selfish and malicious vehicles(Uzma Khan & Abstract, 2015). SVs take the network to send/receive data packets. Due to limited computing power/bandwidth and communication resources, they deny forwarding data packets to others; in addition, they are not directly harming other vehicles or the entire network. However, vehicles have a considerable impact on the complete function of the network(Lu et al., 2010).

In contrast, malicious vehicles aim to damage and disrupt the regular operation of the network. These vehicles intentionally hurt other honest hosts, even the entire network, while conserving resources is not their primary goal(Shan et al., 2021). Therefore, malicious vehicles are not included in this thesis. Two types of selfishness can be defined, static and dynamic selfishness. An SV that continues the desire, unvaried during the whole network process, is categorized as a static SV(Komali et al., 2008).

Oppose to a dynamic SV fluctuates its selfishness according to circumstances. Therefore, this thesis assumes the SVs have dynamic selfish behavior and will behave back to normal after the motivation mechanism applies. However, in general, selfish misbehavior of vehicles leads to several impacts; as such, throughput degradation, causing network disconnection. In either case, poor vehicle behavior will significantly affect network performance, which still requires intensive communication among vehicles(Mantas et al., 2018).

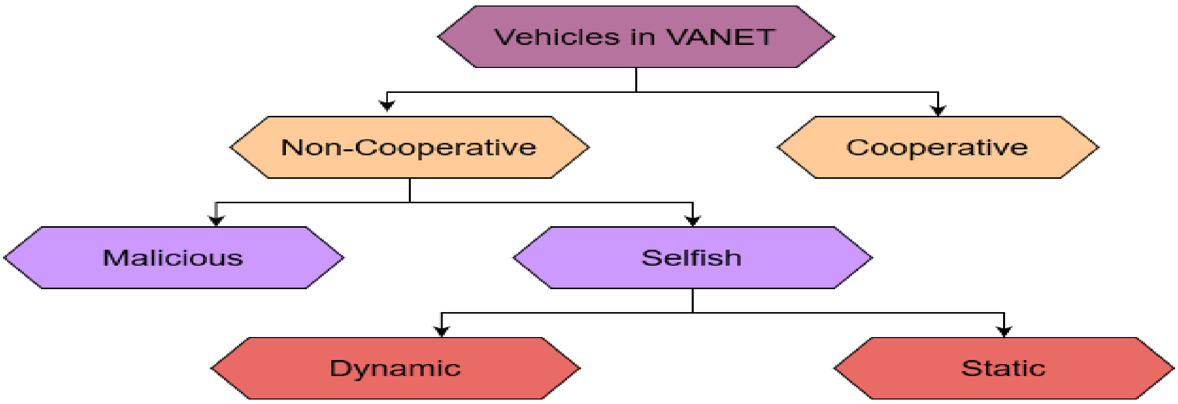


Figure 2-3:Vehicle behavior in VANET

2.6.1 MAC layer Selfish Behavior of Vehicles

VANET applications are often limited by bandwidth and computing resources. An SV may be unwilled to use its computing power and bandwidth to exchange information or increase its channel probability, which is not directly beneficial to it(Pandey, 2015).

An SV can pointedly abuse the MAC protocol to obtain more network resources than honest vehicles. For example, IEEE802.11p requires vehicles competing for the channel to wait for an idle time slot before each transmission. Vehicles can choose to wait for a smaller back-off interval, which increases their access to the channel, reducing the percentage of bandwidth received by well-functioning stations. Some typical strategies adopted by SVs are highlighted below(Al-Terri et al., 2017):

a) CW cheating strategy: SV misuses the back-off value by selecting a lower value in a range of [0, αCW] instead of [0, CW] and obtain a smaller back-off value, and gained a small interval than of normal vehicles.

b) Fixing CW strategy: The SV chooses a fixed size smaller than CWlower and refuses to double its CW when the transmission is unsuccessful. So, it will get a high probability for the subsequent trial.
c) Back-off cheating strategy: An SV follow the rule of CSMA/CA, but it manipulates the attained back-off value to a lower value.

2.7 Schemes for Solving Nodes' Selfishness Misbehavior

For resolving the destructive impact of the existence of selfish nodes, it is essential to encourage nodes to work together to ensure the regular operation of the network. Information is spread in VANET through the cooperative interaction of vehicles. The messages transmitted in the vehicular network include essential information such as traffic jams, urgent braking, road conditions, accident reports, severe weather, and weather conditions.

The misbehavior manipulation of the messages may be very dangerous(Uzma Khan & Abstract, 2015). Hence, uncooperative node misbehavior is a critical challenge for the VANET network. Several works have been used to identify and prevent the selfish behavior of vehicles in VANET. According to ad-hoc-network, commonly used schemes for identifying selfish nodes are generally categorized into two: node-oriented and data-oriented identification schemes. In such a scheme, the node-oriented pays special attention to the node sending the message, not the sent data. It depends on how the node behaves and its reliability in relaying messages. The node-oriented misbehavior identification scheme is divided into two methods, behavioral-based and trust-based. As opposed to a node-oriented, a data-oriented scheme is related to the link between messages than the behavior of individual nodes. The information disseminated by nodes in the network is analyzed and measured with information received from other nodes to check the destination node's messages(Uzma Khan & Abstract, 2015).

Different incentive mechanisms are proposed to encourage nodes to cooperate and reduce the unwilling behavior in the network. The following table describes mechanisms that are categorized under node-oriented schemes. These are credit-based, reputation-based, and barter-based(van der Heijden et al., 2013). Encouraging SVs to cooperate is a key motive to ensure the network's performance. For this case, in this thesis, we attempt to design a reputation-based incentive solution. The solution identifies the selfish behavior using watchdog approaches. It motivates the identified SVs to get a high reputation score when their credit is incremented based on their well-behaving behavior. The proposed solution only focuses on the behavior of vehicles while channel accessing.

Table 2-4: Node-Oriented schemes

Schemes	***Working Principle***	***Advantage***	***Challenge***
Credit-Based (Nuglet)(Uzma Khan & Abstract,	Reward mechanisms implement to the nodes to indicate cooperativeness. If the node acts cooperatively, it receives a specific credit.	Giving credit motivates the SN to cooperate better.	Message-over flow and the overhead ratio are high.
Reputation-Based (Watchdog)(van der Heijden et al.,	The more the node behaves cooperative, the higher reputation points it receives.	The behavior of nodes is more assessed and managed.	The scheme is a bit traditional
Barter-Based (Game theory approach)(Al-Terri et al., 2017)	The Tit-for-tat (TFT) strategy applies to discipline the non-cooperative node's node takes cooperative or selfish actions based on the actions of the previous node.	Social ties make more collaboration.	Lack of faithfulness makes the scheme not work well

2.8 Related Work

In this section, the research areas related to the selfishness behavior of CSMA/CA of IEEE802.11p was discussed and assessed. Also, the relevant gap and improvements were distinguished based on this thesis work. Presumably, many researchers who studied the impact of the selfish misbehavior in MAC protocol, specifically in CSMA/CA of VANET, mainly focus on investigating the impact of selfish behavior and detecting the SVs, which misuses the channel accessing mechanism. Most of the papers did not include any motivation or punishment mechanism for making the SV cooperate in the network again.

(Cheng et al., 2018) presented a time series analysis approach for detecting MAC layer malfunctions in IEEE802.11p, emphasizing selfish behavior on the sender and receiver sides generated while the nodes are accessing the channel. Three types of attacks have been presented in this document: back-off value manipulation, carrier sense manipulation, and RTS drop attacks. Two methods were used: the detection method and the diagnostic method. The detection process is active for screening the selfish behavior, while the diagnostic process is ready after the suspected selfish behavior. Even though the proposed method detected selfish behavior in the network flow and differentiated its effects on the overall network performance based on parameters such as throughput, delay, and packet interval, no countermeasures were taken to motivate or punish those detected selfish nodes.

In the research work, (Al-Terri et al., 2017) have endeavored to correct selfish behavior in the IEEE 802.11p CSMA/CA protocol in VANET. By collaborating tit-for-tat strategies with a group reputation-based and collaborative detection strategies, motivational mechanisms had been proposed to encourage selfish nodes to behave normally under threat of retaliation. Furthermore, the mechanisms were proposed to encourage nodes, which behave selfishly, to perform regularly under threat of retaliation. Five tit-for-tat-based methods aimed at dealing with greedy node problems were presented, particularly (1) classic tit-for-tat, (2) generous tit-for-tat, (3) reputation-based tit-for-tat, (4) group reputation-based tit-for-tat, and (5) cooperative detection based tit-for-tat. All strategy has improved the selfish behavior and implemented cooperation on the MAC layer in VANET. Although this paper improved the network throughput, some other standard parameters were not considered, such as end-to-end delay and latency. In addition, the process taken by the nodes to maximize their payoff and get some reputation will cause a high-end to end delay and high latency; because of these reasons, the low delivery ratio was expected.

(Thambi & Sakthivel, 2015) have developed a Markov-RET-FT technique to fix a node's misbehavior in CSMA/CA on a wireless network. The proposed model monitored selfish behavior and penalized these selfish nodes using the flow trust value calculation. This calculation was used to ensure the legitimate flow. Without prior knowledge of statistics on selfish behavior, the proposed solution was discovered in the misconduct node. However, the proposed solution detects and penalizes selfish nodes. It didn’t include a reliable motivation scheme for the selfish nodes. In this case, there are still more nodes encouraged to misbehave.

In GDVAN (Mejri & Ben-Othman, 2017) (Greedy Detection for VANETs), Mejri et al. proposed a greedy algorithm to discover malicious behavior of VANETs, IEEE802.11p. In GDVAN, three lately evolved metrics used to pick out nodes' misbehavior in high mobility environments in VANET, malicious nodes deliberately adaptively regulate the back-off parameters because of the short-range communication and confined bandwidth. The proposed solution is based totally on linear regression and fuzzy logic, which includes two phases, suspicion and decision. The rules can look for greedy nodes by retaining track of network traffic. GDVAN is passive, did not require loads of resources, and did not require any changes to the MAC layer. Despite GDVAN, using the suspicion and decision phase, it detected all existing greedy vehicles in a very high probability. No other motivational or punitive mechanisms were used to encourage or mitigate the greedy vehicles to re-collaborate in the network.

Ben-Othman et al. (Mejri & Ben-Othman, 2014) proposed a detection algorithm to detect greedy behavior in IEEE802.11p from VANETs. The detection technique requires no modification of the initial IEEE 802.11p and can be performed from any node within the network, detected CSMA /CA backoff manipulation at EDCA. The authors proposed an algorithm composed of a watchdog monitoring tool and a linear regression method based on the network traffic. The proposed algorithm evaluates the network to determine whether there is greedy behavior or not. No alternative incentive mechanism was used to motivate or punish the greedy vehicles discovered, and these greedy nodes were not mentioned in the network traffic.

The authors (Sadek et al., 2021) proposed a mechanism for detecting greedy nodes on the IoT of the IEEE802.15.4 standard. Especially greedy behavior in CSMA/CA The proposed method involves the assessment of eight parameters that are considered essential for this type of attack. A node is considered greedy only if its eight parameters are outside the specified threshold. Three main steps: the first is gathering network traffic and information and evaluating the gathered information; Finally, the third stage is the detection stage, which detects the accused nodes. If they are greedy or not, which is based on the threshold value mechanism. However, the method proposed by the author has shown that the efficiency of greedy nods was demonstrated. All other motivating or punitive mechanisms were not considered in this work. Moreover, the techniques for identifying these greedy nodes had taken a long time and caused a high probability of end-to-end delay.

Table 2-5 Related work summary

NO	***Authors and Year***	***Problem Addressed***	***Objective***	***Method used***	***Gap***
1	Ben-Othman et al. (2017)	Greedy Behavior Attacks in VANET	To suspect and identify greedy behavior on IEEE802.11p.	Linear Reaggregation and Watchdog supervision tool	No other motivational or punitive mechanisms used
2	D.Al-Terri et al. (2017)	Node misbehavior in CSMA/CA, MAC protocol of VANET	To identify greedy behavior in CSMA/CA and motivate the greedy nodes to re-cooperate.	Tit-for-Tat strategies with a group reputation-based	High-end to end delay and high latency
3	M.X Chang et al. (2018)	MAC layer malfunctions in IEEE802.11 of WLAN	To detect MAC lay misbehavior	Time series analysis approach	No countermeasures were taken to motivate or punish those detected selfish nodes.
4	Sadek et al. (2021)	Greedy nodes misbehavior IEEE 802.15.4 of IoT network	To detect greedy nodes on IEEE 802.15.4	Analysis of network traffic information and threshold mechanism	There is no motivating or punishment mechanism High end-to-end delay

2.8.1 Related Work Summary

The CSMA/CA of IEEE802.11p has been proposed and overviewed in many papers. However, a few papers have studied the nodes' misbehavior under this standard. That is focused on studying the impact of selfish behavior and proposing a detecting mechanism for abnormal behavior. Degrading the overall network performance was the central issue of selfish behavior. The main task which was taken by those papers to resolve the above issue was to detect the SN which misused the channel accessing method. However, except for one paper, all the related works mentioned above did not propose any countermeasure mechanism which forces or allows the SN to be cooperative again in the network. In other words, no punishment or motivation mechanism was used to revive the detected SN. Therefore, we propose a solution based on the above-discussed gaps of the related works. It identifies the SV by monitoring the node's (vehicles') behavior and motivating the identified SV to revive in the network again by giving a high score to the nodes willing to balance their waiting time and connection trial and getting credit when necessary increased. Since this vehicle gets good credit, it will access the channel and transmit data.

CHAPTER THREE

3. PROPOSED SOLUTION

3.1 Overview of WC-CSMA/CA

The proposed solution, WC-CSMA/CA, has two main phases: identification and motivation. First, a group of vehicles forms a neighborhood within the RSU in the region(R). Since the transmission range is restricted, the scale of the neighborhood is also limited. There are roughly 50-100 vehicles in one R. The vehicles move at high speed along with street lanes in the same and opposite direction. The R neighborhood members speedily tie and leave the region at a known short interval. Therefore, the proposed solution works in region R with 25,50,100 vehicles, respectively, with one RSU by considering vehicle density. Since the distance between each RSU is one up to Three km. We assume the coverage area of the R is no longer than 1km. The vehicles under this region with the same WIBSS (Wave Independent Basic service set) can share and access the communication medium, using the CSMA/CA access method controlled by MAC protocol to guarantee all connected vehicles. So, we set each vehicle with the same WIBSS since the proposed solution is concerned with channel accessing.

Before identifying the selfish behavior, first, it is better to select vehicles that have a task to monitor the SV. The primer thing to do is to select a verifier vehicle. In this case, RSU or other vehicles are assumed as verifier vehicles (VV) as the Watchdog, which monitors the vehicle's abnormal behavior. The VV vehicles are chosen among the total vehicles based on their noteworthy value, distance, and load. The identification phase will be activated after the VV selection task is done. a VV With a specified threshold will identify each SV individually by monitoring the neighbor vehicles. For identifying the SV, three primary metrics will be used: the vehicle backoff value, average load, and average network throughput that the vehicle handled. The SV will be identified by calculating the above metrics that wrongly access the channel. Afterward, the last phase, the motivation phase, revives those SVs to behave in the usual way by incrementing their credit if they get a good score. For determining the vehicle's score, two countermeasure metrics are taken and calculated, which are average waiting time and total connection trial.

The proposed solution has the following three steps

a) Verifier Vehicle selection
b) Identification Phase
c) Motivation Phase

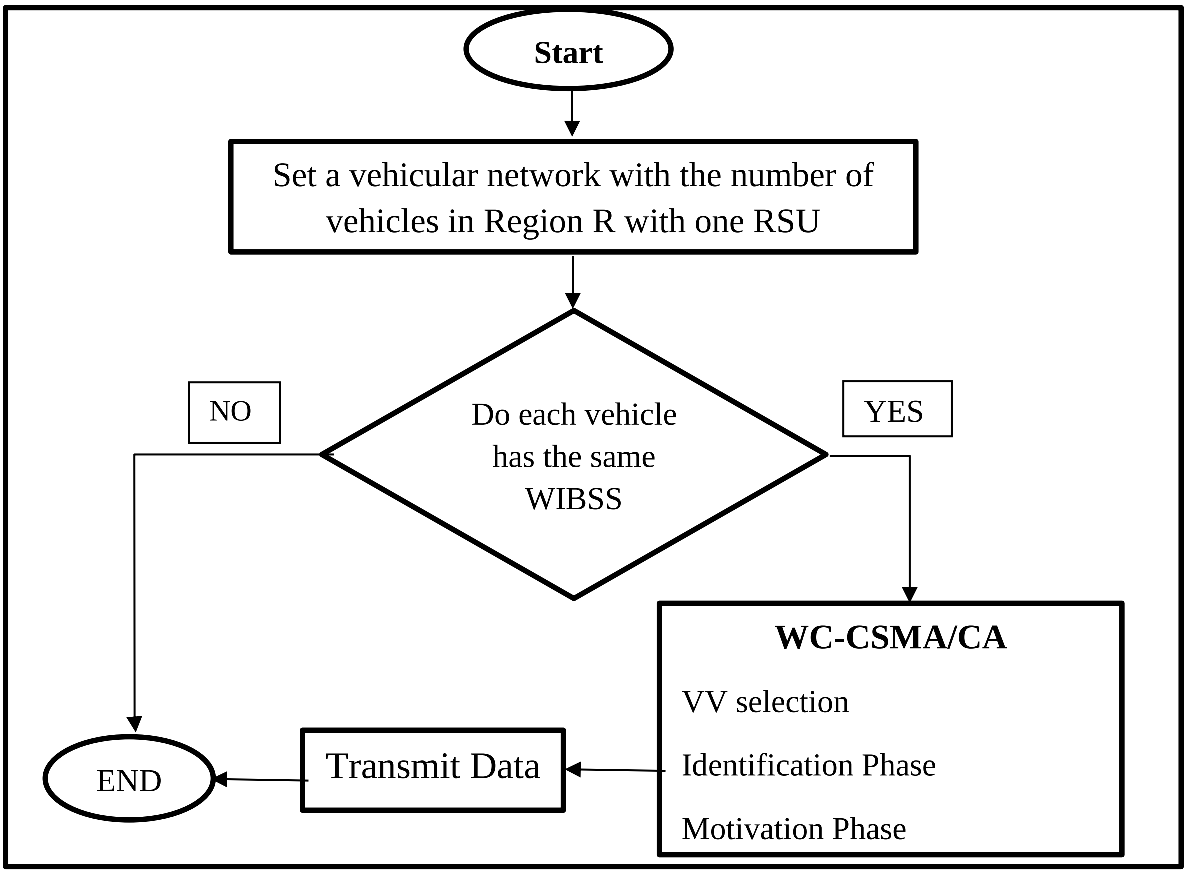


Figure 3-1: Overall flow of the proposed solution

Overall pseudocode for the proposed solution

Step 1: /* Set a vehicular network */

Step 2: Define region R with one RSU

Step 3: for all vehicle v ∊ V, do

Step 4: if vehicle v is a set of $V_{WS,}$ Then

/* where Vw is a set of vehicles that have the same WIBSS */

Step 5: WC-CSMA/CA applied to the range of transmission

Step 6: vehicle v accesses the channel in order and Transmit data

Step 7: END

Algorithm 1: Overall pseudocode for the proposed solution

3.2 General Research Assumption

These are the general assumption for the proposed WC-CSMA/CA

- The proposed solution works in Region R with 25,50,100 vehicles, with one RSU including the SVs.
- For demonstrating the problem, 3, 5, and 10 number of SVs are added in the network respectively for 25, 50, and 100 vehicles
- Since the distance between each RSU is located in 1 up to 3km, we assume the coverage area of the R is no longer than 1 km.
- The vehicles under this region with the same WIBSS can share and access the communication medium and transfer the packets.
- The vehicles are moving in bidirectional linear Highways in the same and opposite direction; therefore, the traffic motion is created in a street lane using the Addis-to-Adama expressway road map from OpenStreetMap.

