Non-Cooperative Beaconing Control in Vehicular Ad hoc Networks

Submitted for the Degree of

Doctor of Philosophy

by

Forough Goudarzi

Department of Electronic and Computer Engineering

College of Engineering, Design and Physical Sciences

Brunel University London

February 2017

To the memory of my father

To my family

ABSTRACT

The performance of many protocols and applications of Vehicular Ad hoc Networks (VANETs), depends on vehicles obtaining enough fresh information on the status of their neighbouring vehicles. This should be fulfilled by exchanging Basic Safety Messages (BSMs) also called beacons using a shared channel. In dense vehicular conditions, many of the beacons are lost due to channel congestion. Therefore, in such conditions, it is necessary to control channel load at a level that maximizes BSM dissemination. To address the problem, in this thesis algorithms for adaptation of beaconing to control channel load are proposed.

First, a position-based routing protocol for VANETs is proposed and the requirement of adaptive beaconing to increase the performance of the protocol is indicated. The routing protocol is traffic-aware and suitable for city environments and obtains real-time traffic information in a completely ad hoc manner without any central or dedicated control, such as traffic sensors, roadside units, or information obtained from outside the network. The protocol uses an ant-based algorithm to find a route that has optimum network connectivity. Using information included in small control packets called ants, vehicles calculate a weight for every street segment that is proportional to the network connectivity of that segment. Ant packets are launched by vehicles in junction areas. To find the optimal route between a source and destination, a source vehicle determines the path on a street map with the minimum total weight for the complete route. The correct functionality of the protocol design has been verified and its performance has been evaluated in a simulation environment. Moreover, the performance of the protocol in different vehicular densities has been studied and indicated that in dense vehicular conditions the performance of the protocol degrades due to channel load created by uncontrolled periodic beaconing.

Then, the problem of beaconing congestion control has been formulated as non-cooperative games, and algorithms for finding the equilibrium point of the games have been presented. Vehicles as players of the games adjust their beacon rate or power or both, based on the proposed algorithms so that channel load is controlled at a desired level. The algorithms are overhead free and fairness in rate or power or both rate and power allocation are achieved without exchanging excess information in beacons.

Every vehicle just needs local information on channel load while good fairness is achieved globally. In addition, the protocols have per-vehicle parameters, which makes them capable of meeting application requirements. Every vehicle can control its share of bandwidth individually based on its dynamics or requirements, while the whole usage of the bandwidth is controlled at an acceptable level. The algorithms are stable, computationally inexpensive and converge in a short time, which makes them suitable for the dynamic environment of VANETs. The correct functionality of the algorithms has been validated in several high-density scenarios using simulations.

ACKNOWLEDGEMENTS

I would like to express my gratitude to Professor Hamid Asgari for his invaluable advice and insightful comments and encouragement during my PhD study. I am also thankful to him for his continued support after my PhD study and during the publication of my research outcomes.

I would like to thank Professor Hamed S. Al-Raweshidy and Dr Maysam Abbod for their continuous support of my PhD study and for their motivation and patience. I am also grateful to Dr Thomas Owens who explained the research area of VANETs to me at the beginning of my PhD.

I am thankful to Dr Ali Mousavi for his support and assistance during my research. His willingness and effort to help postgraduate students in the Department of Electronic and Computer Engineering were unfailing.

I am thankful to the members of Wireless Network and Communication Centre at Brunel University London for their friendship and help they gave to me and the pleasant environment they created.

I would also like to thank the members of my thesis examination board, Professor Mehrdad Dianati and Dr Rajagopal Nilavalan, for their brilliant comments and suggestions that improved this thesis.

Furthermore, I would like to offer my special thanks to my family for supporting me throughout this PhD and my life in general. They were always keen to help my progress and encouraged me.

DECLARATION

I confirm that this thesis is an original report of my research except where references are cited, and has been completely written by me. I declare this work has not been submitted for any other degree or professional qualification.

Forough Goudarzi
February 2017

List of Contents

List of Figures

List of Tables

List of Acronyms

ACO	Ant Colony Optimisation
AIMD	Additive Increase Multiplicative Decrease
AODV	Ad hoc On-demand Distance Vector
AODV-ETX	AODV Augmented with the Expected Transmission Count
AQRV	Adaptive and Opportunistic QoS-based Routing in VANET
A-STAR	Anchor-based Street and Traffic Aware Routing
BAHG	Backbone-Assisted Hop Greedy Routing
BSM	Basic Safety Message
CAR	Connectivity-Aware Routing
CBR	Channel Busy Ratio
CCA	Clear Channel Assessment
CCC-MAC	Congestion-Controlled-Coordinator-based MAC
DCC	Decentralized Congestion Control
D-FPAV	Distributed Fair Power Adjustment for Vehicular environments
DSR	Dynamic Source Routing
DSRC	Dedicated Short Range Communication
DYMO	Dynamic MANET On-demand Routing
FABRIC	Fair Adaptive Beaconing Rate for Inter-vehicular Communications
FCC	Federal Communications Commission
FPAV	Fair Power Adjustment for Vehicular Environments
GPS	Global Positioning System
GPSR	Greedy Perimeter Stateless Routing
GSR	Geographic Source Routing
GyTAR	Improved Greedy Traffic Aware Routing
HLAR	Hybrid Location-based Ad hoc Routing
iAODV	Irresponsive AODV
IDR	Information Dissemination Rate

IGRP	Intersection-based Geographical Routing Protocol
INTERN	Integration of Congestion and Awareness Control
LAN	Local Area Network
LIMERIC	Linear Message Rate Integrated Control
MAC	Medium Access Control
MANET	Mobile Ad hoc Network
MAR-DYMO	Mobility-aware Ant Colony Optimisation Routing DYMO
MAZACORNE	Mobility Aware Zone based Ant Colony Optimisation Routing For VANET
NE	Nash Equilibrium
NORAC	Non-Cooperative Beacon Rate and Awareness Control
NOPC	Non-Cooperative Power Control
OBU	On Board Unit
OFDM	Orthogonal Frequency Division Multiplexing
EGSR	Efficient Geographic Source Routing
OLSR	Optimised Link State Routing
OMNeT++	Objective Modular Network Test bed in C++
OSI	Open System Interconnection
PARC	Power and Rate Control
PFQ-AODV	Fuzzy Constraint Q-Learning Algorithm Based on AODV Routing
PHY LAYER	Physical Layer
POCA	Position Aware Reliable Broadcasting
PULSAR	Periodically Updated Load Sensitive Adaptive Rate Control
QoS	Quality of Service
R2V	Roadside-to-Vehicle
RBVT	Road-Based Traffic Aware Routing
RREP	Routing Reply
RREQ	Routing Request
RSU	Roadside Unit

SBCC	Statistical Beaconing Congestion Control
SBCC-C	Channel-Busy-Time-based SBCC
SBCC-N	Neighbour-based SBCC
STAR	Spatial and Traffic-Aware Routing
SUMO	Simulation of Urban Mobility
TACR	Trust Dependent Ant Colony Routing
TDMA	Time Division Multiple Access
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VACO	Vehicular Routing Protocol Based on Ant Colony Optimisation
VANET	Vehicular Ad hoc Network
VC	Vehicular Communication
VNAODV	Virtual Node AODV
VNAODV+	Virtual Node AODV+
VNLayer	Virtual Node Layer
WAVE	Wireless Access in Vehicular Environment
WSM	Wave Short Message
WSMP	Wave Short Message Protocol

Chapter 1

Introduction

Vehicular Ad hoc Network (VANET) is an emerging technology that aims to provide wireless communication between moving vehicles as well as between vehicles and infrastructure stations. Each vehicle is equipped with an On-Board Unit (OBU) to provide this communication. The main motivation to use this technology is its potential safety applications. Vehicles exchange status information, such as speed, acceleration, and braking status, to increase safety and reduce accidents. Diverse non-safety applications are also expected for VANET, ranging from transportation efficiency to commercial and convenience applications, to provide road information and entertainment for road travellers to make their journeys more pleasant.

VANET is a special type of traditional Mobile Ad hoc Network (MANET), where the vehicles are the mobile nodes. However, it has characteristics that differentiate it from MANET. When employed, VANET will be the largest MANET ever implemented in terms of the number of network nodes and the geographical extent. Due to the high mobility of the nodes, the network topology and connections between nodes experience rapid and frequent changes. On the other hand, the network nodes' movement is restricted by the road topology and the requirements to follow road signs and respond to other moving vehicles. Unlike MANET, there is no limitation with respect to energy or processing capabilities. Vehicles can be equipped with an adequate number of computational resources, such as processors, memories, and Global Positioning System (GPS).

Regarding the distinctive features of VANET, challenges emerge to develop novel protocols and techniques for such a dynamic and massive network. As the network size is huge both spatially and in terms of the number of nodes, centralized approaches might be impractical for many applications. Although the future of the network might not be completely ad hoc [1], infrastructure-less and autonomous [2] solutions are attractive as they place the decision-making burden on the individual nodes and can

off-load the infrastructure traffic and enhance protocols scalability. The network must have capabilities of self-organization and survivability [3] to be able to continuously provide its services. The dynamic nature of the network in terms of traffic distribution, bandwidth demand patterns, and channel and network conditions requires techniques that are adaptive to the highly dynamic network behaviour. The other challenge is that all these requirements should be met using scarce network resources (e.g., bandwidth) and disruptive communication channels.

A problem that is very likely to happen in city environments is channel congestion due to periodic broadcasting of Basic Safety Messages (BSMs). Channel congestion degrades the performance of the network protocols and services as well as data routing. BSMs which carry the status information of vehicles are broadcast periodically using a shared channel. The performance of many applications of VANETs relay on every vehicle obtaining fresh information on the status of its surrounding vehicles. Therefore, it is crucial to have high BSMs dissemination rate. In dense vehicular conditions, many of the BSMs are lost due to channel congestion. In such conditions, it is necessary to control channel load at a level to maximize BSM dissemination. Current beaconing congestion control mechanisms suffer the following problems:

- Most of these mechanisms rely on the exchange of extra information in beacons over one or two hops to obtain fairness or to operate. Such mechanisms:
    - consume the network bandwidth and are error-prone due to the loss of information.
    - lose fairness if the scenario is extended to more than the range that the information is exchanged.
    - might reduce the beacon rate of vehicles that have no contribution to the congestion.
- The safety application requirement has not been addressed or if it has been stated, two processes of congestion control and application requirements work separately and thus, there is no guarantee that the channel load remains below a desired level.

Another example of necessity of new protocols for VANETS is the problem of data routing. Diverse applications have been defined for VANETS that make use of Vehicle

to Vehicle (V2V), vehicle to road-side units (RSUs), and Vehicle to Infrastructure (V2I) communication technologies [4]-[6]. These applications range from infotainment, such as media downloading, to safety applications, such as collision risk warning. The performance of the applications that forward messages in a multi-hop manner, depends on how efficiently the routing of data takes place in the network. Routing of data depends on the routing protocols used in the network. VANET is characterized by potentially highly mobile nodes, subject to the constraints of road topology and an unbounded network size. These characteristics make conventional routing protocols inappropriate for VANET.

Routing protocols face challenges in VANET. Regarding the large size of the network, these protocols should be scalable. Scalability means that, by increasing the number of nodes, an increase in overhead does not prevent efficient working of the protocol. As the wireless channel is shared between vehicles, the aggregated usage of the channel in areas with dense vehicular traffic is very high, leading to a high packet collision and degradation of performance of the protocols. Traditional routing schemes are not sufficiently scalable to provide effective services because of the overhead created by the large number of flooded control messages and the computation and memory requirements of the large routing tables. City structures, such as buildings, block communications between vehicles; thus, routing protocols that work well in highways might encounter problems in cities [7].

Position-based routing seems to be a very promising routing method for VANET but still has problems when they are used in city environments. As greedy forwarding mechanism used in position-based routing protocols do not consider the vehicular traffic information in selection of forwarder nodes, these protocols are frequently forced to run in perimeter mode [8]. Working in this mode reduces the protocols performance. There are protocols that obtain vehicular traffic information by using traffic sensors or road side units at every junction which seems impractical at least at the beginning of using of VANET.

To provide solutions for challenges in VANET, in this thesis, tools were used that can exhibit appropriate properties: Ant Colony Optimisation (ACO) for data routing and game theory [9], [10] for congestion control. Biologically inspired algorithms

offer desirable attributes such as self-organization, scalability, and resilience to failure [3], [11] that make them well suited to address challenges in VANET. Among bio-inspired algorithms, ACO has been widely used for routing in networks. Biological ants can find the shortest path to food sites; this ability has inspired considerable work in the specific domain of network routing. The rationale behind this interest is that the problem of finding the optimised route in a network can be defined as a shortest path problem, while the weight of the edges are dynamic values depending on delay, traffic, or bandwidth.

The second tool, game theory, is a mathematical technique aimed at modelling situations in which decision makers must take actions that have mutual, possibly conflicting outcomes. Game theory was widely applied to wireless communications in areas such as power control [12]-[14], data rate and congestion control [15]-[17], security [18] and load balancing [19]. The underlying motivation is that it can provide scalable, robust, and autonomous solutions [20].

1.1 Aim and Objectives

The aim of this thesis is to propose beaconing congestion control algorithms that can control channel load at a desired level. The algorithms should possess key characteristics of congestion control mechanisms such as fairness in resource allocation, stability and short convergence time. The other desired characteristic is that the algorithms should be able to allocate the scarce bandwidth to vehicles based on their safety application requirements or dynamics. All these goals should be met with as low overhead as possible.

1.2 Contributions

The contributions of this research are outlined below.

1.2.1 Efficient Geographical Source Routing for City Environments

To solve the traffic-blindness of position-based routing, in this thesis ant colony optimisation has been employed to select the streets that have optimum traffic condition for data routing. Small control packets called ants are launched by the vehicles at junction areas. Using information included in the ant packets, every vehicle obtains information on the traffic of surrounding streets and can calculate a weight for the street segments.

To compute the best path for data routing, every vehicle considers a graph of the city map so that junctions are vertices of the graph and the street segments between two junctions are edges of the graph. The weight of every edge of the graph is proportional to its length and network connectivity condition of the street provided by ant packets. Whenever a vehicle wants to send a packet to a destination, it initially adds two vertices, which correspond to the source and the destination, to the graph. Then, it computes the shortest path using Dijkstra's algorithm, adds the ordered list of junctions (anchor points) to the packet header, and then sends it. Between junctions, vehicles forward the packet using greedy forwarding, and at junction areas the packet is forwarded towards the next junction that has been included in the packet header.

1.2.2 Non-Cooperative Beacon Rate and Awareness Control Protocol

A protocol for beacon rate adjustment has been developed so that, the bandwidth usage is restricted to a desired level. The proposed protocol employs non-cooperative game theory. Interestingly, non-cooperative games do not rely on communication between nodes. Every node decides individually, and the system ends up at an equilibrium point. In a wireless network, this is a desirable characteristic because it results in bandwidth saving. For the protocol, the existence and uniqueness of Nash Equilibrium (NE) is proved, and the condition for the stability of the NE is derived mathematically. A distributed mechanism is proposed to find the equilibrium point of the game.

1.2.3 Non-Cooperative Beacon Power Control Protocol

As the problem of channel congestion due to beaconing activity can be addressed by beacon power adjustment, a protocol based on a non-cooperative game is proposed as the solution. It is proven that a unique NE exists for the game. An algorithm is presented to find the equilibrium point in a distributed manner, and the stability and convergence of the algorithm have been validated using simulation. This approach differs from previous works in this area for two main reasons: First, the fairness is obtained without exchanging control information between nodes, which results in bandwidth saving. The fairness in this protocol is obtained based on the fairness concept of the NE. Second, weighted fairness in power allocation is achieved, which is useful to meet application requirements.

1.2.4 Joint Beacon Rate and Power Control Protocol

In very dense traffic conditions, decrease in both beacon rate and power might be required to control the bandwidth usage at an appropriate level. Joint beacon rate and power control problem is modelled as a non-cooperative game in which the strategy space of the players (vehicles) is two dimensional. The existence and uniqueness of NE of the game are proved mathematically. The protocol is stable and converges to the NE from any initial rate and power and it can provide fairness in power and weighted fairness in rate. Weighted fairness is useful when, in a congested situation, different vehicles require different beaconing rates. As the protocol has per-vehicle parameters, every vehicle can control its share of the bandwidth while the whole usage of the bandwidth is controlled at a desired level.

1.3 Thesis Structure

This thesis is presented in a further six chapters:

Chapter 2 states a brief description of vehicular network concepts, technology and standards that are related to this thesis. As for this research, a simulator is used to

evaluate the performance of the developed protocols, in this chapter a comparison between four open source simulators that are widely used for simulation of VANETs is presented. Then a simulator has been selected for experiments in this thesis. Related work in beaconing congestion control and routing protocols in VANETs is reviewed. A short review of ACO and game theory that are the analytical tools used in this thesis is also presented.

Chapter 3 presents an Efficient Geographic Source Routing (EGSR) protocol for city environments. The performance of this protocol has been compared with other protocols using simulations.

Chapter 4 presents a non-cooperative mechanism for beacon rate and awareness control for VANETs. The existence and uniqueness of the equilibrium point of the mechanism are proved. Then, the proposed mechanism is compared with other rate control mechanisms using simulation. Furthermore, it is indicated that the mechanism has the ability of awareness control.

In Chapter 5, a non-cooperative mechanism for beacon power control is developed.

In Chapter 6, joint control of beacon power and rate is formulated as a non-cooperative game. The existence and uniqueness of the equilibrium point of the game are proved. Then it is indicated that by appropriate selection of the parameters of the algorithm, weighted fairness in rate and fairness in power are achieved while the congestion is controlled at a desired level.

Chapter 7 concludes the thesis.

Chapter 2

Background

This chapter provides a brief description of VANETs technology and standards that are related to this thesis and knowledge of them is required to find solutions, propose protocols and conduct experiments based on simulation. In addition, beaconing congestion control and routing protocols in VANETs, which are the problems studied in this thesis are reviewed. The tools used in this thesis to develop protocols, ACO and game theory, are also reviewed in this chapter.

There are three major types of vehicular communications (VC) in VANETs [21]. 1) Vehicle-to-Vehicle (V2V) communication: This type of communication is completely distributed without involving any centralized coordinator. Vehicles through OBUs create a MANET and communicate with each other. If a direct wireless connection is available between two vehicles they communicate through a single-hop V2V communication, otherwise a routing protocol is used to forward packets over a multi-hop V2V connection to deliver them to the destination. 2) Roadside-to-Vehicle (R2V) communication: RSUs are fixed network nodes that can provide special applications or form vehicle-to-infrastructure communication (V2I). Utilization of RSUs can also increase the connectivity of the network. 3) Infrastructure-to-Vehicle (I2V) communication: RSUs can be connected to the infrastructural networks or to the Internet, enabling vehicles to access the infrastructure network. Vehicles can also communicate with other hosts for non-safety applications, using communication to the cellular radio network.

2.1 VANETs Technology

In this section technologies related to vehicular communications are reviewed.

2.1.1 VANETs Spectrum

A Wireless Access in Vehicular Environments (WAVE) system is a radio communication system intended to provide seamless, interoperable services to transportation [22]. WAVE standards developed by the IEEE aimed at setting out specifications, procedures and guidelines for communication between vehicles and roadside infrastructures, offering transportation safety, user comfort and convenience. The term Dedicated Short Range Communications (DSRC) is used to refer to the radio spectrum or technologies associated with WAVE. The US Federal Communications Commission (FCC) has allocated 75 MHz of spectrum for DSRC communication, from 5.850 GHz to 5.925 GHz, which is commonly referred to as the 5.9 GHz band. The lower 5 MHz is reserved as a guard band. The remaining 70 MHz is divided into seven 10 MHz channels.

2.1.2 Protocol Stack

The components of the DSRC protocol stacks are illustrated in Figure 2.1 along with the well-known Open Systems Interconnection (OSI) protocol stack for comparison. For simplicity, the functions of the three upper layers of the OSI stack, are usually referred as the application layer.

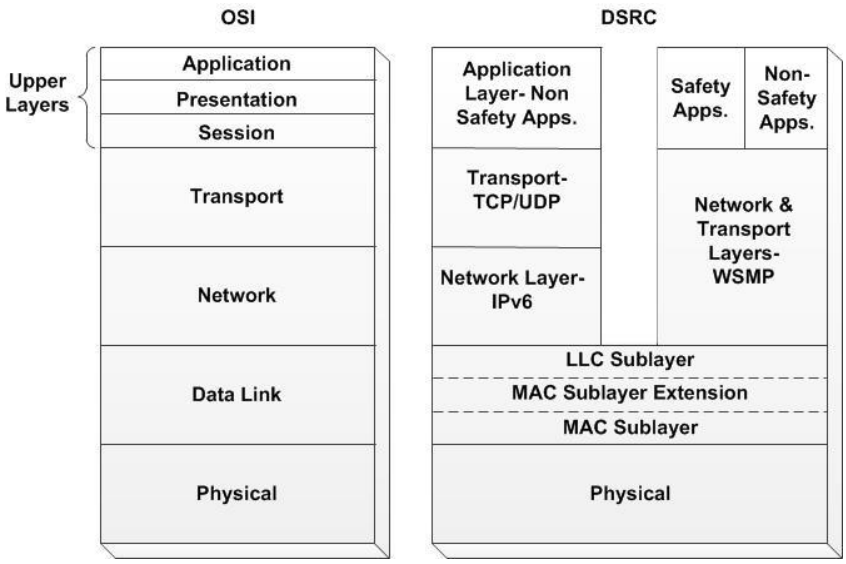

Figure 2.1 DSRC and OSI protocol stack [23]

At the PHY layer and the Medium Access Control (MAC) sublayer of the data link layer, DSRC exploits IEEE 802.11p, more precisely; the well-known IEEE 802.11[24] wireless Local Area Network (LAN) standard has been amended to support DSRC. In the physical layer, it deploys the Orthogonal Frequency Division Multiplexing (OFDM) technique, which was initially added to 802.11 in the 802.11a amendment. An IEEE 802.11 device implementing the OFDM 10 MHz channel spacing can transmit and receive data at rates of 3, 4.5, 6, 9, 12, 18, 24 and 27 Mb/s. Data communication capabilities at data rates of 3, 6 and 12 Mb/s are mandatory, support for the other rates is optional [24]. Most DSRC testing in the U.S. has utilized the 6 Mb/s configuration because it provides an acceptable balance between channel load and signal-to-noise requirement [25].

The Logical Link Control (LLC) sublayer of the link layer uses the established IEEE 802.2 standard. In the upper parts of the stack and MAC sublayer extension, DSRC utilizes a class of standards defined by the IEEE 1609 Working Group. MAC sublayer extension supports the multichannel functionality by employing IEEE 1609.4 [26], which defines how a device switches among DSRC channels.

Above the link layer, the IEEE 1609.3 [27] networking services standard defines two protocol stacks (sharing the same lower stack at the physical and data link layers): the WAVE Short Message Protocol (WSMP) and the standard Internet Protocol Version 6 (IPv6). The WSMP has been designed to support network and transport layer functions in a wireless vehicular environment. Packets sent by WSMP are referred to as WAVE Short Messages (WSMs). One of these messages is BSM, by which vehicles periodically broadcast their key status information such as position, speed, acceleration and heading.

Application layer protocols could support both safety and non-safety applications. Non-safety applications can be provided by the Internet and DSRC communication system [23]. IEEE 1609.11 [28] is an example of a non-safety application layer standard. IEEE 1609.11 Over-the-Air Electronic Payment Data Exchange provides a common interoperable service for device identity, payment authentication and payment data transfer. The IEEE 1609.2 [29] Security Services standard is another standard from the 1609 class of standards, which defines encryption and authentication techniques for WSMs.

2.1.3 Power Limits

The shared DSRC spectrum is used by OBUs and RSUs, consequently, there is the potential of interference between the signals of different transmitters. If the interferers are transmitting on the same channel, this is known as co-channel interference. If the interferers are transmitting in different (spectrally near) channels this is known as cross-channel interference. The probability that a given transmission suffers a collision in a given area is proportional to the number of potential interferers in that area, which is itself proportional to the transmit power applied by each device. The FCC regulates the transmit power of DSRC devices to control both co-channel and cross-channel interference. The FCC defines four groups, A–D, to classify devices according to the maximum allowed transmit power at the antenna, as shown in Table 2.1. Each class has a desired transmission range. Devices participating in V2V safety will normally fall in Class C [25].

Table 2.1 FCC device classification [25]

Device Class	Max. Output Power (dBm)	Communication Zone (meters)
A	0	15
B	10	100
C	20	400
D	28.8	1000

2.1.4 VANETs Simulation

VANETs protocols and services must be tested before they can be used in the real world. To evaluate them, real experiments are expensive and highly complex as they need to include many kinds of situations. Hence, they practically can only be undertaken for the purposes of verification. For primarily design purposes, computer modelling and simulations are vital in studying VANETs. Computer modelling and simulation is the reproduction of the operation of a system that allows engineers to

study and analyse complex systems. Thus, an important issue in studying VANET is the selection of a suitable simulator.

Network simulators are usually designed in accordance with the OSI stack model [30]. The validity of a simulation is dependent on the proper modeling of these layers. In addition, different networks need different models for their layers, hence a simulator might show a valid result for a particular network but it might not be good for other networks. Therefore, to have accuracy in simulation results a simulator must be selected that uses models for different layers that best describe the network under consideration, which will be a vehicular ad hoc network herein.

The standards for different layers of a VANET device were reviewed in the above sections. In [31] the necessity for using 802.11p model for VC has been demonstrated by simulating VC with three different MAC and PHY layer models: 802.11p, 802.11b and a tuned 802.11b to match the 802.11p model. It was indicated that in scenarios with very low channel usage the differences in the performance of applications when using different IEEE 802.11 MAC layer models are negligible. However, with increasing channel load, the performance of the applications varies greatly. It was also demonstrated that changing the parameters of a Wi-Fi model to comply with the 802.11p settings does not improve the accuracy of the model.

An important issue in the modeling of the PHY layer is the propagation model employed by the simulator. These models could be very simple, like the free space propagation model or very complicated. The performance and efficiency of routing protocols are thoroughly influenced by the selection process of the neighbouring nodes that are candidates to disseminate the information from source to destination, the density of neighbouring nodes and the radio link reliability. These operational parameters are heavily influenced by the received signal level in VANETs, so it is crucial to model the radio propagation conditions appropriately and accurately to understand, design, optimise and evaluate VANETs routing protocols. The impact of the radio channel modeling on the performance of VANET communication protocols has been evaluated in [32]. That work indicated the considered radio propagation models strongly influence the simulated VANETs routing performance and the simulated protocol’s operation.

Hence, to select a simulator for VANETs, a comparison between the features of some widely used network simulators is made in the following paragraphs. ns-2, ns-3, OMNeT++ [33] and JiST [34] are the open source network simulators that have been most used over the period 2009–2011 for simulation studies of VANETs [35]. The characteristics and protocol stack models used in these open source network simulators are the followings:

- **Radio propagation model:** Several radio propagation models exist in ns-2 but none of them is designed for vehicular communications [36]. ns-3 has 15 radio propagation models in its model library, from simple free space, two rays, lognormal, to more complicated propagation models [37]. In [38], different propagation models for VANETs have been studied and it was concluded that the Nakagami model [39] is a suitable and realistic radio propagation model for VANET in highway and urban scenarios. This model exists in the ns-3 library. The two main MANETs frameworks, INET and MiXiM, benefit from a propagation model specially designed for the V2V channel [36].

- **802.11p MAC/PHY layer**: A model for 802.11p has been designed for ns-2 but it has not been included in the official release of ns-2 and it needs to be applied as a patch [36]. ns-3 did not provide a model for 802.11p in its released library of models in May 2013 [37]. Vehicles in Network Simulation (Veins) is an OMNeT++ framework that completely supports 802.11p. JiST/SWANS uses the 802.11b model for its MAC layer [34].

- **Performance and Scalability:** In [40], [41] the performance of ns-2, ns-3, OMNeT++ and JiST have been compared with different numbers of nodes (up to 2000 nodes in [41] and 3035 nodes in [40]). ns-3 was reported to be the most efficient in memory usage, then OMNeT++, ns-2 and JiST. ns-2 memory and CPU consumption do not allow practical usage of the simulator for scenarios with more than a few hundred nodes [36]. In addition, ns-2's computation time increased rapidly with increasing numbers of nodes, which means ns-2 is not scalable. The CPU utilization of ns-2 and ns-3 was very similar (5% variation) and was much higher than that of OMNeT++ (no data on CPU utilization was provided for JiST).

- **Traffic Simulators:** mobility models provided by network simulators are not usually suitable for VANETs research. However, there are traffic simulators that can generate vehicle traces for network simulators. VanetMobiSim is based on Java and can generate movement traces in different formats, supporting different simulators, including ns-2 [42]. STRAW (STreet RAndom Waypoint) provides accurate simulation results using vehicular mobility models for real U.S. cities based on the operation of actual vehicular traffic. STRAW's current implementation is written for the JiST/SWANS simulator and its mobility traces can be directly used by it [42].

  SUMO (Simulation of Urban MObility) [43] is an open source, microscopic road traffic simulation package designed to handle large road networks. Its main features include collision free vehicle movement, different vehicle types, single-vehicle routing, multilane streets with lane changing, junction-based right-of-way rules, a hierarchy of junction types, a graphical user interface (GUI) and dynamic routing. SUMO can manage large environments, up to 10000 streets. However, because SUMO is a pure traffic generator, its generated traces cannot be directly used by the available network simulators, which is a serious shortcoming. Some frameworks have been designed to import traffic data from SUMO to network simulators. For example, Veins couples SUMO and OMNeT++, and iTetris is a framework that integrates SUMO and ns-3 [36].

- **Upgrade and Maintenance:** JiST/SWANS' most recent version was released in 2004. The ns-2 project is no longer active after 2010. ns-3 and OMNeT++ are active projects.

In conclusion, OMNeT++ is a reasonable choice for the simulation of VANETs because, it has models that best describe VANETs characteristics. However, ns-3 is an active project and might add these kinds of models to the simulator; but currently, OMNeT++ is the best open source simulator for VANETs. OMNeT++ is scalable and more user-friendly than the other open source simulators that could be used. ns-2 and JiST/SWAN are not active projects, this reason is enough not to select them.

2.2 Beaconing Congestion Control

The primary motive for using VANETs is to enhance safety in transportation. This goal is achieved by messages exchanged among vehicles. One of the most important messages is the BSM, also called beacons, which includes vehicle status data such as position, speed, and acceleration. Frequent broadcast of BSM provides awareness about nearby vehicles. Thus, beaconing with the highest rate (10 Hz) is desirable from the viewpoint of providing fresh information and ensuring that vehicles have high levels of awareness. However, in dense traffic environments, a high beaconing rate increases packet collision, which reduces the number of received beacons, and thus, reduces vehicles' awareness of surrounding vehicles. In addition, channel congestion reduces the performance of event-driven messaging due to high collision rate. The maximum beacons are received, when the Channel Busy Ratio (CBR), is around 0.65 [44]. Therefore, considerable efforts have been dedicated toward designing congestion control mechanisms to limit the channel load around 0.65 for VANETs by controlling either beacon rate or transmission power or both [45]-[56]. In chapter, we consider beacon rate control.

Congestion control is an important issue in computer networks because congestion degrades network performance. The key characteristics that have traditionally been used to evaluate congestion control mechanisms include efficiency in keeping channel load under a desired level, fairness among network users, convergence time of the mechanism, and oscillation size [57]. Given the special features of VANETs, the requirements in terms of these characteristics are distinctive; sometimes, even new requirements emerge. Congestion control in VANETs should work in a distributed manner without involving any infrastructure. The overhead due to the control mechanism should be as low as possible. Owing to the highly dynamic nature of such a network, convergence time of the congestion control mechanism should be short.

Several notions of fairness have been defined in computer networks [58]. In this thesis whenever the term fairness has been used it refers to the condition that all network users, utilize the same beacon rate and/or power. For this definition of fairness Jain Index [59] is appropriate indicator of the notion. We also use the term weighted fairness to refer to the condition that network users, utilize network resources proportional to some weighting factors. Fair access to the wireless channel for

vehicles, creates awareness with respect to surrounding vehicles in a fair manner which is necessary to make the safety applications of VANETs reliable.

Regarding fairness in VANETs, in many works, fairness simply has been considered the condition in which all vehicles in a congested condition should use the same beacon rate apart from their dynamics. Even with this simple definition, several protocols fail to achieve fairness [53]. Moreover, such fairness cannot meet awareness and safety application requirements in VANETs [60], [61]. A scenario in which there is congestion on one direction of a highway and free flow on the other direction exemplifies that vehicles have different beaconing requirements. Vehicles on both sides of the highway might experience the same CBR, but those running at higher speeds should have higher beaconing rates than stationary vehicles to create a high level of awareness. Actually, in a congested scenario, when the overall bandwidth is inadequate to allow vehicles to transmit beacons with the highest allowed beaconing rate, the bandwidth should be shared among vehicles proportional to their dynamics or requirements while maintaining the CBR below the desired level.

Generally, approaches in resolving the unfairness of congestion control mechanisms are based on piggybacking excess information in beacons (such as current beacon rate or experienced CBR) and propagating it over one or two hops [50], [53], [54], [60], [62], [63]. Broadcasting such information both creates overhead and makes the mechanism error-prone owing to channel fading and loss of information. In addition, when the size of a congested area is larger than the range that this information is shared the unfairness problem appears again.

2.2.1 Congestion Control

Congestion in a computer network refers to the condition in which the load in the network is greater than the network resources [64]. Such a condition leads to high packet loss, delay, or both. Congestion control algorithms are resource management techniques that recover a network from this condition. They include two main mechanisms: a) congestion detection, and b) congestion mitigation.

– Congestion in a network can be detected in four ways [64]:

- MAC queue length: When the queue occupancy exceeds a certain level, the algorithm infers a congestion state. This is a simple method that does not require many resources from the node. However, it is dependent on the MAC protocol. If the MAC protocol is not efficient, it is possible for collisions to exist in the medium and not be detected by this method.
- Wireless channel load: In this method, the packet load in the medium is measured and if it is higher than a threshold, actions are taken. Their limitation is that they cannot react if buffers are fully occupied and start dropping packets.
- A combination of queue length and channel load methods.
- Packet transmission time: These methods count packet service time and packet inter-arrival time (or a combination of them) and, if they exceed a limit, infer that congestion is imminent. These methods may detect and trigger congestion control falsely. For instance, it is possible to have packet loss due to reasons other than congestion, such as environmental or physical causes.

– Congestion mitigation can be implemented in the following ways [64]:
  - Traffic control: This method involves reducing the traffic that is injected into the network, and can be implemented by reducing the data rate or increasing the contention window of MAC, or by decreasing transmission power.
  - Resource control: With this method, the traffic is routed to other paths that are not congested.

Due to the one-hop broadcast nature of BSMs and the lack of acknowledgment, some of the above mechanisms are not applicable in detecting or mitigating congestion due to beaconing. In most of the congestion control mechanisms for VANETs, channel load is used to detect congestion. Channel load can be measured based on the IEEE 802.11 Clear Channel Assessment (CCA) function [24].

2.2.2 Related Work in VANETs

The existing approaches for congestion mitigation in VANETs can be classified as rate adaptation, power adaptation, modifying CSMA/CA parameters, or a combination of them.

2.2.2.1 Rate Adaptation

Periodically Updated Load Sensitive Adaptive Rate Control (PULSAR) protocol [62] uses Additive Increase Multiplicative Decrease (AIMD) technique to adapt the beacon rate of vehicles. Vehicles communicate the measured CBR as well as the maximum CBR they received from the nodes in their neighbourhood. If the maximum reported CBR is higher than a threshold level, the rate is decreased by a multiplicative factor; otherwise it is increased by an additive factor.

Linear Message Rate Integrated Control (LIMERIC) [49] is a beacon rate control algorithm in which, each vehicle measures CBR and then updates its rate proportional to the error between the desired CBR and the measured value. To ensure the convergence of the algorithm in dense traffic situations, a gain saturation approach is introduced; if the magnitude of the updates exceeds a specified threshold, the updates will be limited to the magnitude of the threshold. The algorithm does not require exchange of control information between vehicles. It assumes that all the vehicles measure the same CBR which can be unrealistic, even when all the vehicles are in the communication range of each other, due to channel fading. The advantage of LIMERIC over the approaches that use binary control (such as PULSAR) is that it avoids the limit cycle behaviour (oscillation around the target level) that happens in binary control. Both LIMERIC and PULSAR suffer unfairness in rate allocation [53].

To solve the unfairness problem in beacon rate control, mean-checked threshold-based control is proposed in [53] which relies on piggybacking excess information on beacons. Every vehicle includes its current beacon rate in its beacons. Before applying any change to the rate, a vehicle compares its rate to the average rate of its immediate neighbouring vehicles to avoid too much difference between the rates of neighbouring vehicles. A similar mechanism is used in [54] in which vehicles exchange their state information instead of beacon rate. Three different states are defined and in each state, vehicles use different transmitter power levels, beaconing rates, receiver sensitivities, and physical layer bit rates. Numerical values of 0, 1, and 2 are assigned to the states. Each vehicle piggybacks its current state number on its beacons. The average value of states of neighbouring vehicles is used as a criterion for changing state.

In Integration of Congestion and Awareness Control (INTERN) protocol [63], the safety application sets the minimum and maximum rate and required power for

transmission of beacons. Then every vehicle adjusts its beacon rate within the specified interval. To achieve fairness, vehicles exchange information on the measured CBR and their excess rate with respect to the minimum they use, over two hops. This is similar to the approach used in PULSAR. Since each vehicle sets a minimum beacon rate that is required by an application, the aggregated channel utilization may violate the maximum desired level of CBR.

The authors in [65] suggest that the beacon frequency should be adjusted dynamically based on the current traffic situation, so that appropriate accuracy of the status of vehicles is maintained. They propose a situation-adaptive beaconing, in which the beaconing rate depends on a vehicle's own movement and the movement of surrounding vehicles. The latter is divided into two categories: macroscopic traffic situations, such as vehicle density; and microscopic traffic situations, such as relative vehicle speed. The paper further discussed that every framework should also consider weighting of the schemes where microscopic traffic-situation-based adaptation has the greatest impact. However, no approach has been developed for the proposed scheme.

Fair Adaptive Beaconing Rate for Inter-Vehicular Communications (FABRIC) algorithm [50] is based on network utility maximization in which every vehicle piggybacks information on the computed Lagrange multipliers and its current beaconing rate in its beacons. Vehicles use this information from their immediate neighbours to update their rates and the Lagrange multipliers. The speed of convergence of the algorithm is dependent on the initial values of the Lagrange multipliers which are not controllable in practice because over time, vehicles change these parameters. Although it has been stated that the algorithm can meet the application requirements, there are no experimental results given in the paper to verify it. An algorithm should have parameters per-vehicle to be able to present this feature, while such parameters do not exist in the algorithm. The other problem with FABRIC is that it has not considered fading and collision in defining constrains in the utility maximization problem. This results in inefficient usage of bandwidth.

2.2.2.2 Power Adaptation

In [46] a centralized approach, named Fair Power Adjustment for Vehicular environments (FPAV), was presented that requires synchronization between nodes.

The algorithm first computes the maximum power level for which the channel load remains under a specified level, and then all the nodes increase their transmission power simultaneously by the same amount, so long as the beaconing network load remains under the specified level. In this algorithm, a formula is used to compute the channel load, in which it is assumed that all beacons have the same size, and that the vehicle density is already known.

In Distributed FPAV (D-FPAV) [48], in order to provide enough information for every vehicle to be able to compute the channel load, excess information is piggybacked in beacons. This information includes the position and power level that every node computes using the FPAV algorithm over multi-hop (maximum carrier scene range). Then, every node utilizes the minimum power level that was computed by the nodes. The problem with this approach is that a large amount of information must be exchanged between nodes, which increases with the number of neighbours. In order to mitigate this problem, the authors proposed piggybacking this information on every tenth beacon, which, of course, affects the speed of the algorithm.

Statistical Beaconing Congestion Control (SBCC) has two variants, Channel-Busy-Time-(CBT) based SBCC (SBCC-C) and Neighbour-based SBCC (SBCC-N) that were proposed in [45]. Each node includes only the power in use in its beacons, which, compared to the previous approach, creates much less overhead. In this work, using a Nakagami-m path loss model, a formula was derived for channel load that is a function of transmission power, average beacon size and rate, and channel parameters including the path-loss exponent and the shape parameter (m). Since channel parameters and communication ranges are not known and change over time, vehicles should periodically estimate them from the information that they receive in beacons. For estimation of the path-loss exponent, a simple path-loss model is assumed for the channel. Then, using the received power, the transmission power that is included in beacons, and the distance between sender and receiver, the path-loss exponent is estimated. For estimation of the shape parameter, the method proposed in [66] was used. Communication range is estimated by a proposed formula. For beacon rate and size, constant values of 10 Hz and 500 bytes, respectively, have been assumed.

SBCC algorithms also need to estimate the density of vehicles in the communication range. For this purpose, SBCC-N obtains the number of neighbour vehicles using a

neighbour table, while SBCC-C estimates the density of vehicles using CBR. Then, based on the derived formula, every node computes the maximum power that will keep the channel load below the maximum accepted level. In this work, the fairness in power assignment was not studied.

2.2.2.3 MAC Adjustment

The congestion control approach in [67] is based on adjusting the minimum contention window size (CW_{min}) for each of the four different access categories (ACs) in MAC. The channel congestion is measured based on length of queue and the number of transmission failures. For the *i*th AC (AC_i), the congestion condition $C_m(i)$ is defined as:

$$C_m(i) = \left(N_{queue}(i) + N_{fail}(i)\right) \Big/ N_{total}(i) \tag{2.1}$$

where $N_{fail}(i)$ is the number of packets transmitted unsuccessfully, $N_{total}(i)$ is the total number of packets the AC_i has generated and $N_{total}(i) = N_{queue}(i) + N_{fail}(i) + N_{sus}(i)$ where $N_{sus}(i)$ is the number of packets transmitted successfully. $C_m(i)$ is measured continuously during specified time intervals. If $C_m(i)$ exceeds a threshold, then the AC should increase its CW by multiplying a scaling factor a = 2. In the broadcasting transmission mode with no successful transmission acknowledgement, the amount of the unsuccessful transmission is substituted with the failure amount of the received packets [51]. On the other hand, if $C_m(i)$ is less than threshold, the AC reduces the CW by dividing by a scaling factor a = 2, until it reaches the initial value. The congestion thresholds are adjusted dynamically with the vehicular network density, and are different for each AC. The optimised congestion threshold was determined by experiment, with the aim of guara0nteeing the transmission chance for the highest priority traffic as much as possible, while restricting the collision probability to remain under a determined constraint value (5%) in the simulation. In this work, fairness was not studied; in addition, it is not clear whether increasing the contention window can meet the acceptable latency for delivering safety messages.

Congestion-Controlled-Coordinator-based MAC (CCC-MAC) [68] is a coordinator-based Time-Division Multiple ACCESS (TDMA) MAC that uses multiple data rate to reduce BSM loss due to congestion. Roads are divided into segments, and vehicles using a digital map are aware of which segment they are in and transmit their beacons

in the time slots that belong to that segment. In each segment, a vehicle is selected as a coordinator to assign time slots for transmission of beacons to other vehicles in that segment to prevent beacon collisions. In normal conditions, vehicles use a 6 Mb/s data rate, but when the load is higher, this rate and a higher data rate are used to shorten the transmission time. Specifically, the data rate is increased until the remaining load is accommodated in the channel. If there is time slot shortage in one segment due to a high number of vehicles, unused slots from neighbouring segments can be borrowed by an intersegment slot transfer mechanism, which is accomplished through request/reply messaging.

2.2.2.4 Hybrid Approaches

In [90], a protocol for adaptive beaconing rate and power based on the dynamics of a vehicular network is proposed. A vehicle increases its beacon rate when it suspects the estimated tracking error of neighbouring vehicles towards its position has increased. For this purpose, in every defined time step, vehicles compute transmission probability based on suspected tracking error on neighbouring vehicles toward its own position in a Euclidean sense. If the suspected error is smaller than a threshold, there will be no transmission. Otherwise, if the suspected error is larger than this threshold, the transmission in that time step occurs with a probability proportional to the magnitude of the suspected error. For the power control, two levels of CBR are defined; CBR_{max} and CBR_{min}. If the CBR measured by each vehicle is greater than CBR_{max}, minimum transmission power is used; if it is lower than CBR_{min}, maximum transmission power is used. Otherwise, the transmission power is selected between the maximum and minimum values using a linear function. In this work, fairness was not studied, and there is no guarantee that the CBR remains below the desired level.

In [55] a vehicle computes a target distance within that, beacons should be received. The target distance is computed based on the vehicle and its neighbours dynamics (speed, acceleration, etc.), system delay and driver reaction time. Then the required power to cover the target distance is found using a lookup table. The beacon rate is adapted in an allowed range to keep channel load under a maximum allowed load. The protocol lack details for implementation and the performance of the proposed approach has not been evaluated.

ETSI [69] proposes several techniques for Decentralized Congestion Control (DCC) which are: 1) transmit rate control, 2) transmit power control, 3) receiver sensitivity control, 4) transmit data rate control, and 5) transmit access control. DCC can be implemented by applying one or combination of several of these techniques. To implement DCC using the mentioned techniques, ETSI defines a State Machine (SM) that has three states: RELAXED, RESTRICTIVE and ACTIVE [70]. The ACTIVE state can have multiple sub-states. In each state vehicles use different transmission or receiver parameters. The transition between the states happens based on the measured CBR. ETSI DCC suffers unfairness and oscillation [54], [71], [72]. In [54] a congestion control scheme that used the techniques 1 to 4 simultaneously was tested and indicated that ETSI DCC is instable and unfair. In [72] by adapting just transmit rate control, the instability of the method was demonstrated. In [71] DCC mechanisms considering a single technique of 1, 2, 3 and when the three techniques are simultaneously active, evaluated. In all of them, instability in the state of vehicles was observed.

2.3 Routing Protocols in VANETs

VANETs have some characteristics that differentiate them from other types of MANETs, including rapid node movement, a large network, and constrained mobility imposed by roads topology. Due to such differences, topology-based MANETs routing protocols, such as AODV [73], OLSR [74], and DSR [75], perform less efficiently in VANETs [7], [76]. Typically, in MANET protocols, when the network nodes move, the established paths are subject to breaking, in which case the routing protocol must dynamically search for other feasible routes. Therefore, using the existing routing protocols for MANETs, maintaining connectivity in VANETs with rapidly changing topology is very difficult. Furthermore, these routing protocols use the broadcast mechanism to find and maintain routes. In VANETs, the movement of vehicles causes the communication links between them to be broken frequently. Such link failures increase the broadcasting and routing control overheads, and lead to degradation of the protocol performance. Therefore, considerable efforts have been dedicated toward

designing routing protocols that can cope with the problems of routing in VANETs. In the following sections, a review of these protocols is presented.

2.3.1 Modified MANET Routing Protocols

Certain modifications to topology-based routing protocols have been proposed [77]-[80] to qualify them for routing in dynamic VANETs environments. In [78], the development of a modified AODV protocol called AODV-ETX is described that, in combination with greedy forwarding, attempts to solve the scalability problem of topology-based routing. AODV-ETX adopts the Expected Transmission Count (ETX) metric [81] instead of the minimum hop count metric used in AODV. In AODV-ETX, all nodes in the network send periodic beacon packets in order to allow nodes to estimate the ETX metric and to identify their neighbours. AODV-ETX also differs from AODV in that it allows intermediate nodes to repair broken routes locally. The combined AODV-ETX and greedy forwarding protocol is called Hybrid Location-based Ad-hoc Routing (HLAR) [78]. HLAR uses greedy forwarding, rather than the broadcast mechanism, to discover or repair a route. When greedy forwarding fails to find a neighbour node closer to the destination, e.g., in a local maximum or when the location information degrades, the protocol reverts to reactive routing and uses broadcast to recover from the situation. A comparison of HLAR and AODV-ETX protocols showed HLAR has higher performance in terms of the packet delivery ratio, and of end-to-end delay and routing overhead.

HLAR does not prevent local maxima in greedy forwarding. It just uses a different approach (broadcast) to recover from that condition. In city environments with non-uniform traffic, such conditions are encountered repeatedly, requiring broadcast for recovery in each instance. The performance of the HLAR protocol has not been tested in such scenarios. The ETX metric in [81] is defined to evaluate the quality of links in a network with stationary nodes, and so using this metric for a network with a short link lifetime does not appear to be an optimal choice. As described in [78], at least ten seconds are required to evaluate the quality of a link. Ten seconds could be more than total lifetime of a link in a VANET with fast-moving nodes.

Virtual Node AODV+ (VNAODV+) [80] is a cluster-based routing protocol in which every cluster has a leader and a backup leader. The whole geographical area of a network is divided into square regions, whose size is chosen so that every node in a given region can, at least, send and receive messages to and from every other node in the neighbouring regions. This protocol attempts to address overhead and route instability issues in AODV in high-mobility scenarios. It alleviates these issues by permitting only cluster leaders to participate in route finding and route repair mechanisms. VNAODV+ is an enhanced version of VNAODV [82], qualified for communication in VANETs. One feature of these protocols is that they introduce a layer between the link layer and network layer, called the Virtual Node Layer (VNLayer), which is responsible for cluster management (leader selection, etc.). The enhanced version of the VNLayer is called VaNetLayer, and creates less overhead for management of clusters, and reduces the time that clusters are out of leader. The layer creates overhead to manage the clusters and the overhead is expected to be directly proportional to the speed of vehicles; however, in the simulation results presented in [80], the speed of the vehicles was not specified.

In the irresponsive AODV (iAODV) protocol [83], the broadcast mechanism for route discovery is replaced with a probabilistic forwarding. Every node retransmits a route discovery packet with a computed probability, reducing the routing control overhead. The transmission probability increases by increasing the distance between sender and a potential rebroadcast and decrease with increasing node density.

In [79], the authors attempted to tune the OLSR configuration parameters, including the intervals of periodic messages (hello interval, topology control interval, etc.) and the hold times for VANET. They defined a cost function that is a weighted sum of three Quality of Service (QoS) metrics: the packet delivery ratio, the network routing load, and the end-to-end delay. They applied five different metaheuristic techniques to minimize the cost function: 1) particle swarm optimisation; 2) differential evolution; 3) a genetic algorithm; 4) simulated annealing; and 5) a random search algorithm. A simulator (ns-2) interacted automatically with the optimisation procedure. The optimisation procedure generated new routing configuration parameters in every run, then invoked the simulation procedure to evaluate them over a defined urban VANET

scenario. After the simulation, the computed cost function is returned to the optimisation procedure.

The problem with this approach lies in offline optimisation. The optimised parameters are obtained by optimisation on a selected VANETs scenario, and therefore cannot produce optimal performance in every scenario. As it is showed, OLSR with new configuration parameters did not exhibit higher performance than the standard configurations (RFC 3626), for all VANETs scenarios. For example, in the scenario used for optimisation, vehicles had speeds between 10 and 50 km/h. Thus, the optimised hello intervals obtained by all five optimisation algorithms were greater than the standard value. It does not appear that these hello intervals could be ideal choices for scenarios with faster vehicles. The other key result is that, in all the experiments, the average length of the routing paths was less than two hops. The question arises as to whether the tuned parameters can work perfectly to find longer paths. In the end, the authors of [42] did not propose a set of optimised configuration parameters for OLSR, since every optimisation technique resulted in different set of parameters, but no one set exhibited performance that is superior to the others, or even to the standard one, in all the experiments.

Fuzzy Constraint Q-Learning Algorithm Based on AODV (PFQ-AODV) [77] is a routing protocol based on AODV, but employs metrics other than hop count to select paths. It uses available bandwidth, link quality and relative vehicle movement to evaluate a given path. The author proposed an approach for calculating these metrics that is independent from lower layer or location information. Each node broadcasts hello messages periodically. Each hello message includes the available bandwidth of the sender node (calculated according to a formula the author proposed) and the addresses of all one-hop neighbours. By receiving hello messages, each node maintains its two-hop neighbour information. Using this information, the nodes calculate the bandwidth factor, mobility factor, and link quality factor for every link to their one-hop neighbours. Then, numerical values (fuzzy values) representing the quality of every link is derived from these three metrics for each. According to the author, the benefits of this approach using fuzzy logic are first that the link status can be evaluated without deriving a complex mathematical model, and second that it is possible to tune the fuzzy membership functions and rules to make the protocol more

suitable for a particular scenario (flexibility). Every node maintains a fuzzy value for every one-hop neighbour and a Q-value, also computed using fuzzy values, for up to two-hop neighbours. The protocol uses the same mechanism as AODV to find a route, with the difference that when a node rebroadcasts a Route Request (RREQ), it attaches its maximum Q-value for the source node to RREQ. The destination node chooses the node that has the maximum Q-value as the next hop for the Route Reply (RREP) message. The next hop node also chooses its next hop according to its Q-table. Upon receiving the RREP message, the source node employs the route indicated by the RREP message. The author compared this protocol with AODV and two other AODV-based protocols with different metrics, and showed that it exhibits higher performance.

The PFQ-AODV protocol [45] is actually AODV with a different metric. Therefore, it faces the problems of AODV for VANETs, for instance, using broadcast to find routes, which creates scalability problems. Its advantage over AODV is that, because it considers the mobility of vehicles to compute the metric, it may select links with longer lifetimes, and so requires less route recovery.

2.3.2 Position-Based Routing Protocols

Position-based routing is an alternative approach for routing in VANETs. These protocols do not select a fixed set of nodes between the source and destination for routing packets, and consequently do not suffer route instability. They also do not require route discovery and management, so they are more scalable and suitable for VANETs than are MANETs protocols. Position-based routing protocols are able to solve such problems as high mobility and transmission delay, because they maintain only local information on their neighbours instead of per-destination routing entries.

In position-based routing protocols, a greedy mechanism, such as Greedy Perimeter Stateless Routing (GPSR) [8], is used to forward packets. With such a mechanism, each node obtains its current location, via, for example, a GPS receiver, and learns the position of its one-hop neighbours by receiving periodic beacon messages. To route packets, a node sends them to the neighbouring node that is nearest to the destination. This mechanism does not require route discovery and management, and thus is more scalable and suitable for large and highly dynamic networks. However, packets can

reach a node that has no neighbour that is closer to the destination than the node itself. This problem, known as a local maximum, is likely to occur in cases of sparse networks or uneven traffic distribution. In such cases, GPSR switches to a recovery strategy called perimeter mode that employs the right-hand rule algorithm of planar graph traversal to route the packets out of the region of the local maximum (a planar graph is one in which no two edges cross). Basing the GPSR perimeter mode on the right-hand rule biases it to a specific direction when it selects the next hop, without consideration of network connectivity. This recovery procedure is abandoned as soon as it is practicable to revert to the greedy strategy, since it can decrease the performance of the network when used frequently.

Despite the advantages of position-based routing protocols, they have open issues when they are used for routing in city environments [7]. City scenarios, where almost the entire area between streets is covered with buildings, frequently force GPSR to run in perimeter mode. As a result, the performance of GPSR can deteriorate dramatically in these scenarios, and it therefore may not be suitable for inter-vehicle communications. Due to the presence of city structures, nodes that would have seen each other in free space cannot communicate, leading to more local maxima, and causing the protocol to operate in perimeter mode. In this situation, nodes might no longer send packets to the neighbour with the maximum forward progress. Thus, compared to greedy routing, GPSR may require many more nodes to be traversed, which leads to more delay and greater hop counts. In addition, mobility can induce routing loops for packets that are routed in perimeter mode [7].

Geographical Source Routing (GSR) [7] tries to overcome the disadvantages of the position-based routing protocols developed for MANETs, when they are applied to VANETs in urban scenarios. In GSR protocols, every node is equipped with a city map. The source node computes the shortest path to the intended destination using Dijkstra's algorithm [84], based on the street map. The computed path consists of a sequence of junction IDs known as Anchor Points (APs). The list of junctions is then inserted into the header of each data packet sent by the source. Using this list, source-based routing is employed across junctions, while greedy-based routing is used for packet forwarding in the street segments between the junctions.

The disadvantage of GSR is that it does not consider the vehicular traffic conditions

of the streets along a route to support connectivity. Figure 2.2 depicts an example where, along one road segment, the packets face a local maximum situation that prevents them from progressing towards the next junction. In such a situation, the packets are discarded at the local maximum even though a longer alternative path exists. In the following section, a number of traffic-aware or connectivity-aware routing protocols are described that have been designed to address this issue.

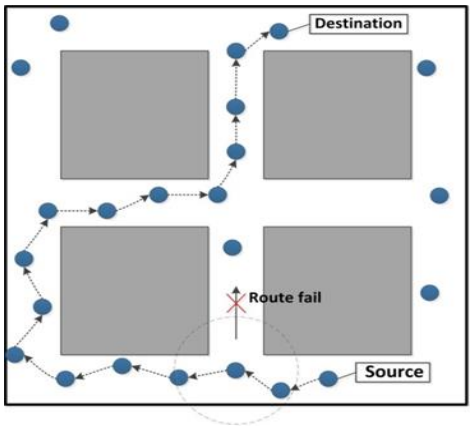

Figure 2.2 GSR fails to find the route, because the shortest path does not have sufficient connectivity.