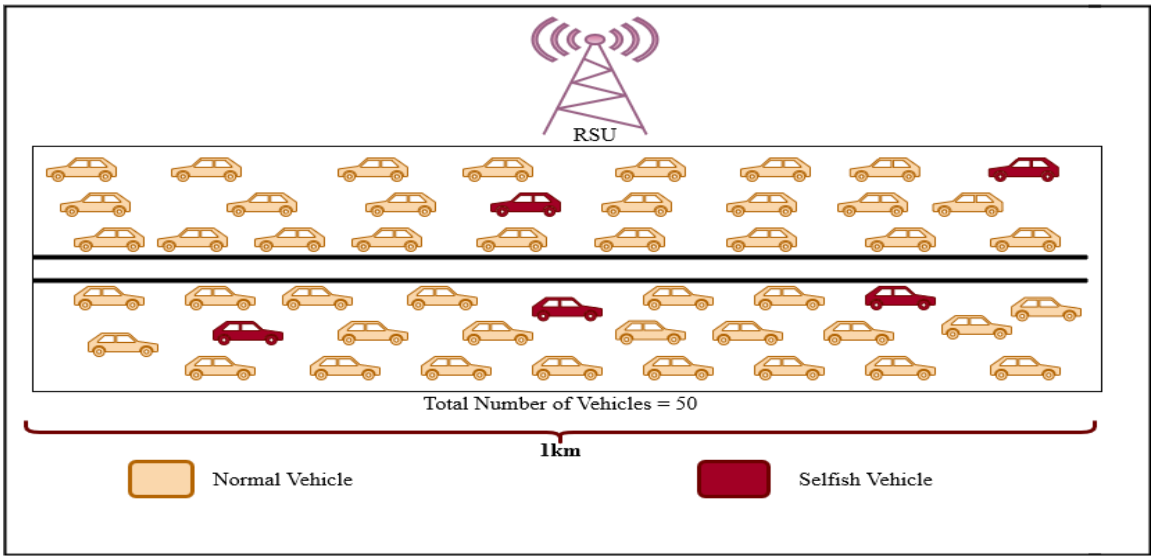


Figure 3-2: General network assumption for WC-CSMA/CA

3.3 Description of the proposed WC-CSMA/CA

3.3.1 Verifier Vehicle Selection

For the sake of suspecting whether there is a selfish behavior or not in the defined region R, Trustier vehicles that serve as a verifier are needed. The RSU selects these verifier vehicles as central entities since untrusted vehicles are assigned to the network. Those selected verifier vehicles are used to distinguish selfish behavior. Therefore, based on three basic metrics among the region R, specified verifier vehicles (VV) must be picked by the RSU first.

For choosing these verifier vehicles, three fundamental parameters are used. These are:

- Noteworthy value (NT)
- Load (Lo)
- Distance (Ds)

The vehicles are selected as a verifier (VV) when the value of the decision metrics (DM) is smaller than the Picking Threshold (Pt). Amongst neighboring vehicles located in region R, This mechanism cultivates the choosing means of VV and hence benefits to avoid wasting the network bandwidth and consequently booster network performance. Therefore, vehicles are well chosen for being verifiers in the vehicles area region(R).

The three parameters for choosing the VV are:

- Load (Lo): The number of nodes a vehicle V is already monitoring, so a vehicle, with the less loaded than other vehicles, gets a higher chance of being chosen as a verifier.
- Noteworthy value (NT): Refers to a vehicle's degree of trustworthiness. If a vehicle V scores a lower distrust value, it's considered the reliable node. If it shows abnormal behavior, this value gets higher, and appropriate decisions are made compared with the threshold.
- Distance (Ds): If the distance from a node to the vehicle V to be monitored is smaller, the node stays longer in the transmission range of the vehicle, which offers a margin for better observations and decisions.
- Decision Metrics (DM): It is calculated for all nodes considered for the selection of VV into account the value of the Load, the Distance, and the Noteworthy of the node by the following Equation (1).

$$\mathbf{DM} = \alpha * Lo + \beta * NT + \theta * Ds \tag{2.1}$$

Where α, β, and θ are the weighting factor of the Load, Noteworthy, and Distance respectively, as such $\alpha + \beta + \theta = 1$.

The decision metric DM calculated for the nodes under consideration is compared with the picking threshold Pt for selecting a VV. If the value of the DM of a node is less than the picking threshold (DM < Pt), then the vehicle is assigned as the verifier.

Pseudocode for selecting a verifier vehicle (VV)

Step 1: vehicle V joins the Network

Step 2: Get the keys

Step 3: Retrieve Lo, Ds of the vehicle V

Step 4: Compute Nt of vehicle V in the region R for selecting verifier by RSU as a central Entity

Step 4: Compute the DM for selecting verifier

$$DM = \alpha * Lo + \beta * NT + \theta * Ds$$

Where $\alpha + \beta + \theta = 1$ where the α, β, and θ are the weight factor of Lo, NT, and Ds.

The Value of the three-waiting factor is 0.33

Step 5: If (DM <Pt) then,

Step 6: Allocate vehicle V as a verifier to the recently joined vehicle

Step 7: /* Verifier vehicle VV start monitoring */

Step 8: Else

Step 9: Vehicle V has abnormal behavior and stays in the network to be monitored

Step 10: END

Algorithm 2: Pseudocode for verifier vehicle selection

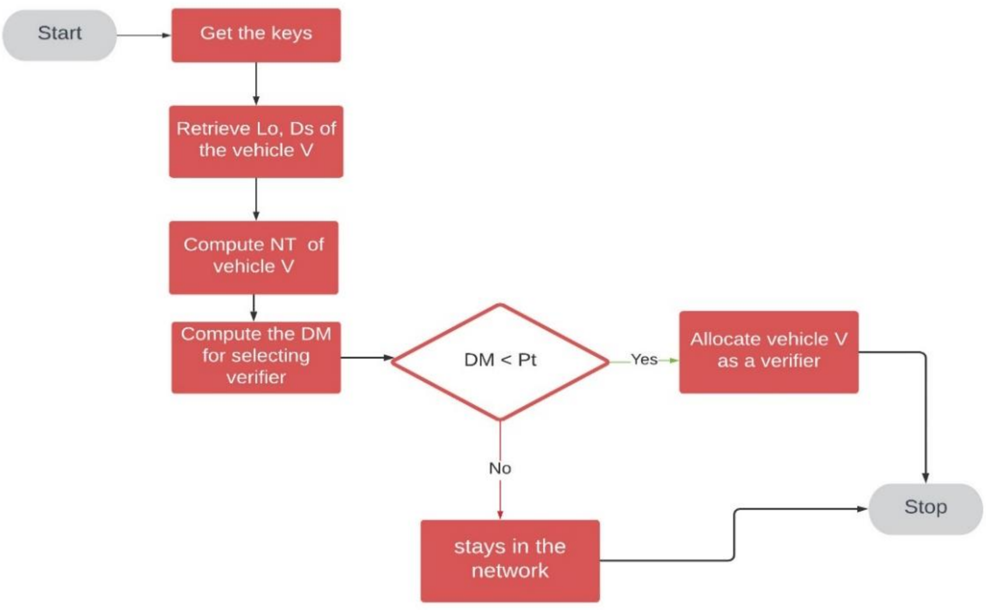


Figure 3-3:Flow chart for verifier selection

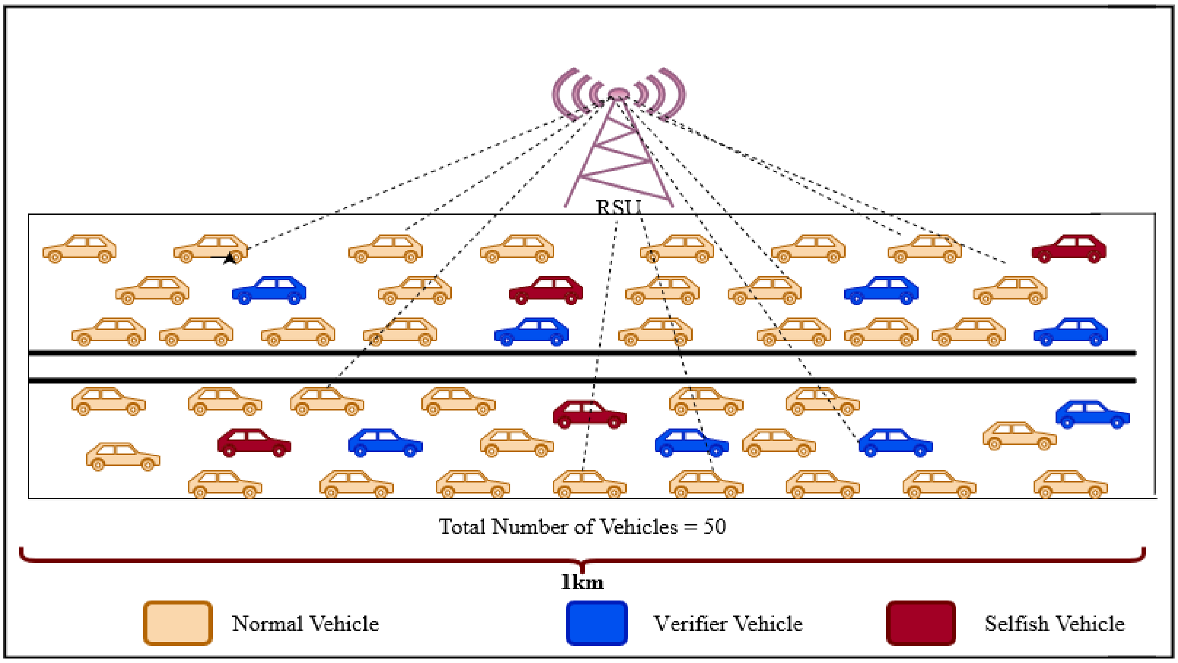


Figure 3-4: Network scenario with verifier vehicles

3.3.2 Identification Phase

After selecting the verifier vehicles, the next step will be identifying the SV that abuses the channel and degrade the network performance. The identification phase continues to identify which vehicle or vehicles of the network are SV. Finally, each VV must monitor its neighbor's behavior and report to share the medium and the load being handled. In this case, three main attributes are observed and calculated, the total back-off time, average network throughput, and the average load of the vehicle.

Major factors:

1. Back-off value (backoff v): To get a high probability of accessing the channel, the SV slightly lowers their backoff value and time, leading to lower end-to-end delay for packet transmission.
2. A load of vehicle (Lv), measured in byte/s: To have a high probability of channel reservation, the SV as a source or forwarder in multi-hop networks can handle more load than the regular vehicle within the network.
3. Network throughput(T), measured in Kbit/sec: The SV can increase its payout (throughput) by changing the CW contention window to a smaller value. The SV throughput increases monotonically as the CW value decreases.

As the vehicle becomes more selfish, its back-off time decreases, its load increases, and its payoff or throughput increases since the vehicle changes its CW value to lower. These two properties, load, and throughput, negatively react with selfishness most of the time and are inversely proportional to the backoff value. For combining the effects of these inversely proportional properties, it is necessary to calculate the interchangeability of one of them with the other. Therefore IM (Identification Metrics) is computed according to the following equation. The values of each weighting factor have to be selected based on the essentials of each attribute. In other words, the attributes that vary from the expected behavior should be given a more considerable value to exaggerate its contribution in IM and help distinguish the selfish vehicle.

$$\mathbf{IM} = \delta * \frac{backoff\ v}{\max \forall i \in V, backoff\ i} + \gamma * \frac{1}{\frac{L\ v}{\max \forall i \in V, L\ i}} + \tau * \frac{1}{\frac{T\ v}{\max \forall i \in V, T\ i}} \tag{2.2}$$

Where backoff v: the backoff value of vehicle v

Lv: the average load of vehicle v in byte/s

V: the total number of vehicles in the network which has verifier vehicles that watch and monitor their neighbors and other vehicles which are near to them

Tv: the average network throughput of a vehicle in byte/s

δ, γ, and τ: are the waiting factors, where $\delta + \gamma + \tau = 1$

The range of backoff slots is the primary aspect that captures the primary difference among the node's behavior because, majorly, SV is regarding their backoff value. Thus, deciding on δ could have a better rank than γ and τ, after computing the IM of each vehicle for the suspected network. The selfish behavior is considered by comparing its value with a specific threshold by the verifier vehicles that watch their neighbor’s behavior. The VV will announce to the other neighbor vehicles and notify them to be ready and adjust their behavior.

Pseudocode for selfish vehicle Identification

Step 1: Define Verfier_Count = 0

Step 2: / * The VV monitors the vehicles for each Tm period */

Step 3: For ∀v ∊ V do /* for all vehicles including the VV, do the following task*/

Step 4: Retrieve backoff v, Lv, and Tv

Step 5: Start to compute Total backoff v, Total Lv, and Total Tv

Step 6: end for

Step 7: For ∀v ∊ V do /* for all vehicles including the VV, do the following task*/

Step 8: Retrieve backoff v, Lv, and Tv

Step 9: Compute IM

Step 11: If IM < λ then /* where λ a threshold, which λ < 1 */

/* Start counting how many consecutive Tm periods the network is considered selfish */

Step 12: Verfier_count = 1

Step 13: Vehicle v is the selfish vehicle

Step 14: End If

Step 15: else

Step 16: Vehicle v is the normal vehicle

Step 17: /* The VV announce to its neighborhood about the behavior status of the vehicles *