2.3.3 Traffic-Aware Routing Protocols

The Anchor-based Street and Traffic Aware Routing (A-STAR) protocol [85] utilizes city bus routes to find a path with high connectivity for packet delivery in a city environment. It assumes that connectivity on a street with more bus routes is higher, due to the higher density of vehicles, and that traffic density is more stable, due to the regular presence of city buses. It assigns a weight to each street based on the number of bus routes that traverse it. Lower weights are assigned to street segments that have more bus routes. It is assumed that the digital street map used by the vehicles is equipped with bus route information. Thus, a routing path can be computed using Dijkstra's least-weight path algorithm. This path comprises the sequence of junctions that a packet must pass through to reach its destination. To send a packet, A-STAR inserts the computed sequence of junctions into the packet. Between the junctions, greedy forwarding is used. There are two problems with this approach. First, it uses static information on city bus services, so that the weights of streets are constant, while, in reality, traffic conditions can be very dynamic. Second, streets with higher traffic

density are not always optimum paths, because packet loss occurs due not only to low connectivity, but also to collisions along congested paths.

Spatial and Traffic-Aware Routing (STAR) [86] exploits both street topology information provided by geographic information systems and information about vehicular traffic to make routing decisions. It attempts to detect two extreme situations: either the presence of a high number of vehicles, or the total absence of vehicles. Monitoring and propagation of vehicular traffic conditions is performed through the exchange of network-level beacons that carry observations of node neighbourhoods. The observations are maintained in data structures managed by the traffic monitoring module. Every node maintains four node counters, which represent the number of neighbours it has in the directions of the four cardinal points. When a counter exceeds an upper threshold, it indicates a high concentration of vehicles in the corresponding direction, while a counter below a lower threshold indicates scarce vehicular traffic in the corresponding direction. If these critical situations are persistent (criteria exist to detect persistent situations, in order to guarantee that a temporary abnormal condition is not registered), they are recorded in the traffic table. Each node periodically broadcasts a beacon that contains a sender identifier, sender coordinates, and the critical vehicular traffic conditions present in the sender's traffic table. Each entry in the traffic table has a time-to-live (TTL) value, which determines how far the traffic information will be spread. When a source has a packet to send to a destination, the source builds a weighted graph using the street map and traffic information. Edges corresponding to streets without traffic are assigned a high weight. In contrast, edges corresponding to streets with high levels of vehicular traffic are assigned low weights to bias the choice in favour of the street as a route path even though it could be longer in length than other paths. In this protocol, streets with high traffic density are assigned a low weight, and the weights of streets can change dynamically. However, the second problem highlighted above still exists; the algorithm tends to use more congested streets, because it weighs the streets according to the number of nodes and not according to the packet-relaying properties of the streets. Another problem with this protocol is that, using only propagated information, the street along which a certain condition exists cannot be exactly determined.

Improved Greedy Traffic Aware Routing (GyTAR) [87] selects junctions one by one

through which a packet must pass to reach its destination. When selecting the next destination junction, a node (the sending vehicle or an intermediate vehicle at a junction) looks for the positions of the neighbouring junctions using a map. A score is assigned to each junction based on the traffic density and the curve metric distance to the destination. The best destination junction is then the junction with the highest score. Once the destination junction is determined, the improved greedy strategy is used to forward packets between the sending and the next destination junctions. According to this strategy, when a packet is received, the forwarding vehicle computes the new predicted position of each neighbour using its velocity, direction, and latest known position information, and then selects the next hop neighbour. In this routing protocol, it is assumed that every vehicle obtains the vehicular traffic information (number of vehicles between two junctions) from traffic sensors installed beside the junctions. Apart from the need for a large number of sensors, such protocols face problems such as limited coverage of detection equipment, high deployment and maintenance costs, and a great deal of time being consumed in collecting, processing, and disseminating traffic-related information [88]. Furthermore, due to the junction-by-junction routing approach, the protocol might not always find a path to a given destination.

Road-Based Traffic Aware Routing (RBVT) [89] is another traffic-aware routing protocol that uses a beaconless mechanism in order to overcome network congestion. Backbone-Assisted Hop Greedy Routing (BAHG) [90] is a position-based connectivity-aware routing protocol which tries to select paths between source and destination consisting of a minimum of intermediate intersections (intersections in which the direction of the path is changed), since such intersections result in higher hop counts and poor connectivity due to buildings around the intersection. The protocol assigns a weight to each road segment that is the sum of the hop count and a second parameter, the "delta count". The hop count is the required number of hops for a packet to traverse a road segment. The delta count signifies the degree of disconnection along a road segment, and is calculated as the product of a constant and the hop count. The delta count is set to zero for roads with four or more lanes (which are considered fully connected). The protocol defines major roads as those that have more than two lanes. The city map is divided into zones, such that every zone is surrounded by major roads. The intersections at the corners of each zone act as entry

points for the packets sent to a zone. The problem with this protocol is that the delta count does not represent the real-time connectivity conditions of the roads. Therefore, the calculated path sometimes encounters void regions. In this case, the path must be recalculated from that point, which causes more hop counts and greater delay.

Intersection-based Geographical Routing Protocol (IGRP) [91] is a routing protocol for forwarding data packets from vehicles to Internet gateways, under the assumption that the Internet gateways have up-to-date information on the positions of all vehicles in their surrounding areas. When a vehicle requires a route to forward data packets to the Internet gateway, it sends a request to it. The gateway then computes the intersections of the path and the transmission range at each street segment that the vehicles are required to use to achieve high connectivity.

In Connectivity-Aware Routing (CAR) [92], the source broadcasts request messages to find a path to the destination. To estimate connectivity, every node forwarding the route request updates the hop count, as well as the average and minimum number of neighbours. The destination decides the routing path and replies to the source. Even though CAR addresses connectivity issues, the gathered information on the number of nodes cannot ensure connectivity in individual road segments along a routing path, because the connectivity depends on both the number of nodes and on their topology.

2.3.4 Ant-Based Routing Protocols

Mobility-aware Ant Colony Optimization Routing DYMO (MAR-DYMO) [93] is a reactive routing protocol for VANETs that is a combination of Dynamic MANET On-demand Routing (DYMO) [94] and ACO. DYMO itself is an improved version of AODV. Using vehicles' position and speed, it predicts their movements to find the path with the longest lifetime. MAR-DYMO has scalability problem as the performance of the protocol drops rapidly with increasing numbers of vehicles. It has higher overhead than DYMO, while the packet delivery is increased slightly. Also, because it is node-based, the overhead increases as the speed of vehicles increases due to greater link breakage and route recovery.

Trust Dependent Ant Colony Routing (TACR) [95] is a reactive ant-based routing protocol in which clusters of vehicles are created by considering direction, position,

and relative speed of vehicles to manage the scalability of the protocol. Only cluster heads contribute to launching ants and finding routes to decrease routing overhead. However, the simulation results did not show much improvement against MAR-DYMO, because managing the clusters creates overhead itself.

Mobility Aware Zone based Ant Colony Optimization Routing (MAZACORNET) [96] is a zone-based ACO routing for VANETs. By using ACO technique, it tries to select the links for routing that have higher lifetimes and quality. The link quality is estimated by using the Nakagami fading model. Inter-zone routing follows a proactive approach and intra-zone routing is on demand. In terms of routing control overhead, it does not show much improvement against AODV due to the inter-zone proactive routing approach. Thus, like other node based algorithms, it has scalability problem when used in VANETs.

Vehicular Routing Protocol Based on Ant Colony Optimization (VACO) [97] uses ant colony optimisation to assess the packet-relaying quality of each street segment located between two junctions in terms of latency, bandwidth, and delivery ratio. It is assumed that there is an RSU at each junction to save routing information and find routes for packets. To set up a route, the source node forwards several ants toward a target RSU, which is the closest one to the destination vehicle. At the target RSU, backward ants are generated and sent back to the source. For route maintenance, VACO utilizes a proactive approach. Using RSU at every junction can be costly and might not be practical, at least during the initial deployment of VANETs. In addition, this causes the protocol to be vulnerable to failure of specific nodes (RSUs).

Adaptive and Opportunistic QoS-based Routing in VANETs (AQRV) [28] is a junction-based QoS routing protocol whose QoS metrics include connectivity probability, packet delivery ratio, and delay. It is assumed that there is a static terminal intersection (TI) with Wi-Fi capability at every junction to store the routing table, launch ants (small control packets), and relay data packets. At the beginning of the data transfer, a source vehicle sends its request to a TI (TIS) and, if the TIS has no route to the destination, launches several forward ants toward the destination TI (TID). At the TID, forward ants are converted to the backward ants and are returned to the TIS. A so-called pheromone table of every TI that the backward ants pass is updated based on the information that the ants carry, with an analytical expression for the three

metrics. The analytical expression assumes that the streets are one-way, and the routing protocol is tested on a one-way simulation scenario. It is also assumed that the TIs know the vehicle density at each street.

2.4 ACO Algorithms

Many algorithms inspired by the behaviour of ant colonies have been developed to solve optimization problems. An ant colony is a multi-agent system where each agent (ant) operates independently by very simple rules. Despite the very primitive behaviour of the agents, the whole system functions in a reasonable way and can fulfil complicated goals. The agents use only local information to interact with other agents, which results in the independence of the system from centralized control.

However, ACO has been implemented in different forms, in general, ACO can solve the problems defined as follows [98].

- A finite set of components $C = \{c_1, c_2, c_3, \dots, c_N\}$ is given where N is the number of components.
- A finite set of possible connections between elements of C is given by $L = \{l_{c_i c_j} |\ (c_i, c_j) \in \hat{C}\ \}$ where $\hat{C} = C \times C$.
- $J_{c_i c_j} = J(l_{c_i c_j}, t)$ is the connection cost function associated to each $l_{c_i c_j} \in L$ that might change over time.
- $Q(C, L, t)$ is the set of constraints assigned over the elements of C and L.
- A sequence $s =< c_i, c_j, \dots, c_k >$ of some elements of C is called a state of the problem. If S is the set of all possible sequences, the set $\hat{S}$ of all possible sequences under the constraints Q is a subset of S and includes the feasible states.
- Φ is a solution of the problem if it is an element of $\hat{S}$ and satisfies all the problem's requirements.

The graph $G = (C, L)$ is associated to the problem defined above and the solutions to the optimization problem are the feasible paths on the graph G. ACO algorithms use a population of ants to search the graph and collectively solve the defined problem.

The information collected by the ants is represented as pheromone deposits associated with connections $l_{c_i c_j}$.

2.4.1 ACO for Data Routing in Computer Networks

Different implementations of ACO for routing in telecommunication networks have been applied; however, in general, they can be described as follows.

- Let sets C and L introduced above correspond to network nodes and communication links between nodes, respectively, and $G = (C, L)$ is the corresponding directed graph.
- Network nodes keep routing tables (also called pheromone tables) to forward data packets to the destinations. The pheromone table of node c_k with set of neighbours N_k is a data structure consisting of probability values P_{id} which expresses the goodness of choosing $c_i \in N_k$ as the next node when the destination node is d with the following condition:

$$\sum_{c_i \in N_k} P_{id} = 1 \tag{2.2}$$

- Network nodes update probabilities P_{id} regularly with two processes, pheromone deposit and evaporation so that the route finding task becomes adaptive to network changes.
- To this aim, network nodes regularly generate ant packets and forward them to a randomly selected destination to evaluate and find paths to the destination. The ant packets collect information on the quality of the links they pass. This information is used by network nodes that ants meet in their paths to update the P_{id} values corresponding to those paths. The process of updating the P_{id} values is called pheromone deposit.
- Probabilities P_{id} are reduced regularly at constant intervals so that network nodes forget an old path which is not a good path any more. This process, which is called pheromone evaporation, can be implemented as follows.

$$P_{id} \leftarrow \alpha . P_{id}, \quad \text{where} \quad 0 < \alpha < 1 \tag{2.3}$$

- To forward data packets to a destination, every node selects one of its neighbouring nodes randomly with the highest probability of selection given to the neighbour that has the highest amount of pheromone to reach the destination.

ACO has been implemented in different variations for routing. For example, ants might deposit pheromone in their backward or forward journey or the pheromone values of the links might represent the delay, traffic flow or other qualities of the links.

2.5 Game Theory

A situation referred to as a game when several entities involved in the situation and the outcome of the situation for an entity depends not only on what the entity does but also what the other entities do. The entities are referred to as decision-makers or players of the game. A non-cooperative game is a game in which players take their actions without any agreement with other players. Game theory is a mathematical study of the interactions between the players who might have conflicting or common interests. Game theory deals with designing interaction models, studying the conditions that some outcome can be achieved and designing strategies to reach desired outcomes [99].

A game can be represented in different types. In this thesis, a non-cooperative game in the strategic or normal form [10] has been used, then just these games are introduced. A strategic form game is a triplet $\mathcal{G} = \{\mathcal{N}, \{\mathcal{S}_i\}_{i\in\mathcal{N}}, \{\varphi_i\}_{i\in\mathcal{N}}\}$ where $\mathcal{N} = \{1, 2, \dots, n\}$ is the set of players, $\mathcal{S}_i$ is the set of strategies of player i and φ_i is the payoff function of player i that gives the player i the value $\varphi_i(s)$ for each strategy profile $s = \{s_1, s_2, \dots, s_n\} \in \prod_{i\in\mathcal{N}} \mathcal{S}_i$.

Nash Equilibrium (NE) is a key concept in game theory. It is the profile of strategies such that each player's strategy is an optimal response to the other players' strategy [10]. In mathematical terms, the vector *s** is an NE if:

$$\forall i \in N, \forall s_i \in S_i, \varphi_i(s_i^*, \mathbf{s}_{-i}^*) \geq \varphi_i(s_i, \mathbf{s}_{-i}^*) \tag{2.4}$$

where $\boldsymbol{s}_{-i}$ is a vector of strategies of all the players except player i. In other words, an NE is the point that no player has incentive to change its strategy unilaterally and it is the solution of the non-cooperative games involving rational players.

2.6 Summary

In VANETs cooperative awareness is created by exchanging beacons periodically. However uncontrolled beaconing reduces the performance of the protocols that rely on the cooperative awareness. Current adaptive beaconing schemes, suffer from unfairness, instability and control overhead. In addition, most of them assign the same beaconing parameters (rate or power) to vehicles, apart from the dynamics of vehicles.

Chapter 3

Efficient Geographic Source Routing (EGSR) Protocol

The performance of position-based routing in city scenarios will improve when real-time traffic information is used as part of the routing metric [86], [87], [89]-[92]. In this chapter, a position-based routing protocol for city environment is developed that is self-adaptive to traffic conditions. The focus is on approaches that do not rely on particular hardware support, such as the traffic sensors that are assumed to exist in GyTAR, or on traffic information obtained from outside the network, such as the information on bus routes used in A-STAR. The proposed protocol is called Efficient GSR (EGSR). It optimises GSR for routing in a city environment with unevenly distributed vehicular traffic by adding traffic awareness to GSR. Like GSR, EGSR uses the street map to compute the shortest path; however, the weight of every street segment is not the length of the street. Instead, the weights are computed and dynamically updated according to the connectivity conditions of the streets. To make the protocol aware of the traffic conditions of the street segments, it uses small control packets, termed "ants", to sample traffic conditions and update vehicles routing information. The approach presented in this thesis is based on ACO.

Recently, bio-inspired networking approaches have received a great deal of interest due to their potential features such as scalability, adaptability, self-organization, robustness, and resilience to failures [3]. The architecture of bio-inspired solutions should implement key principles [100], [101] to achieve these desirable properties. Otherwise, their effectiveness or functionality might be limited [102], [103]. Thus, despite the similarities of some solutions to biological systems they fail to meet the objectives or achieve the advantages of them.

Among bio-inspired techniques, ant colony optimisation has been widely used for routing in networks [95]-[97], [103]-[108]. Most of the proposed ant-based routing protocols try to find entire paths between network nodes. However, this method might

not be suitable for VANETs due to rapid movement of the nodes. In this work, ant colony optimisation has been applied to finding the optimum street segments (the part of a street between two junctions; see Figure 3.1) for routing packets in VANETs.

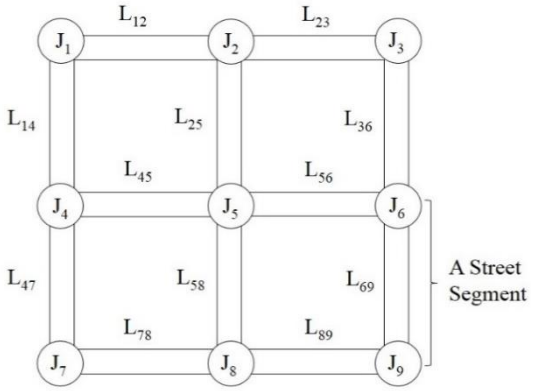

Figure 3.1 A section of a city map as an example

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9
J_1	∞	L_{12}	∞	L_{14}	∞	∞	∞	∞	∞
J_2	L_{12}	∞	L_{23}	∞	L_{25}	∞	∞	∞	∞
J_3	∞	L_{23}	∞	∞	∞	L_{36}	∞	∞	∞
J_4	L_{14}	∞	∞	∞	L_{45}	∞	L_{47}	∞	∞
J_5	∞	L_{25}	∞	L_{45}	∞	L_{56}	∞	L_{58}	∞
J_6	∞	∞	L_{36}	∞	L_{56}	∞	∞	∞	L_{69}
J_7	∞	∞	∞	L_{47}	∞	∞	∞	L_{78}	∞
J_8	∞	∞	∞	∞	L_{58}	∞	L_{78}	∞	L_{89}
J_9	∞	∞	∞	∞	∞	L_{69}	∞	L_{89}	∞

Figure 3.2 Adjacency matrix of the map in Figure 3.1

3.1 System Model

The wireless technology deployed for exchange of packets is DSRC. DSRC uses IEEE 802.11p standard at the PHY and MAC layers. Each vehicle is equipped with a GPS receiver, digital map, and navigation system. Thus, vehicles are aware of their position through the GPS and can map their positions on roads using the navigation system. A sender vehicle obtains the position of a receiver vehicle by querying from a location service [109]-[111]. The clocks of all vehicles are synchronized. Synchronization has been considered in IEEE Std 1609.0-2013 [22] and IEEE Std 1609.4-2016 [26] and is necessary for multi-channel operation and security purposes; it can be provided by

GPS. Vehicles can be equipped with a sufficient number of computational resources, such as processors and a large memory capacity. Thus, no capacity, processing, or power constraints are assumed for vehicles [21].

3.2 The Protocol Design

Using a digital map of the streets each vehicle can obtain an adjacency matrix of the graph that models the city map. As a simple example, Figure 3.1 shows a part of a city map with specified junctions in circles (J_n) and street segments between junction J_i and J_j with lengths L_{ij}. This map can be represented by a graph, with junctions as vertices and streets as edges. Figure 3.2 shows the corresponding adjacency matrix. According to the GSR protocol, whenever a vehicle wants to send a packet to a destination, it initially adds two vertices, which correspond to the source and the destination, to the matrix. Then, it computes the shortest path using Dijkstra's algorithm, adds the ordered list of junctions (anchor points) to the packet header, and then sends it.

In EGSR, like GSR, the source vehicle computes an ordered list of the junctions of the route and stores it in the packet header. The list of junctions is computed using Dijkstra's algorithm on a graph representing the city map in which the weight of every edge (street) is proportional to the connectivity of that street segment. To make the weight of every edge proportional to the network connectivity of the corresponding street and not just its length, the elements of the matrix in Figure 3.2 are redefined as: $\frac{L_{ij}}{P_{ij}}$; where $0< P_{ij}<1$. P_{ij} is a variable showing the connectivity condition of the street segment between junctions J_i and J_j. In other words, P_{ij} is the pheromone value related to the street segment between junctions J_i and J_j. A low P_{ij} demonstrates a poor connectivity due to low traffic density. Vehicles update the P_{ij}s according to the information in the ant packets that they receive. This mechanism is described further in the following sections.

Ant packets are launched by the vehicles in junction areas and are forwarded toward the next junction. On arrival, the junction ID is recorded in the ant packet and the next street is selected randomly with a probability proportional to the number of vehicles in the street in the neighbourhood of the current ant holder. Then, the ant is forwarded

to the junction located at the end of the selected street segment. Between two junctions, ants are broadcast similar to POCA [112] so as to prevent a broadcast storm, but using a simpler approach than POCA. When a node wants to broadcast an ant, it selects its nearest neighbour to the next junction for rebroadcasting it. If an ant packet passes completely through a street segment, there is connectivity in that segment. Every node that receives an ant updates its adjacency matrix: the P_{ij} related to the street segment between the junctions J_i and J_j traversed by the ant will be increased (see section 3.2.2). In other words, the ant deposits pheromone. In this way, vehicles have an adjacency matrix in which the weight of each street is proportional to its length and network connectivity. A mechanism for pheromone evaporation that decreases the P_{ij}s in regular intervals so as to make the adjacency matrix adaptive to traffic changes is presented in section 3.2.2.

The number of ants that traverse a street and the length of time it takes reflect the packet relaying condition of the street segment. If there are not enough vehicles in a street or it is congested, fewer ants and over a longer period of time can pass through that street. When this occurs, the evaporation mechanism decreases the P_{ij} related to that street more rapidly than ants can increase it.

3.2.1 Launching Ants

An area with radius *Ra* at every junction is called an anchor area (see Figure 3.3). The time interval between launching successive ants at a junction is called t_{ant}. If a vehicle in an anchor area during time interval t_{ant}, does not receive a new ant (an ant that has been launched in this junction), it creates one and broadcasts it toward the next junction. This vehicle also selects one neighbour as the next ant forwarder.

For example, in Figure 3.3, vehicle *V1* located in the anchor area of street S1 launches an ant and broadcasts it toward junction J_2. It selects *V2*, which is its closest neighbour to the next junction (J_2), to rebroadcast the ant. Then, only one vehicle will rebroadcast an ant in every forwarding step. When the ant reaches junction J_2, J_2 is recorded in the ant packet. Then, the ant is sent to one of the street segments J_2-J_3, J_2-J_4, or J_2-J_5, selected randomly, with a higher probability of selection given to the street

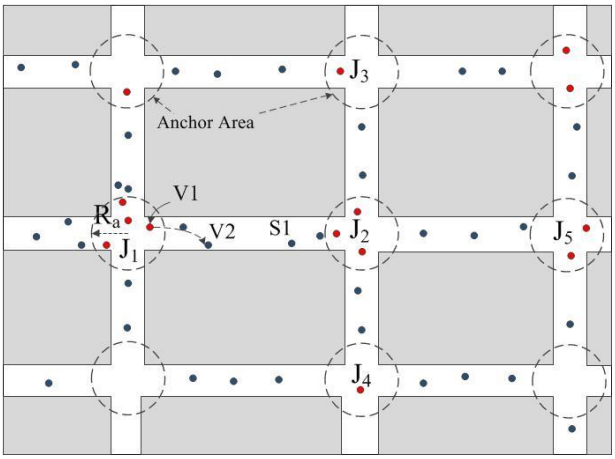

Figure 3.3 Anchor areas at every junction

that has more vehicles in the neighbourhood of the current ant holder. Algorithm 3.1 presents the ant-launching process. t_{ant} is the time between launching successive ants. It determines how quickly the algorithm adapts to changes in vehicular and data traffic of the streets. A too-small t_{ant} causes the network to be flooded by ants, and with a

Algorithm 3.1 Launching ants

//Vehicle Vi upon entering anchor area of J_i

1: Set timer $T_a = t_{ant}$
2: **if** *Vi* received an ant **then**
3: **if** (the only junction recorded in the ant == J_i) **then**
4: Reset timer T_a (timer $T_a = t_{ant}$)
5: **end if**
6: **end if**
7: **if** (timer $T_a == 0$) **then**
8: Launch ant
9: Reset beacon timer
10: Set timer $T_a = t_{ant}$
11: **end if**
12: **if** *Vi* left the anchor area **then**
13: Cancel timer T_a
14: **end if**

large t_{ant}, the protocol cannot adapt to the changes in the network. In both cases, performance of the protocol decreases. Analysis of the parameters of ant-based protocols in ad hoc networks has been presented in [113]. We have selected this parameter experimentally and using the results in [113].

3.2.2 Updating the Adjacency Matrix

When a vehicle receives an ant, it updates its adjacency matrix. If junctions J_i and J_j have been recorded in the ant packet as two consecutive junctions, it means there was connectivity to pass the ant through the street segment between junctions J_i and J_j so P_{ij} and P_{ji} will be updated according to (3.1) and (3.2). In [105] the formula (3.1) was used to update pheromone of links between two nodes. Herein, it is used to update pheromone for every street segment between two junctions.

$$P_{ij} = P_{ji} = \frac{P_{ij} + \Delta P_{ij}}{1 + \Delta P_{ij}} \tag{3.1}$$

$$\Delta P_{ij} = A + \arctan \frac{\min\_delay_{ij}}{delay_{ij}} \tag{3.2}$$

where $delay_{ij}$ is the time it takes the ant to traverse the street between junctions J_i and J_j, min_delay_{ij} is the minimum delay for ants that the vehicle has recorded for that street, and A is a constant.

Because min_delay_{ij} is less than or equal to $delay_{ij}$, $\frac{2}{\pi}\ atan(min\_delay_{ij}/delay_{ij})$ is between 0 and 0.5. Therefore, for A less than 0.5, ΔP_{ij} will be less than 1. The initial value for P_{ij} is selected to be less than 1 and therefore, P_{ij} will always be less than 1. Every time P_{ij} is renewed according to (3.1), it will be increased so the value $\frac{L_{ij}}{P_{ij}}$ will be decreased. ΔP_{ij} is greater if the delay the ant encounters in a street is lower. As a result, the weight of that street would be decreased with decreasing delay. Similarly, if three junctions J_i, J_j, and J_k have been recorded in the ant packet as three consecutive junctions, P_{ij} , P_{ji}, P_{jk}, and P_{kj} will each be updated according to (3.1) and (3.2). If the vehicle that has received the ant is the next forwarder, it then selects its closest neighbour to the next junction as the next forwarder and rebroadcasts the ant. If it is

not the next forwarder, it just renews its adjacency matrix according to the ant's information. Every t_{eva} seconds, each vehicle decreases the pheromone (P_{ij}) of all the streets, using the following formula:

$$P_{ij} = \alpha P_{ij}; \qquad 0 < \alpha < 1 \tag{3.3}$$

With the proposed mechanisms, pheromone increase and decrease, every vehicle regularly updates the weights of the edges of the graph representing the map of the surrounding area proportional to the connectivity of those streets. Thus, the route that every source vehicle computes for its data packets is adaptive to the traffic conditions on the streets. Every node broadcasts its ID and position in beacons regularly. When a node broadcasts an ant, it also includes beacon information in the ant packet and resets its beacon timer so fewer beacons are required. The purpose of this is to reduce the congestion in the network; because sending two packets will contribute to congestion more than sending one packet with the sum of their sizes [114].

3.2.3 Ant Packets

An ant packet consists of the following fields:

- ***Type***: Indicates the type of the packet.
- ***Sender_Id***: ID (or address) of the first node that issued the ant.
- ***Serial_Number***: Every node assigns numbers sequentially to the ants it creates.
- ***Version***: First node that creates an ant sets this field to zero. Every time a node adds a junction ID to *Sequence_Of_Junctions* field, it increases this field by one.
- ***Street_Id***: ID of the street that the ant is traversing.
- ***Next_Junction_Position***: Position of the next junction that the ant should be sent toward.
- ***Sequence_Of_Junctions***: Sequence of junctions that the ant has traversed.
- ***S_Delays***: Time stamps showing the times that the ant has passed each junction.
- ***Next_Forwarder***: ID (address) of the next node that should forward (broadcast) the ant.

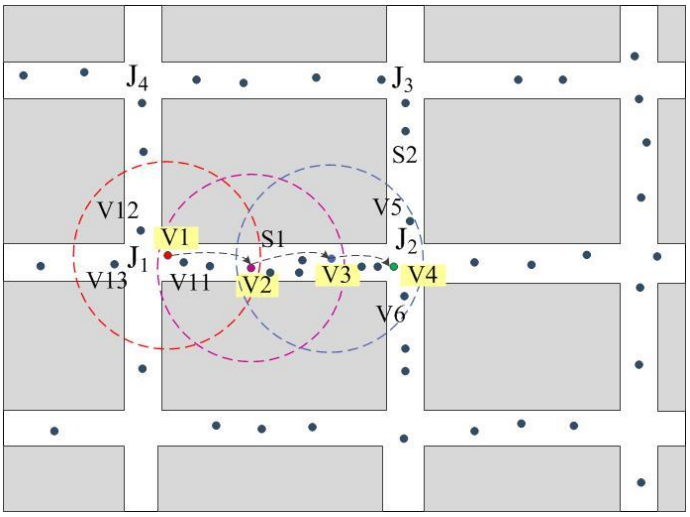

Figure 3.4 Ant launching and forwarding by vehicles. Dashed circles show communication range of vehicles V1, V2 and V3

- ***LastSender_Id***: ID of the last node that has broadcast the ant.
- ***LastSender_Position***: Position of the last node that has broadcast the ant.

Sender_Id and *Serial_Number* have the same functionality as in any other regular routing protocol, that is, to ignore repeated packets. Every node that receives an ant checks the *Serial_Number*, *Sender_Id*, and Version fields of the ant. If it has not already received one with the *Serial_Number* and *Sender_Id*, it uses the ant's information to update its adjacency matrix. If it has received the ant with the same *Serial_Number* and *Sender_Id*, but a lower Version, it just uses that part of the information of the packet that it has not already received, to update its adjacency matrix. For example, if a node has received an ant with version number 2, it means that the ant has passed three junctions. If it then receives an ant with the same *Serial_Number* and *Sender_Id* and version number 3, it just updates the P_{ij} related to the street between the last two junctions.

If the node that receives the ant is the next forwarder, it selects the subsequent next forwarder and then broadcasts the ant, whether or not it has received it before. For example, in Figure 3.4, vehicle *V1* creates and sends an ant toward junction J_2. *V1* sets the Version field to zero, *Sequence_Of_Junctions* to J_1, *S_Delays* to current time, *Next_Forwarder* to *V2*, *Next_Junction_Position* to the coordinates of J_2, *Street_Id* to *S1*, and *LastSender_Id* and *LastSender_Position* to its ID and position, subsequently broadcasting the ant. All nodes *V11*, *V12*, *V13*, and *V2* receive the ant and update their

adjacency matrices if required, which is not necessary in this situation because the ant has only one junction in its *Sequence_Of_Junctions* field. The ant is then broadcast by *V2* and *V3* in turn. Assuming that *V3* has selected *V4* as the next forwarder, *V4* checks the *Street_Id* of the ant and selects one of its neighbours that is not in the street *S1*, for example *V5*. Subsequently, *V4* changes the *Street_Id* field to *S2*, *Next_Junction_Position* to coordinates of *J3*, *Next_Forwarder* to *V5*, and *LastSender_Id* and *LastSender_Position* to its ID and position, respectively. It also records J_2 to *Sequence_Of_Junctions*, current time to *S_Delays*, and increases the Version number by one. Consequently, vehicles receiving this packet, can compute the time this ant took to travel from J_1 to J_2.

The requirement of the Version field can be explained as follows. When vehicle *V3* broadcasts the ant, both *V5* and *V6* are within its communication range and receive it. When *V4* updates the ant and broadcasts it, if the Version field does not exist, *V5* and *V6* ignore the new information that has been added by *V3*. In addition, by comparing the new and previous versions of the ant, they use just that part of the information they have not used before (newly added junctions). In the protocol, there is a limit on the number of junctions ants can travel and after that limit, the nodes kill them. *LastSender_Id* and *LastSender_Position* are required because the ant can have the same functionality as a beacon, with the benefit being that fewer beacons are required. Algorithm 3.2 presents the ant forwarding and the pheromone updating mechanisms.

3.3 Performance Evaluation

The performance of EGSR has been compared with GSR and VACO. GSR is a routing protocol that almost every position-based routing protocol was compared with that in the related literature. VACO uses ACO but it relies on RSUs at every junction thus, it has been selected for comparison to indicate EGSR can work efficiently even without RSUs. OMNeT++ and SUMO have been used to simulate the network and generate vehicular traffic mobility. The propagation model is lognormal shadowing [39].

Algorithm 3.2 Launching ants

//Vehicle Vi received ant A_i with Street_Id =S_n

//Street segment S_m is between junctions J_i and J_j

1: **if (**A_i not received before) **then**
2: **if** (*Vi == Next_Forwarder*) **then**
3: **if** (V_i is in the anchor area of J_i) **then**
4: **if** (J_i has not been recorded in A_i) **then**
5: Record J_i in *Sequence_Of_Junctions*
6: **if** (size of *Sequence_Of_Junctions* <maximum size of *Sequence_Of_Junctions*) **then**
7: Record current time in *S_Delay*
8: Select next forwarder V_j in S_m $(n \neq m)$
9: *Version* ← *Version + 1*
10: *Street_Id* ← S_m
11: *Next_Junction_Position* ← position of J_i
12: **end if**
13: **end if**
14: **else**
15: Select the next forwarder V_j using greedy mechanism
16: **end if**
17: Update pheromones using (3.1) and (3.2)
18: *LastSender_Id* ← V_i
19: *LastSender_Position* ← *position of* V_i
20: *Next_Forwarder* ← V_j
21: Transmit A_i
22: Reset beacon timer
23: **else**
24: Update pheromone using (3.1) and (3.2)
25: **end if**
26: **end if**
27: Update neighbour table

Table 3.1 Simulation parameters

Parameter	**Value**
Scenario Area	2000 m × 2000 m
Communication Range	300 m
MAC Protocol	IEEE 802.11p
Simulation Time	800 s
Vehicle Velocity	45, 50, 55, 60, 65, 70 Km/h
Number of Concurrent Connections	10
Carrier Frequency	5.89 GHz
Bit Rate	18 Mbps
Beacon Frequency	2 Hz
Data Packet Size	512 Byte unless specified
EGSR parameters	t_{ant} = 1.5 s, α = 0.92 [113]

The simulation parameters are indicated in Table 3.1. For The mobility model and urban map topology, the Manhattan model [115] has been employed. The routing protocols use greedy forwarding between two junctions and their difference is in the way that they select street segments for packet forwarding. In Manhattan model at every junction four street segments meet, therefore this model is an appropriate model to evaluate the street selection of the routing protocols. The simulation area covers a 2000 × 2000 meter grid in which the distance between two adjacent junctions is set to

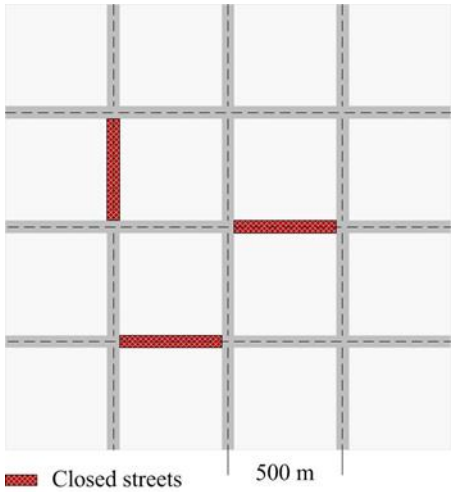

Figure 3.5 Simulation map

500 meter. The streets are two-way, with two lanes in each direction. Three of them have no traffic, as shown in Figure 3.5.

3.3.1 Packet Delivery Ratio

Figure 3.6 shows the packet delivery ratio of the protocols for a packet rate of 5 Pkt/s for different vehicle speeds. EGSR performs better than the other protocols by at least 10% up to a speed of 70 km/h. VACO needs more control packets for route set up and maintenance, because the ants deposit pheromone on the backward journey, while in EGSR the forward ants deposit pheromone, so a backward journey is not required. In General, longer journeys for ants increase both overhead and packet loss due to collision. In EGSR, the control packets take a probabilistic path and the data packets have a deterministic path because the source node determines the junction IDs of the path for the data packets. In VACO, both kinds of packets have a probabilistic path, which might result in sub-optimal choices for data packets. While the pheromones deposited by the ants increase the probability of selecting the optimal path for data packet, there is a possibility a sub-optimal path will be selected. These are the two reasons for the lower delivery ratio of VACO compared to EGSR.

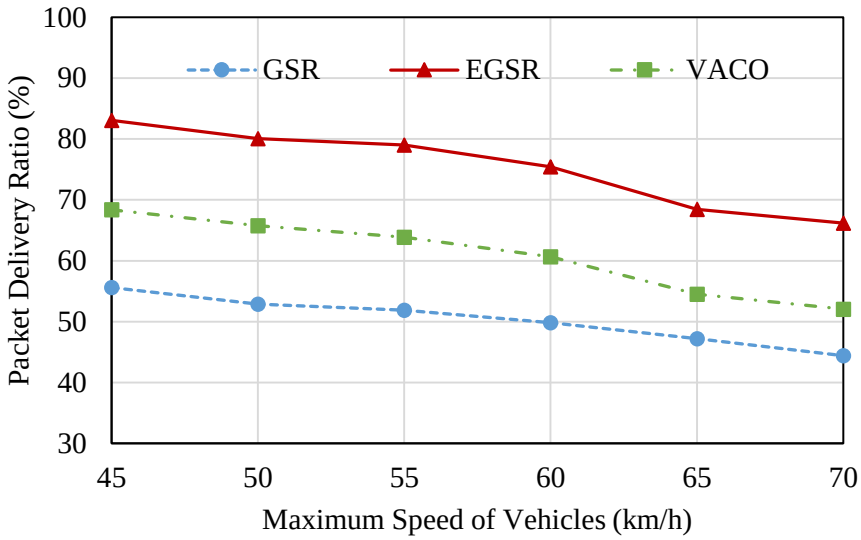

Figure 3.6 Data packet delivery ratio for different vehicle speed

GSR selects the shortest path without considering the packet relaying quality of the path, which leads to the lowest packet delivery ratio among them. The reason for the increase in dropped packets at higher speeds is that the position of vehicles changes rapidly. By using the greedy mechanism to select the next hop, the node that is closest to the destination is selected. Such nodes are usually close to the border of the communication range and can leave it in a shorter time when the speed is higher. If beaconing provides more information like speed and direction of nodes or more than one-hop neighbour information (two or three hops), and this information is used to select the forwarder node, better results would be obtained [116]-[118].

Figure 3.7 shows packet delivery ratios for different packet rates. Figure 3.8 shows the packet delivery ratios for different packet sizes from 256 bytes to 1280 bytes. Larger packets are more susceptible to loss due to higher probability of collision. By increasing the packet size, the packet delivery ratio of EGSR drops less than that of VACO because it uses fewer control packets and thus the protocol suffers fewer collisions.

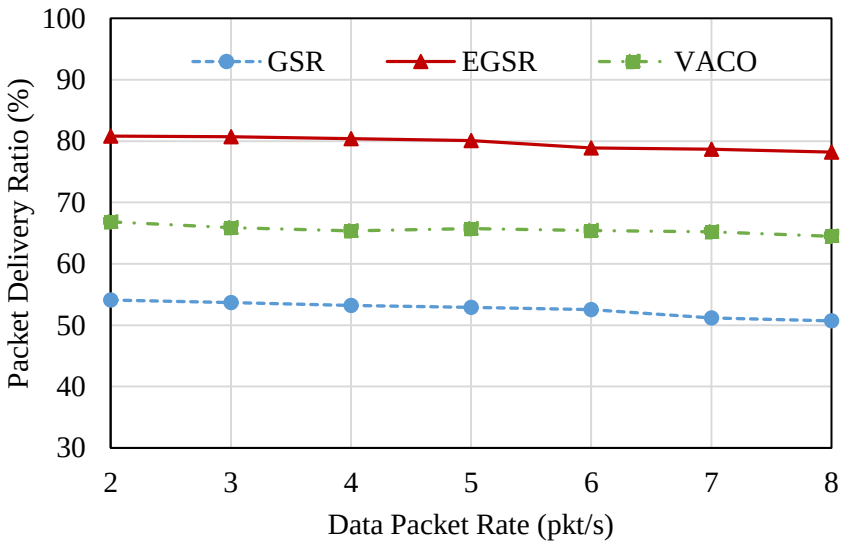

Figure 3.7 Packet delivery ratio for different data rates; maximum vehicle speed of 50 km/h

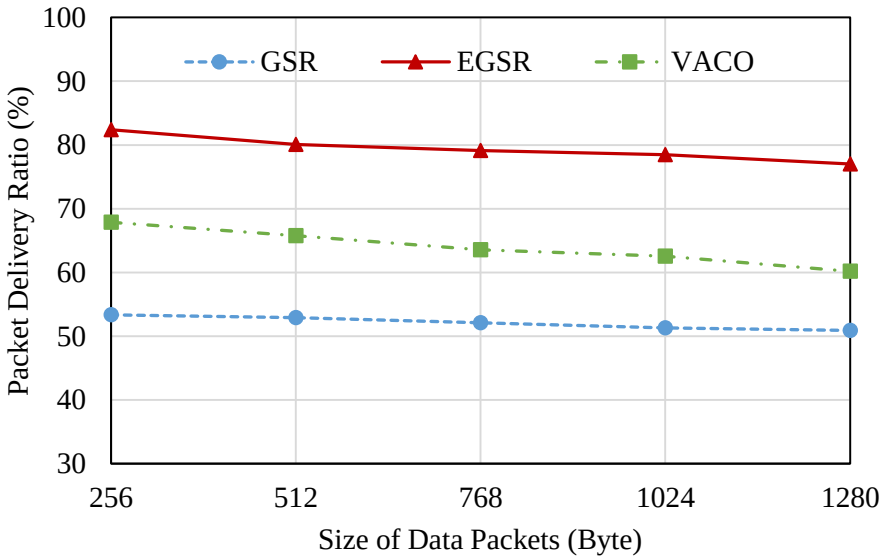

Figure 3.8 Packet delivery ratio with different packet sizes; maximum vehicle speed of 50 km/h

3.3.2 Routing Control Overhead

To compare the overhead of the routing protocols, the number of beacons per second per vehicle in GSR has been compared with the sum of the beacons and ants per second per node (including vehicle and RSU) in EGSR and VACO. The results are shown in Figure 3.9. With the EGSR protocol, when a vehicle forwards an ant it includes beacon information, so fewer beacons are required. This mechanism has been devised to reduce the routing control overhead of the protocol.

In EGSR, at different speeds the number of overhead packets is almost constant. This is because EGSR is road-based, not node-based, and it evaluates connectivity between junctions rather than between nodes. Even when there are many vehicles in an anchor area, which might occur at junctions, the network is not flooded with ants, because the vehicles cooperate on launching ants. A vehicle issues an ant if it does not receive a new one in a specified time (t_{ant}).

VACO is a road-based protocol as well, but it has increasing overhead with increasing velocity. In VACO a communication session is established between source and destination vehicles through RSUs. The source vehicle sends its packet to the first RSU, and the packet is delivered to the destination vehicles by the last RSU. Because of the movement of source and destination vehicles, the first and the last RSU changes

during communication. This requires a new reactive route set up. At higher velocities, this happens more frequently, which contributes to more overhead at higher velocities.

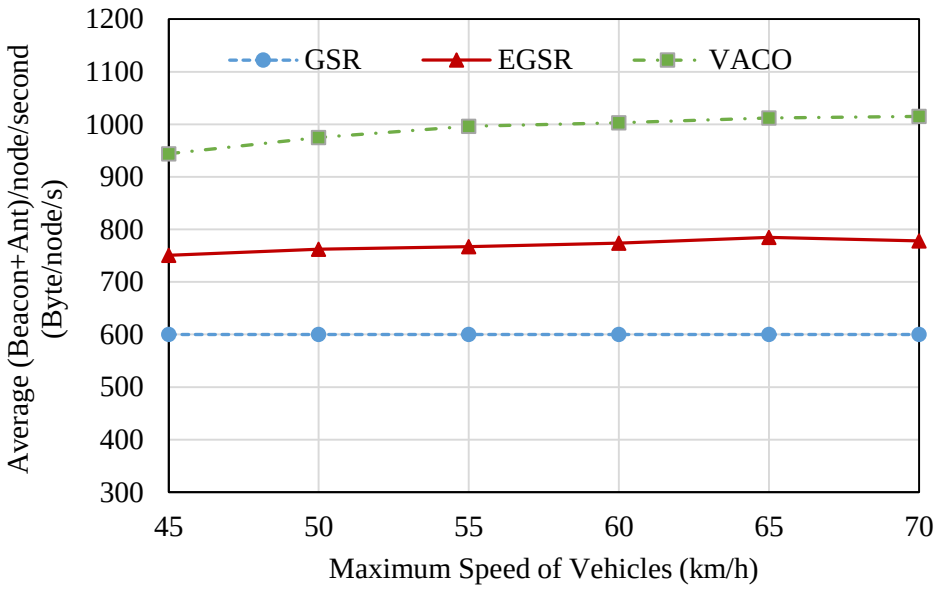

Figure 3.9 Control Packet rate for different vehicle speed

Figure 3.10 shows the total number of routing control packets created by the protocols during the simulation time for different numbers of flows. VACO creates both reactive and proactive ants and they should travel a route two times (forward and backward) to deposit pheromone. Every flow of data requires route set up and maintenance thus the overhead increases by increasing the number of flows.

Figure 3.11 shows the total number of routing control packets created by the protocols during the simulation time for different numbers of vehicles, while the data rate is 5 Pkt/s. As every vehicle creates beacons at 2 Hz frequency, by increasing the vehicle density the number of control packets grows for all the protocols. The figure shows that the excess overhead in EGSR due to ants does not grow with increasing vehicle density, so it is not sensitive to the number of vehicles. Therefore, EGSR is scalable and can work well for different vehicle densities.

As a vehicle moves in the city, it obtains the connectivity information of the surrounding streets and thus it can compute the most connected path up to a few junctions away (in our simulation seven junctions is the longest path). For longer paths, the last vehicle (which through the same mechanism has the connectivity information

of its surroundings) computes the remaining path and thus, scalability in terms of the number of junctions a packet can traverse is also obtainable.

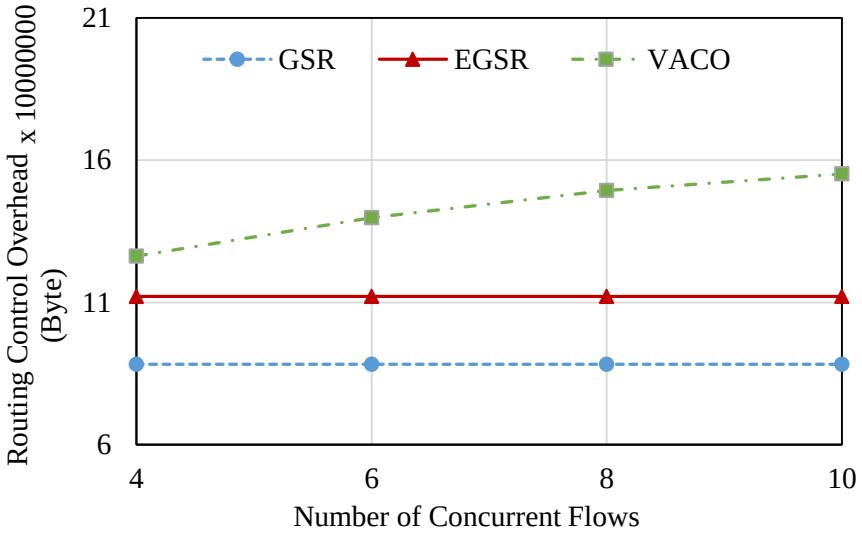

Figure 3.10 Overhead for different numbers of flows

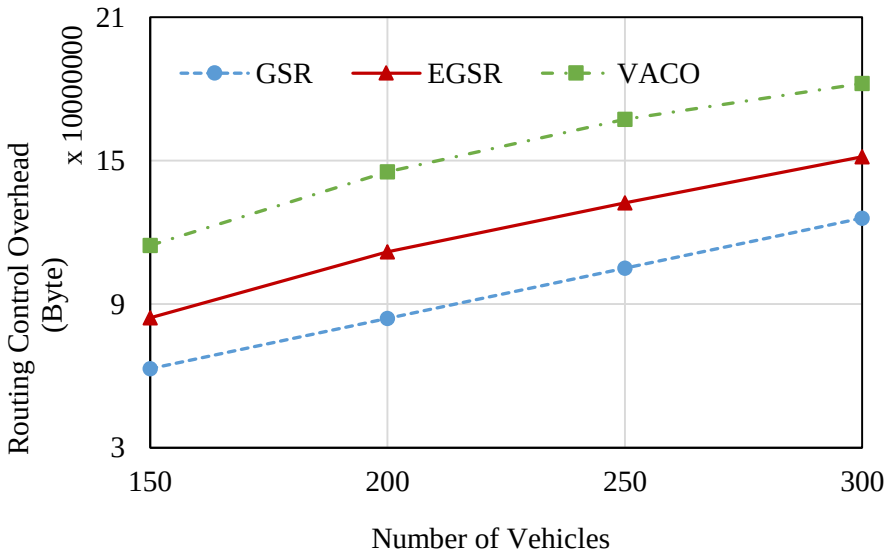

Figure 3.11 Overhead for different numbers of vehicles

3.3.3 End-to-End Delay

Figure 3.12 shows the average end-to-end delay against Euclidean distance between source and destination for data packets. GSR selects road segments to forward packets based only on the position of source and destination. Therefore, a road segment might

be congested and the packets face delay. VACO and EGSR consider delay to select road segments. In VACO, due to the required time to set up a route by reactive forward and backward ants, end-to-end delay is much higher than in EGSR and GSR. The higher delay of EGSR compared to GSR is due to successful delivery of packets for which the shortest path between their source and destination does not have connectivity. These packets traverse longer distance, leading to a higher average end-to-end delay.

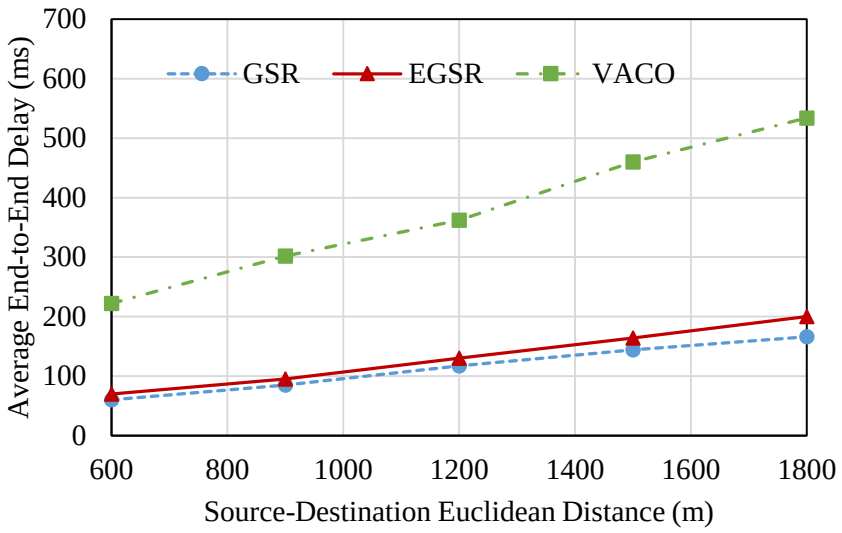

Figure 3.12 Average end-to-end delay for data packets

3.4 Summary

The proposed routing protocol (EGSR) has shown better performance than GSR and VACO. Its packet delivery ratio is at least 10% higher than that of the other protocols up to a speed of 70 km/h. By using small control packets, called ants, vehicles regularly evaluate the connectivity of the streets in their vicinity. In EGSR protocol, ants are broadcast by an efficient broadcasting mechanism to control broadcast storm.

The protocol is road-based and not sensitive to the movement of nodes. Moreover, it does not need additional hardware like traffic sensors or RSUs at every junction. By defining an area around every junction, called an anchor area and having vehicles in this area cooperate to launch ants, RSUs are not needed at every junction to evaluate

the connectivity of the streets. The increased overhead due to ants is not sensitive to the speed or number of vehicles and thus the protocol shows scalability.

In the proposed routing protocol, a constant beaconing rate of 2 Hz has been used. Figure 3.6 shows that this cannot provide correct information on vehicle position at high speed and so the performance of the routing protocol drops as the vehicles speed increases. The nominal transmission rate of beaconing in VANETs is 10 Hz, but vehicles might decrease it to below this rate [23]. It would seem that with a higher beaconing rate, vehicles can receive more up-to-date information on the positions of surrounding vehicles. However, in dense traffic environments as Figure 3.11 shows, a lot of bandwidth is consumed by beacons. Therefore, packet collision increases and consequently the number of successfully received beacons, reduces and, this has a reverse effect on vehicles' awareness of surrounding vehicles. Therefore, an adaptive beaconing scheme that adapts beaconing according to channel load and dynamics of vehicles could increase performance of a routing protocol. In the next chapters protocols for adaptive beaconing are presented.

Chapter 4

Beacon Rate and Awareness Control

In this chapter, a beacon rate and awareness control mechanism based on non-cooperative game theory [99] is proposed. Non-cooperative game theory deals with interactions among several entities that might have conflicting preferences. Every entity selects a strategy individually to increase its pay-off selfishly, while its pay-off is affected by other entities' strategies. This theory matches the problem we face with congestion control in VANETs. Every vehicle tries to work with the highest beaconing rate to make the surrounding vehicles aware of its presence. However, in situations with dense traffic, when every vehicle works with the highest rate, the level of awareness decreases due to loss of beacons. Interestingly, non-cooperative games do not rely on communication between nodes. Every node decides individually, and the whole network ends up at an equilibrium point. In a wireless network, this is a desirable characteristic because it results in scarce bandwidth saving. In our proposed congestion control mechanism, a price function [119] is used to limit bandwidth usage by each network node and reduce the beaconing rate in congested situations. The existence and uniqueness of the NE is proved, and the condition for the stability of NE is derived mathematically. A distributed method is used to find the equilibrium point of the congestion control mechanism.

Two features make this work distinctive from other congestion control mechanisms for VANETs. Firstly, the proposed mechanism does not need to share control information between vehicles for the operation or to achieve fairness. In other words, it is fully distributed and non-cooperative. This leads to saving of valuable network bandwidth and enhances robustness to error because it avoids control information exchanges over a wireless channel. The proposed mechanism achieves fairness based on the fairness concept of the NE. If there is no fairness at the equilibrium point, some vehicles can change their strategy unilaterally to obtain higher payoff, and this is in

contradiction with the NE point concept. Secondly, it provides an efficient congestion control mechanism that can satisfy safety application requirements [60], [61]. Bandwidth is shared among vehicles in proportion to their requirements, while fairness is achieved among vehicles with the same requirements. The proposed mechanism uses parameters that every vehicle can set individually without communicating with the other vehicles, and the entire system ends up being in the desired condition.

4.1 Non-Cooperative Beacon Rate Control

This section explains the non-cooperative beacon rate and awareness control in mathematical terms. Let $\mathcal{G} = \{\{\mathcal{N}, \{\mathcal{R}_i\}_{i\in\mathcal{N}}, \{\varphi_i\}_{i\in\mathcal{N}}\}$ denote the Non-cooperative beacon Rate and Awareness Control (NORAC) game, where $\mathcal{N} = \{1, \dots, n\}$ is the set of players (vehicles), and $\mathcal{R}_i$ is the set of possible beacon rates for player i and is called the strategy set of player i. The payoff function of player i is denoted by φ_i. The beacon rate $r_i \in \mathcal{R}_i$ is referred to as the strategy of player i. Each player i selects a strategy independently. The vector $\boldsymbol{r} = (r_1, r_2, \dots, r_n) \in \boldsymbol{R}$ denotes the selected beacon rates of all the players, where $\boldsymbol{R} = \prod_{i=1}^{n} \mathcal{R}_i$. The resulting payoff function for the ith player is given as $\varphi_i(\boldsymbol{r}) = \varphi_i(r_i, \boldsymbol{r}_{-i})$, where $\boldsymbol{r}_{-i}$ represents the vector consisting of the beacon rates of all the players except the ith player.

Every player creates a beacon with a rate between 1 and 10 Hz [23]. Thus, the strategy set of player i is $\mathcal{R}_i = [1, 10]$. The players create beacons to make aware other players of their presence. Higher awareness about a player enhances that player's safety. Thus, it should result in higher payoff. As explained in the beginning of this chapter, a higher beacon rate is desirable because it creates higher awareness under normal conditions, but it has a negative effect on awareness in congested situations. Then, the desirable payoff function would yield lower payoff with the same beacon rate in situations with high levels of congestion. To achieve this objective, the payoff function is modelled as the difference between a utility and a price function. Accordingly, the payoff for player i is as follows:

$$\varphi_i\left(r_i, \boldsymbol{r}_{-i}\right) = U_i\left(r_i\right) - P_i\left(r_i, \boldsymbol{r}_{-i}\right) = u_i \ln\left(r_i + 1\right) - \frac{c_i}{\left(1 - CBR_i\left(r_i, \boldsymbol{r}_{-i}\right)\right)} \tag{4.1}$$

where u_i and c_i are positive parameters, and $CBR_i(r_i, \boldsymbol{r}_{-i})$ is the channel busy ratio that player i senses, and it is a function of all players' beacon rates.