Step 18: end For

Algorithm 3: Pseudocode for the identification phase

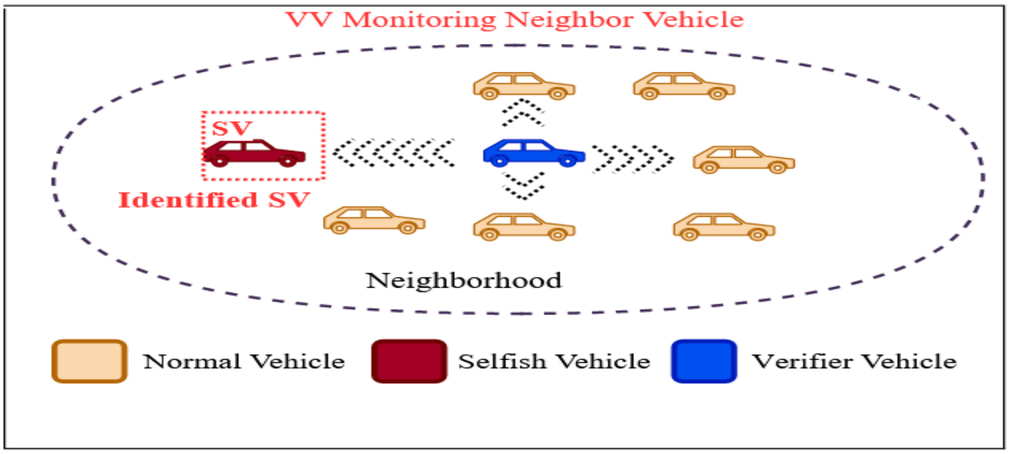


Figure 3-5: SV identification phase

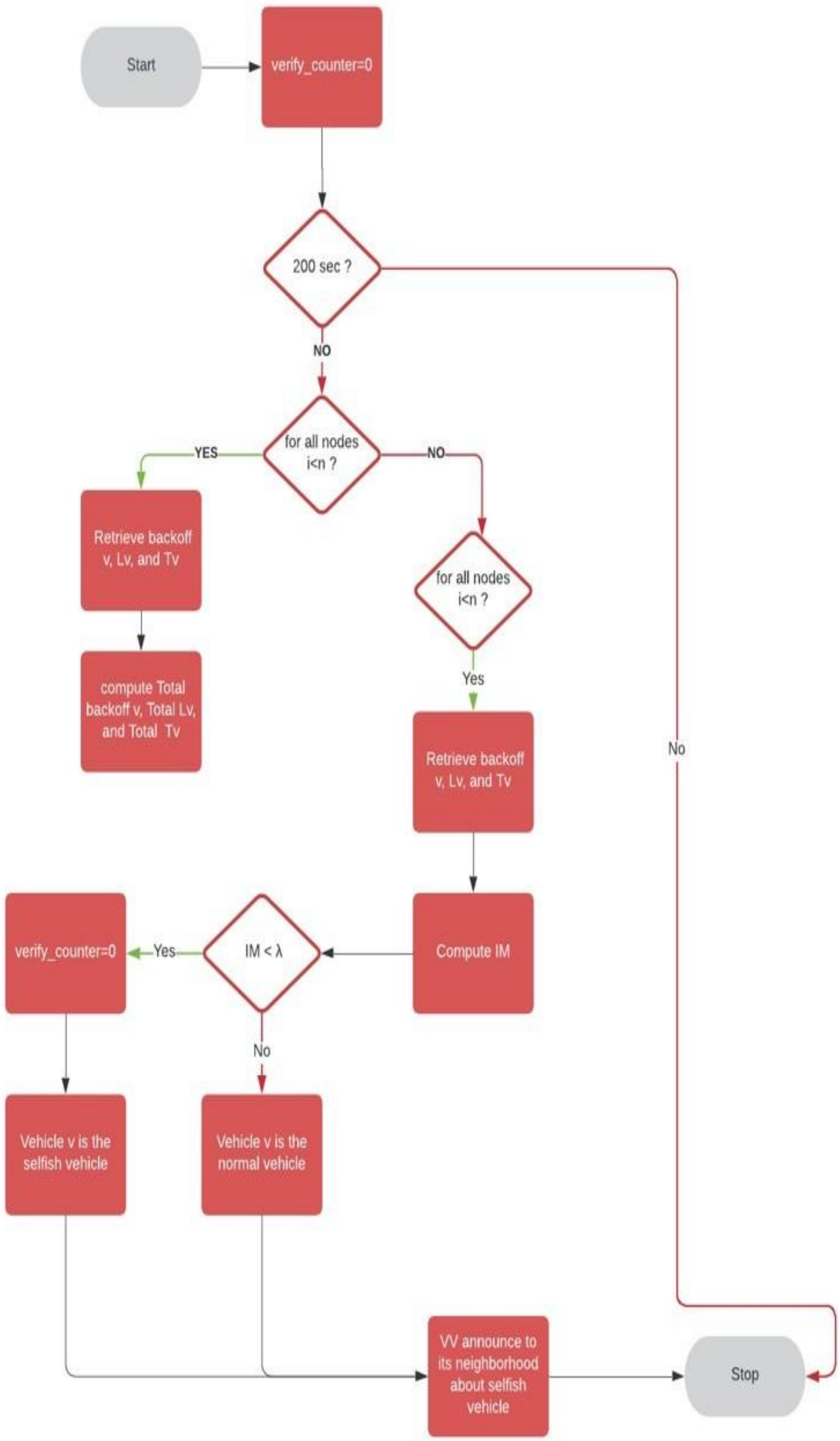


Figure 3-6:Identification phase flowchart diagram

3.3.3 Motivation Phase

The VV announces to all neighborhood vehicles, including the RSU, about the status of its neighboring vehicles. It attached the status of the vehicle with the ID of the vehicles. Therefore, each vehicle stores those items, and if the vehicle state is selfish, no data will be sent to these vehicles. In other words, the vehicles cannot request for accessing the channel. Those vehicles should behave well and act like regular vehicles for accessing the channel. Therefore, by comparing their score, those SV get credit. The SV with the higher credit will get a chance to access the medium. The credit of a vehicle is incremented once a vehicle respects the back-off mechanism without manipulating the process. Thus, the SV will seek to induce pay to extend its credit at the cost of the average vehicle by balancing the following terms.

- **By averaging its waiting time:** To get a higher priority to access the channel, the SV decreases its waiting time (Wt) to almost close to zero. For this reason, to get better credit and to get a chance to access, the SV should have to average its' Waiting time. If its' average waiting time is more significant than a given threshold, its credit will be incremented.
- **By averaging its number of connection attempts:** The SV tries much more than other vehicles to connect the network, which means it establishes a higher number of connections to occupy the channel. Upon that, to get credit and back to normal, the SV should balance its connections like the regular vehicles.

If SV balances the above terms, it will get a high score. Also, its credit will be incremented, and this behavior will be considered as SV is motivated and obeyed to cooperate in the transmission again. In consideration of that, SV will get permission to access the channel.

As mentioned earlier, the status of the neighborhood vehicles announced by the VVs has four items, and it is organized in a tabular form. Showed in the following figure

Vehicle ID	Vehicle State	Vehicle Score	Vehicle Credit

Figure 3-7: Neighbor vehicle status

Vehicle-ID: The identification number of the neighbor vehicle will store in this entity

Examples: 001, 002.

Vehicle-State: It shows the current behavior state of the neighbor vehicle. Two identifiers are used, “S” and “N.” “S” stands for SV, and “N” is a regular vehicle.

Vehicle-Score: The targeted vehicle that the SV tried to connect will calculate a score based on the behavior state of the neighborhood vehicle. If the vehicle state is “N,” it will give a score of “1”. But for the state that indicates “S,” the score will be calculated based on the above terms, and the value will be stored.

Vehicle-Credit: Vehicles’ credit was initially given as a “1” for the regular vehicle and a “0” to the SV by considering the vehicle's score. But each time, the SV balances Its’ Wt and numbers of an attempt to access the channel and get a good score. Then, its’ credit value will be increased, and it will be assumed that the SV is motivated and obey to cooperate.

The VV calculate the score using the following equation

$$\textbf{Score} = (S_1 * Wt) + (S_2 * Ct) \tag{3}$$

$$\textbf{Wt} = \frac{Wt1 + Wt2 + \ldots\ldots + Wtn}{Cost} \tag{4}$$

$$\textbf{CT} = Ct_1 + Ct_2 + \ldots\ldots\ldots + Ct_n \tag{5}$$

Where:

Wt: The Average Waiting Time of the SV, which waits to access the channel, measured in sec.

CT: The Total Connection Trial that the SV node tries to connect.

S1 and S2: The waiting factor where the sum equals 1. $S_1 + S_2 = 1$

Each VV calculates the score of neighboring vehicles, sends the table value shown in Figure 3.7 to all neighborhood vehicles, updates the table as it receives the score value, and gives a credit based on the score value. Thus, according to their updated table, the regular neighboring vehicles will connect to a vehicle with the highest score and good credit. Therefore, for the SV to connect and access the channel, it should have the highest score and good credit.

Pseudocode for motivating the Selfish vehicle

Step 1: /* Define the table, with ID, State, score, and credit */

Step 2: / * the VV announces the SV to all neighborhood vehicles and the RSU * /

Step 3: Define Score = 0, Credit = 0

Step 4: If the state of the vehicle v is "N" **then**

Step 5: Score = Score + 1

Step 6: Credit = credit + 1

Step 7: Else the state of the vehicle v is "S" **then**

Step 8: Compute Wt, Ct

Step 9: Compute Score

$$Score = (S_1 * Wt) + (S_2 * Ct)$$

Where S1 and S2 are the waiting factors, where s1 +s2= 1

Step 10: If Score >= β **then**

/* Where β is a given threshold */

Step 11: Score = 1

Step 12: Credit ++

Step 13: /* the vehicle v get a score value and its credit incremented */

Step 14: /* the vehicle v status table is updated */

Step 15: /* vehicle v is motivated, and can make a connection */

Step 16: Else

/* vehicle v remains as an SV */

/* vehicle v will not allow to make a connection */

Step 17: End If

Algorithm 4: Pseudocode for motivation phase

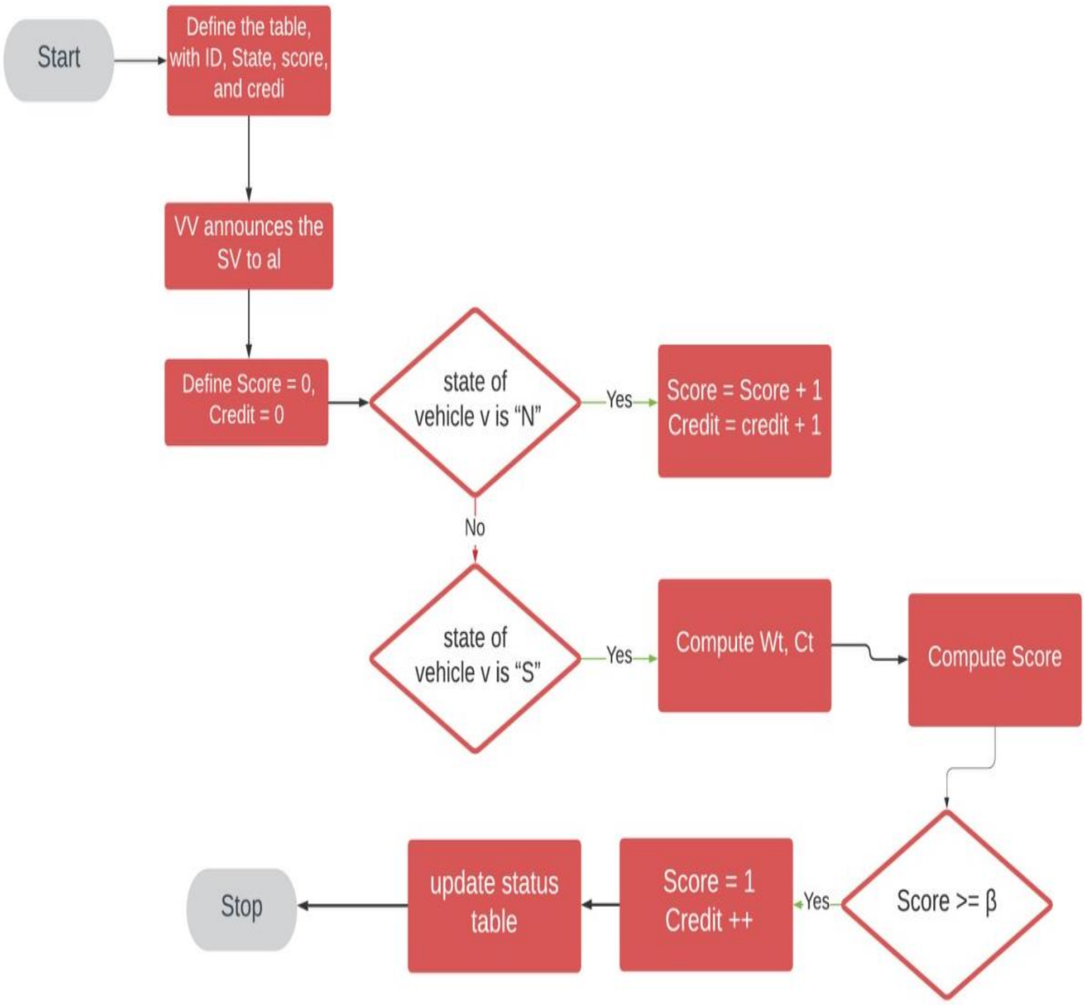


Figure 3-8: Flow chart for the motivation phase

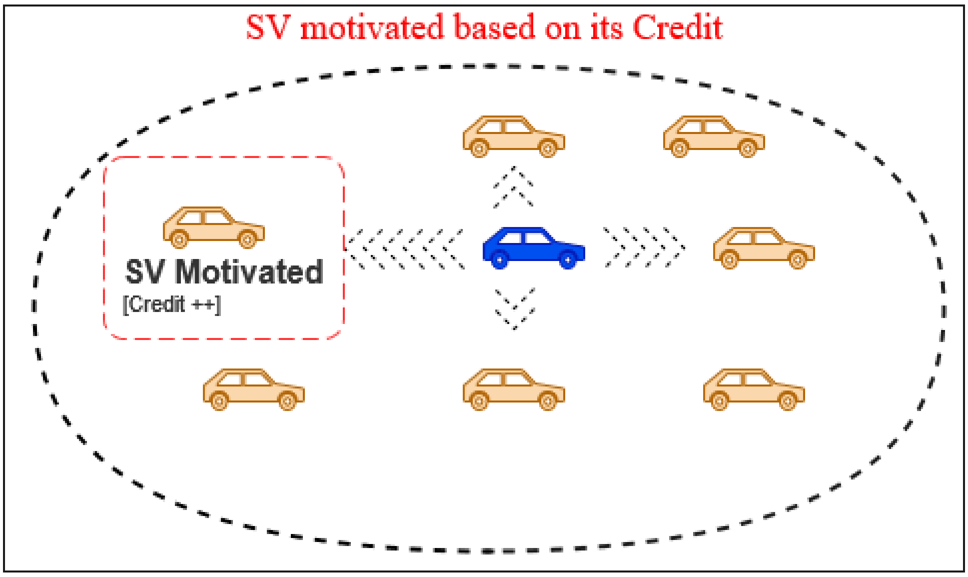


Figure 3-9: SV motivation phase

3.4 Summary

The proposed WC-CSMA/CA motivates the vehicles after Identifying them as SV. The proposed solutions manage backoff manipulation of selfish behavior in CSMA/CA of VANET. Identification and motivation phases are used for detecting and reviving the SVs. In the identification Phase, some vehicles selected as verifiers monitor the vehicles and identify SV using watchdog intensive approaches. After that, they will announce to their neighboring vehicles the status of their monitoring vehicles. Later, the motivation phase proceeds. This phase is designed based on the credit-based approach. The SV that wants to make a connection should get a high credit. The credit is managed based on the SV Score. It is determined by the average waiting time and average connection trial. The SV is considered as a well-behaving vehicle since it increments its credit like other well behave vehicle and become cooperative again

CHAPTER FOUR

4. SIMULATION AND PERFORMANCE EVALUATION

4.1 Overview

As the chapter title indicated, different VANET simulation tools, including mobility generators and network simulators, were described, discussed, and evaluated in this chapter. Concerning implementing the proposed WC-CSMA/CA, suitable tools to simulate the proposed solution were selected. Also, the overall simulation process was addressed. Last but not least, the proposed solution's performance based on CSMA/CA of IEEE802.11p was described and discussed. This section compares the proposed WC-CSMA/CA with the existing CSMA/CA based on the simulation result and basic evaluation metrics. For performance evaluation, average throughput, average waiting time, and average queuing delay were used.

4.2 VANET Simulation Tools

Deploying and testing VANET takes a massive cost and high labor. As an alternative solution, simulation is essential and less expensive, and also it can serve as a prior trial before the actual implementation. Two basic simulators are needed to simulate VANET effectively; mobility simulator and network simulator. An essential aspect of a VANET simulation study is the requirement of a mobility model, which carefully reflects the actual behavior of traffic vehicles. Mobility simulators are used in particular to generate the movement pattern of vehicles in a specific lane. In addition, a network simulator is operated to simulate messages exchanging between the linked nodes. Supposing of a VANET, it typically consists of vehicles and RSUs and primarily involves wireless communication. Ideally, all components of the communication system should be modeled. The network model describes each of the network components and events. Nodes, routers, switches, and links are examples of components. In turn, events can include data transfers and packet errors. Network simulators are liable for designing communication protocols and exchanging messages, whereas mobility simulators monitor the activity of each vehicle(Kumar & Shukla, 2019). This section describes the main mobility generators, network simulators, and VANET simulators and frameworks, which are mainly used.

4.2.1 Mobility Generators

Various traffic simulator mobility generators are available to generate the mobility of vehicles in real-time. For generating mobility, many factors are required, such as vehicle density, road topology, traffic signs, number of lanes, vehicle speed, and road obstacles. However, the performance varies from generator to generator. So, it is not compatible with all network simulators. Specific mobility simulators are listed below, which fits VANET mobility(Martinez et al., 2011).

1. **Simulation for Urban Mobility (SUMO)**

SUMO is a highly portable, open-source, microscopic software for continuous road traffic simulation developed for large road networks. SUMO is used to model intermodal transport structures consisting of road vehicles, public transport, and pedestrians. Pedestrians and public transport are modeled explicitly. Traffic lightmaps can be imported from SUMO or generated regularly. Supported import formats: OpenStreetMap, and VISUM. There are no artificial barriers in the size of the network, and the number of simulated vehicles and SUMOs are implemented in C ++, only with transportable libraries. Each vehicle is explicitly modeled microscopically, has its path, and moves individually through the network.

2. **Mobility model generator for Vehicular Environment (MOVE)**

MOVE is a tool for quickly generating practical mobility models for user VANET simulations. It is based on a SUMO traffic microstimulator, and it generates a mobility tracking file as output. It contains actual data. Hence the movements of the vehicles can be integrated and used by most of the simulation tools called NS2. In addition, MOVE afford a set of graphical user interfaces without writing simulation scripts and learning the internal details of the simulator. Allow the user to create realistic simulation scenarios rapidly.

3. **VanetMobiSim**

VanetMobiSim is an extended form of the CANU mobility simulation environment (CanuMobiSim) software, a flexible framework for modeling user mobility in VANET. The VanetMobiSim extension mainly focuses on vehicle mobility and offers new realistic models of vehicle movement at both macroscopic and microscopic levels. It also implements new mobility models and offers realistic V-to-V and V-to-I interaction. Nearby vehicles overtake and, if there are intersections, they behave according to the traffic lights.

Table 4-1: Comparison of VANET mobility generators

Mobility Generators	Map Support	Open Source	Ease of Use	Compatible Network Simulator
SUMO	Real and User-defined	YES	Hard	NS2, NS3, and OMNET++
MOVE	Real and User-defined	YES	Moderate	NS2
VanetMobiSim	Real and User-defined	YES	Moderate	

4.2.2 Network Simulators

Network simulators are used to inspect the performance of different VANET routing protocols in different scenarios. In addition, network simulators can be simulated in a short time. Therefore, researchers can test different scenarios inexpensively. These simulators, which are described in the following format, are compatible with the new IEEE 802.11p standard for data exchange(Martinez et al., 2011)(Khairnar & Pradhan, 2010).

1. **NS2**

Network Simulator 2 (NS2) is a stand-alone, object-oriented open-source simulation tool on Linux platforms. NS2 is written in C ++ and Object-Oriented Tools Command Language (OCTL). It provides extensive support for simulation routing, multicast protocols, and IP protocols, including TCP, UDP, SRM, and RTP for wired and wireless networks such as MANET and VANET. In addition, it offers support for multiple protocols and can graphically represent network traffic as a helpful tool.

2. **NS3**

Network Simulator 3 (NS3) is a discrete event network simulator that's open-source software written in C ++ that doesn't assist NS2, is a brand-new simulator, and does now no longer assist NS2 API. NS3 affords models of ways packet data networks work and feature in distinctive scenarios. It provides users with a simulation engine to carry out simulation experiments. Create distinctive varieties of community connections which include point-to-point, wireless, and so forth among nodes.

3. OMNET++

Objective Modular Network Testbed in C++ (OMNET++) is a C++ component-based simulation framework and library. It is used for network simulation, especially for VANET. OMNET++ Provides domain-specific capabilities for sensor networks, ad hoc wireless networks, general performance modeling, internet protocols, and more. Also, It offers an Eclipse-based IDE and a graphical runtime environment. OMNET++ is under the academic public license.