The first term ($u_i \ln(r_i + 1)$) in the payoff function is called utility, and it increases with increasing beacon rate and indicates the preference of players to have higher rate. In addition, in [120], it was proved this utility function leads to proportional fairness in data rate.

The second term $\left(c_i/\left(1 - CBR_i(r_i, \boldsymbol{r}_{-i})\right)\right)$ in the payoff function is the price function. Pricing [119] in computer networks is a way to motivate efficient use of network resources. When there is congestion in a network, an efficient pricing mechanism discourages resource usage by competitive nodes. This term is a function of CBR because CBR is a good indicator of successful information dissemination [44]; high CBR, results in poor inter-vehicle awareness. The price function becomes larger in scenarios with higher levels of congestion, resulting in a lower pay-off. Furthermore, it increases more rapidly at higher CBR values than at lower values, which leads to a faster decrease of rate in higher CBRs.

The marginal payoff of player i is $\nabla_i \varphi_i(\boldsymbol{r}) = \partial \varphi_i(\boldsymbol{r})/\partial r_i$ and the vector of marginal payoffs of all players is given as $\nabla \varphi(\boldsymbol{r}) = \left(\nabla_1 \varphi_1(\boldsymbol{r}), \nabla_2 \varphi_2(\boldsymbol{r}), \ldots, \nabla_{\boldsymbol{n}} \varphi_{\boldsymbol{n}}(\boldsymbol{r})\right)^T$ and its Jacobian as $G(\boldsymbol{r})$. For $CBR_i(\boldsymbol{r})$, the mathematical model developed in [121] (see Appendix A), given below, is used.

$$CBR_i\left(\boldsymbol{r}\right) = \sum_{j=1}^{n} h_{ij}\, r_j \tag{4.2}$$

where

$$h_{ij} = T_{frame} \times \frac{\Gamma\left(m, \dfrac{m\, C_{Tt}}{\Omega_{ij}}\right)}{\Gamma\left(m\right)} \tag{4.3}$$

$$\Omega_{ij} = \frac{P_t\, \lambda^2}{\left(4\pi\right)^2 d_{ij}^{\gamma}} \tag{4.4}$$

$\Gamma(.)$ is gamma function, $\Gamma(.,.)$ is upper incomplete gamma function, C_{Tt} is the threshold power level of carrier sense, P_t is transmitter power, d_{ij} is the distance between the jth and the ith players, m is Nakagami fading parameter, λ is the

wavelength, γ is the path loss exponent, and T_{frame} is the time required to send a BSM packet.

4.1.1 Nash Equilibrium

In this section, we prove the proposed NORAC game has a unique NE point and then derive a sufficient condition for stability of the NE.

4.1.1.1 Existence and Uniqueness of the NE

The game $\mathcal{G}$ has twice differentiable pay-off functions and according to [122] it is a submodular game if and only if:

$$\forall i, j \in N \; i \neq j \;\; \frac{\partial^2 \varphi_i}{\partial r_i \, \partial r_j} \leq 0 \tag{4.5}$$

For NORAC we have

$$\frac{\partial^2 \varphi_i}{\partial r_i \, \partial r_j} = -\frac{2 c_i h_{ii} h_{ij}}{\left(1 - CBR_i(\boldsymbol{r})\right)^3} < 0 \tag{4.6}$$

thus, it is a submodular game. In addition, according to Theorem 3.1 in [122] the set of equilibrium points of such game is not empty and a least and a greatest equilibrium point exist. Therefore, NORAC has at least one NE and we require to prove that the NE is unique. Equilibrium uniqueness is a desirable property in non-cooperative games because, in such games, players make their decisions independently, and in the case of several equilibriums, the game might end up at a non-equilibrium point [123].

Assume the greatest NE is $\boldsymbol{r_1} = (r_{11}, r_{12}, \dots, r_{1n})$ and the least is $\boldsymbol{r_2} = (r_{21}, r_{22}, \dots, r_{2n})$ thus, we have:

$$\forall i \in N, \; r_{1i} \geq r_{2i} \tag{4.7}$$

at equilibrium points:

$$\frac{\partial \varphi_i}{\partial r_i} = \frac{u_i}{r_i + 1} - \frac{c_i h_{ii}}{\left(1 - CBR_i(\boldsymbol{r})\right)^2} = 0 \tag{4.8}$$

thus,

$$\forall i \in N,\ r_{1i}+1=\frac{u_i\left(1-CBR_i\left(\boldsymbol{r_1}\right)\right)^2}{c_i h_{ii}} \quad \text{and} \quad r_{2i}+1=\frac{u_i\left(1-CBR_i\left(\boldsymbol{r_2}\right)\right)^2}{c_i h_{ii}} \tag{4.9}$$

Considering (4.7) and (4.9) we can write,

$$CBR_i\left(\boldsymbol{r_1}\right)\leq CBR_i\left(\boldsymbol{r_2}\right) \tag{4.10}$$

As CBR is an increasing function with respect to all r_i, (4.10) contradicts (4.7) unless we have:

$$\forall i \in N,\ r_{1i}=r_{2i} \tag{4.11}$$

thus, the NE of the game is unique.

4.1.1.2 Stability of the NE

In this section, we try to derive a sufficient condition for the stability of the NE. Theorem 9 in [124] proved that in a concave game, sufficient condition for stability of an equilibrium point is that the matrix $G(\boldsymbol{r})+G^T(\boldsymbol{r})$ be negative definite, where $G^T(\boldsymbol{r})$ is the transpose of $G(\boldsymbol{r})$. One way to guarantee stability of the equilibrium point of NORAC is to find the conditions for u_i and c_i such that the above matrix is always negative definite. Because this provides sufficient conditions for stability, it imposes great restrictions on the game parameters however using Fact 1 in [125], less restrictive conditions are obtained. According to Fact 1 in [125], the equilibrium point is stable under gradient projection method [126], if the corresponding Jacobian is a stable matrix at the NE. In NORAC, $-G(r)$ is an $n \times n$ matrix with elements:

$$g_{ii}=\frac{u_i}{\left(r_i+1\right)^2}+\frac{2\,c_i {h_{ii}}^2}{\left(1-CBR_i\left(\boldsymbol{r}\right)\right)^3} \tag{4.12}$$

and

$$g_{ij}=\frac{2c_i h_{ii} h_{ij}}{\left(1-CBR_i\left(\boldsymbol{r}\right)\right)^3} \quad i \neq j. \tag{4.13}$$

At equilibrium, $\nabla\varphi(\boldsymbol{r})=0$, which gives

$$\frac{u_i}{r_i+1}=\frac{c_i h_{ii}}{\left(1-CBR_i\left(\boldsymbol{r}\right)\right)^2} \tag{4.14}$$

thus,

$$1-CBR_i\left(\boldsymbol{r}\right)=\left(\frac{\left(r_i+1\right)c_i h_{ii}}{u_i}\right)^{1/2} \tag{4.15}$$

so the elements of the matrix can be written as

$$g_{ii}=\frac{u_i}{\left(r_i+1\right)^2}+\frac{2h_{ii}{u_i}^{3/2}}{\left(r_i+1\right)^{3/2}\left(p_i h_{ii}\right)^{1/2}} \tag{4.16}$$

and

$$g_{ij}=\frac{2h_{ij}\,{u_i}^{3/2}}{\left(r_i+1\right)^{3/2}\left(p_i h_{ii}\right)^{1/2}} \tag{4.17}$$

Therefore, we should find the condition that all eigenvalues of the matrix $-G(r)$ be positive. According to the Gershgorin circle theorem [127], the eigenvalues of a matrix lie in a circle centred at the diagonal elements, with a radius equal to the sum of the absolute values of the off-diagonal elements. Thus, a strictly diagonally dominant matrix with positive diagonal elements is positive definite. Alternatively, in [128], it was proved that a matrix with positive row averages and all off-diagonal elements bounded above by their corresponding row averages has a positive determinant. Such a matrix is called a B-matrix, and it is positive definite [129]. Using the results in [128], a weaker condition is obtained for stability of the NE compared to the one obtained using the former approach. Thus,

$$\forall i \quad \frac{1}{n}\times\left(\frac{u_i}{\left(r_i+1\right)^2}+\sum_{j=1}^{N}\frac{2h_{ij}{u_i}^{3/2}}{\left(r_i+1\right)^{3/2}\left(c_i h_{ii}\right)^{1/2}}\right)>\max_{j\neq i}\left(\frac{2h_{ij}u_i^{3/2}}{\left(r_i+1\right)^{3/2}\left(c_i h_{ii}\right)^{1/2}}\right) \tag{4.18}$$

then,

$$\forall i \quad \left(\frac{c_i h_{ii}}{u_i\left(r_i+1\right)}\right)^{1/2}>2\left(n\times\max_{j\neq i}\left(h_{ij}\right)-\sum_{i=1}^{N}h_{ij}\right) \tag{4.19}$$

Because h_{ij} is maximum when $j=i$ ($d_{ij}=0$ in (3.4)), $n\,h_{ii}-\sum_{i=1}^{n}h_{ij}$ is considered the upper bound of the right-hand side of (4.19). Then,

$$\frac{u_i}{c_i h_{ii}}<\frac{1}{4\left(r_i+1\right)\left(n\,h_{ii}-\sum_{i=1}^{N}h_{ij}\right)^2} \tag{4.20}$$

It worth noting that condition specified in (4.20) is a sufficient condition for the stability of the NE under gradient projection method, not a required one. Therefore, even if this condition is violated the NE might be stable.

4.2 Congestion Control Process

As has been indicated in section 4.1.1.2, the game is gradient-stable under sufficient condition (4.20). In NORAC game, every vehicle updates its beacon rate according to the gradient method as follows.

$$\frac{dr_i}{dt} = \frac{\partial \varphi_i}{\partial r_i} = \frac{u_i}{r_i + 1} - \frac{c_i h_{ii}}{\left(1 - CBR_i\left(\boldsymbol{r}\right)\right)^2} \tag{4.21}$$

From now on, $c_i h_{ii}$ is considered as a single parameter pc_i for simplicity. Algorithm 3.1 shows the NORAC mechanism. Where r_{max} and r_{min} are 10 Hz and 1 Hz, respectively. As Algorithm 3.1 shows, every vehicle updates its BSM rate according to the locally measured CBR in each iteration, and vehicles do not require to exchange control information.

Algorithm 3.1 Beaconing updates based on gradient method

---

1. Every vehicle measures CBR
2. Every vehicle updates the beacon rate according to:

$$r_i = \left[r_i + \frac{u_i}{r_i + 1} - \frac{pc_i}{\left(1 - CBR_i\left(\boldsymbol{r}\right)\right)^2} \right]_{r_{\min}}^{r_{\max}}$$

4.3 Selection of NORAC Parameters

In this section, we attempt to find a numerical representation of the right-hand side of inequality (4.20) and appropriate values of pc_i and u_i. For ease of referring to the right-hand side of (4.20), it is denoted by $O(u, pc)$. Most congestion control mechanisms have been tested in a network with a bit rate of 6 Mbps, since this bit rate provides a good trade-off between channel load and signal to noise requirements [25].

According to (4.3) and (4.4), $h_{ii} = T_{frame}$ (because $d_{ii} = 0$); thus, for VANETs with a bit rate of 6 Mbps and a maximum BSM size of 500 bytes [49], $h_{ii} = 6.6 \times 10^{-4}$. Both terms nh_{ii} and $\sum_{i=1}^{n} h_{ij}$ grow with increasing number of vehicles (n), but the first term grows faster. It is expected that the term $\left(n\, h_{ii} - \sum_{i=1}^{n} h_{ij}\right)^2$ would be a small number considering the value of h_{ii} and the term's power two. Thus, with the maximum $r_i = 10\ Hz$, the game has stable equilibrium for a broad range of $\frac{u_i}{pc_i}$, as shown in the simulation results.

The value of $\sum_{i=1}^{n} h_{ij}$ depends on network topology because according to (4.4), h_{ij} is a function of d_{ij}, so it is not possible to find its value generally. However, in order to get a sense of the value of $O(u, pc)$, it was computed using simulation for a scenario with n = 120 vehicles on a 300-m-long track with three lanes. Considering (4.2), it is evident the term $\sum_{i=1}^{n} h_{ij}$ is equal to CBR when all vehicles' beacon rate is 1 Hz, so it was measurable in the simulation. The minimum measured CBR by the vehicles in the scenario and the maximum $r_i = 10\ Hz$ were used to compute $O(u, pc)$ in order to obtain the minimum $O(u, pc)$. For such a scenario

$$\frac{u_i}{pc_i} < 109.5 \tag{4.22}$$

To evaluate the effect of u_i and pc_i on the CBR and beacon rate, the results of simulation performed for a track measuring 400 m in length and with n =159 vehicles are shown in Figures 4.1 and 4.2. All vehicles have the same pc_i and u_i. Figure 4.1 shows the results when u_i is constant and equal to 5 and pc_i has values of 0.1, 0.3, 0.7, and 1. As expected, an increase in pc_i increases the price of using bandwidth; then, players use lower beacon rates, so CBR is controlled to a lower level.

In Figure 4.2, pc_i is constant and equal to 0.2 and u_i has different values of 1, 3, 5, and 20. By increasing u_i, the algorithm ends up with higher CBR and beacon rate because the players' payoff increases according to (4.1). Figure 4.2, also shows that for $pc_i = 0.2$ and u_i between 5.0 and 20 CBR is controlled within the desired range. These values are used in the simulation runs reported in the next sections too.

Figure 4.3 shows the beacon rate in every iteration of NORAC when all vehicles have a beacon rate of 10 Hz at the start of the simulation. For every pair of u_i and pc_i,

changes in the beacon rate are shown for two vehicles—one at the middle of the scenario (x = 205) and one at the edge (x = 0). For larger values of u_i and pc_i, the algorithm converges faster. As an example, with $pc_i = 0.2$ and $u_i = 5$, it converges in fewer than 10 iterations.

In the formulation of the mechanism, it was never assumed that u_i and pc_i are equal for players. Thus, every vehicle can select its parameters individually according to its safety application and awareness requirements and yet congestion is controlled. This is demonstrated in the simulation results reported in the next sections.

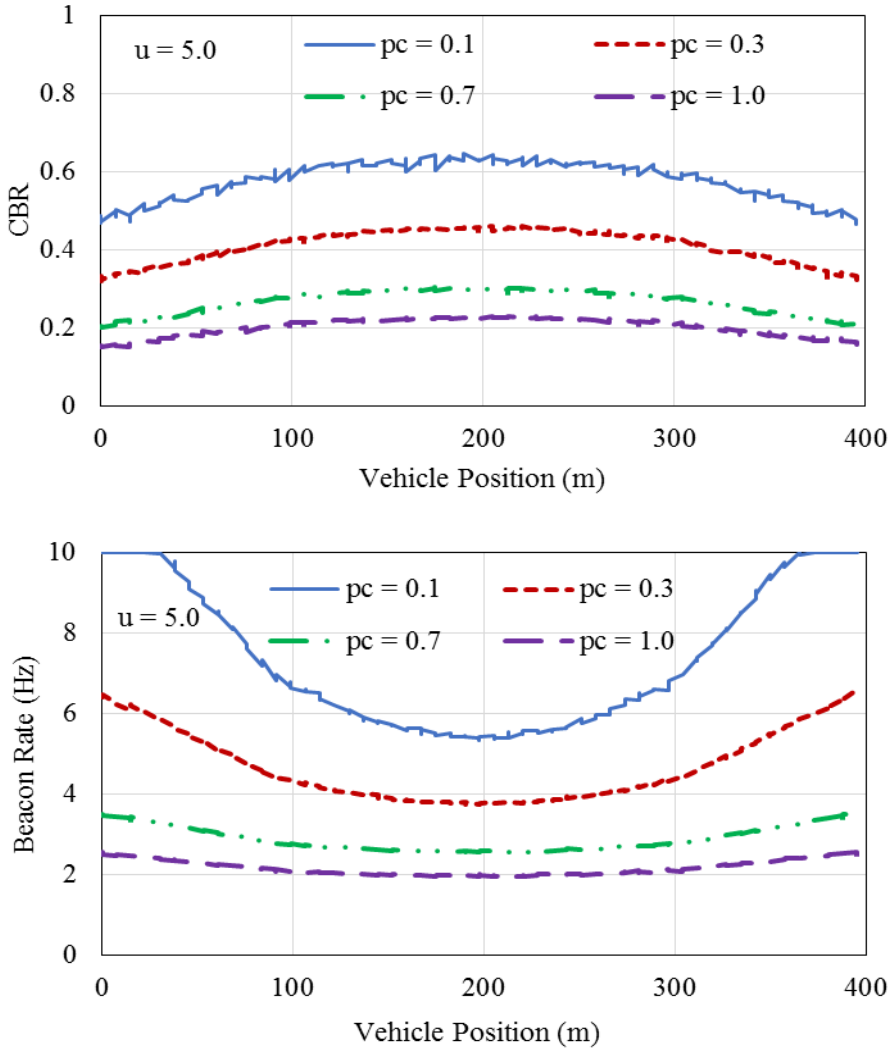

Figure 4.1 Beacon rate and CBR for a track measuring 400 m in length with total n =159 vehicles. Effect of changes in price when utility factor is constant and equal to 5

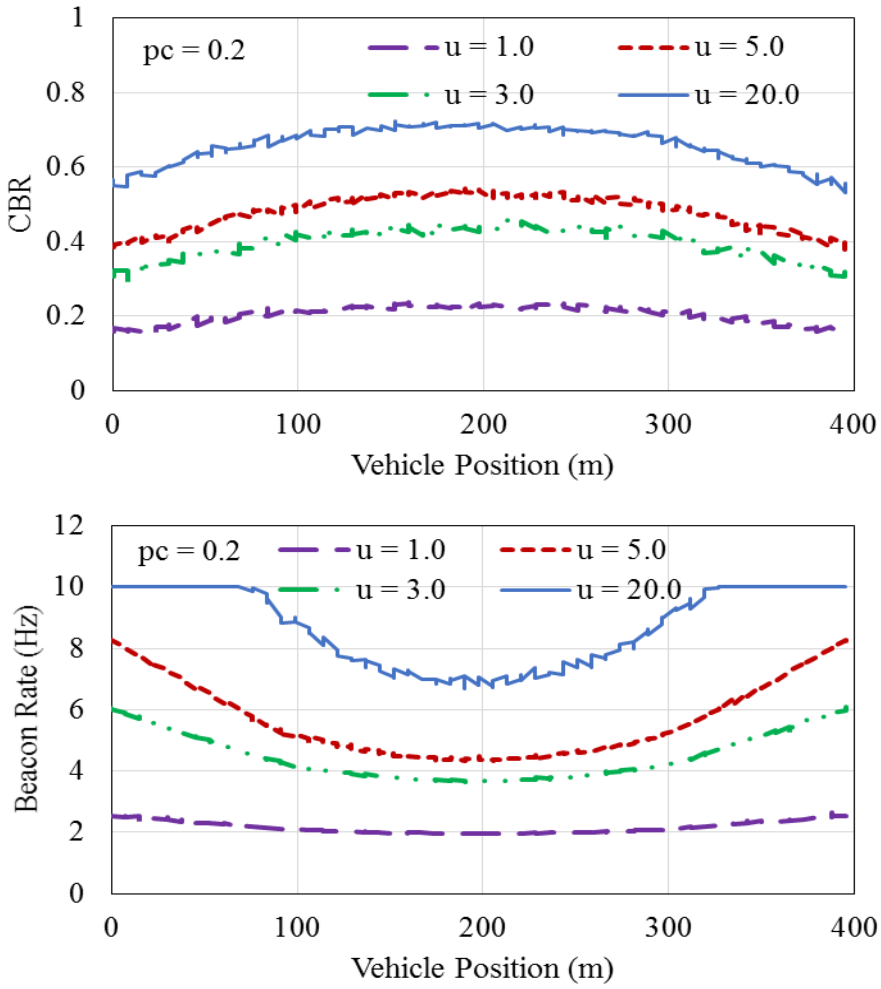

Figure 4.2 Beacon rate CBR for a track measuring 400 m in length with total n =159 vehicles. Effect of changes in utility when price factor is constant and equal to 0.3

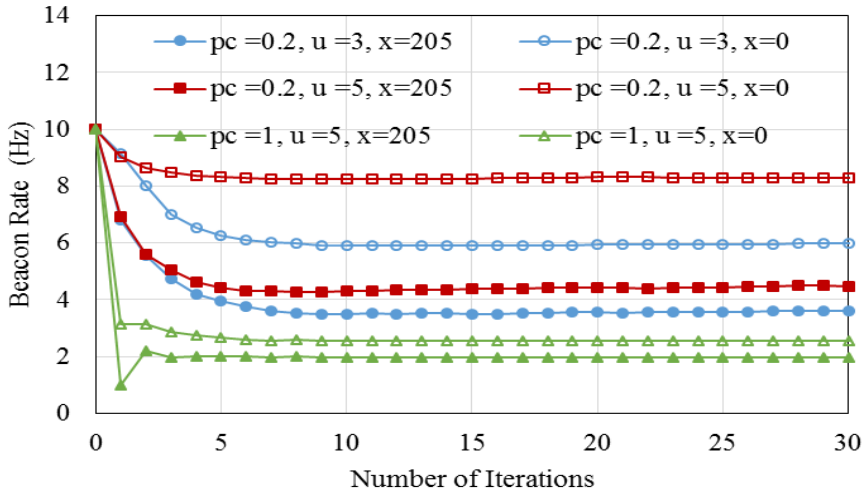

Figure 4.3 Beacon rate updates for vehicles at x = 0 m and x = 205 m for different values of pc and u

Table 4.1 Simulation parameters

Parameter	**Value**
Communication Range	300 m
MAC Protocol	IEEE 802.11p
Carrier Frequency	5.89 GHz
Bit Rate	6 Mbps
Beacon Size	500 bytes
pc_i	0.2
u_i	5.0 unless specified otherwise
Sampling Time	500 ms
Propagation Model	Nakagami m = 2.0
α (FABRIC)	1, 2
β (FABRIC)	2.8×10^{-5}
π_i^0 (FABRIC)	1.252×10^{-3}
Maximum Channel Capacity (FABRIC)	781.25 beacons/s
Anti-flapping Parameter (FABRIC)	0.022
α (LIMERIC)	0.1
β (LIMERIC)	1/150
Gain Saturation Parameter X (LIMERIC)	0.005

4.4 Performance Evaluation

The performance of NORAC was evaluated in several high-density scenarios using OMNeT++ as network simulator and SUMO for generating traffic mobility. The simulation parameters are summarized in Table 4.1. FABRIC [50] and LIMERIC [49] were selected for comparison. FABRIC is one of the most recent works in this area, and it works based on network utility maximization. Every vehicle piggybacks information such as the current beaconing rate and the computed Lagrange multiplier on its beacons. Vehicles use this information from their one hop neighbours to update their rates and Lagrange multipliers. LIMERIC is based on linear control, where each vehicle measures CBR locally and adapts its rate linearly with respect to the difference between the current channel load and the desired. LIMERIC was selected for

comparison because, similar to NORAC, vehicles do not communicate their algorithm parameters with each other.

The parameters of LIMERIC and FABRIC are the same as those suggested in [49] and [50]. Parameters u_i and pc_i have been selected so that the congestion level is controlled between 0.4 and 0.8, and a reasonable speed of convergence is achieved. For simplicity, in all simulations in this section, pc of all vehicles was considered constant and equal to 0.2 and that the vehicles change only their u parameter.

The results in the previous section were obtained under the assumption that vehicles are synchronized and update their rates at the same instant. This is a valid assumption because in VANETs, devices should be synchronized, and it has been considered in IEEE Std 1609.0-2013 [22] and IEEE Std 1609.4-2016 [26]. This synchronization is necessary for multi-channel operation and security purposes, and it can be achieved by GPS, as has been mentioned in said standards. However, conventionally, congestion control mechanisms have been tested under asynchronous conditions too. From now on, we assume that vehicles are not synchronized. The simulation results show that asynchronous update simply increases the convergence time of NORAC, but it is still faster than the other mechanisms selected for comparison.

4.4.1 Single-hop Scenario

The first scenario is a single-hop scenario with n = 120 vehicles on a 3-lane track measuring 300 m in length and with homogenous distribution of vehicles. With a communication range of 300 m all, vehicles are within range of each other. Figure 4.4 shows the beacon rate and CBR of the vehicles after convergence. While all congestion control algorithms control CBR well, LIMERIC is not fair in beacon rate allocation. As the Figure shows, vehicles have different beacon rates ranging from 4 to 10 Hz. The unfairness of LIMERIC has been indicated in [53], too. In NORAC similar to LIMERIC vehicles do not exchange the algorithm parameters, but NORAC is superior in terms of fairness. In FABRIC, with both $\alpha = 1$ and 2, all vehicles converge to the same rate, however convergence is faster with $\alpha = 1$. Moreover, because all vehicles are within the range of each other, they can receive parameters of all other vehicles;

thus, they converge to the same beacon rate, as shown in Figure 4.4. NORAC has good fairness, too, with beacon rates between 5.5 and 7 Hz all over the track.

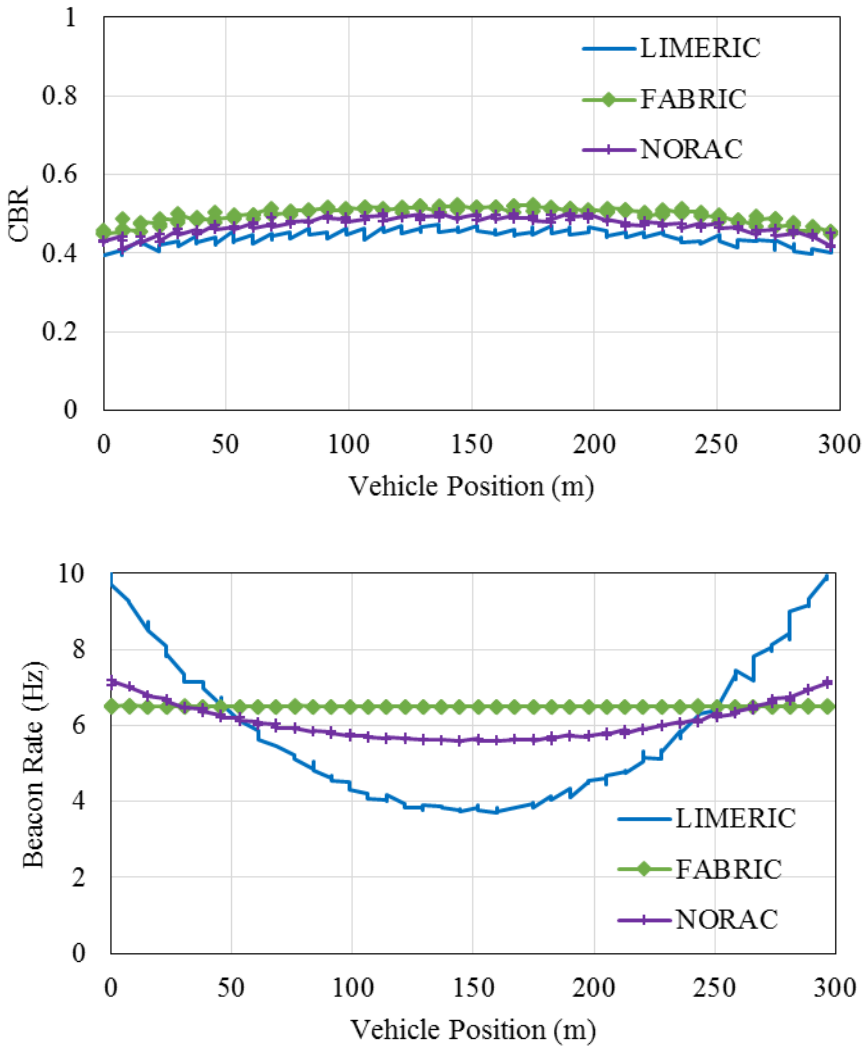

Figure 4.4 Beacon rate and CBR for a single-hop scenario with 120 vehicles

Figure 4.5 shows the changes in the beacon rate of a vehicle at position x = 152 m (almost the centre of the track). Despite its unfairness, LIMERIC converges in 20 iterations. NORAC converges in fewer than 15 iterations, which with a sampling time of 500 ms, it is equal to 7.5 s. Actually, after the first few iterations of the algorithm, the beacon rate is very close to the final value, which signifies the congestion level is controlled rather rapidly. This makes NORAC suitable for congestion control in dynamic VANETs scenarios. There is a jump in beacon rate in the first iteration of FABRIC because it is updated in every step as

$$r_i = \frac{1}{\left(\sum_n \pi_i\right)^{\frac{1}{\alpha}}} \tag{4.23}$$

Thus, the size of this jump depends on the initial values of $\pi_i(\pi_i^0)$, α, and the number of vehicles. The recommended value of π_i^0 in [50] is given in Table 4.1. Using this value, at the first step, the algorithm for $\alpha = 1$ jumps to a point close to the final rate for this scenario and converges fast. Assuming that all vehicles have equal π_i^0 is not realistic because vehicles change their π_i over time, and when they contribute to a congestion control scenario, they might have different π_i than the recommended value.

In FABRIC, the beacon rate updates with $\alpha = 1$ when every vehicle i has a random value of π_i^0 between 0.001252 and 2×0.001252, was shown in Figure 4.5 for comparison. At every step in FABRIC, π_is is increased or decreased by β, which is a very small number (2.8×10^{-5}), and this generally results in a high number of iterations before convergence, and convergence speed becomes heavily dependent on π_i^0. In subsequent simulations, it is still assumed that π_i^0 values are identical and equal to the recommended value, which seems to result in the best convergence time for FABRIC.

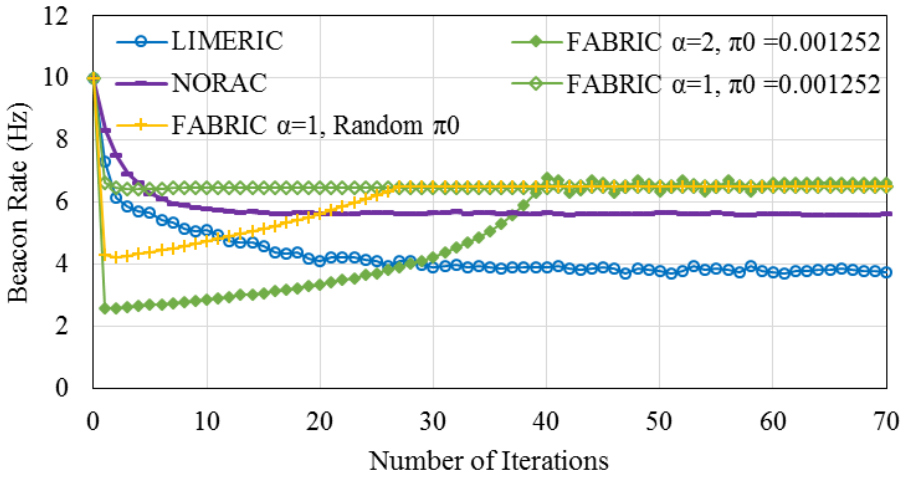

Figure 4.5 Beacon rate against number of iterations of the algorithms for a vehicle at x = 152 m on a track measuring 300 m in length

4.4.2 Static Multi-Hop Scenarios

This scenario comprises a track measuring 1000 m in length with three lanes and 399 vehicles distributed homogenously along the track. Figure 4.6 shows beacon rates and CBR of the algorithms. The unfairness of LIMERIC worsened in this scenario, but it did control congestion efficiently and converged in about 30 iterations.

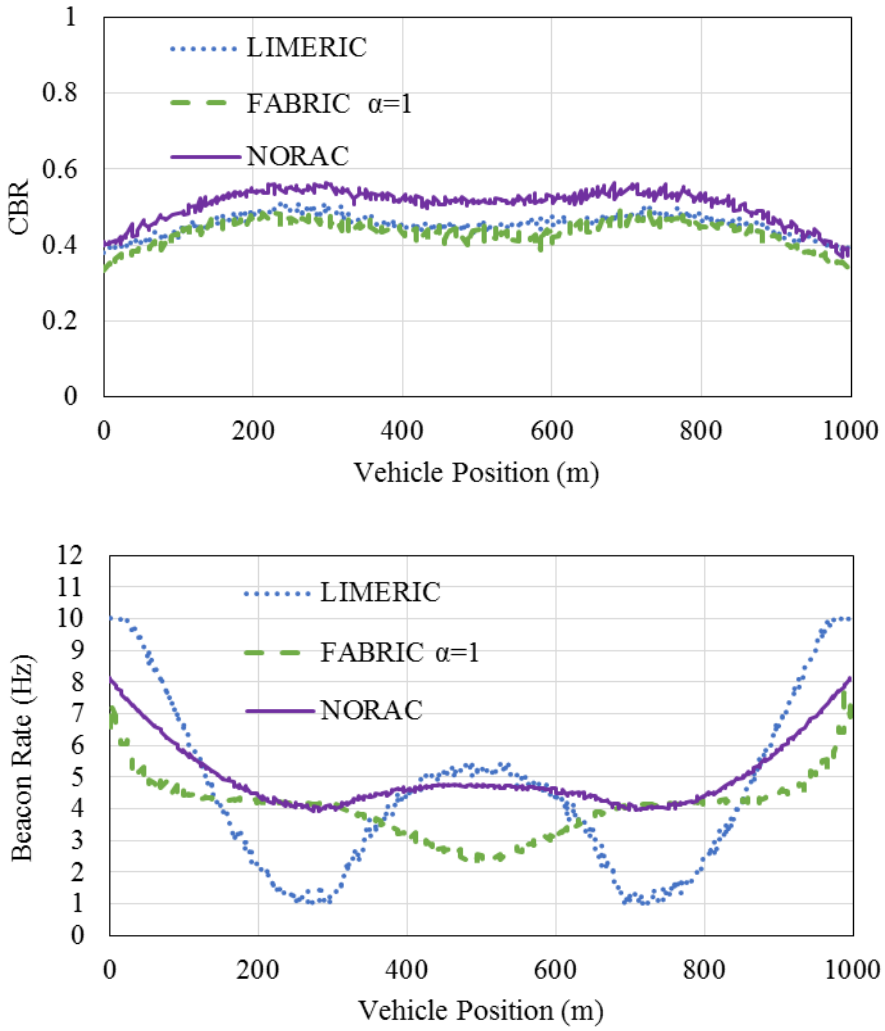

Figure 4.6 Beacon rate and CBR for multi-hop scenario. n =399 vehicles on a track of length 1000 m with three lanes

With both $\alpha = 1$ and $\alpha = 2$, FABRIC almost converges to the same beacon rate and CBR after an adequate number of iterations. Hence, to keep the figures clear, only the results for $\alpha = 1$ have been shown. FABRIC loses its fairness when the scenario is longer than one hop. Because vehicles communicate their parameters with their one-

hop neighbours and the rate that they achieve cannot be fair as in one-hop scenario. Similar to the previous scenario, FABRIC with $\alpha = 1$ is faster than $\alpha = 2$, but as was explained, convergence speed depends on the initial values of π_i, which are not controllable in realistic situations.

With NORAC, vehicles all over the track converge to a rate between 4 and 8 Hz. Vehicles far enough from the ends of the track, which sense almost the same CBR, have rates between 4 and 5 HZ.

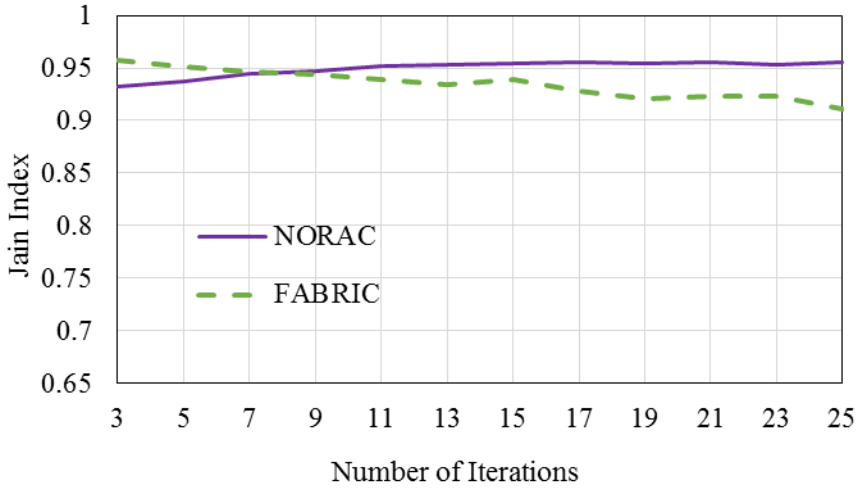

Figure 4.7 Jain Index for NORAC and FABRIC against the iteration of the algorithms

Figure 4.7 shows the Jain Index [59] at different iterations of NORAC and FABRIC with α =1. The Index was computed using the following formula.

$$Jain\ Index = \frac{\left(\sum_{i=1}^{n} r_i\right)^2}{n \times \sum_{i=1}^{n} r_i^2} \tag{4.24}$$

When all the nodes have the same rate (maximum fairness) the Index would be 1 otherwise it is less than 1 and higher values shows better condition regarding fairness. At the beginning of the experiment all the nodes have the rate of 10 Hz thus the Jain Index is 1 for both the algorithms. After enough iterations, the Jain Index for NORAC is 0.96 and for FABRIC is 0.91 which shows better fairness with NORAC. Furthermore, the figure shows after the seventh iteration there is no considerable

change in vehicles rate for NORAC while change in rates continues for a longer time with FABRIC. Figure 4.8 shows changes in the beacon rate of a vehicle at the centre of the track at position x = 501 m. NORAC converges faster than the other algorithms considered herein.

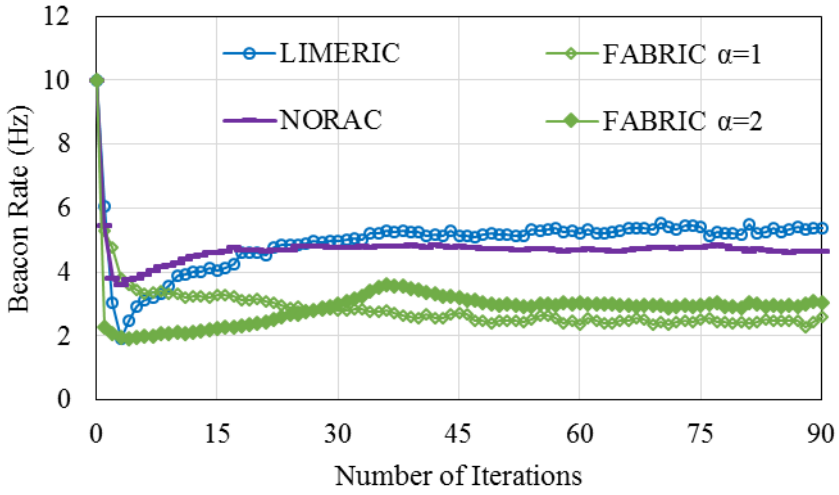

Figure 4.8 Beacon rate against number of iterations for the three algorithms for a vehicle at x = 501 m on a track of length 1000 m

Figures 4.9 and 4.10 show the same results for n = 792 vehicles on a track of length 1500 m with four lanes. Again, for vehicles far enough from the ends of the track, NORAC is fairer in terms of beacon rate. Both FABRIC and NORAC can control CBR below 0.6 however NORAC is more efficient than FABRIC in using available bandwidth.

Figure 4.10 shows the beacon rate updates of the algorithms when the initial beacon rate is 10 HZ. It also shows the result for NORAC when the initial rate is 1 Hz. In both conditions, NORAC converges in about five iterations. The results for LIMERIC are not shown because it fails to provide fairness.

Figure 4.11 shows beacon rate for the same scenario when vehicles have random initial rate. For each algorithm, rate of two vehicles V1 and V2 at the middle of the track are shown. Vehicle V1 initial rate is almost 4.5 Hz and vehicle V2 initial rate is almost 10 Hz. NORAC converges in 10 iterations and after that, beacon rates are almost constant while, with FABRIC beacon rates change for a longer time.

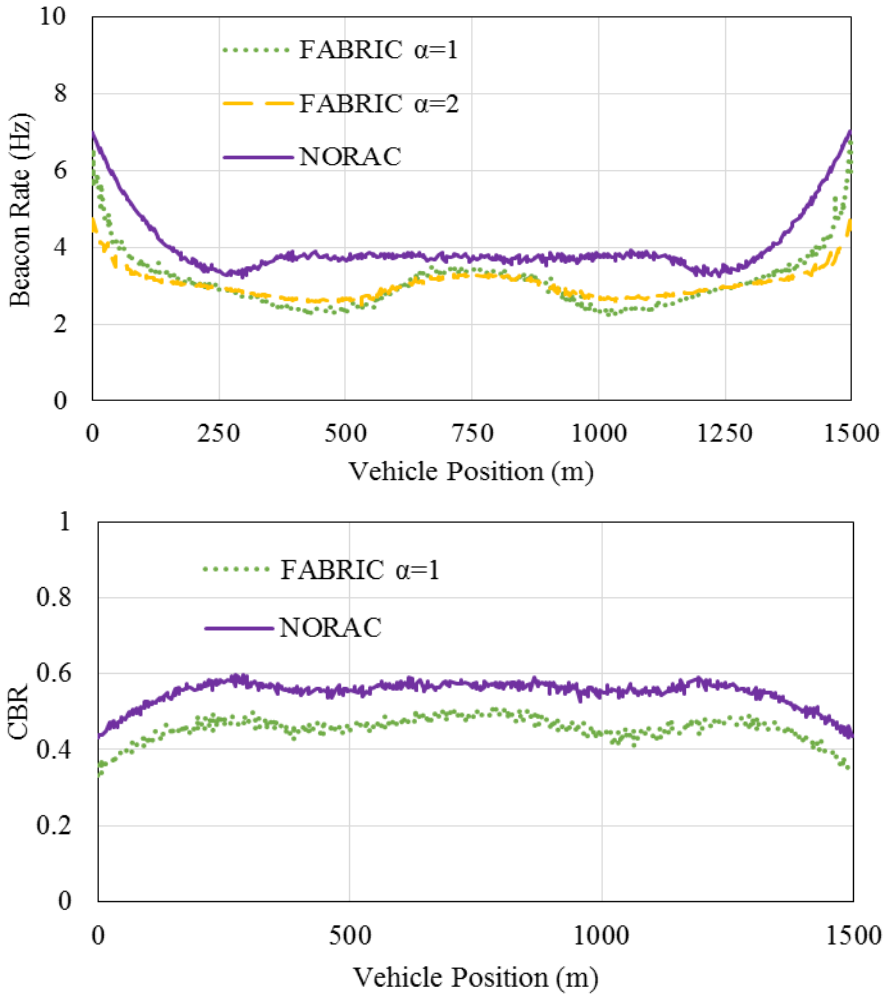

Figure 4.9 Beacon rate and CBR for a multi-hop scenario with 792 vehicles on a track of length 1500 m with four lanes

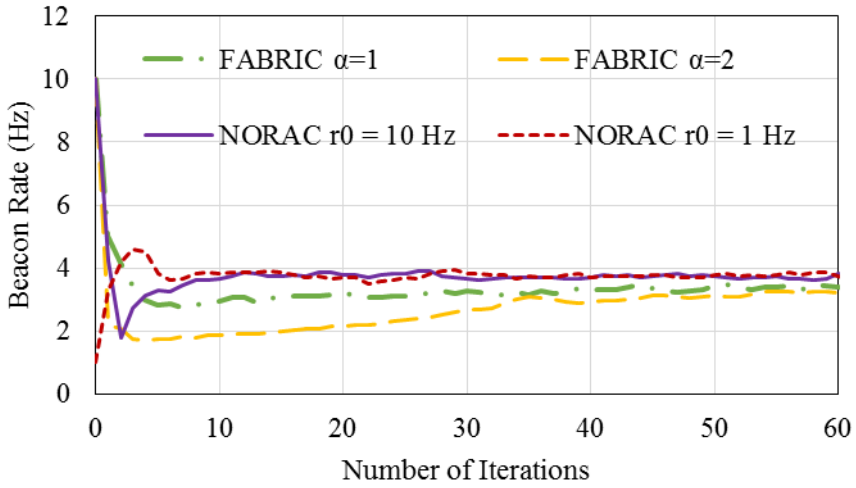

Figure 4.10 Beacon rate against number of iterations for NORAC and FABRIC for a vehicle at x = 752 m on a track of length 1500 m with four lanes and 792 vehicles

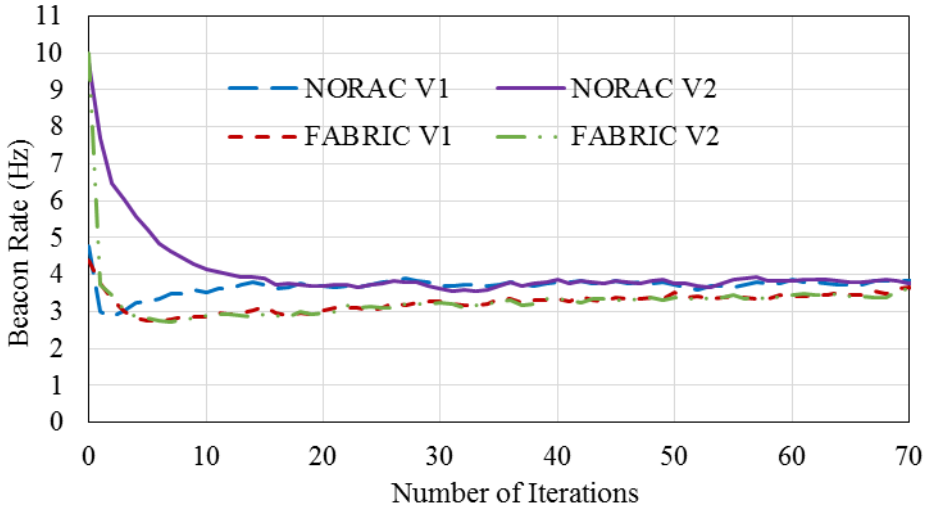

Figure 4.11 Beacon rate against number of iterations of NORAC and FABRIC (α=1) for vehicles V1 and V2 at x ≈ 750 m on a track of length 1500 m with four lanes and 792 vehicles when the vehicles have random initial beacon rate

For the above scenarios, average Information Dissemination Rate (IDR) [44] over all the vehicles has been compared in Figure 4.12. IDR indicates how many beacons per seconds are delivered successfully. In Figure 4.12, the scenarios with track lengths 300 m, 1000 m and 1500 m are indicated with scenarios 1, 2 and 3, respectively. Only in scenario 1 does FABRIC have higher IDR than NORAC.

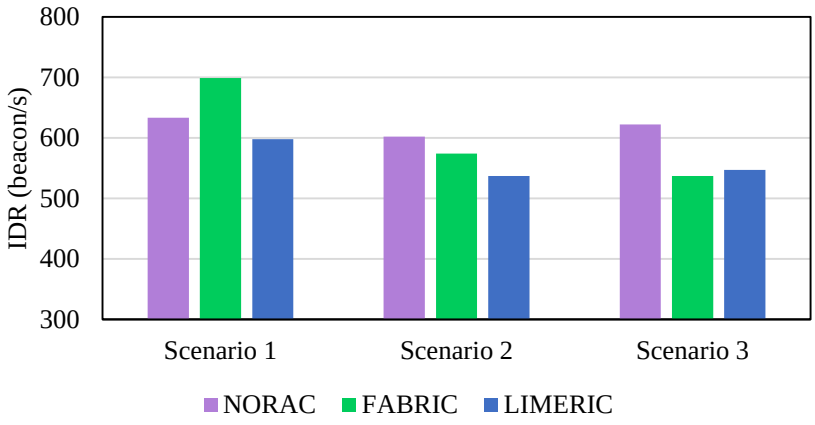

Figure 4.12 IDR for the static scenarios

4.4.3 Dynamic Scenario 1

Thus far, u_i was assumed to have the same constant value for all vehicles. However, this parameter (and pc_i) can be set per-vehicle to meet the application and awareness

requirements of that vehicle, which might be different from those of others. For example, in a dynamic scenario, it is desirable that vehicles with higher speed use a higher beaconing rate to create a higher level of awareness. Vehicles moving at a speed of 15 m/s change their positions thrice as fast as vehicles moving at a speed of 5 m/s. Then, in a congested scenario where the congestion level should be maintained around 0.65, vehicles moving at a higher speed should have a higher beaconing rate. In this section, it is shown that NORAC has awareness control property, too. To this end, the following simple function is used instead of constant u_i.

$$u_i = \lfloor v_i \rfloor_4 \tag{4.25}$$

where v_i is the speed of vehicle i. In this way, every vehicle sets its utility parameter equal to its speed, and the minimum value of the utility parameter is 4. The minimum value was selected based on the experiments in section 4.3 and to prevent vehicles with very low speed from always using the minimum beaconing rate (1 Hz). Two important points are worth noting. First, u_i can be selected by the application layer as a function of speed, acceleration, or even based on vehicle position (for example, at junctions, a higher beacon rate is desirable). The utility based on (4.25) is selected as an example to show how the algorithm functions. The design of u_i and pc_i could be based on the application requirements and is out of the scope of this work. Second, every vehicle can set its parameters u_i and pc_i individually and need not communicate it with other vehicles while the algorithm works well.

Three different scenarios are compared to show how NORAC can control awareness. All scenarios comprise a track of length 1200 m with vehicles moving at speeds of 0, 10, 15, and 20 m/s. The first scenario has two lanes with stationary vehicles (316 stationary vehicles) and three lanes with vehicles moving at speeds of 10, 15, and 20 m/s; the vehicles set their u_i according to (4.25). In the second scenario, there are twelve lanes, six of them are with stationary vehicles (3 × 316 stationary vehicles). The vehicles use the same u_i as in the first scenario. The third scenario has the same number of vehicles as in the first scenario but with $u_i = [v_i/2]_4$. The beacon rates and CBRs of these scenarios are shown in Figures 4.13–4.15, respectively.

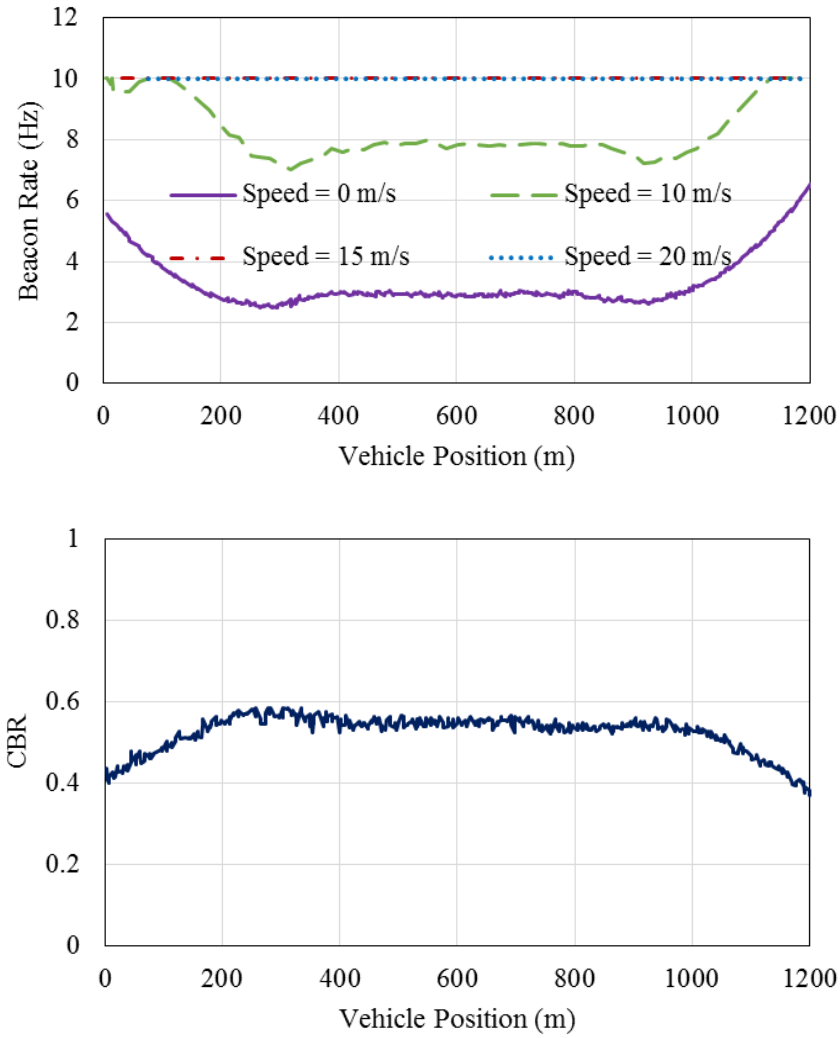

Figure 4.13 Beacon rate and CBR for a track of length 1200 m with two lanes of stationary vehicles - vehicles in the various lanes have speeds of 0, 10, 15, and 20 m/s and $u_i = [v_i]_4$

In the three aforementioned scenarios, congestion was controlled efficiently and in each of them, vehicles with higher utilities (speeds) achieved higher beacon rates. Moreover, fairness in beaconing rate was maintained among vehicles with the same utility. In the first scenario, vehicles with speeds of 15 and 20 m/s did not contribute to congestion control because their utility was higher than those of the others, and congestion was not so high as to warrant their contribution. This can be explained as follows: at the NE point for every vehicle,

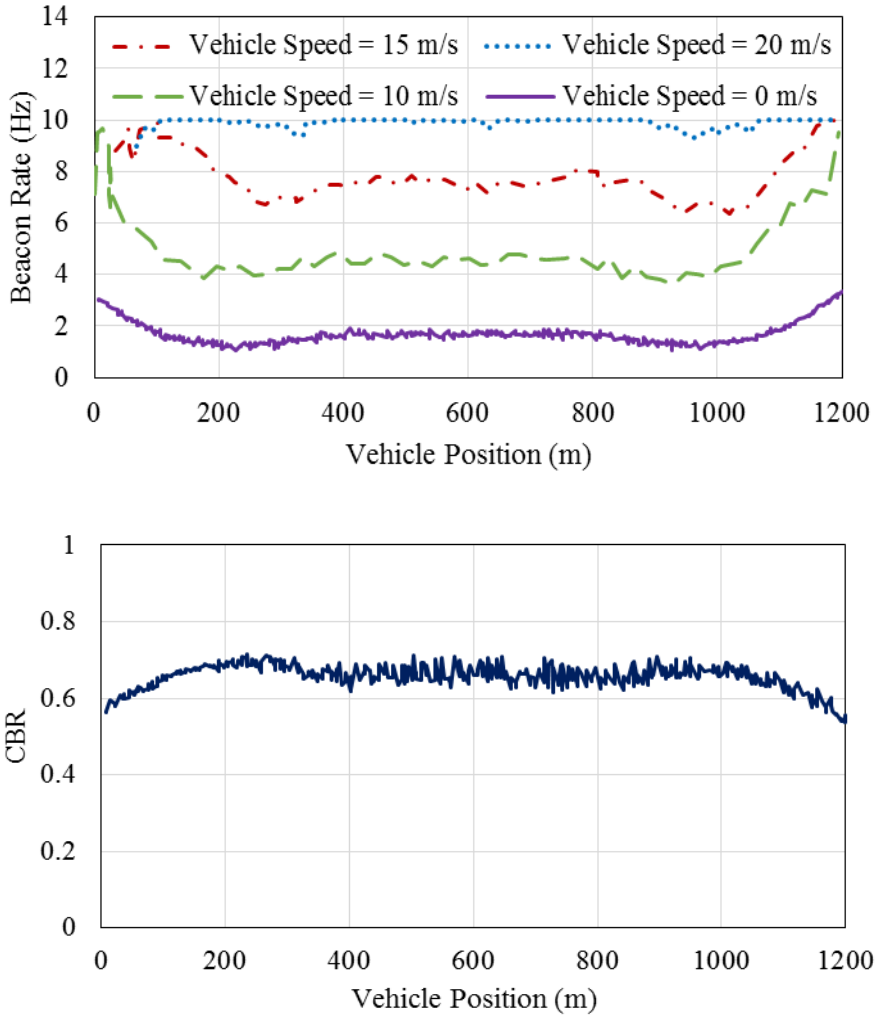

Figure 4.14 Beacon rate and CBR for a track of length 1200 m with 12 lanes - vehicles have different speeds of 0, 10, 15, and 20 m/s and $u_i = [v_i]_4$

$$\frac{\partial \varphi_i(\mathbf{r})}{\partial r_i} = 0 \tag{4.26}$$

thus,

$$r_i + 1 = \frac{u_i \left(1 - CBR_i(\mathbf{r})\right)^2}{pc_i} \tag{4.27}$$

Thus, for vehicles i and j with the same measured CBR,

$$\frac{r_i}{r_j} \approx \frac{u_i pc_j}{u_j pc_i} \tag{4.28}$$

For these scenarios, the same pc_i was used for all vehicles. Thus, the following relation is expected for the beacon rates of the vehicles at the same x-position because, those vehicles sense the same CBR.

$$\frac{r_i}{r_j} \approx \frac{\lceil v_i \rceil_4}{\lceil v_j \rceil_4} \tag{4.29}$$

In the first scenario, (4.29) is true for vehicles with speeds of 0 and 10 m/s. For example, at position x = 600 m, these vehicles have beacon rates of almost 3 Hz and 8 Hz, respectively. The ratio of these rates is proportional to the ratio of their utilities. For vehicles moving at higher speeds, the beacon rate is constrained by the algorithm to the maximum accepted rate (10 Hz).