Table 4-2: Comparison of VANET network simulator

Network Simulator	**Simulation Languages**	**Open Source**	**Ease of Use**	**802.11p Support**
NS2	C++ and OTCL	YES	YES	NS-2.33
NS3	C++ and Python	YES	YES	YES
OMNET++	C++	YES	YES	YES

4.2.3 VANET Simulators and Frameworks

The tools described below are frameworks that integrate the VANET mobility pattern with the network simulator. With the support of these tools and frameworks, it is not that hard to build an integrated platform to test the simulation of VANET in a particular place (Weber et al., 2021),(Kabir et al., 2014).

1. VEINS

VEINS (Vehicles in Network Simulation) is open-source software. It helps to make vehicle network simulations as realistic as possible without affecting the speed, offers a complete set of models, and is compatible with two recognized simulation software; OMNET++ network simulator for simulation and SUMO for the generation of traffic mobility. Allows online reconfiguration and redirection of vehicles in response to network packets; various metrics are used in the Veins Simulator to increase performance.

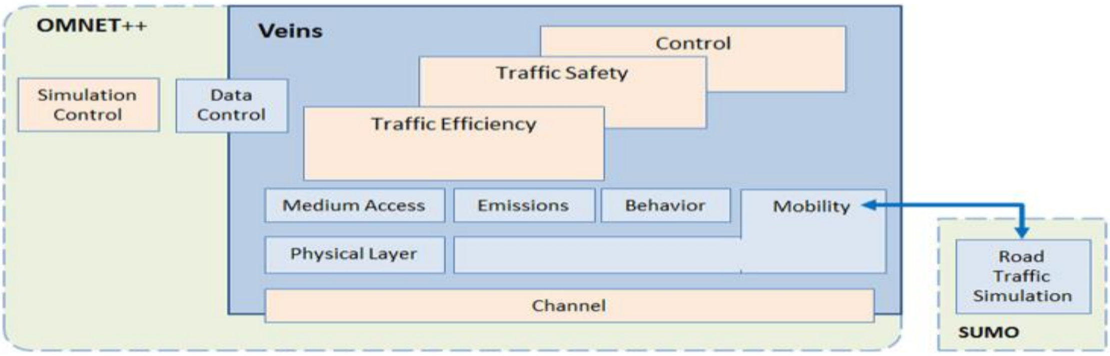


Figure 4-1: Architecture of the Veins platform

2. VENTOS

VENTOS (Vehicular Network Open Simulator) is an integrated C ++ simulator designed to study VANET traffic flow Especially, for V-to-I communication through DSRC wireless communication capabilities to reduce average delay. As well as, there is the fluctuation of the average speed in the entire network. VENTOS uses automatic traffic accident detection and MATLAB scripts to represent different aspects of simulation scenarios.

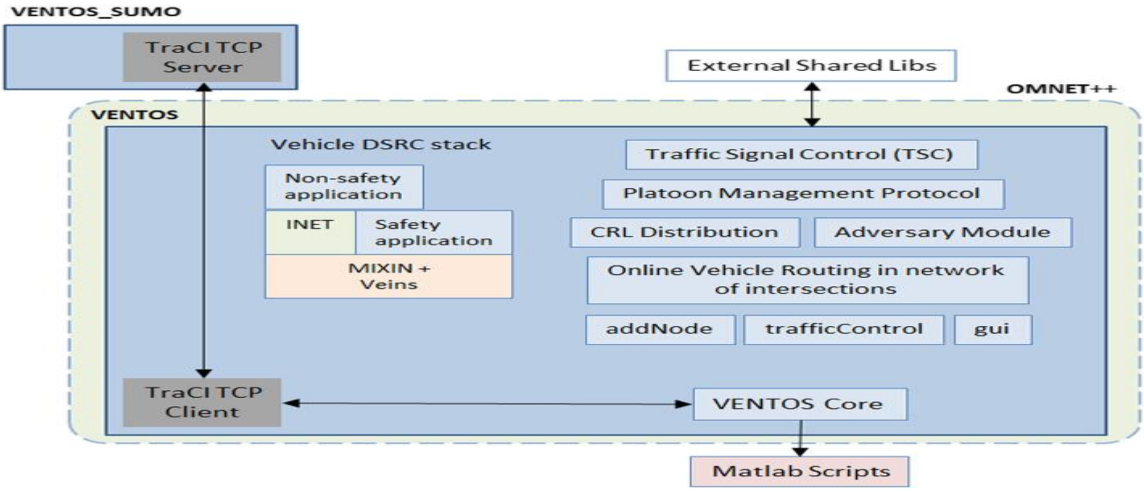


Figure 4-2: Architecture of the VENTOS platform

3. EstiNet

EStiNet is a commercial network simulator and emulator required for high accuracy, So it uses a popup method called kernel reentering to combine emulation and simulation approaches on the network. Vehicle network simulation is an optional additional module for the network simulator and emulator EstiNet. It offers vehicle traffic simulation for V-to-V and V-to-I network simulation.

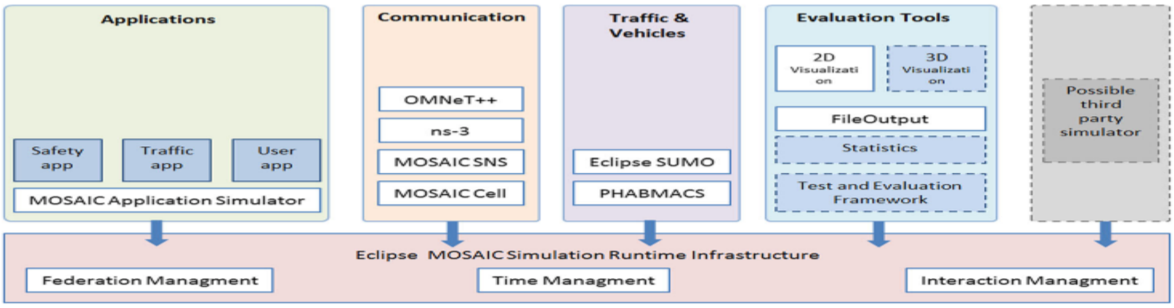


Figure 4-3: Architecture of the EstiNet platform

Table 4-3: Comparison of VANET simulator

VANET Simulator	Mobility Generators	Network Simulator	Active Development	Features
VIENS	SUMO	OMNET++	YES	Multiple flow models, online configuration
VENTOS	SUMO	OMNET++	YES	Vehicle movement models, Dynamic traffic routing
EstiNET	Inbuilt	EstiNET	YES	Multiple Traffic flow modules

Following the above comparison tables; Table 4-1, Table 4-2, and Table 4-3, we simulated the proposed solution using:

SUMO: As a mobility generator, because it supports a complex network and has a feature of Interoperability. The capability to interact with various network simulators was one of its natures. Also, realistic vehicular movement can be easily implemented using this mobility generator. In addition, importing a road map to SUMO can be done simply.

OMNET++: As a network simulator, it supports parallel distributed simulation. It could be easy to simulate a network while generating traffic mobility. Also, different wireless ad hoc modules (i.e., for MANET and VANET) are available.

Veins Framework: As a VANET simulator integrating SUMO and OMNET++, it contains the IEEE802.11p model, including QoS channel access conforming to CSMA/CA.

4.3 Performance Evaluation Metrics and Simulation Setup

4.3.1 Performance Evaluation Metrics

Various network and traffic simulators were available for simulating a network scenario. To apply and integrate a new mechanism in CSMA/CA of VANET. OMNET++, including SUMO integrated with Veins framework, is a suitable simulator for vehicular communication like VANET. After implementing the proposed solution into a simulation, the result will be evaluated using different performance metrics(Alkahtani & Alturki, 2021),(Kaur & Kaur, 2015). The following evaluated metrics are used to measure the newly proposed mechanism, WC-CSMA/CA, compared with the existing CSMA/CA of VANET.

Average Throughput: Average number of bits that can be transferred over the channel in units of time(bit/sec).

$$\text{Average Throughput} = (\Sigma\ \mathrm{BTC})/(\mathrm{st} - \mathrm{St}) \quad (4.1)$$

Average Waiting-time: The amount of time a host waits to connect the channel or the total number of connection requests a host tries until the channel accepts it.

$$\text{Average Waiting-time} = \Sigma(\mathrm{NC}) \quad (4.2)$$

Average Queuing Delay: Average amount of time between when the channel accepts a connection request and when a request process begins to sense/ accesses the channel(milli-second). It is a delay that a connection spends in a connection queue.

$$\text{Average Queuing delay} = (\mathrm{st} - \mathrm{St})/\ (\mathrm{NC}) \quad (4.3)$$

NB: BTC; Bits transferred over a channel, (St, st): Start Time and Stop Time, and NC: Number of connections.

4.3.2 Simulation Setup

By integrating two basic simulators, OMNET++ as a network simulator and SUMO as a road traffic generator, the proposed WC-CSMA/CA simulation was conducted using the Veins framework to run the V-to-I and V-to-V vehicular network simulation. The first task is to create a traffic motion for a bidirectional linear road. We used the Addis-to-Adama expressway from the actual map of the Ethiopia road and customized it using OpenStreetMap. Since the Addis-to-Adama expressway was far too long to test the simulation, we specifically used around 1km road near Addis-to-Bishoftu getaway and generated the map.sumo.cfg file for integrating the traffic file to the Veins framework. Different modules in veins were used to simulate the proposed solution. In the ieee80211p module, the Mac1609_4.ned file is used. This module is a backbone to create a network since the CSMA/CA of IEEE802.11p is located in the MAC layer of VANET. PhyLayer80211p.ned was used for representing the ieee802.11p of the PHY layer. Also, RSU.ned, CAR.ned, and Scenario.ned files are used for generating and adding the RSU and the vehicles in the vehicular network scenario. To determine the impact of the SV for low, medium, and high traffic dense scenarios, we use 25,50, and,100 vehicles respectively by adding the SV to them. MMM mobility model is taken for vehicles movement because it is presented to simulate urban road environment and guide the vehicles in the original path. Other simulation setups and parameters used to simulate the proposed solution are listed in the following table 4.4.

Table 4-4 Simulation setup parameters

Parameter	Value
Network Simulator	OMNET++ 5.6.2
Mobility Generator	SUMO 1.10.0
VANET Framework	Veins 4.4
Mode of Communication	V-to-V and V-to-I
Number of Vehicles	20,50,100 vehicles
No of SV	3,5,10
Mobility Model	MMM
Road length	1km
Simulation Time	300 sec
Traffic Direction	Linear/ Bidirectional
Transmission Range	300m
Simulation Area	2500m X 2500m
MAC, PHY layer	802.11p
Routing protocol	AODV

4.4 Simulation Experiment

As mentioned earlier, to simulate the proposed solution, first, we had created the traffic motion using SUMO and network editor (Net Edit). Then, we took an actual road map located in Ethiopia using OpenStreetMap. It is the Addis-to-Adama expressway, the only highway supporting high-speed vehicles. To conduct the simulation of the proposed solution, we had chosen around 1Km road length, which is near to Addis-to-Bishoftu gateway. Since the main expressway is far too long to simulate and locate an RSU, we used a short lane sample. The main expressway includes a sub highway that leads to Bishoftu. Figure 4-4 shows a map taken and downloaded from OpenStreetMap.To make the map entirely ready for traffic simulation, the map.osm file was enhanced and edited using JOSM; java base editor used to customize and edit the actual map. Figure 4-5 shows the specific lane map selected, and Figure 4-6 shows the edited josm map file.

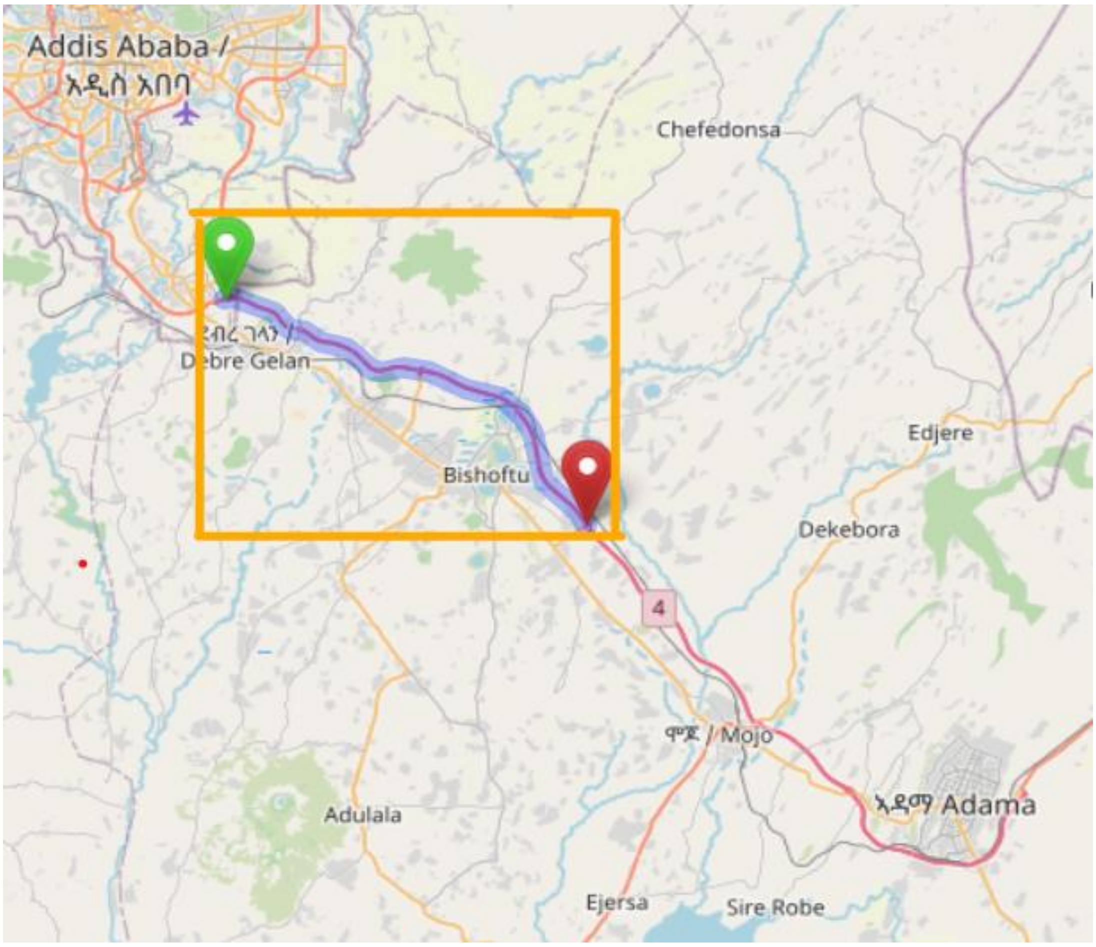


Figure 4-4 Addis-to-Bishoftu sub-expressway (tested area)

Figure 4-5 Expressway tested area

Figure 4-6 Edited map.josm file of the tested area

For generating a vehicle traffic motion for the proposed solution, the vehicle network must be created first, for that reason, the map.josm file was imported to the net-edit software located in the SUMO directories. By creating different XML files, map.net.xml and map.rou.xml files are the network and routing files. Finally, the SUMO configuration file (sumoconfig.sumo.cfg) was generated to run the realistic traffic motion for the road scenario. Figures 4-7 and 4-8 illustrate the images of the generated traffic scenario. The first image showed the Addis-to-Bishoftu expressway with vehicles from all directions. The second image illustrated the exact simulation tasted road for the proposed solution. The SUMO map.rou.xml , and the code for generating (*.cfg) configuration files are illuminated in **Appendix A** and **Appendix B.**

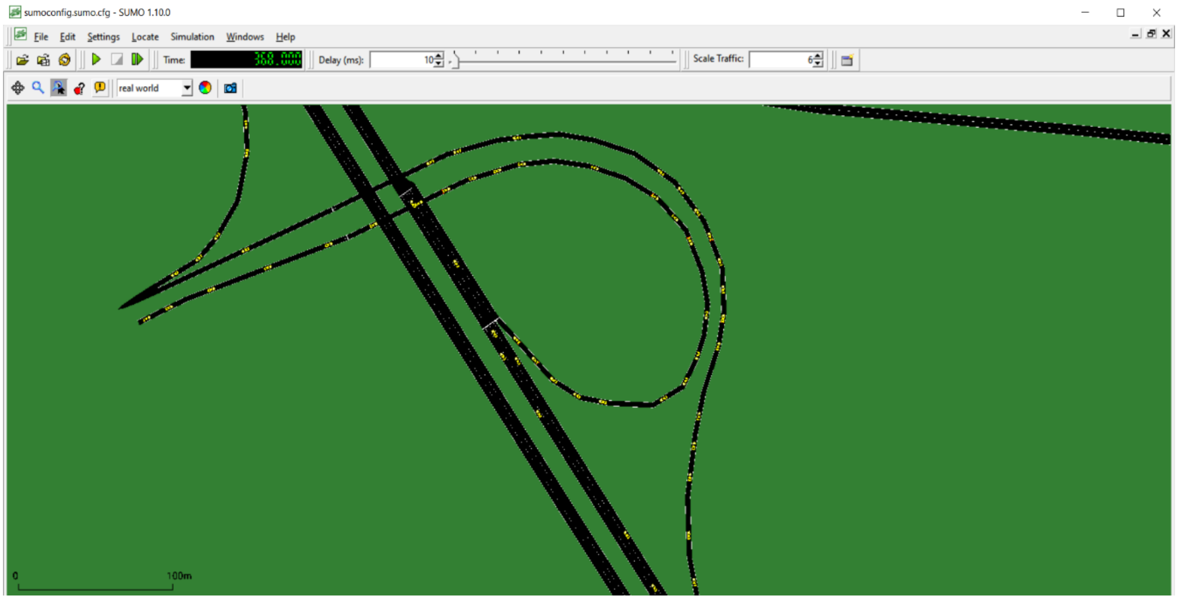


Figure 4-7: Zoom out image of the generated traffic motion

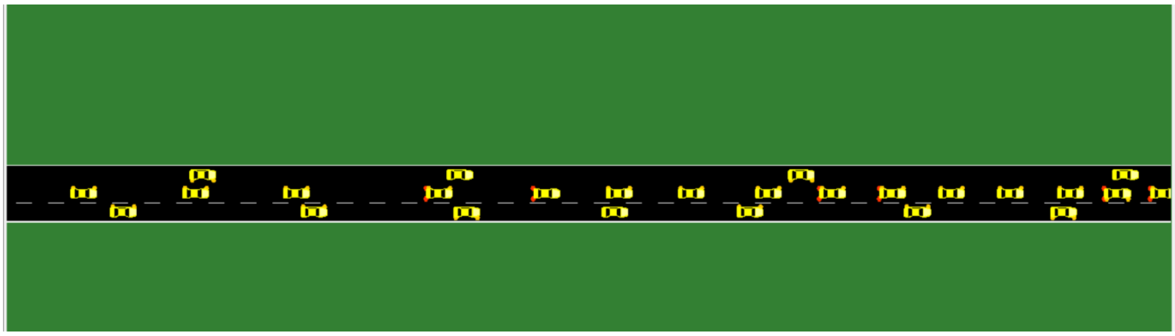


Figure 4-8: Linear bidirectional road scenario

After the SUMO traffic motion was created, the next task was to transfer the custom SUMO file into Veins framework to implement the simulation using OMNET++. Inet framework(another framework) was needed to simulate vehicular communication under this simulator. An open-source library model provides a routing protocol for ad-hoc, OSI layer implementation, and wireless interface. For applying an AODV routing protocol and designing a MAC and PHY layer to the proposed solution, we used both the Veins and Inet frameworks. We were using a working directory called VanetTutorial. We created our VeinsInet project. And then, we had set map.net.xml, map.rou.xml, and sumoconfig.sumo.cfg file into the project and a network scenario was created using the custom SUMO file.