In the second scenario with a larger number of vehicles, the beacon rates of vehicles with speed of 20 m/s is still the maximum, and vehicles moving at speeds of 15 m/s contribute to congestion control by reducing their beacon rates to control the channel usage (almost around 0.65). The relation (4.29) is observable among the beacon rates of vehicles moving at speeds of 0, 10 and 15 m/s.

In the third scenario, where vehicles have smaller utility than those in scenario 1, vehicle with speeds of 0, 10, 15 m/s contribute to congestion control in smaller CBRs. In Figure 4.15, relation $\frac{r_i}{r_j} \approx \frac{\lceil v_i/2 \rceil_4}{\lceil v_j/2 \rceil_4}$ is observed among the beacon rates of vehicles with speeds of 0, 10, 15 m/s as none of these rates has the extreme values.

These experiments show that with NORAC, vehicles can share bandwidth based on their application requirements (utility parameter), while congestion is controlled to a desired level and fairness is ensured among the vehicles with the same requirements (utility).

In LIMERIC, the parameters α and β can be set per-vehicle [130], so vehicles with different parameters converge to different rates. However, in this situation, fairness is worse than that in the case where all vehicles have the same parameters, and the results are not comparable to those of NORAC. FABRIC has parameter α, but the above experiments showed that for different values of α, vehicles converged to almost the same beacon rate with different convergence times. Moreover, for larger α such as

$\alpha = 6$, even for the static scenarios in sections 4.4.1 and 4.4.2, the beacon rate oscillates; thus, FABRIC is not comparable with NORAC in this aspect.

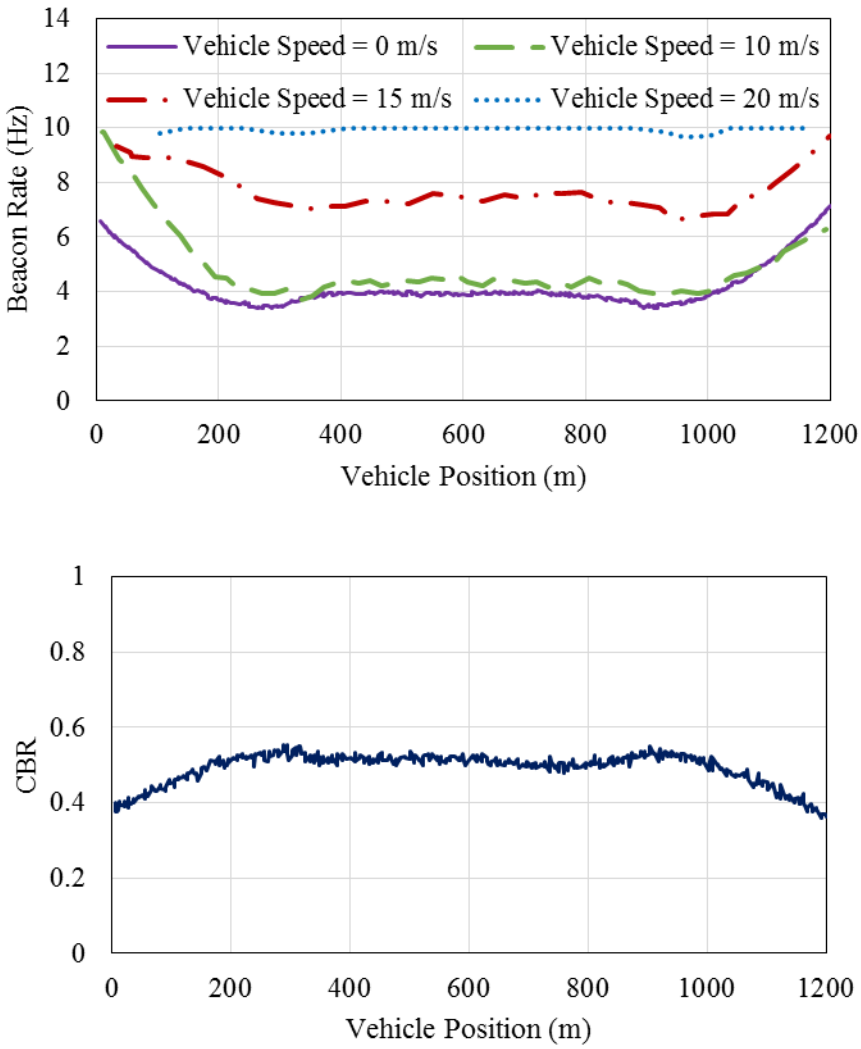

Figure 4.15 Beacon rate and CBR for a track of length 1200 m - vehicles have different speeds of 0, 10, 15, and 20 m/s and $u_i = [v_i/2]_4$

4.4.4 Dynamic Scenario 2

In this scenario, two clusters of vehicles with the number of vehicles 63 and 80 at speeds of 15 m/s and 10 m/s, respectively, move toward each other on a highway of length 1200 m. The utility introduced in (4.25) was used for this scenario too. Figure 4.16 shows the beacon rate and the congestion level at different times. At $t = 5\ s$, the vehicles in both the clusters use their maximum beaconing rate. At this

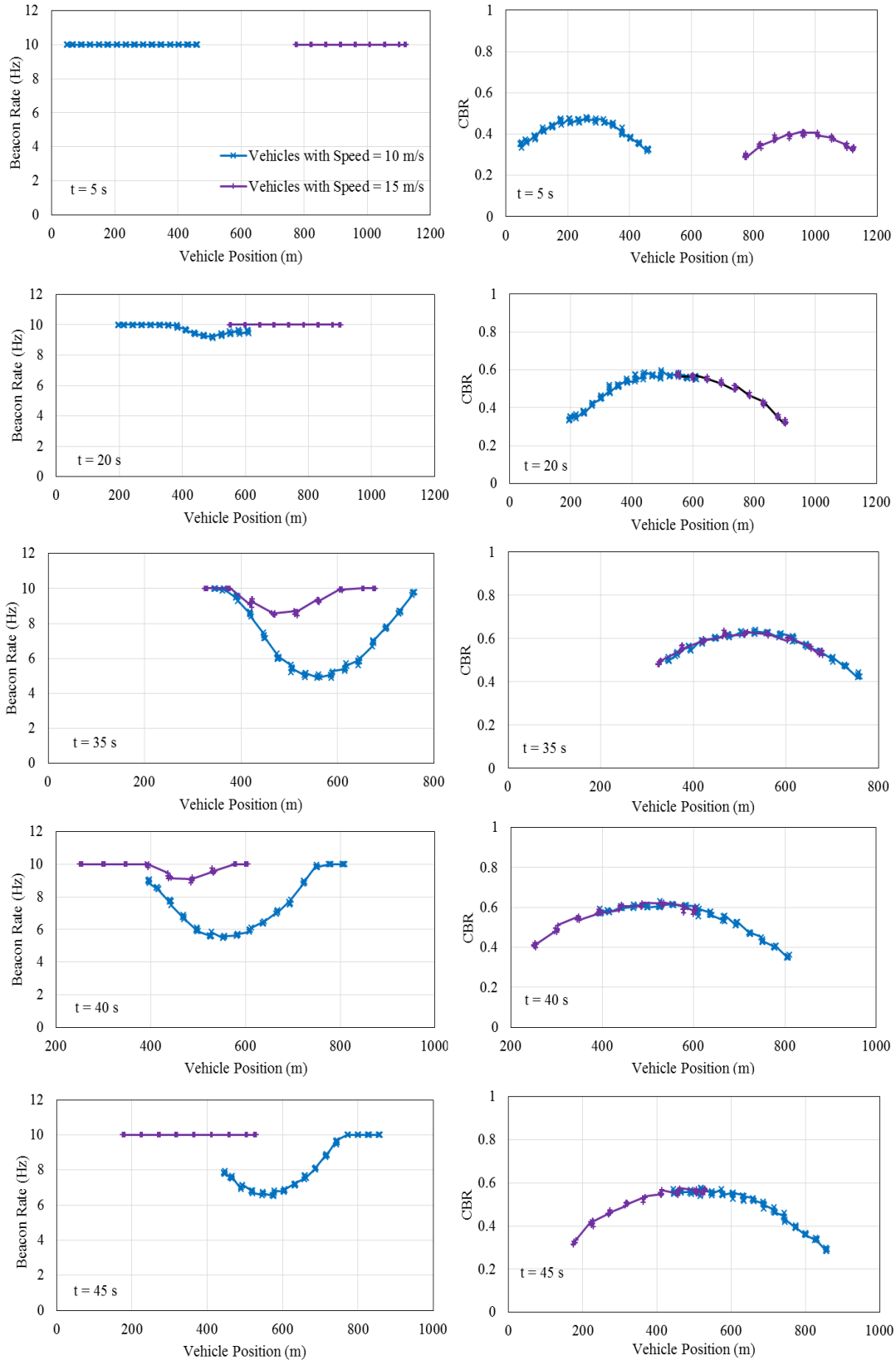

Figure 4.16 Beacon rate and CBR for two clusters of vehicles with speeds of 10 and 15 m/s and $u_i = [v_i]_4$

time, the two clusters are far enough to have not any effect on each other's CBR and the number of vehicles in each cluster is not so large that they require to reduce their beacon rates. As the two clusters move closer, first, the vehicles in the cluster with the lower utility start to reduce their rate as it is indicated in the figure at $t = 20\ s$. At $t = 35\ s$, where the clusters have the maximum overlap, (4.29) can be observed for the nodes in the middle of the scenario. When the clusters move farther apart, the vehicles with the higher utility increase their rate earlier than those with the lower utility. While bandwidth is shared between two clusters in a manner proportional to their utilities, CBR is maintained below 0.65 throughout. This experiment indicates that NORAC is fast enough to be suitable for dynamic scenarios encountered in VANETs. Every vehicle as a non-cooperative player tries to maximize its payoff and changes its strategy (beacon rate) for this purpose, without exchanging information with other players, and the entire system ends up at a desirable condition (controlled CBR).

4.5 Summary

In this chapter, a beacon rate and awareness control algorithm called NORAC, based on a non-cooperative game, was proposed. A payoff function for the game was presented. The existence and uniqueness of NE was proved, and a sufficient condition for stability of NE was derived mathematically. The gradient method was used to find NE in a distributed way. The presented algorithm was tested in several static and dynamic scenarios and compared to state-of-the-art rate control algorithms.

In the comparison, characteristics such as fairness, efficiency in controlling congestion, and algorithm speed were considered. All the compared algorithms could control congestion at a desired level, although NORAC was considerably better in terms of fairness than the others. NORAC has a short convergence time both in static and dynamic scenarios. In the experiments, it converged in less than 20 iterations. In very rare cases, FABRIC can be faster than NORAC.

In addition to the above criteria used for comparison, which are the system outputs, NORAC is superior in its design to congestion control mechanisms that achieve fairness by exchanging information between nodes. Information exchange creates

overhead and makes the system error-prone. Moreover, it cannot always solve the problem. As the simulation results showed, when a scenario was longer than the range over which information can be shared, the unfairness problem appeared in FABRIC. Furthermore, in algorithms that exchange the information over more than one hop, the beacon rate or power of vehicles that do not contribute to congestion might be reduced unnecessarily. Such scenario has been discussed in [50].

NORAC achieves fairness because NE is unique and at the NE point, players with the same payoff function will have the same rate. If there is no fairness at the equilibrium point, some vehicles can change their strategy unilaterally to obtain higher payoff, and this is in contradiction with the NE point concept.

NORAC can also meet safety application requirements and assign a rate based on the requirement to every single vehicle, while controlling CBR and ensuring fairness among vehicles with the same requirement. This feature was evaluated in a number of dynamic scenarios where utility was a function of speed, so vehicles with higher speeds could achieve higher beacon rates.

Chapter 5

Non-Cooperative Beacon Power Control

Adapting beaconing power is another option to limit the channel usage to around 0.65 (ideally within a range between 0.4 and 0.8) [44] thus, this chapter specifically focuses on beaconing transmission power control. Several approaches have been proposed to reduce the beaconing power during the congested situations [45]-[48]. All of them rely on including excess information in beacons to obtain fairness while, in the previous chapter it was explained why this is not a good approach. In this chapter, the problem of beacon's power control is presented as a non-cooperative game. It is proven the NE exists for the game and that the NE is unique and an algorithm is presented to find the equilibrium point in a distributed manner.

Like other beacon power control approaches for VANETs, it is assumed that there is no power restriction and every node transmits its beacons with the maximum allowed power level. When there is congestion in the network, vehicles reduce their power level to prevent BSM loss due to collision.

The proposed approach differs from previous works in this area in two aspects: Firstly, the fairness is obtained whiteout exchanging information between nodes, which results in bandwidth saving. The fairness in this protocol is obtained based on the fairness concept of NE. Secondly, weighted fairness in power allocation is achieved which is useful to meet application requirements [61]. Some safety applications require that the status of vehicles be disseminated longer distances thus, assigning the same power to vehicles with different requirements cannot meet this goal. As an example of such situations similar to Chapter 4, a scenario in which there is traffic jam in one side and free flow of vehicles on the other side is considered. As the simulation results indicate, the proposed protocol can provide weighted fairness in such conditions.

5.1 Non-Cooperative Power Control Game

Let $\mathcal{G} = \{\mathcal{N}, \{\mathcal{P}_i\}_{i\in\mathcal{N}}, \{F_i\}_{i\in\mathcal{N}}\}$ denotes the Non-cooperative Power Control (NOPC) game, where $\mathcal{N} = \{1, \dots, n\}$ is the set of players (vehicles), and $\mathcal{P}_\mathrm{i}$ is the set of possible beaconing powers for player i. $\mathcal{P}_\mathrm{i}$ is called the strategy set of player i and the power $p_i \in \mathcal{P}_i$ is called the strategy of player i. Each player selects its strategy independently. The vector $\boldsymbol{p} = (p_1, p_2, \dots, p_n) \in \boldsymbol{P}$ shows the selected power of all the players, where $\boldsymbol{P} = \prod_{i=1}^{n} \mathcal{P}_i$. $\mathcal{F}_i$ is the payoff function of player i and is indicated as $F_i(\boldsymbol{p}) = F_i(p_i, \boldsymbol{p}_{-i})$, where $\boldsymbol{p}_{-i}$ denotes the vector consisting of the beacon powers of all the players except the ith player.

Every vehicle transmits its beacons with a power between 1 and 100 mW [25]. Thus, the strategy set of vehicle i is $\mathcal{P}_i = [1, 100]$. A higher power is desired because the beacon is disseminated over larger distance thus, it creates higher awareness under normal conditions. But high power has a negative effect on awareness in congested situations. Therefore, the desirable payoff function would yield lower payoff with the same power in situations with high levels of congestion. To fulfil this goal, the payoff function is modelled as the difference between a utility function ($U_i(p_i)$) and a price function ($J_i(p_i, \boldsymbol{p}_{-i})$). Accordingly, the payoff for player i is defined as follows:

$$F_i\left(p_i, \boldsymbol{p_{-i}}\right) = U_i\left(p_i\right) - J_i\left(p_i, \boldsymbol{p_{-i}}\right) = u_i \ln\left(p_i + 1\right) - \frac{c_i\, p_i}{1 - CBR_i\left(\boldsymbol{p}\right)} \tag{5.1}$$

where u_i and c_i are positive parameters, $ln(.)$ is natural logarithm, and $CBR_i(\boldsymbol{p})$ is the channel busy ratio that player i experiences, and it is a function of all the players' power level.

The first term in the payoff function ($u_i \ln(p_i + 1)$) is called utility, it is an increasing function of BSM power level. A logarithmic function has been selected as utility because it is increasing, concave and differentiable. The second term $\left(c_i\, p_i / \left(1 - CBR_i(\boldsymbol{p})\right)\right)$, is the price function. Which indicates that a user should pay more price to use the network resource (bandwidth), in higher congestions. This term is a function of CBR because CBR is a good indicator of successful information dissemination in VANETs [44]; high CBR results in poor inter-vehicle awareness. The

price function becomes larger in scenarios with higher levels of congestion, yielding a lower payoff.

Again for $CBR_i(\boldsymbol{p})$, the mathematical model developed in [121], given below, is used. However, unlike what was defined for NORAC, here the beaconing rate r is constant (10 Hz) and p_j transmission power of jth player is variable.

$$CBR_i\left(\boldsymbol{p}\right)=\sum_{j=1}^{n}h_{ij}r \tag{5.2}$$

where

$$h_{ij}=T_{frame}\times\frac{\boldsymbol{\Gamma}\left(m,\frac{m\,C_{Tt}}{\Omega_{ij}}\right)}{\boldsymbol{\Gamma}\left(m\right)} \tag{5.3}$$

$$\Omega_{ij}=\frac{p_j\,\lambda^2}{\left(4\pi\right)^2 d_{ij}^{\gamma}} \tag{5.4}$$

the other variables are similar to what described for NORAC in the previous chapter. $\nabla_i F_i(\boldsymbol{p})=\partial F_i(\boldsymbol{p})/\partial p_i$ is the marginal payoff of player i. The vector of marginal payoffs of all the players is given as $\nabla F(\boldsymbol{p})=\left(\nabla_1 F_1(\boldsymbol{p}),\nabla_2 F_2(\boldsymbol{p}),\dots,\nabla_{\boldsymbol{n}} F_{\boldsymbol{n}}(\boldsymbol{p})\right)^T$ and its Jacobian as $G(\boldsymbol{p})$.

5.2 Nash Equilibrium of the Games

5.2.1 Existence and Uniqueness

A game $\mathcal{G}$ with twice differentiable pay-off functions F_i, is submodular if and only if [122]:

$$\forall\, i,j\in N \quad i\neq j \quad \frac{\partial^2 F_i}{\partial p_i\,\partial p_j}\leq 0 \tag{5.5}$$

In addition, according to Theorem 3.1 in [122] the set of equilibrium points of such game is not empty and a least and a greatest equilibrium point exist. Considering the NOPC game, the pay-off functions are twice differentiable and

$$\frac{\partial F_i}{\partial p_i} = \frac{u_i}{p_i + 1} - \frac{c_i}{1 - CBR_i(\boldsymbol{p})} \tag{5.6}$$

$$\frac{\partial^2 F_i}{\partial p_i \partial p_j} = -\frac{c_i r T_{frame}}{\Gamma(m)\left(1 - CBR_i(\boldsymbol{p})\right)^2} \times \frac{\partial \Gamma\left(m, \frac{m C_{Tt}}{\Omega_{ij}}\right)}{\partial p_i}$$

$$= -\frac{c_i r T_{frame}}{\Gamma(m)\left(1 - CBR_i(\boldsymbol{p})\right)^2} \times \frac{(k_{ij})}{p_j^{m+1}} e^{-\frac{k_{ij}}{p_j}} < 0 \qquad i \neq j \tag{5.7}$$

where

$$k_{ij} = \frac{m C_{Tt} (4\pi)^2 d_{ij}^{\gamma}}{\lambda^2} \tag{5.8}$$

Therefore, NOPC is a submodular game and has a greatest and a least NE. The proof of uniqueness of the NE is similar to the proof in Chapter 4.

5.2.2 Stability

In [124] it was proved that for a strictly concave game, the unique equilibrium of the game is globally stable and gradient method converges to the NE. In NOPC, $-G(\boldsymbol{p})$ is an $n \times n$ matrix with elements:

$$g_{ii} = -\frac{\partial^2 F_i}{\partial p_i^2} = \frac{u_i}{p_i^2} \tag{5.16}$$

and

$$g_{ij} = -\frac{\partial^2 F_i}{\partial p_i \partial p_j} = \frac{c_i r T_{frame}}{\Gamma(m)\left(1 - CBR_i(\boldsymbol{p})\right)^2} \times \frac{\partial \Gamma\left(m, \frac{m C_{Tt}}{\Omega_{ij}}\right)}{\partial p_j}$$

$$= \frac{c_i r T_{frame}}{\Gamma(m)\left(1 - CBR_i(\boldsymbol{p})\right)^2} \times \frac{(k_{ij})^m}{p_j^{m+1}} e^{-\frac{k_{ij}}{p_j}} \tag{5.17}$$

Localizing the eigenvalues of $-G(\boldsymbol{p})$ using analytical methods if not impossible, is very difficult. In such conditions, Numerical-based or simulation-based techniques are

used to conclude stability of the system [17]. We use simulation-based technique for a large number of vehicles to show the stability of the system under gradient method in the next sections.

5.3 Congestion Control Process

To find the NE of NOPC in a distributed manner, using the gradient method, every vehicle updates its beacon power as follows.

$$\frac{dp_i}{dt} = \frac{\partial F_i}{\partial p_i} = \frac{u_i}{p_i + 1} - \frac{c_i}{1 - CBR_i(\boldsymbol{p})} \tag{5.18}$$

Algorithm 5.1 shows the NOPC mechanism. In the algorithm p_{max} and p_{min} are 100 mW and 1 mW, respectively [25]. As the Algorithm shows, every vehicle updates its BSM power, according to the locally measured CBR in each iteration of the algorithm, and vehicles do not communicate excess information in their beacons.

Algorithm 4.1 Beacon's power updates in NOPC based on the gradient method

1. Every vehicle measures CBR
2. Every vehicle updates the beacon power according to:

$$p_i = \left[p_i + \frac{u_i}{p_i + 1} - \frac{c_i}{1 - CBR_i(\boldsymbol{p})} \right]_{p_{\min}}^{p_{\max}}$$

5.4 Selection of the Parameters

As mentioned before, the purpose of the NOPC algorithm is to control CBR around 0.65 (according to [44] between 0.4 and 0.8); thus, simulations are run, in order to find the appropriate values for u_i and c_i. For this purpose, OMNeT++ as network simulator and SUMO as mobility generator have been used. The simulation parameters are summarized in Table 5.1.

Table 5.1 Simulation parameters

Parameter	**Value**
Thermal Noise	-100 dBm
Carrier Sense Threshold	-90 dBm
MAC Protocol	IEEE 802.11p
Carrier Frequency	5.89 GHz
Bit Rate	6 Mbps
Beacon Size	500 Byte
Beacon Rate	10 Hz
Sampling Time	500 msec
Propagation Model	Nakagami m = 2.0
N_{max} (SBCC-N)	98.3
C_{max} (SBCC-C)	0.7

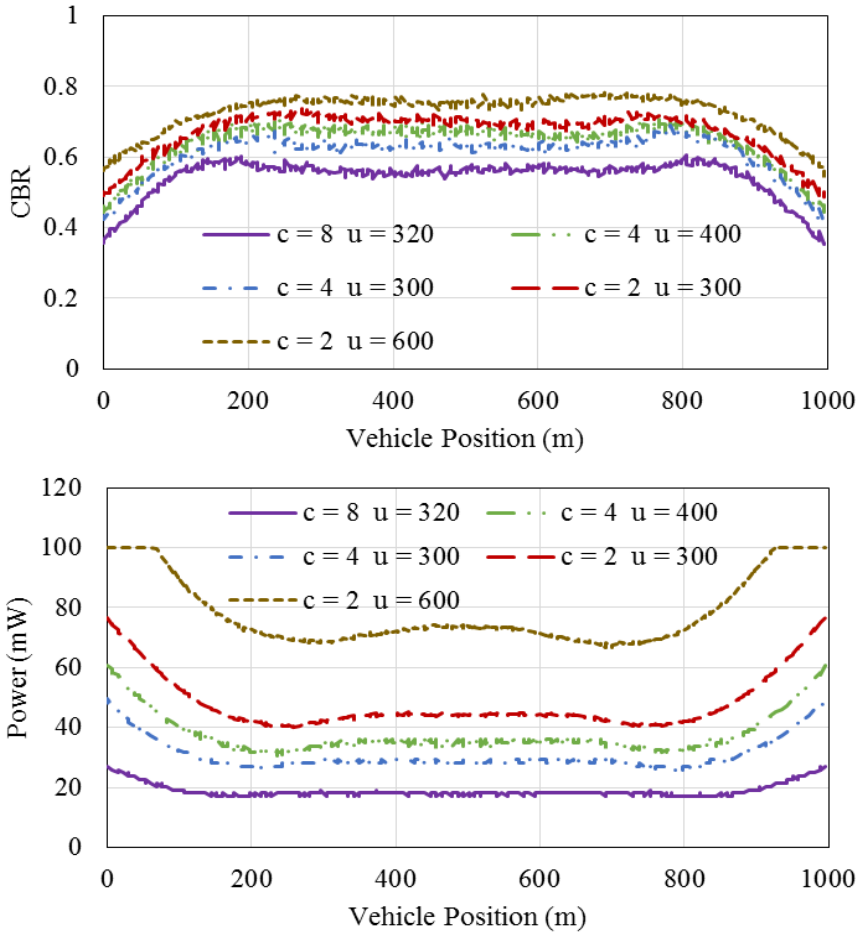

Figure 5.1 Beacon power and CBR for NOPC with different values of u and c parameters on a 1000 m track with three lanes and homogenous distribution of 396 vehicles

Simulations were run for a scenario of a track with three lanes and a total number of vehicles N = 396, with a homogeneous distribution. Figure 5.1 shows the beacon power and CBR for the different values of the parameters of the game. By increasing c_i, the CBR is controlled at a lower level and vehicles tend to use lower power. The increase of u_i has the reverse effect. Figure 5.1 also shows that for all the indicated values, the CBR is controlled around the desirable level 0.65, thus for a wide range of parameters the algorithm works efficiently. In these experiments all the vehicles have the same parameters however, later it is shown that vehicles can change their u_i parameter individually, in order to meet their application requirements, while they do not need to communicate their parameter with other vehicles and the algorithm works properly and is stable. The Figure also shows that for $c_i = 2$ and $u_i = 300$, the CBR is controlled around the desired level 0.65. These values are used to compare NOPC algorithms with SBCC-N and SBCC-C [45] in a static scenario. In Section 5.5.2 it is shown that vehicles can change their u_i parameter individually in order to meet their application requirements; and they do not require to exchange their parameters with other vehicles for the algorithm to work properly and remain stable.

5.5 Performance Evaluation

5.5.1 Static Scenarios

The same scenario in the previous section; the track with length 1000 m and n = 396 vehicles is used to compare NOPC algorithm with SBCC-N and SBCC-C. Figure 5.2 shows beacon power and CBR for the vehicles in the scenario; as it is evident, the NOPC algorithm is fairer in power allocation. The Jain Index [59] for the allocated power for SBCC-N and SBCC-C and NOPC is 0.57, 0.90 and, 0.96 respectively, which indicates NOPC is fairer than the others. The average IDR for SBCC-N and SBCC-C and NOPC is 517 beacon/s, 537 beacon/s and 591 beacon/s respectively which indicate with NOPC vehicles receive more information from their neighbouring vehicles. Figure 5.2 also shows that the CBR over the track has more fluctuations with SBCCs than NOPC does. In SBCC algorithms, vehicles require to compute average transmission power used by neighbouring nodes. They also estimate channel parameters such as path loss component and shape parameter in Nakagami fading

model. In SBCC-N the number of neighbouring vehicles should be estimated too. Because different vehicles might estimate different values for the mentioned parameters, unfairness happens in beacon power. In addition, the functionality of SBCC algorithms relies on the exchange of excess information in beacons; every vehicle should include the beacon transmission power in its beacons. Thus, NOPC is better, in bandwidth usage too.