A total of 100 vehicles as a node were included in the simulation. Therefore, we studied the impact of the SV by generating 25, 50, and 100 vehicles. Figure 4-9 shows the GUI of our simulation in the OMNET++ simulator with 50 vehicles with an average speed of 50km/hr. To run the network scenario omnetpp.ini was used. OMNET++ command for opening the working directory for IEEE802.11p is exhibited in **Appendix C**.

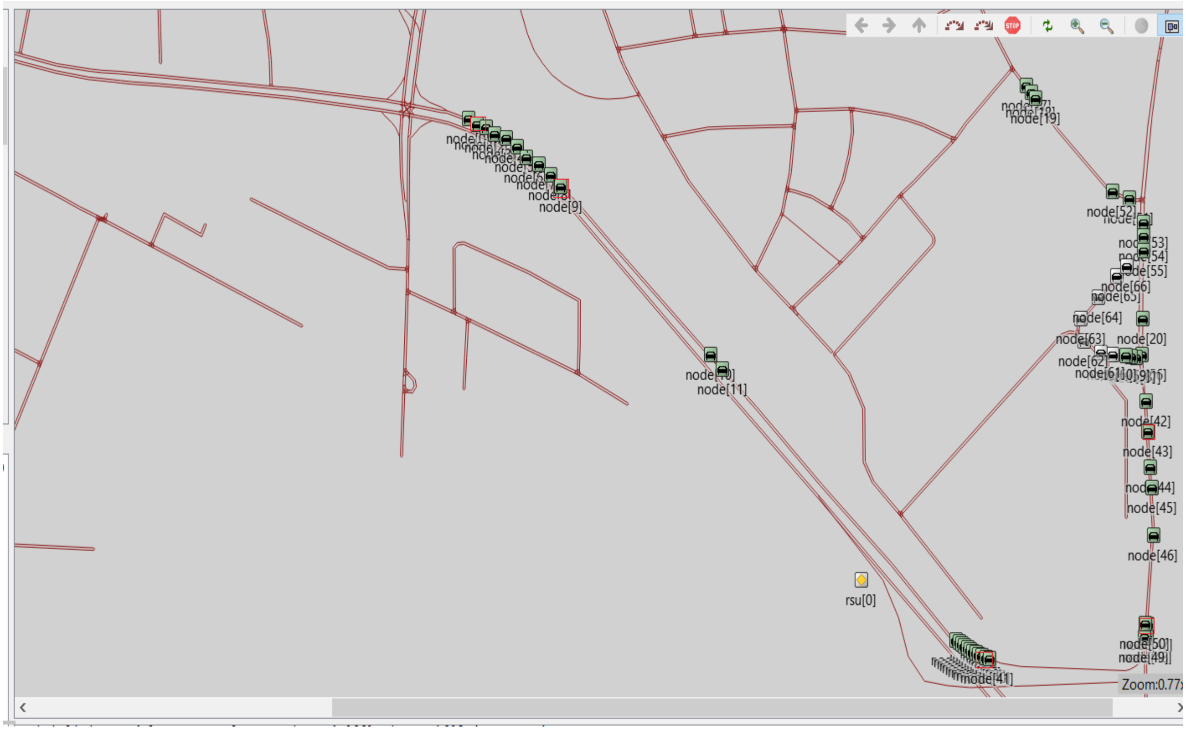


Figure 4-9: Simple GUI simulation for 50 nodes including the SV

The purpose of taking 25, 50, and 100 vehicles was to analyze the selfish behavior at minimum, medium, and maximum traffic density. For minimum road dense 25 vehicles are utilized, and for maximum and medium 50 and 100 vehicles respectively. The performance, as mentioned earlier metrics evaluated the SV impact in the existing CSMA/CA; Average throughput, average waiting time, and average queuing delay. The SV is assigned by their contention window lower value in the simulation process. In this case, the SV means a node that manipulates its CW_{lower} value into 5 for getting a higher probability for accessing the channel. Therefore, the vehicle which has $CW_{lower} = 5$ is considered as SV. Other vehicles are referred to as regular vehicles, which behave cooperatively. The code for assigning vehicle parameters is shown in **Appendix D.**

Figure 4-10 shows the GUI of the simulation, which included the selfish vehicles, which are marked by a red rectangle.

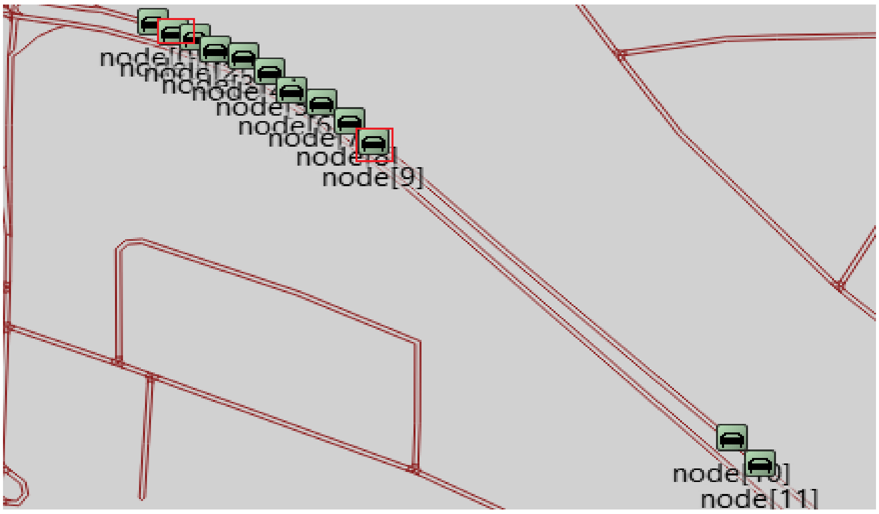


Figure 4-10: SV GUI simulation

4.5 Simulation Result

To comprehensively analyze and understand the impact of the selfish vehicle's behavior in different traffic dense scenarios, we constructed a vehicular network that handled 25,50,100 vehicles, respectively. Because a lane with a single RSU can serve and communicate 25-100 vehicles in its region. The simulation was first conducted to study the effect of selfishness misbehavior by different vehicle densities. We tested the simulation using three different scenarios, which are described as follows;

- **Scenario one:** It is considered as "**Min-dense**" for minimum vehicle density. In this scenario, we use 25 vehicles and assume three vehicles are SV.
- **Scenario Two:** It is considered as "**Med-dense**" for medium vehicle density. 50 vehicles are implemented for this scenario by assuming five vehicles are SV.
- **Scenario Three:** It is considered as "**Max-dense**" for maximum vehicle density. 100 vehicles are conducted for this scenario by including ten SV in the network.

To get a high chance of accessing the channel before any other vehicles, the SV slightly lowers their backoff time by lowering the CW_{lower} value. If the channel is occupied, the SV vehicle may increase its CW_{upper} value, to be considered as it waits much time other than any other vehicle. In this thesis, the SV has assumed it chose a lower backoff time by taking the minimum value of $CW_{lower} = 5$ slots from a random value of [0 to CW+1]. Before attempting the subsequent transmission and serving first, the SV manipulates the back-off time by taking a smaller CW size. The following algorithm showed the parameters that could be set and adjusted to determine the selfish behavior in channel access.

```
std: queue<BaseFrame1609_4*> queue;
int aifsn; // queue; number of aifs slots
int cwMin; // minimum contention window size
int cwMax; // maximum contention window size
int cwCur; // current contention window size
int64_t currentBackoff; // queue; current Backoff value
```

Algorithm 5: Code for channel access priorities

By applying the above three scenarios in the existing CSMA/CA of IEEE802.11p, This Section addresses the impact of SVs; the result is evaluated in the case of Min-dense, Med-dense, and Max-dense in terms of average throughput, average waiting time, and average queuing delay. The following graph shows the simulation result.

4.5.1 The Effect of SV in terms of Average Throughput

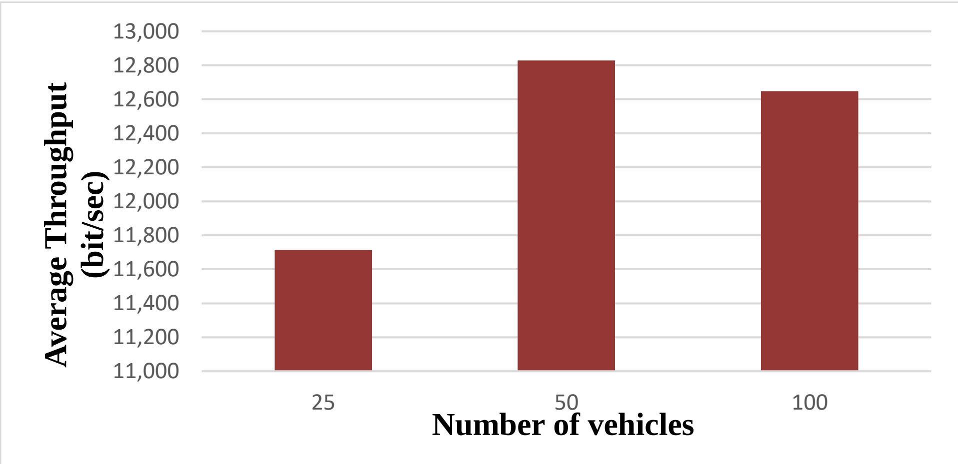


Figure 4-11: Throughput Vs. SV effect

Figure 4-11 showed the acquired result of the three scenarios described earlier in average throughput. In the case of Min-dense with three selfish vehicles, the average throughput has decreased compared to Med-dense and Max-dense vehicle density scenarios. The total percentage of the SV in Min-dense is 15% and 10% for both Max-dense and Med-dense. As the percentage of selfish behavior increase in the network, the average network throughput decrease. The SV increased their channel accesses and transmission probability and forced other well-behaving vehicles to double their backoff value for the next transmission trail, significantly decreasing the regular vehicles' accesses probability and later degrading their throughput. Even if both Med-dense and Max-dense had the same SV percentage, the average throughput of Max-dense was slightly decreased as compared with Med-dense. It is because, as the number of vehicles increases, the average throughput of the network decreases. The percentage of the average throughput for the three scenarios is 11.7%, 12.8%, and 12.6%.

4.5.2 The Effect of SV In terms of Average Waiting Time

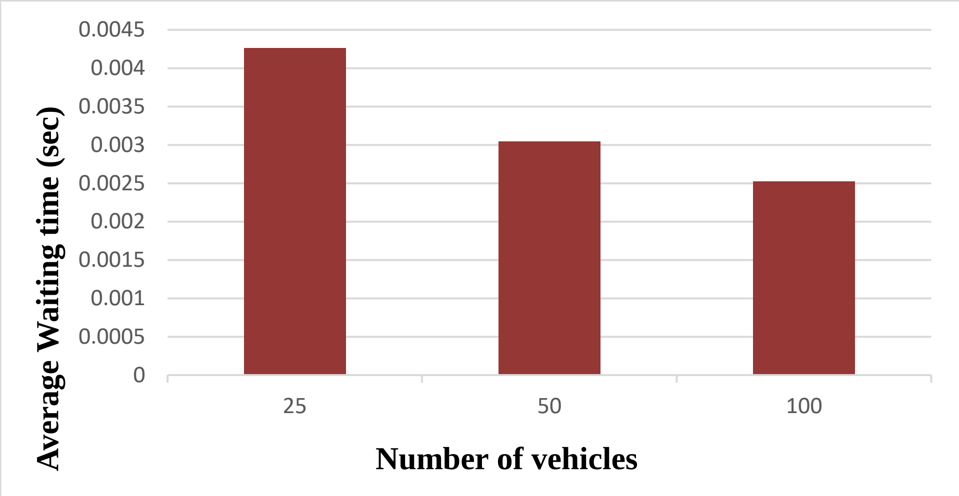


Figure 4-12: Average waiting time Vs. SV effect

Figure 4-12 demonstrated the impact of the SV in terms of average waiting time based on the three-vehicle density scenarios. As shown in the figure in the case of Min-dense, the average waiting time is higher than compared to the other two scenarios. The result indicated that when the SV chooses a lower backoff time from the interval of [0, CW +1], it can access the channel before the other well-behaving vehicles or get a higher chance of being served to access the channel in the queue. However, once the selfish vehicle occupies a channel, it takes a long time and causes other vehicles to back off many times before successfully transmitting their packet. Therefore, it causes a busy channel time to increase and forces the regular vehicles to wait long until they can access the channel. Thus when the number of selfish vehicles increases, the waiting time of the overall vehicles increases. For example, since the number of SV is 20% in Min-dense, its waiting time is 42.6% higher than in other scenarios. On the other hand, the waiting time for the other two scenarios is 24% and 30%.

4.5.3 The Effect of SV In terms of Average Queuing Delay

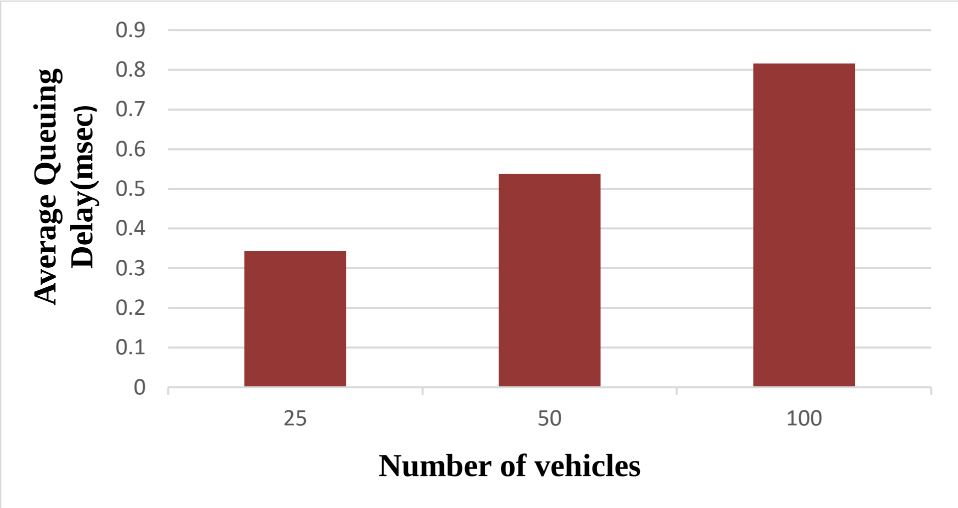


Figure 4-13: Average queuing delay Vs. SV effect

As for the average queuing delay, Figure 4-13 illustrates that the impact of SV in the case of Max-dense has a higher value than two other scenarios in terms of average queuing delay. Also, the Med-dense value is slightly increased compared with the Max-dense. Since SV has a lower contention window value, it gets a chance to access the channel first. For that matter, the SV almost entirely receives the channel and heads of the other vehicles to access the medium and increases their staying time in the queue until their connection request is accepted. And also, as a vehicle density increases, it leads to a delay in the same case. Therefore, the average queuing delay of the network will decrease as the vehicles' selfish behavior and vehicle density increase. The total percentage of the average queuing delay for Min-dense, Med-dense, and Max dense was 34.3%, 53.7%, and 81.6%.

4.6 Simulation Result Analysis

The following Simulation graphs show the comparison of the existing CSMA/CA with the proposed solution WC-CSMA/CA in the presence of selfish vehicles. The presence of SV in the CSMA/CA results in decreased average throughput and increased average queuing delay and average waiting time. Comparing the performance of the existing CSMA/CA with WC-CSMA/CA was measured by adding a different number of SV in different vehicle density scenarios. The vehicle that selects a lower backoff value [CWlower = 5 slots] is considered selfish. The result was measured by conducting 25,50,100 vehicles by adding 3,5,10 selfish vehicles. The following Section described the proposed CSMA/CA comparison with the original CSMA/CA.

4.6.1 Result Comparison In terms of Average Throughput

Figure 4-14 demonstrated the proposed WC-CSMA/CA comparison with the existing CSMA/CA in average throughput. As the graph showed, the overall network throughput of the proposed WC-CSMA/CA is higher than the existing CSMA/CA with the existence of the SVs.It is due to WC-CSMA/CA having a cooperative mechanism by motivating the SVs to be well-behaved and cooperative again. For instance, after the proposed WC-CSMA/CA was implemented, the average throughput for 20 vehicles, including the SV, was upgraded from 11,712.32 bit/sec to 13,107.87 bit/sec. As the consequence of the motivation phase of the proposed system encourages the identified SV to behave normally by giving credit if it adjusts its waiting time and averaging its connection trial.

For this reason, once the SV returns to its well-behaved behavior and chooses a random backoff interval from [0, CW+1] as the regular vehicle does, the overall average throughput of the network will increase. Because it does not manipulate the backoff time and the channel accesses probability by well-behaving like other vehicles. The following figure shows that the percentage of average throughput of the proposed WC-CSMA/CA is 13.1%, 14.7.%, and 17.3% for 20, 50,100 vehicles, which is higher than the percentage of the existing CSMA/CA.

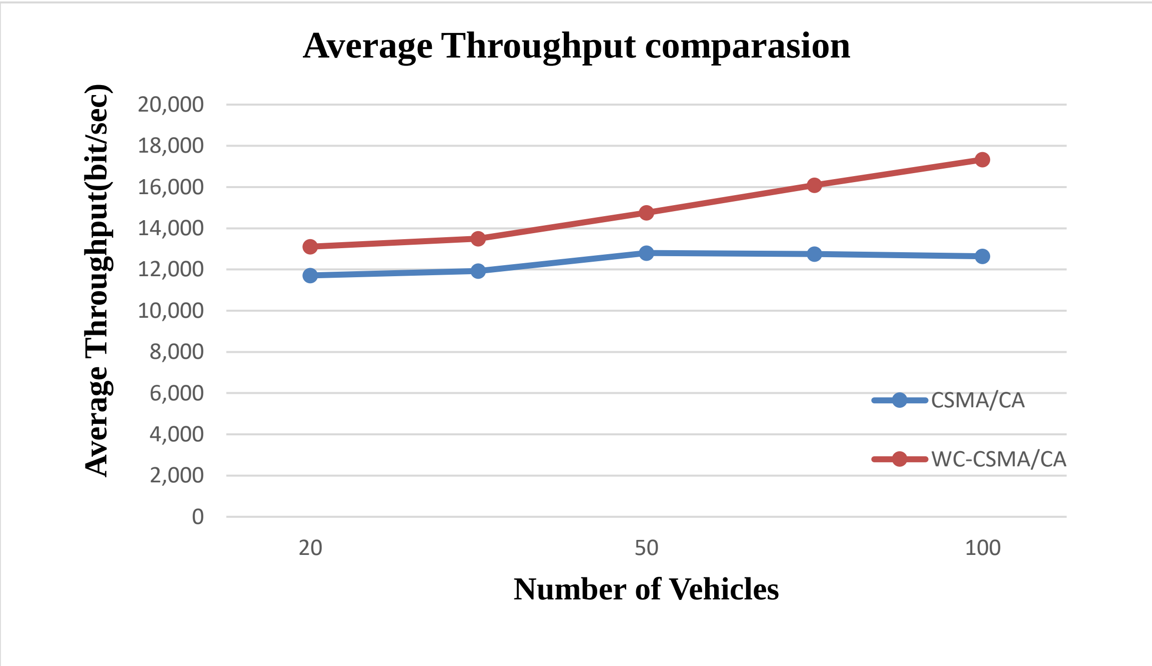


Figure 4-14: Average throughput comparison

Even if the proposed WC-CSMA/CA had reached the better throughput performance, major costs were happening during the attending of enhancing the overall throughput. These costs were overhead ratio and simulation complexity. This is due to the mechanism of motivating those identified SVs leads a massage storm and earning credit by averaging their waiting time and connection trial creates a bit complexity in the channel accessing prosses.

4.6.2 Result Comparison In terms of Waiting time

Figure 4-15 showed the simulation result for the average waiting time of the original CSMA/CA compared with the WC-CSMA/CA for the different number of vehicles[25,50,100]. As shown in the graph, the average waiting time of the proposed WC-CSMA/CA is lower than the original CSMA/CA in the presence of SVs. As a consequence of the motivation mechanism of WC-CSMA/CA orders the SVs to revive in the network again and decrease the total waiting time of the other host vehicles, which forces them to stay in the line till they get a chance to access the channel. In the proposed WC-CSMA/CA, the SV was assumed to be a cooperative vehicle since it gains a good credit like other vehicles, determining if it chooses a random backoff value and averages its waiting time without scuttling the channel access process. After the SV becomes cooperative and well behaved, the impact of the channel busy time is decreased. Also, the waiting line of other serving vehicles becomes average, which means the whole vehicle's average waiting time decreases. For example, in the case of Med-dense, the average waiting time of the original CSMA/CA was 0.00462 sec, and it has reduced and improved to 0.00368 when the proposed WC-CSMA/CA applied. The following graph illustrated that the waiting time of the original CSMA/CA is 43.6%, 30.4%, and 25.2% and for the WC-CSMA/CA is 36.6%, 24.9%, and 16.5% in the presence of SVs.

While the proposed WC-CSMA/CA has gained a better performance in terms of waiting time, other QoS awareness costs were attained. Which are overhead ratio and massage -overflow. In the motivation mechanism, the competition of gaining a high score and incrementing a credit makes an SVS broadcast many messages.

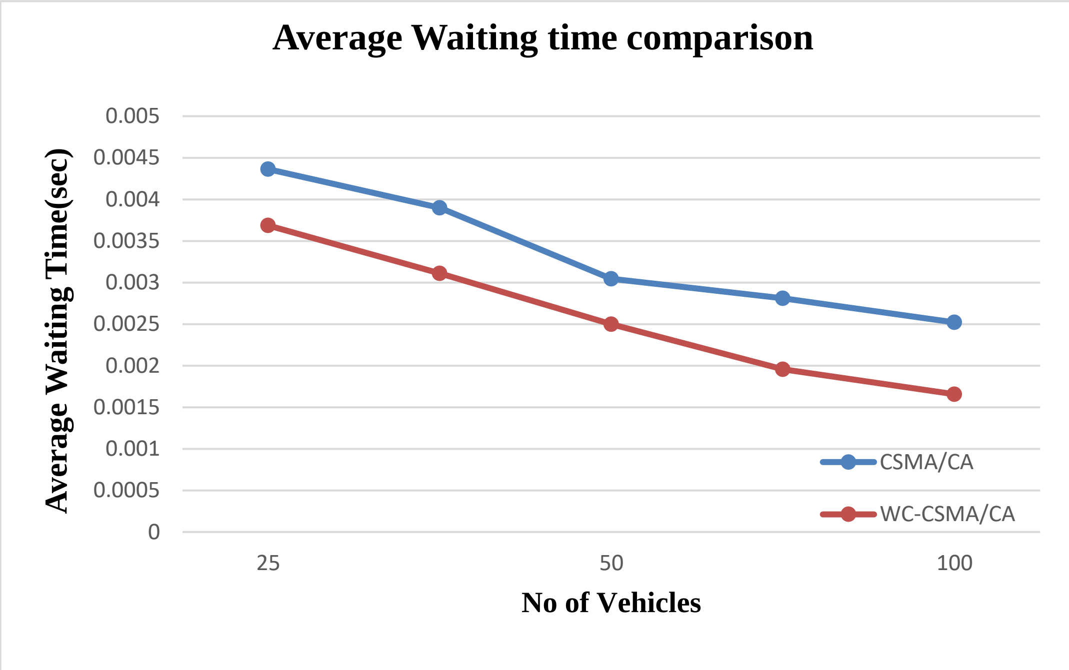


Figure 4-15: Average waiting time Comparison

4.6.3 Result Comparison In terms of Average Queuing Delay

As Figure 4-16 illustrated, the simulation result of the average queuing delay of the proposed WC-CSMA/CA is lower than the existing CSMA/CA in the existence of the SV. Since the proposed system motivating mechanism changes the selfish behavior of the vehicles into well-behaved, the delay caused by the SV occupying the channel is decreased. It means regular vehicles spend staying in the queue until the channel accepts the connection request is improved. Since the proposed WC-CSMA/CA mandates the SV to be motivated and behave well, it increases the probability of the other regular vehicle accessing the channel without suffering. As a result, the average queuing delay of the existing CSMA/CA is higher than the proposed WC-CSMA/CA. As an illustration, the average queuing delay of the existing CSMA/CA in the case of Max-dense was 0.8166 msec, and It reduced to 0.722 msec after WC-CSMA/CA was applied. The average queuing delay present of the WC-CSMA/ CA is 22.2%, 45.3% 72.2% for Min, Med, and Max vehicle density scenarios. And 81.6%, 53.7% and 81.6% for the original CSMA/CA.

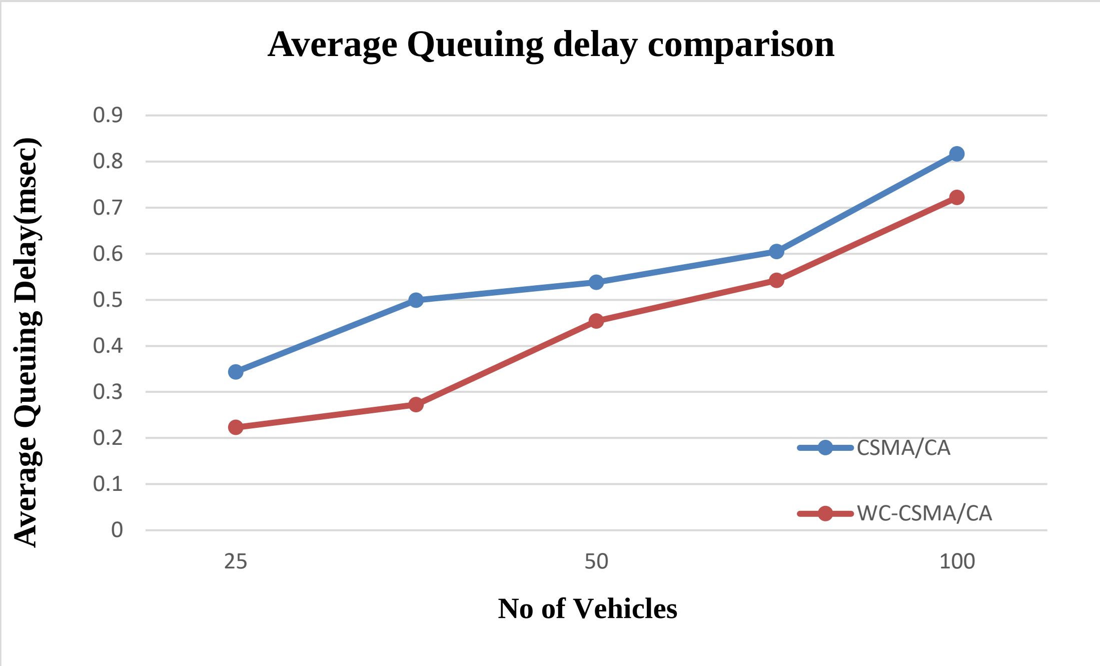


Figure 4-16: Average queuing delay comparison

4.7 Evaluation Result

The following table includes the performance evaluation result for the existing CSMA/CA and the proposed WC-CSMA/CA. We have simulated the result frequently, but the results were not changed. We are building the simulation whenever we have tried to simulate the network. Testing or outright simulation helps ensure the parameters are running without any errors. The output data is acquired from the simulation report directory E:\Dev\VANETTutorials\simulations\veins\results and OMNET++/Qtenv GUI event log. Based on the result, we distinguished how the proposed WC-CSMA/CA improves the impact of SVs compared with the existing CSMA/CA in the presence of SVs. As a result, the performance metrics; Average throughput altered, average waiting time, and average queuing delay reduced in the case of WC-CSMA/CA. The table shows how the WC-CSMA/Ca has a better performance than CSMA/CA. For instance, we took a result for Max-dense[100 vehicles], including 10 SVs. The enhancement occurs due to the motivation mechanism used by the proposed solution. The below parameters value assures that WC-CSMA/CA has a better improvement.

Table 4-5: Simulation report

Parameter	CSMA/CA	WC-CSMA/CA
Simulation time	200 sec	200 sec
Average throughput	12,647 Mbit/Sec	17,332 Mbit/Sec
Average waiting time	0.00252 Sec	0.00165 Sec
Average queuing delay	0.8163 msec	0.722 msec

From the above table, we have surmised that. Since the original CSMA/CA has no countermeasure that identifies and motivates the SVs, these SVs degrade the overall throughput of the network. However, the proposed WC-CSMA/CA identified and motivated those SVs and made them cooperate again. As a result, the average throughput of the network was enhanced from 12.6% to 17.3%. Furthermore, in terms of average waiting time, the proposed WC-CSMA/CA has been improved in the case of the proposed solution compared with the existing CSMA/CA. As shown in the table, the average waiting time of the WC-CSMA/CA is 0.00165 sec, whereas 0.00252 for the original CSMA/CA. The result was improved since it was reduced in the case of the proposed solution as the average waiting time has a hostile retaliation. Therefore, the improvement is considered when the waiting time is decreased. The motivation phase used by the proposed solution revives the SVs to be well behaved again. And it helps other regular vehicles be adequately served and reduces the time they spend while waiting in the line during requesting. Likewise, the average queuing delay is lower regarding the proposed WC-CSMA/CA than the original CSMA/CA. The existence of the SV increased a delay in the process of channel accessing Since it took much time in the channel and forced the other vehicles to hold in the queue. The original CSMA/CA has a higher average queuing delay value than the proposed solution because the SVs were not motivated and ordered to behave well again. The result was improved from 0.8163 msec to 0.712 msec.

Table 4-6:Total results of CSMA/CA and WC-CSMA/CA

Performance Evaluation Metrics	**Number of Vehicles**					
	***Min-Dense [25 Vehicles, 3 Svs*]**		***Med-Dense [50 Vehicles, 5 Svs]***		***Max-Dense [100 vehicles, 10 Svs]***	
	CSMA/CA	**WC-CSMA/CA**	**CSMA/CA**	**WC-CSMA/CA**	**CSMA/CA**	**WC-CSMA/CA**
Average throughput (Mbit/Sec)	11712.32	13107.87	12800.18	15752.37	12647.32	17332.23
Average waiting time (Sec)	0.004362702	0.003685413	0.003044662	0.002498234	0.002520709	0.00165672
Average queuing delay (mSec)	0.343768	0.222967	0.537736	0.453937	0.816366	0.722009

4.8 Result Summary

As the previous Section described and reported, the existing CSMA/CA and the proposed CSMA/CA were Simulated based on the same number of vehicles, number of SVs, simulation time, and run in a similar IDE environment. According to the three metrics mentioned earlier, the proposed WC-CSMA/CA was more effective than the existing CSMA/CA. The significant difference between the proposed WC-CSMA/CA and the original one was the motivation mechanism to encourage the SVs to revive the network again. The existence of the SVs in the existing CSMA/CA was done by lowering their backoff value. And this selfish behavior degree the network throughput and increases the waiting time and the queueing delay of the typical vehicle while the SVs take the channel for too long. Therefore, applying the identification and motivation mechanism for detecting and ordering those SVs to be cooperative again, the proposed solution has a better performance than the original CSMA/CA. The following table shows how the proposed WC-CSMA/CA has a better performance than the original CSMA/CA using the percentage result of the three-evaluation metrics.

Table 4-7: CSMA/CA VS. WC-CSMA/CA

		Average Throughput	***Average Waiting Time***	***Average Queueing Delay***
CSMA/CA	Min-dense [25]	11.7%	42.6%	34.3%
	Med-dense [50]	12.8%	30.4%	83.7%
	Max-dense [100]	12.6%	35.2%	81.6%
WC-CSMA/CA	Min-dense [25]	13.1%	36.6%	22.2%
	Med-dense [50]	14.7%	24.9%	45.3%
	Max-dense [100]	17.3%	16.5%	72.2%

All in all, the graph shows the percentage of each evaluation matrices in the case of Min-dense, Med-dense, and Max-dense for the proposed WC-CSMA/CA and the original CSMA/CA. Since vehicle density and the number of SV increased, The value of each metrics' percentage varies. The graph demonstrated a relative variance between the existing CSMA/CA and the proposed WC-CSMA/CA in all evaluation matrices. The overall network performance gets lower when the number of SVs increases. By motivating the SVs to be cooperative in the network again, the common issues were improved by our proposed WC-CSMA/CA. ***Average Throughput;*** From our study, the proposed WC-CSMA/CA improves the average throughput by 11.9% in the case of Min-dense,23.06% in the case of Med-dense, and 37.04% in the case of Max-dense compared with the existing CSMA/CA. ***Average Waiting Time;*** From this study, the proposed WC-CSMA/CA average waiting time is decreased by 15.52%, 17.94%, and 37.27% in Min-dense, Med-dense, and Max-dense vehicle density from the existing CSMA/CA**.** ***Average Queuing Delay;*** From our study, the average queuing delay of the proposed WC-CSMA/CA compared with the existing CSMA/CA is reduced by 35.14% in the case of Min-dense and 15.5% in both Men-dense and Max-dense.

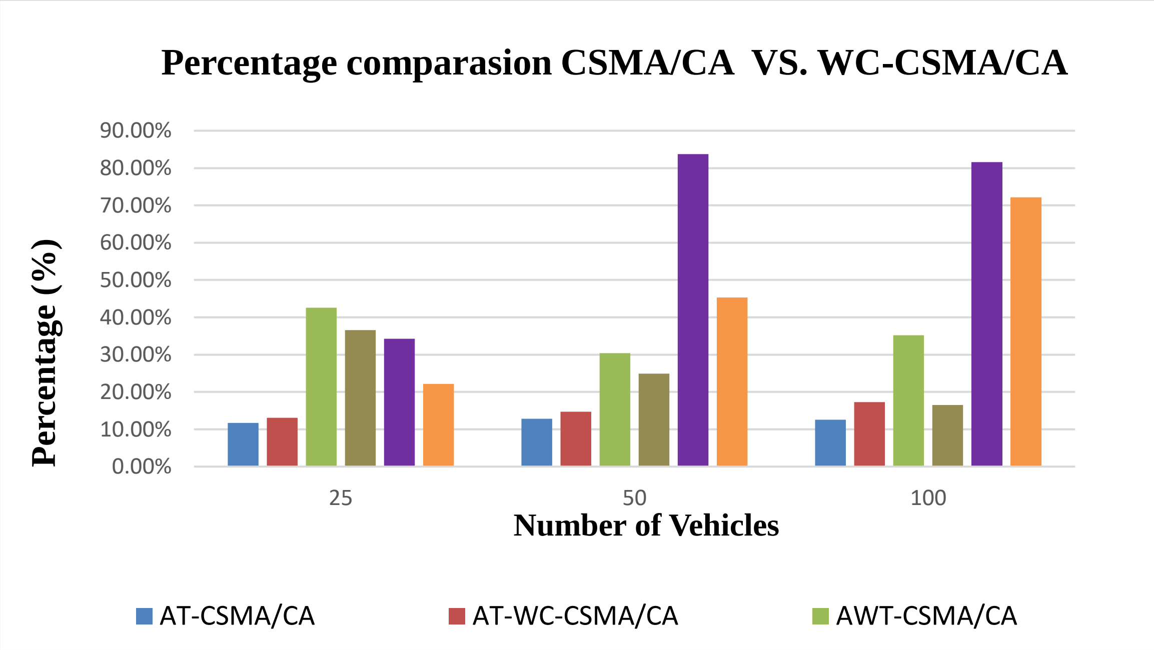


Figure 4-17: CSMA/CA VS. WC-CSMA/CA

NB: AT: Average Throughput, **AWT**: Average Waiting Time, **AQD**: Average Queuing Delay

4.9 Discussion

To compare the result based on the three primary metrics, the proposed WC-CSMA/CA and the original CSMA/CA were simulated in the same simulation tool, OMNET++, using the same number of SVs and similar methods vehicle density. By considering different vehicle densities, we studied the impact of the SVs in maximum, medium, and minimum vehicle density. As a result, the proposed solution is better than the existing CSMA/CA because a cooperation mechanism motivates the SVs to behave and act regularly in the network. The result of the performance evaluation matrices illustrated in Tables 4.6 and 4.7 assures that the proposed WC-CSMA/CA has a relative improvement compared to the existing CSMA/CA.

The research questions; stated in section 1.4 and aimed to be answered and addressed by our study based upon the result and the performance evaluation metrics described are going to be discussed as follow ;

RQ1: *How to resolve the SVs effect in CSMA/CA IEEE802.11p by collaborating a watchdog approach with a credit-based approach?*

To resolve the SVs effect in CSMA/CA, we have to propose an adequate solution. For that matter, different incentive mechanisms for detecting and elevating selfish behavior in ad hoc networks were assessed and evaluated. The two reputation-based approaches were selected to propose WC-CSMA/CA: the watchdog approach and the credit-based approach.
Since the proposed solution has two main phases, for the identification phase watchdog approach was used to monitor and identify the selfish behavior of vehicles. A credit-based approach was adapted in the motivation phase to encourage the SVs by giving credit if they are willing to cooperate again. By collaborating these two approaches to work consecutively, WC-CSMA/CA rectifies the SVs to be well-behaved like other honest vehicles.

RQ2: *How to design a new mechanism for Selfish Vehicles to identify and motivate them?*

After finding out selfish behavior issues in CSMA/CA IEEE802.11p of VANET. The next task for this study was to design and propose a new solution that resolves those issues by assessing different incentives Schemes for solving vehicle' selfishness misbehavior. We have proposed a solution, described briefly in section 3.3, by applying the node-oriented intensive mechanism. The proposed WC-CSMA/CA identifies the SVs using the watchdog approach and motivates them using the Credit-based approach. By calculating the backoff, load, and throughput of the vehicles, the SVs are identified by the VVs; after that, to revive those SVs using the motivation phase, the SV gets higher credit if it gained a good score cooperating again in the network. Suppose it averages its waiting time and relatively reduces its connection trial in the case of channel access. Then, it will be allowed to transmit data.

RQ3: *How can we assess the performance of the newly designed WC-CSMA/CA solution with the existing CSMA/CA IEEE802.11p of VANET?*

The performance evaluation was conducted by taking the simulation result of both CSMA/CA and WC-CSMA/CA in the presence of SVs. Then we assessed the performance of the newly designed WC-CSMA/CA solution compared with the existing CSMCA using three evaluation metrics described in sec 4.3.1. As described in sections 4.6.1, 4.6.2, and 4.6.3, since the proposed WC-CSMA/CA has a cooperative mechanism, the effect of SVs is improved relatively. Also, average throughput is increased in the performance evaluation metrics, and the average waiting time and queuing delay is reduced compared with the existing CSMA/CA. The following figure shows how much the proposed WC-CSMA/CA improves those performance metrics.

Table 4-8 Performance assessment of WC-CSMA/CA

Evaluation metrics	Min-dense improvement	Med-dense improvement	Max-dense improvement
Average throughput	↑ 11.9%	↑23.06%	↑37.04%
Average waiting time	↓15.2%	↓17.94%	↓34.27%
Average queuing delay	↓35.14%	↓15.5%	↓11.55%

CONCLUSIONS AND FUTURE WORK

Conclusion

This study resolved and improved the effect of selfish behavior in CSMA/CA that leads to throughput degradation, network partitioning, and channel crowdedness. Furthermore, various VANET and CSMA area-related literature and papers were reviewed and assessed to formulate the problem caused in CSMA/CA with the existence of SVs.Afterward, we proposed WC-CSMA/CA solution that solves the problem and improves the existing CSMA/CA. The proposed WC-CSMA/CA motivates the SVs and revives them to perform regularly like other typical vehicles by creating a cooperative mechanism. The proposed solution has three basic steps, Verifier vehicle selection, Identification phase, and motivation phase. After the SVs are identified by the verifier vehicles in the proposed WC-CSMA/CA, the next task is to make them cooperative again in the network using the motivation phase. By crediting those SVs to fulfill the standard criteria using two countermeasure metrics(Average waiting time and Connection Trial) and behave as a regular vehicle. The conceptual framework is briefly described in section three by including the flowcharts, pseudocodes, and basic assumptions. The simulation is conducted using veins framework by integrating OMNET++ as a network simulator and SUMO as a traffic generator to test and elaborate the proposed solution reliability.

The outcome of the simulation, The following observations are acquired. First, the average throughput of WC-CSMA/CA is slightly increased and varies a bit from min-dense to max-dense. However, the average waiting time was reduced moderately, and the average queuing delay is decreased highly from Min-dense, Med-dense, and max-dense.

Study Output

As a Study output, the performance of the proposed wC-CSMA/CA is increased with the existence of SVs. By adding the SVs in both CSMA/CA and WC-CSMA/CA, we evaluated the performance of both mechanisms. Those SVs are assigned by lowering their CWlower value. Since the proposed WC-CSMA/CA has an identification and motivation mechanism, the Svs can cooperate again in the network. Due to this reason, the following performance metrics are enhanced better when compared to the existing CSMA/CA.

Average Throughput: Average Throughput was increased in the proposed solution.

Average Waiting Time: Average Waiting Time was reduced in the proposed solution.

Average Queuing Delay: Average Queuing Delay was decreased in the proposed solution.

Contribution

The main contribution of this study is to rectify the selfish vehicles that misuse the BoM during channel accessing in CSMA/CA ieee802.11p of VANET. By identifying the SVs using VVs and motivating them by giving credit once they are willing to cooperate again, the selfish misbehavior that occurs in CSMA/CA was resolved.

Future Work

The proposed WC-CSMA/CA motivates the SVs to be cooperative. However, the proposed solution is conducted in a linear bidirectional linear highway. Moreover, it includes only one RSU in the interval of 1km. for the future type of road scenarios as such, multidirectional lanes should have also been considered for future work. Also, the Km interval and the number of RSU need to be mediated.

SPECIAL ACKNOWLEDGMENT

This research work was funded by Adama Science and Technology University under grant number:

ASTU/SM-R/225/21
Adama, Ethiopia

REFERENCE

Al-Terri, D., Otrok, H., Barada, H., Al-Qutayri, M., & Al Hammadi, Y. (2017). Cooperative based tit-for-tat strategies to retaliate against greedy behavior in VANETs. *Computer Communications*, *104*, 108–118. https://doi.org/10.1016/j.comcom.2017.02.008

Alkahtani, S. M., & Alturki, F. (2021). Performance Evaluation of Different Mobile Ad-hoc Network Routing Protocols in Difficult Situations. *International Journal of Advanced Computer Science and Applications, 12*(1), 158–167. https://doi.org/10.14569/IJACSA.2021.0120119

Ayyappan, B., & Mohan Kumar, P. (2016). Vehicular Ad Hoc Networks (VANET): Architectures, methodologies and design issues. *2016 2nd International Conference on Science Technology Engineering and Management, ICONSTEM 2016*, 177–180. https://doi.org/10.1109/ICONSTEM.2016.7560946

Busson, A. (2014). *Modeling CSMA / CA in VANET Modeling CSMA / CA in VANET. June 2012*. https://doi.org/10.1007/978-3-642-30782-9

Cárdenas, A. A., Radosavac, S., & Baras, J. S. (2004). Detection and prevention of MAC layer misbehavior in ad hoc networks. *Proceedings of the 2004 ACM Workshop on Security of Ad Hoc and Sensor Networks, SASN'04*, 17–22. https://doi.org/10.1145/1029102.1029107

Chang, C. Y., Yen, H. C., & Deng, D. J. (2016). V2V QoS Guaranteed Channel Access in IEEE 802.11p VANETs. *IEEE Transactions on Dependable and Secure Computing*, *13*(1), 5–17. https://doi.org/10.1109/TDSC.2015.2399912

Cheng, M. X., Ling, Y., & Wu, W. B. (2018). MAC Layer Misbehavior Detection Using Time Series Analysis. *IEEE International Conference on Communications*, *2018-May*, 1–7. https://doi.org/10.1109/ICC.2018.8422724

Chouhan, P., Kaushal, G., & Prajapat, U. (2016). Comparative Study MANET and VANET. *International Journal Of Engineering And Computer Science, 05*(16079), 16079–16083. https://doi.org/10.18535/ijecs/v5i4.04

Eze, E. C., Zhang, S. J., Liu, E. J., & Eze, J. C. (2016). Advances in vehicular ad-hoc networks (VANETs): Challenges and road-map for future development. *International Journal of Automation and Computing*, *13*(1), 1–18. https://doi.org/10.1007/s11633-015-0913-y

Filali, F., & Bonnet, C. (2006). Mobility Models for Vehicular Ad Hoc Networks : A Survey. *IEICE Transactions on Communications*, *E89-B*(6), 1888–1891. http://www.scopus.com/inward/record.url?eid=2-s2.0-33745440978&partnerID=40&md5=566c2685569f5b8bd3d3d6a049a6d3d3

Ghori, M. R., Zamli, K. Z., Quosthoni, N., Hisyam, M., & Montaser, M. (2018). Vehicular Ad-hoc Network (VANET): Review. *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, 1–6.

Gopinath, A. J., Nithya, B., Mogalapalli, H., & Khanna, P. K. (2020). Channel Status based Contention Algorithm for Non-safety Applications in IEEE802.11p Vehicular Network. *Procedia Computer Science*, *171*(2019), 1479–1488. https://doi.org/10.1016/j.procs.2020.04.158

Goyal, P., Parmar, V., & Rishi, R. (2011). *MANET : Vulnerabilities , Challenges , Attacks , Application*. *August 2016*. https://doi.org/10.1007/978-3-642-17878-8

Hamdi, M. M., Audah, L., Rashid, S. A., Mohammed, A. H., Alani, S., & Mustafa, A. S. (2020). A Review of Applications, Characteristics and Challenges in Vehicular Ad Hoc Networks (VANETs). *HORA 2020 - 2nd International Congress on Human-Computer Interaction, Optimization and Robotic Applications, Proceedings*. https://doi.org/10.1109/HORA49412.2020.9152928

Hartenstein, H., & Laberteaux, K. (2010). *VANET Vehicular Applications and Inter-Networking Technologies - Hannes Hartenstein, Kenneth Laberteaux*. 466.

Hussein, S., Mohamed, M. S., & Krings, A. (2016). A new hybrid jammer and its impact on DSRC Safety Application reliability. *7th IEEE Annual Information Technology, Electronics and Mobile Communication Conference, IEEE IEMCON 2016*. https://doi.org/10.1109/IEMCON.2016.7746090

Ismail, T., Touati, H., Hajlaoui, N., Hadded, M., Bouzefrane, S., Saidane, L., Ismail, T., Touati, H., Hajlaoui, N., Hadded, M., Mühlethaler, P., Saidane, L. A., & Gobert, B. T. (2021). *Impact Analysis of Greedy Behavior Attacks in Vehicular Ad hoc Networks To cite this version : HAL Id : hal-03480486 Impact Analysis of Greedy Behavior Attacks in Vehicular Ad hoc Networks*.

Jafari, A., Al-Khayatt, S., & Dogman, A. (2012a). Performance evaluation of IEEE 802.11p for vehicular communication networks. *Proceedings of the 2012 8th International Symposium on Communication Systems, Networks and Digital Signal Processing, CSNDSP 2012*, 3–7. https://doi.org/10.1109/CSNDSP.2012.6292712

Jafari, A., Al-Khayatt, S., & Dogman, A. (2012b). Performance evaluation of IEEE 802.11p for vehicular communication networks. *Proceedings of the 2012 8th International Symposium on Communication Systems, Networks and Digital Signal Processing, CSNDSP 2012*, *July*, 6–11. https://doi.org/10.1109/CSNDSP.2012.6292712

Jain, J., & Chahal, N. (2016). a Review on Vanet, Types, Characteristics and Various Approaches. *INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY: 5 (2016) Pp. 239-245 (9).*, *4*(1), 1–11. https://zenodo.org/record/61605#.X6iWW51KjeR

Kabir, M., Islam, S., Hossain, M., & Hossain, S. (2014). Detail comparison of network simulators. *International Journal of Scientific & Engineering Research*, *5*(10), 203–218.

Kamble, S. J., & Kounte, M. R. (2019). Routing and scheduling issues in vehicular Ad-hoc networks. *International Journal of Recent Technology and Engineering*, *8*(3), 4272–4283. https://doi.org/10.35940/ijrte.C5168.098319

Karagiannis, G., Altintas, O., Ekici, E., Heijenk, G., Jarupan, B., Lin, K., & Weil, T. (2011). Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *IEEE Communications Surveys and Tutorials*, *13*(4), 584–616. https://doi.org/10.1109/SURV.2011.061411.00019

Kaur, R., & Kaur, H. (2015). Performance Evaluation of Routing Protocols in VANET. *International Journal of Future Generation Communication and Networking*, *8*(6), 239–246. https://doi.org/10.14257/ijfgcn.2015.8.6.23

Khairnar, P. V. D., & Pradhan, D. S. N. (2010). Comparative Study of Simulation for Vehicular Ad-hoc Network. *International Journal of Computer Applications*, *4*(10), 15–18. https://doi.org/10.5120/864-1214

Khan, U. A., & Lee, S. S. (2019). Multi-layer problems and solutions in VANETs: A review. *Electronics (Switzerland)*, *8*(2). https://doi.org/10.3390/electronics8020204

Komali, R. S., MacKenzie, A. B., & Gilles, R. P. (2008). Effect of selfish node behavior on efficient topology design. *IEEE Transactions on Mobile Computing*, *7*(9), 1057–1070. https://doi.org/10.1109/TMC.2008.17

Kumar, P., & Shukla, S. (2019). A Survey of Simulation Tools for VANET. *Journal of Analysis and Computation (JAC) , XII*(I), 1–12. www.ijaconline.com,

Kyasanur, P., & Vaidya, N. H. (2005). Selfish MAC Layer Misbehavior in Wireless Networks. *IEEE Transactions on Mobile Computing*, *4*(5), 502–516. https://doi.org/10.1109/TMC.2005.71

Lamssaggad, A., Benamar, N., Hafid, A. S., & Msahli, M. (2021). A Survey on the Current Security Landscape of Intelligent Transportation Systems. *IEEE Access*, *9*(Vlc), 9180–9208. https://doi.org/10.1109/ACCESS.2021.3050038

Li, M., Salinas, S., Li, P., Sun, J., & Huang, X. (2015). MAC-layer selfish misbehavior in IEEE 802.11 ad hoc networks: Detection and defense. *IEEE Transactions on Mobile Computing*, *14*(6), 1203–1217. https://doi.org/10.1109/TMC.2014.2348560

Li, Y. (2015). *An Overview of the An Overview of the DSRC/WAVE Technology*. *15*(March), 9–21.

Lu, Z., Wang, C., & Wang, W. (2010). On the impact of backoff misbehaving nodes in IEEE 802.11 networks. *IEEE International Conference on Communications*. https://doi.org/10.1109/ICC.2010.5502302

M.Shahid Anwer. (2017). *Improving TCP behaviour to non-invasively share spectrum with safety messages in VANET*. 183 Pages.

Ma, M., Liu, K., Luo, X., Zhang, T., & Liu, F. (2020). Review of mac protocols for vehicular ad hoc networks. *Sensors (Switzerland)*, *20*(23), 1–28. https://doi.org/10.3390/s20236709

Madi, S., & Al-Qamzi, H. (2013). A survey on realistic mobility models for Vehicular Ad Hoc Networks (VANETs). *2013 10th IEEE International Conference on Networking, Sensing and Control, ICNSC 2013*, 333–339. https://doi.org/10.1109/ICNSC.2013.6548760

Malik, S. A., Shah, M. A., Khan, S. A., Jahanzeb, M., Farooq, U., & Adnan Khan, M. (2010). Performance evaluation of IEEE 802.11p MAC protocol for VANETs.

Australian Journal of Basic and Applied Sciences, *4*(8), 4089–4098.

Mantas, N., Louta, M., Katsaros, K., & Kraounakis, S. (2018). Social CLWPR: A socially enhanced position based routing protocol for handling misbehaviour inVANETs. *2017 8th International Conference on Information, Intelligence, Systems and Applications, IISA 2017*, *2018-Janua*, 1–6. https://doi.org/10.1109/IISA.2017.8316449

Martinez, F. J., Toh, C. K., Cano, J. C., Calafate, C. T., & Manzoni, P. (2011). A survey and comparative study of simulators for vehicular ad hoc networks (VANETs). *Wireless Communications and Mobile Computing*, *11*(7), 813–828. https://doi.org/10.1002/wcm.859

Mejri, M. N., & Ben-Othman, J. (2014). Detecting greedy behavior by linear regression and watchdog in vehicular ad hoc networks. *2014 IEEE Global Communications Conference, GLOBECOM 2014*, *ii*, 5032–5037. https://doi.org/10.1109/GLOCOM.2014.7037603

Mejri, M. N., & Ben-Othman, J. (2017). GDVAN: A New Greedy Behavior Attack Detection Algorithm for VANETs. *IEEE Transactions on Mobile Computing*, *16*(3), 759–771. https://doi.org/10.1109/TMC.2016.2577035

Menouar, H., Filali, F., & Lenardi, M. (2006). A survey and qualitative analysis of MAC protocols for vehicular ad hoc networks. *IEEE Wireless Communications*, *13*(5), 30–35. https://doi.org/10.1109/WC-M.2006.250355

Miao, L., Djouani, K., Wyk, B. J. Van, & Hamam, Y. (2011). A Survey of IEEE 802 . 11p MAC Protocol. *Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT)*, *29*(1), 113–128.

Pandey, P. (2015). Effect of selfish behavior on network performance in VANET. *Proceedings - 2015 5th International Conference on Communication Systems and Network Technologies, CSNT 2015*, 693–697. https://doi.org/10.1109/CSNT.2015.63

Pattnaik, O., & Pattanayak, B. K. (2017). Performance analysis of MANET and VANET based on throughput parameter. *International Journal of Applied Engineering Research*, *12*(18), 7435–7441.

Potnis, N., & Mahajan, A. (2006). Mobility models for Vehicular Ad Hoc network simulations. *Proceedings of the Annual Southeast Conference*, *2006*, 746–747. https://doi.org/10.1145/1185448.1185615

Raja, L., & Baboo, S. S. (2014). An Overview of MANET : History , Challenges and Applications. *Indian Journal of Computer Science and Engineering*, *3*(1), 121–125. http://www.ijcse.com/docs/INDJCSE12-03-01-144.pdf

Raut, C. M., & Devane, S. R. (2018). Intelligent transportation system for smartcity using VANET. *Proceedings of the 2017 IEEE International Conference on Communication and Signal Processing, ICCSP 2017*, *2018-Janua*, 1602–1605. https://doi.org/10.1109/ICCSP.2017.8286659

Reddy, R. G., & Ramanathan, R. (2018). An Empirical study on MAC layer in IEEE 802.11p/WAVE based Vehicular Ad hoc networks. *Procedia Computer Science*, *143*, 720–727. https://doi.org/10.1016/j.procs.2018.10.443

Ritesh Gupta, & Patel, P. (2016). A Survey on Vehicular Ad hoc Networks. *Engineering and Technology*, *2*(2), 10.

Sadek, F. S., Belkadi, K., Abouaissa, A., & Lorenz, P. (2021). Identifying Misbehaving Greedy Nodes in IoT Networks. *Sensors*, *21*(15), 5127. https://doi.org/10.3390/s21155127

Shan, A., Fan, X., Wu, C., & Zhang, X. (2021). Quantitative study on impact of static/dynamic selfishness on network performance in VANETs. *IEEE Access*, *9*, 13186–13197. https://doi.org/10.1109/ACCESS.2021.3051976

Song, C. (2017). Performance analysis of the IEEE 802.11p multichannel MAC protocol in vehicular ad hoc networks. In *Sensors (Switzerland)* (Vol. 17, Issue 12). https://doi.org/10.3390/s17122890

Sood, M., & Kanwar, S. (2014). Clustering in MANET and VANET: A survey. *2014 International Conference on Circuits, Systems, Communication and Information Technology Applications, CSCITA 2014*, 375–380. https://doi.org/10.1109/CSCITA.2014.6839290

Thambi, S. K., & Sakthivel, N. K. (2015). Real-time MAC-layer selfish misbehavior detection and prevention technique for wireless networks. *Indian Journal of Science and Technology*, *8*(21). https://doi.org/10.17485/ijst/2015/v8i21/71496

Uzma Khan, S. A. and S. S., & Abstract. (2015). A Detailed Survey on Misbehavior Node Detection Techniques in Vehicular Ad Hoc Networks Uzma. *Advances in Intelligent Systems and Computing*, *339*, 11–19. https://doi.org/10.1007/978-81-322-2250-7

van der Heijden, R. W., Dietzel, S., & Kargl, F. (2013). Misbehavior Detection in Vehicular Ad-hoc Networks. *Proceedings of 1st GI/ITG KuVS Fachgespräch Inter-Vehicle Communication (FG-IVC 2013)*, *May*, 23–25. http://www.ccs-labs.org/bib/pdf/vanderheijden2013misbehavior.pdf

Weber, J. S., Neves, M., & Ferreto, T. (2021). VANET simulators: an updated review. *Journal of the Brazilian Computer Society*, *27*(1). https://doi.org/10.1186/s13173-021-00113-x

APPENDIXES

Appendix A: Sample SUMO map.rou.xml fileFew lines of codes are presented here as a sample Since Sumo XML file is bulky


```
<?xml version="1.0" encoding="UTF-8"?>
<!-- generated on 2021-09-25 03:45:07 by Eclipse SUMO duarouter Version 1.10.0
<configuration>
    <input>
        <net-file value="map.net.xml"/>
        <route-files value="map.trips.xml"/>
    </input>
    <output>
        <output-file value="map.rou.xml"/>
        <alternatives-output value="map.rou.alt.xml"/>
    </output>
    <report>
        <ignore-errors value="true"/>
    </report>
</configuration>
-->
<routes>

    <vehicle id="1" depart="1.00">
        <route edges="304004914#1 304004914#2-AddedOnRampEdge 304004914#2"/>
    </vehicle>
    <vehicle id="2" depart="2.00">
        <route edges="506570059 304400428"/>
    </vehicle>
    <vehicle id="6" depart="6.00">
        <route edges="304004914#0 304004914#0-AddedOffRampEdge 304004914#1"/>
    </vehicle>
    <vehicle id="13" depart="13.00">
        <route edges="304004914#0 304004914#0-AddedOffRampEdge"/>
    </vehicle>
    <vehicle id="19" depart="19.00">
        <route edges="604246163#0 604246163#0-AddedOffRampEdge 304008640"/>
    </vehicle>
    <vehicle id="21" depart="21.00">
        <route edges="304004914#1 304004914#2-AddedOnRampEdge 304004914#2"/>
    </vehicle>
    <vehicle id="22" depart="22.00">
        <route edges="304004914#0 304004914#0-AddedOffRampEdge 304004914#1 304004914#2-AddedOnRampEdge 304004914#2"/>
    </vehicle>
    <vehicle id="23" depart="23.00">
        <route edges="304004914#0 304004914#0-AddedOffRampEdge 304004914#1 304004914#2-AddedOnRampEdge 304004914#2"/>
    </vehicle>
</routes>
```

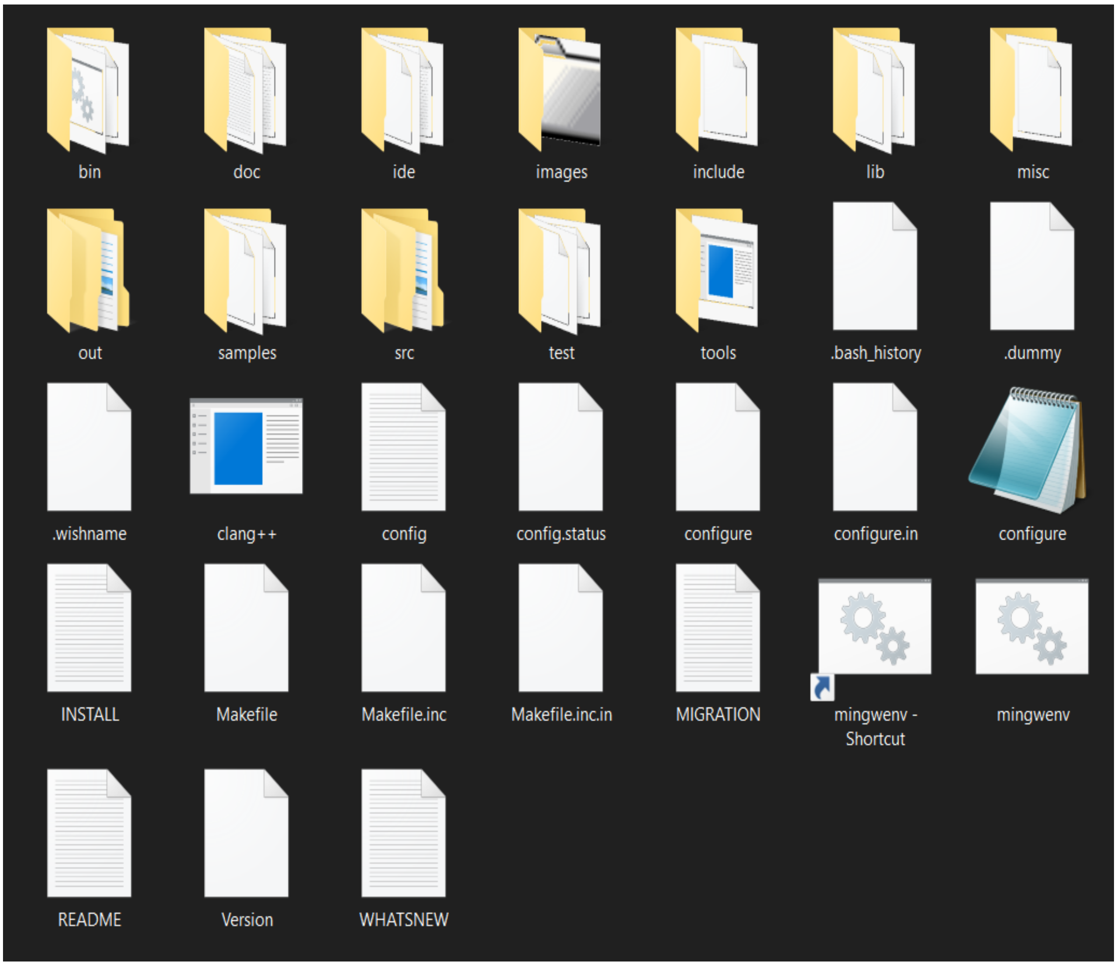
Appendix B: Sample sumoconfig.sumo.cfg File

```xml
<configuration>
    <input>
        <net-file value="map.net.xml"/>
        <route-files value="map.rou.xml"/>
    </input>
    <time>
        <begin value="0"/>
        <end value="10000"/>
        <step-length value="0.4"/>
    </time>
</configuration>
```

Appendix C: Command Prompt for OMNET++ Setups

```
/e/Dev/veins/src/veins/modules/mac/ieee80211p

Welcome to OMNeT++ 5.6.2!

/e/OMNET++/omnetpp-5.6.2$ omnetpp
Starting the OMNeT++ IDE...

/e/OMNET++/omnetpp-5.6.2$ cd ../..

/e$ cd E:/Dev/veins/src/veins/modules/mac/ieee80211p

/e/Dev/veins/src/veins/modules/mac/ieee80211p$
```

Appendix D: OMNET++ Directory



Appendix E: Sample code for parameters assignment for vehicles in **Car.ned** file

```
package vanettutorials.simulations.ieee80211p;
package org.car2x.veins.nodes;
import org.car2x.veins.base.modules.*;
import org.car2x.veins.modules.nic.Nic80211p;
module Car
{
parameters:  // vehicles parameters used by the WC-CSMA/CA
        string applType; //type of the application layer
        string nicType = default("Nic80211p");
        string veinsmobilityType =
default("org.car2x.veins.modules.mobility.traci.TraCIMobility");//type of the
mobility module
        string key;
        bool isAVerifier;
        bool isSelfish;
        bool isNormal;
        bool Vehicle_Status;
        double load;
        double backoffValue;
        double speed;
        double distance;
        double totalWaitingTime;
        double throughput;
        int totalConnectionTrail;
        double cost;
        int cwMin; // minimum contention window
        int cwMax; // maximum contention window

    gates:
        input veinsradioIn; // gate for sendDirect
    submodules:
        appl: <applType> like org.car2x.veins.base.modules.IBaseApplLayer {
            parameters:
                @display("p=60,50");
        }
    nic: <nicType> like org.car2x.veins.modules.nic.INic80211p {
            parameters:
                @display("p=60,166");
        }
          veinsmobility: <veinsmobilityType> like
          org.car2x.veins.base.modules.IMobility
       {
            parameters:
                @display("p=130,172;i=block/cogwheel");
        }
        connections:
               nic.upperLayerOut --> appl.lowerLayerIn;
               nic.upperLayerIn <-- appl.lowerLayerOut;
               nic.upperControlOut --> appl.lowerControlIn;
               nic.upperControlIn <-- appl.lowerControlOut;
               veinsradioIn --> nic.radioIn; }
```

Appendix F: Samples code for VV selection in net.ned file

```
        package vanettutorials.simulations.ieee80211p;
        import vanettutorials.simulations.ieee80211p.Nic80211p;
        import org.car2x.veins.base.phyLayer.BasePhyLayer;
        import vanettutorials.simulations.ieee80211p.Mac1609_4;
        import org.car2x.veins.nodes.RSU;
        import  org.car2x.veins.nodes.Car;

network net extends Nic80211p
{
     Parametrs:
        host host = gethostFromTheNetwork();
        Key = host.key;
        host  verifierhosts[22];
        host  hosts[25];
        int verifierHostsCounter = 0;
        int hostCounter = 0;
  double threshold = 10;
  double load = host.load;
  double distance = host.distance;
  double netWorthyValue = computeNetWorthyValue(host);
  double decisionMatrix = computeDecisionMatrix(load, distance,netWorthyValue);
             }

      host[hostCounter++] = node;

  if(decisionMatrix < threshold){

       host.isAVerifier = true;
       verifierhost[verifierHostCounter++] = host;
               }
     }
       host getHostFromTheNetwork(){

        return null; //get the first node that joins the network
            }

    parameters:
        int numHosts;
        int numRSUs;

        @display("bgb=1100,800");
        @display("bgi=background/matrix");
    submodules:
        host[numHosts]: Host80211p {
            @display("p=126,251");
        }
        rsu[numRSUs]: RSU {
            @display("p=203,284");
        }

    connections allowunconnected:
```

Appendix G: Samples code for motivation Phase in mac1609_4.cc file

```
void announceToNeighbourNodesAboutSelfishNode(Node nodeStatus){
 // send the vehicle status to neighbour vehicles in the network;
}
void updateSelfishNode(Node selfishNode){
    double totalWaitingTime = selfishNode.totalWaitingTime;
    double cost = selfishNode.cost;
    int totalConnectionTrail = selfishNode.totalConnectionTrail;
    double s1, s2 = 0.5;
    double threshold = 8.0;
    double averageWaitingTime = totalWaitingTime/cost;
    double score = (s1 * averageWaitingTime) + (s2 * totalConnectionTrail);

    if(score >= threshold){
    NodeStatus nodeStatus = selfishNode.nodeStatus;
    int credit = selfishNode.credit;

    score = 1;
    credit++;

    nodeStatus.score = score;
    nodeStatus.credit = credit;

    selfishNode.nodeStatus = nodeStatus;
     }//still it is selfish vehicle
}
NodeStatus createNodeStatusForSelfishNode(Node selfishNode){
 NodeStatus nodeStatus;

 nodeStatus.id = selfishNode.key;
 nodeStatus.score = 0;
 nodeStatus.credit = 0;
 selfishNode.nodeStatus = nodeStatus;

 return nodeStatus;
}
class Node {
  public:
    string key;
    bool isAVerifier;
    bool isSelfish;
    bool isNormal;
    double load;
    double backoffValue;
    double speed;
    double distance;
    double totalWaitingTime;
    int totalConnectionTrail;
    double cost;
    NodeStatus nodeStatus
};
class NodeStatus{
   public:
     String id;
     char status; //'N', 'S'
     double score = 1; //by default score is 1 for normal vehicle
     int credit = 1; //by default credit is 1 for normal vehicle

}
```

Appendix H: Total busy time report for CSMA/CA and WC-CSMA/CA

Module	Name	CSMA/CA	WC-CSMA/CA
RSUExampleScenario.node[0].nic.mac1609_4	totalBusyTime	0.011601	0.011601
RSUExampleScenario.node[1].nic.mac1609_4	totalBusyTime	0.011601	0.011601
RSUExampleScenario.node[2].nic.mac1609_4	totalBusyTime	0.011601	0.011601
RSUExampleScenario.node[3].nic.mac1609_4	totalBusyTime	0.081601	0.021601
RSUExampleScenario.node[4].nic.mac1609_4	totalBusyTime	0.011601	0.011601
RSUExampleScenario.node[5].nic.mac1609_4	totalBusyTime	0.011833	0.011833
RSUExampleScenario.node[6].nic.mac1609_4	totalBusyTime	0.012065	0.012065
RSUExampleScenario.node[7].nic.mac1609_4	totalBusyTime	0.012993	0.012993
RSUExampleScenario.node[8].nic.mac1609_4	totalBusyTime	0.013457	0.013457
RSUExampleScenario.node[9].nic.mac1609_4	totalBusyTime	0.013457	0.013457
RSUExampleScenario.node[10].nic.mac1609_4	totalBusyTime	0.089617	0.039617
RSUExampleScenario.node[11].nic.mac1609_4	totalBusyTime	0.014849	0.014849
RSUExampleScenario.node[12].nic.mac1609_4	totalBusyTime	0.006265	0.006265
RSUExampleScenario.node[13].nic.mac1609_4	totalBusyTime	0.005569	0.005569
RSUExampleScenario.node[14].nic.mac1609_4	totalBusyTime	0.005105	0.005105
RSUExampleScenario.node[15].nic.mac1609_4	totalBusyTime	0.004873	0.004873
RSUExampleScenario.node[16].nic.mac1609_4	totalBusyTime	0.008973	0.003973
RSUExampleScenario.node[17].nic.mac1609_4	totalBusyTime	0.005337	0.005337
RSUExampleScenario.node[18].nic.mac1609_4	totalBusyTime	0.005569	0.005569
RSUExampleScenario.node[19].nic.mac1609_4	totalBusyTime	0.005801	0.005801
RSUExampleScenario.node[20].nic.mac1609_4	totalBusyTime	0.010673	0.010673
RSUExampleScenario.node[21].nic.mac1609_4	totalBusyTime	0.005337	0.005337
RSUExampleScenario.node[22].nic.mac1609_4	totalBusyTime	0.005105	0.005105
RSUExampleScenario.node[23].nic.mac1609_4	totalBusyTime	0.009997	0.003997
RSUExampleScenario.node[24].nic.mac1609_4	totalBusyTime	0.003017	0.003017
RSUExampleScenario.node[25].nic.mac1609_4	totalBusyTime	0.003249	0.003249