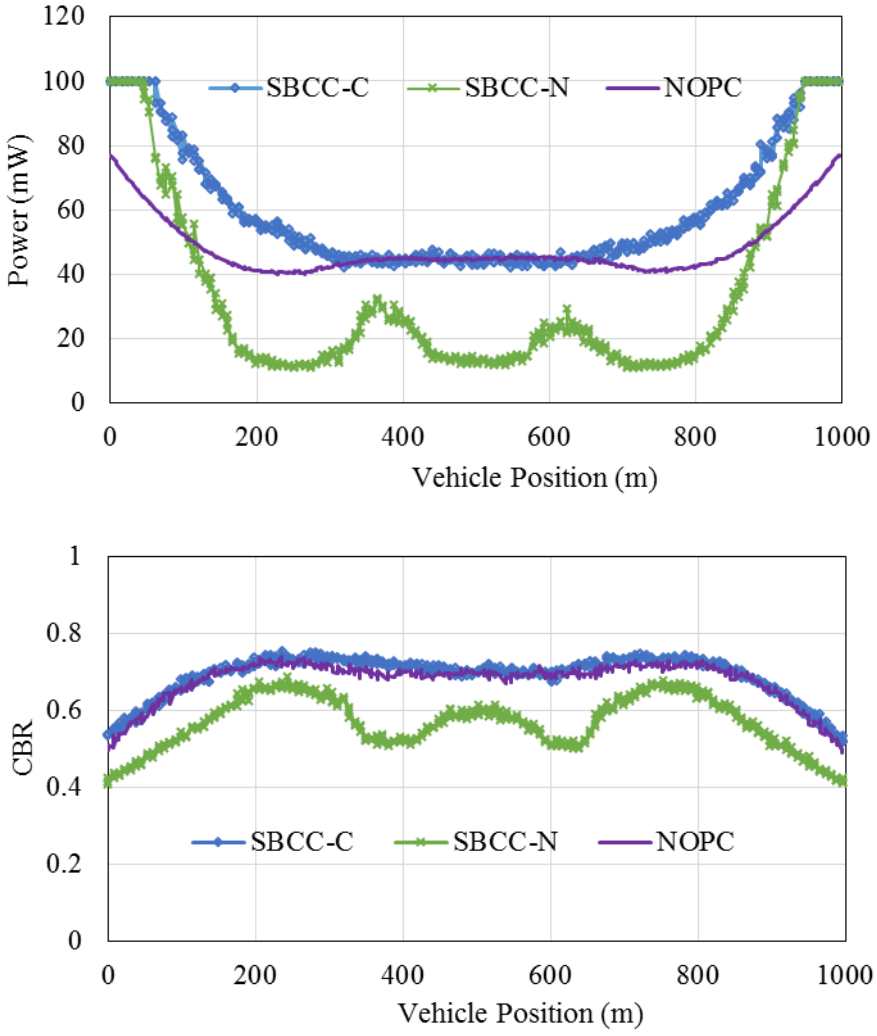

Figure 5.2 Beacon power and CBR for the algorithms

To verify the stability of the algorithm and the uniqueness of the NE in a scenario with a higher number of vehicles, the next scenario is selected so that there are 600 vehicles randomly distributed, over a track with a length of 1400 m and with six lanes. The experiment has been repeated with different initial values of power for vehicles:

when all the vehicles have an initial power 1 mW, 100 mW and when every vehicle has a random initial power between 1 and 100 mW. For all the conditions, NOPC converges to the same level of power and CBR, which signifies the uniqueness and stability of the NE.

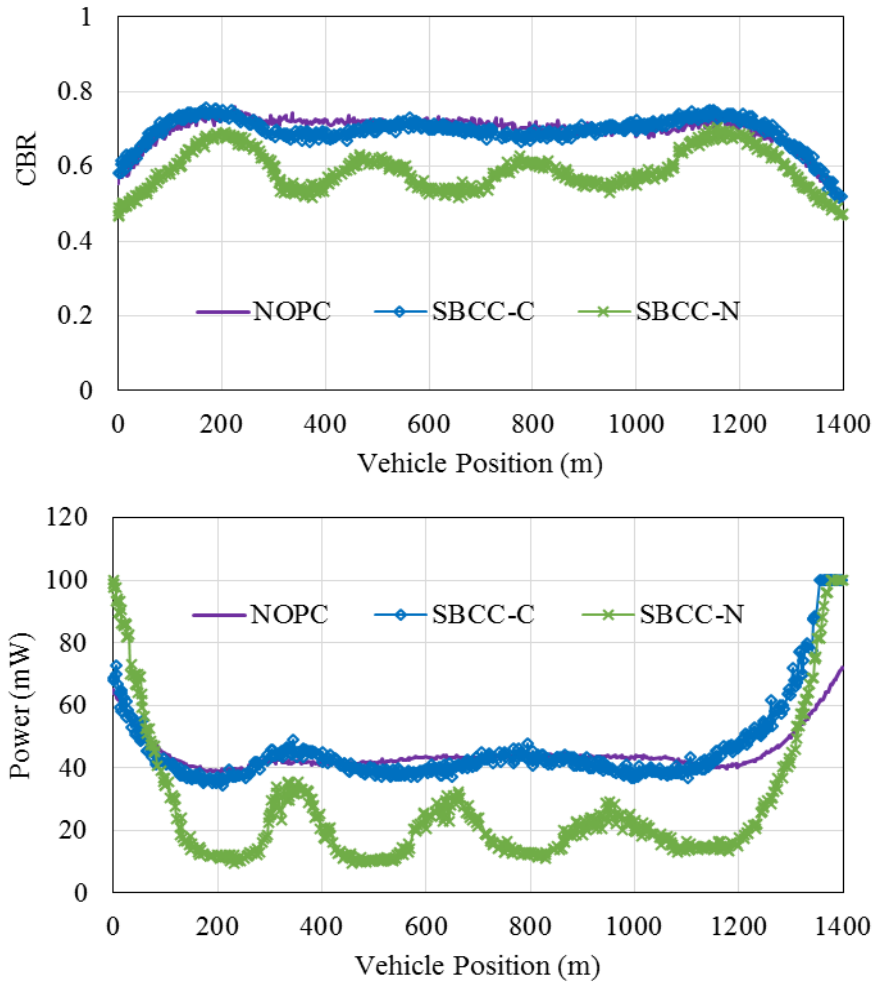

Figure 5.3 Beacon power and CBR for a 1400 m track with six lanes and random distribution of 600 vehicles

Figure 5.3 shows the power and CBR for this scenario, for the three algorithms. It is clear that NOPC is much fairer in terms of power allocation than SBCCs and that CBR is smoother along the track. IDR for NOPC, SBCC-C and SBCC-N is 545 beacon/s, 507 beacon/s and 483 beacon/s respectively. Figure 5.4 shows the changes in beacon power against iterations of the algorithms, for a vehicle at a position almost middle of the track (almost x=700) for SBCC-N and SBCC-C when the initial power is 100 mW

and also for NOPC with initial powers 1 mW and 100 mW. It is observed that NOPC converges in about ten iterations of the algorithm.

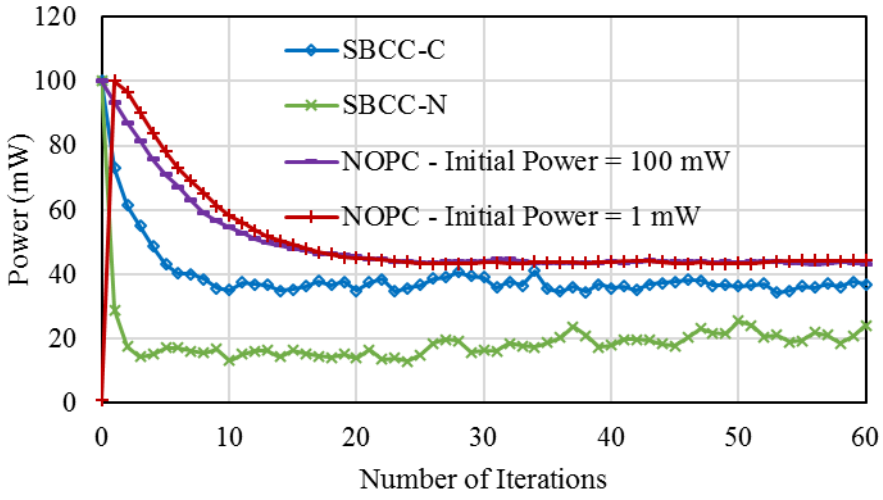

Figure 5.4 Beacon power changes versus the iteration of the algorithms for a 1400 m track with six lanes and random distribution of 600 vehicles

5.5.2 Dynamic Scenarios

In the next experiments, it is indicated how NOPC can assign different power levels to vehicles with different beaconing power requirements. In the proposed power control algorithm, every vehicle can adjust its u_i (and c_i) parameter to meet its application requirement. For example, when there is a traffic jam in one side of a highway and there is free flow on the other side, it is desired that vehicles with higher speed will have higher power. Such a scenario has been simulated in the next experiments. In the scenario, there is a traffic jam on one side of a highway, so vehicles are stationary.

The stationary vehicles are distributed homogenously over two lanes (316 stationary vehicles). On the other side of the highway, vehicles move with speeds of 10, 15 or 20 m/s. All the vehicles have the same c_i ($c_i = c = 2$) and every vehicle adjusts its u parameter proportional to its speed, as follows.

$$u_i = 50 \times \lceil v_i \rceil_4 \tag{5.22}$$

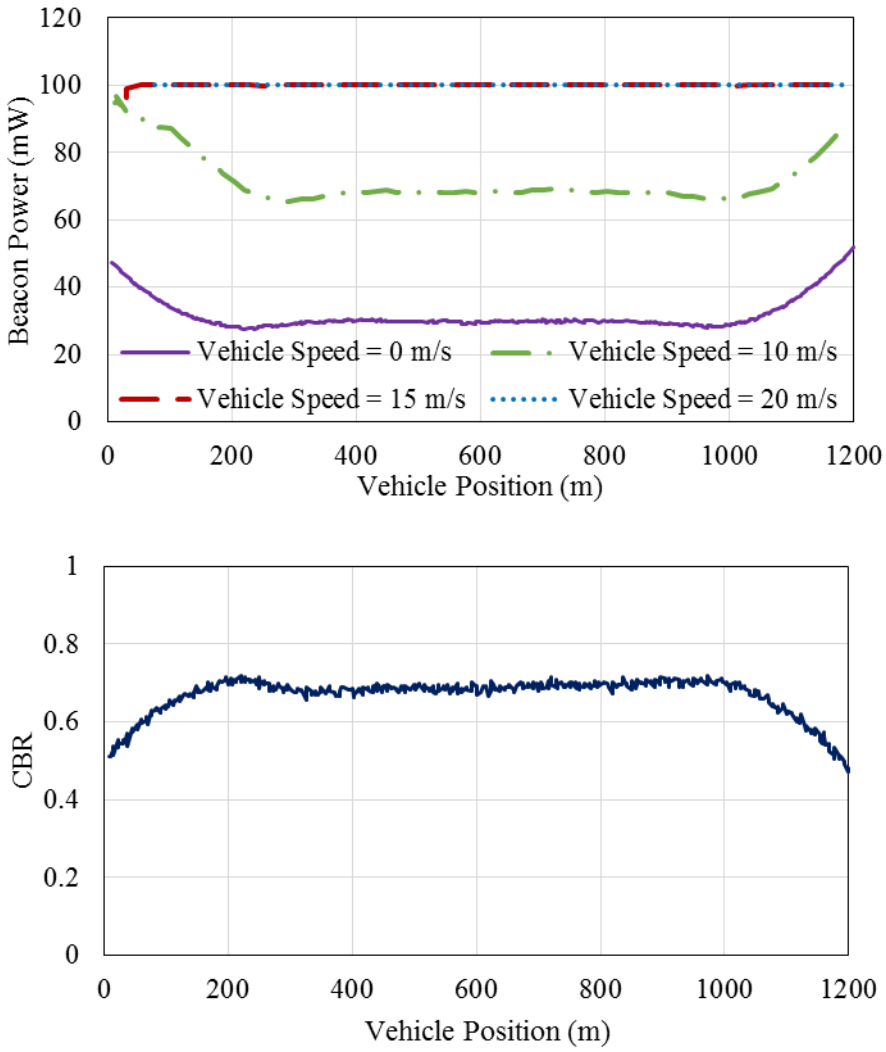

Figure 5.5 Beacon power and CBR for a 1200 m track, with vehicles which have different speeds of 0, 10, 15 and 20 m/s with $u_i = 50 \times [v_i]_4$

where v_i is the speed of the vehicle. Thus, for example, the utility parameter for stationary vehicles would be 50×4 = 200 and, for vehicles with 10 m/s speed it would be 50×10 = 500. Figure 5.5 shows that for vehicles far enough from the edges of the scenario, the vehicles with higher speeds use higher power for beaconing and the CBR is controlled. This could be explained in this way that, at equilibrium point:

$$\frac{\partial F_i}{\partial p_i} = \frac{u_i}{p_i + 1} - \frac{c}{1 - CBR_i(\boldsymbol{p})} = 0 \tag{5.23}$$

thus,

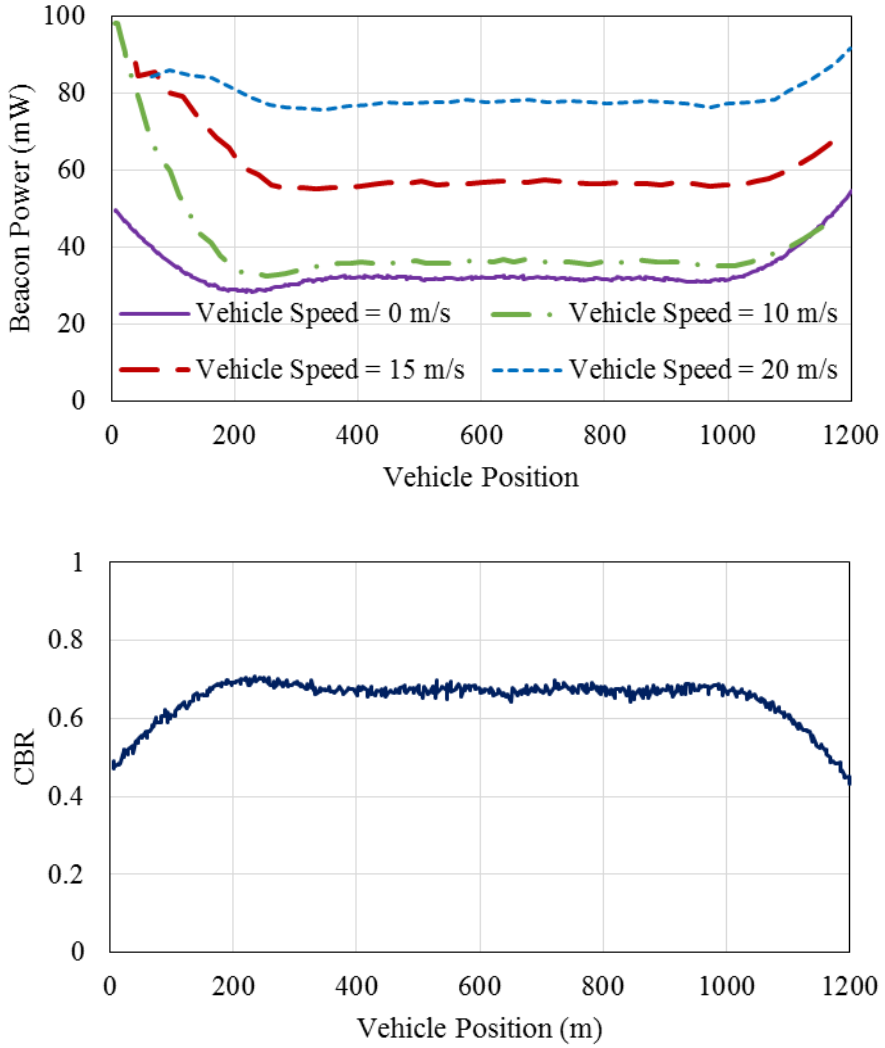

Figure 5.6 Beacon power and CBR for a 1200 m track, with vehicles which have different speeds of 0, 10, 15 and 20 m/s with $u_i = 50 \times [v_i/2]_4$

$$p_i + 1 = \frac{u_i \left(1 - CBR_i\left(\boldsymbol{p}\right)\right)}{c} \tag{5.24}$$

The vehicles i and j at the same x position sense the same CBR, so:

$$\frac{p_i}{p_j} \approx \frac{u_i}{u_j} = \frac{[v_i]_4}{[v_j]_4} \tag{5.25}$$

Thus, the ratio of allocated powers is equal to the ratio of speed of vehicles. In other words, the NOPC algorithm has per-vehicle parameter u_i that every vehicle can change it without communicating it with other vehicles to meet its requirement. Besides, it is seen that there is fairness in power amongst the vehicles that have the

same requirement (in this example the same speed). The parameter u_i could be a function of acceleration, deceleration…. so that the vehicles which are in a status that needs a longer beaconing power, can obtain this by adjusting their u_i parameters, while the CBR is controlled at the desired level.

Figure 5.6 shows the results for the same scenario when $u_i = 50 \times [v_i/2]_4$. In this condition, all the vehicles reduce their beacon power, because they are using lower utility and they start to reduce their power in lower channel loads. In addition, weighted fairness is seen among the power level of them.

5.6 Summary

A distributed algorithm for congestion control, by adapting BSM power for VANET, was proposed. The algorithm is based on non-cooperative game theory for which the uniqueness of the equilibrium point was proved. Using simulation, it was indicated that the algorithm is stable for a large number of vehicles. The algorithm was compared with other power control algorithms and it was indicated that it performs much better in terms of fairness and band width usage. In addition, NOPC can meet application requirements; it has per-vehicle parameter so that every vehicle can obtain appropriate power for its requirements by adapting them, while congestion is controlled.

Chapter 6

Joint Beacon Power and Rate Control

In very dense traffic situations, vehicles might be required to reduce both their beacon power and rate. ETSI Decentralized Congestion Control (DCC) [70] proposes a joint beacon rate and power control mechanism. However, several researches have revealed that ETSI DCC suffers unfairness and oscillation [54], [71], [72] . In this chapter, a protocol for joint beacon rate and power control that is based on game theory is proposed. The protocol models the interaction between vehicles as a non-cooperative game in which the strategy space of every player is two-dimensional (power and rate).

6.1 Non-Cooperative Beacon Power and Rate Control Game

Let $\mathcal{G} = \{\mathcal{N}, \{\mathcal{X}_i\}_{i\in\mathcal{N}}, \{\mathcal{Q}_i\}_{i\in\mathcal{N}}\}$ denote the Non-cooperative Power And Rate Control (PARC) game, where $\mathcal{N} = \{1, \dots, n\}$ is the set of players (vehicles), and $\mathcal{X}_i \subset \mathbb{R}^2$ is the set of two-tuples $x_i = (p_i, r_i)$ of possible beaconing powers and rates for player i. $\mathcal{X}_i$ is called the strategy set of player i , and the tuple $x_i \in \mathcal{X}_i$ is called the strategy of player i. Each player selects its strategy independently. The vector $\boldsymbol{x} = (x_1, x_2, \dots, x_n) \in \boldsymbol{\mathcal{X}}$ shows the selected power and rate of all the players, where $\boldsymbol{\mathcal{X}} = \prod_{i=1}^{n} \mathcal{X}_i$. The expression $\mathcal{Q}_i$ is the payoff function of player i and is indicated as $\mathcal{Q}_i(\boldsymbol{x}) = \mathcal{Q}_i(x_i, \boldsymbol{x}_{-i})$, where $\boldsymbol{x}_{-i}$ denotes the vector consisting of the beacon powers and rates of all the players except the ith player.

As with the previous chapters, the range of beaconing rate and power are $[1, 10]$ Hz and $[1, 100]$ mW, respectively. Higher beaconing power and rate is desired because it creates higher awareness under typical conditions. But high power and rate have negative effects on awareness in congested situations. Therefore, the desirable payoff

function would yield lower payoff in congested situations. To fulfil this goal, the payoff functions are defined as the difference between a utility function ($U_i(x_i)$) and a price function ($J_i(x_i, \boldsymbol{x}_{-i})$). Accordingly, the payoff for player i is defined as follows:

$$Q_i\left(x_i, \boldsymbol{x}_{-i}\right) = U_i\left(x_i\right) - J_i\left(x_i, \boldsymbol{x}_{-i}\right)$$

$$= u_i \ln\left(r_i + 1\right) + w_i \ln\left(p_i + 1\right) - \frac{c_i p_i}{1 - CBR_i\left(\boldsymbol{x}\right)} \tag{6.1}$$

where u_i, w_i, and c_i are the positive parameters of rate utility, power utility, and price, respectively. Similar to previous chapters, the same mathematical function used for $CBR_i(\boldsymbol{x})$ is employed in this chapter. However, every vehicle can change its rate and power. Thus, we have

$$CBR_i\left(\boldsymbol{r}, \boldsymbol{p}\right) = \sum_{j=1}^{N} T_{frame} \times \frac{\Gamma\left(m, \frac{m C_{Tt}}{\Omega_{ij}}\right)}{\Gamma\left(m\right)} \times r_j = \sum_{j=1}^{N} h_{ij} \times r_j \tag{6.2}$$

where

$$\Omega_{ij} = \frac{p_j \lambda^2}{\left(4\pi\right)^2 d_{ij}^{\gamma}} \tag{6.3}$$

The components of (6.2) and (6.3) are as defined in Chapter 4.

6.2 Nash Equilibrium

The payoff functions (6.1) are twice differentiable. Thus, according to [122] the game is a submodular game if (6.4) and (6.5) hold.

$$\forall i \in N, \quad \frac{\partial^2 Q_i}{\partial r_i \partial p_i} \leq 0 \tag{6.4}$$

$$\forall i \neq j, \quad \frac{\partial^2 Q_i}{\partial y_i \partial z_j} \leq 0 \quad y \text{ and } z \text{ could be } r \text{ or } p \tag{6.5}$$

For PARC we have,

$$\frac{\partial^2 Q_i}{\partial r_i \partial p_i} = -\frac{c_i h_{ii}}{\left(1-CBR_i\left(\boldsymbol{x}\right)\right)^2} < 0 \tag{6.6}$$

$$\frac{\partial^2 Q_i}{\partial r_i \partial r_j} = -\frac{2\, c_i p_i h_{ii} h_{ij}}{\left(1-CBR_i\left(\boldsymbol{x}\right)\right)^3} < 0 \tag{6.7}$$

$$\frac{\partial^2 Q_i}{\partial p_i \partial p_j} = -\frac{c_i \dfrac{\partial CBR_i\left(\boldsymbol{x}\right)}{\partial p_j}}{\left(1-CBR_i\left(\boldsymbol{x}\right)\right)^2} = -\frac{c_i r_j T_{frame}}{\Gamma\left(m\right)\left(1-CBR_i\left(\boldsymbol{x}\right)\right)^2} \times \frac{\partial \Gamma\left(m, \dfrac{m C_{Tt}}{\Omega_{ij}}\right)}{\partial p_j}$$

$$= -\frac{c_i r_j T_{frame}}{\Gamma\left(m\right)\left(1-CBR_i\left(\boldsymbol{x}\right)\right)^2} \times \frac{\left(k_{ij}\right)^m}{p_j^{m+1}} e^{-\frac{k_{ij}}{p_j}} < 0 \tag{6.8}$$

$$\frac{\partial^2 Q_i}{\partial r_i \partial p_j} = -\frac{2c_i p_i h_{ii} \dfrac{\partial CBR_i\left(\boldsymbol{x}\right)}{\partial p_j}}{\left(1-CBR_i\left(\boldsymbol{x}\right)\right)^3} = -\frac{2c_i p_i h_{ii} r_j T_{frame}}{\Gamma\left(m\right)\left(1-CBR_i\left(\boldsymbol{x}\right)\right)^3} \times \frac{\left(k_{ij}\right)^m}{p_j^{m+1}} e^{-\frac{k_{ij}}{p_j}} < 0 \tag{6.9}$$

$$\frac{\partial^2 Q_i}{\partial p_i \partial r_j} = -\frac{c_i h_{ij}}{\left(1-CBR_i\left(\boldsymbol{x}\right)\right)^2} < 0 \tag{6.10}$$

where

$$k_{ij} = \frac{m C_{Tt} \left(4\pi\right)^2 d_{ij}^{\gamma}}{\lambda^2} \tag{6.11}$$

Thus, PARC is a submodular game; the set of its equilibrium points is nonempty, and a greatest and a least equilibrium point exist (Theorem 3.1 in [122]). The proof of uniqueness of the equilibrium point is similar to the proof for NORAC in Chapter 4.

The gradient projection method is used to solve the game. Algorithm 6.1 shows the PARC mechanism for rate and power adaptation based on this method. As indicated in the previous chapters, greater utility parameter leads to using higher power or rate. Thus, we select the parameters u_i, w_i, c_i so that typically the vehicles contribute in congestion control simply by reducing their rates, and when the congestion (the number of vehicles) is higher, the vehicles start to reduce their power too. To this

objective, a greater value than what was chosen for power utility in Chapter 5 is selected for w_i in the following experiments.

Algorithm 6.1 Beacon power and rate updates in PARC

1. Every vehicle measures CBR
2. Every vehicle updates the beacon power according to

$$p_i = \left[p_i + \frac{w_i}{p_i + 1} - \frac{c_i}{1 - CBR_i(\boldsymbol{x})} \right]_{p_{\min}}^{p_{\max}}$$

3. Every vehicle updates the beacon rate according to

$$r_i = \left[r_i + \frac{u_i}{r_i + 1} - \frac{c_i \, h_{ii} \, p_i}{\left(1 - CBR_i(\boldsymbol{x})\right)^2} \right]_{r_{\min}}^{r_{\max}}$$

6.3 Simulation Results

The simulation parameters are as indicated in Table 5.1 (Chapter 5) except for the beacon rate, which is not constant and can be adapted in the interval [1,10] Hz.

6.3.1 Static Scenarios

For the experiments of this section, a 1000 m track with stationary vehicles distributed homogenously is modelled. Simulations are run with parameters $u_i = 4, c_i = 3.0,$ and two different values for $w_i, 650$ and 450. The value of c_i is selected so that the term $c_i h_{ii} p_i$ that appears in the beacon rate update of the algorithm (line 3 of algorithm 6.1) instead of the pc_i parameter in Chapter 4, at $p_i = 100\ mW$ becomes almost 0.2. This is because 0.2 is the value that was used for pc_i in Chapter 4 and produced reasonable results. The track has three lanes with 396 vehicles, and all vehicles use the same parameters for the static scenarios. The vehicles at the beginning of the simulation have random rate and power, as Figures 6.1 and 6.2 show, to validate the convergence of the algorithm from any initial point.

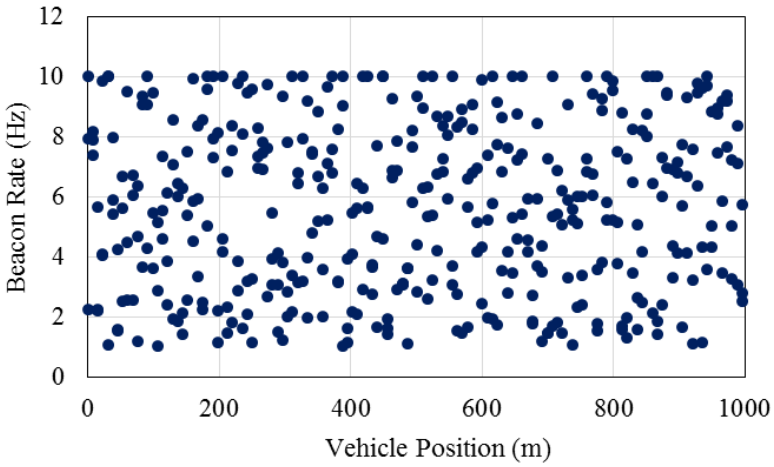

Figure 6.1 Initial beacon rate of the vehicles over the 1000 m track with three lanes

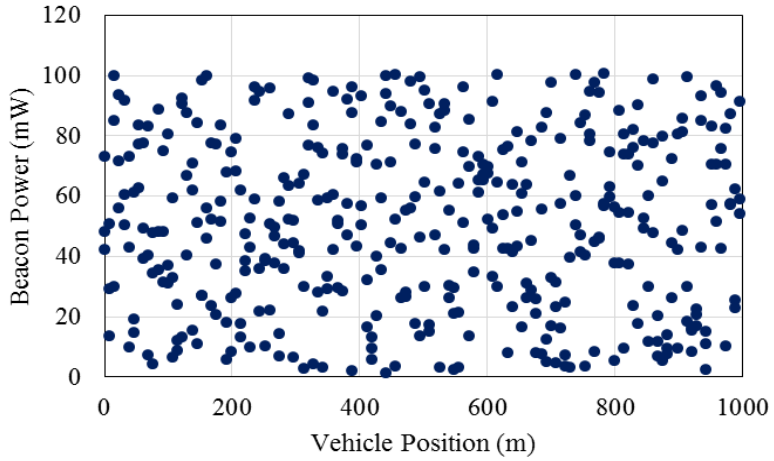

Figure 6.2 Initial beacon power of the vehicles over the 1000 m track with three lanes

Figures 6.3, 6.4, and 6.5 indicate the simulation results for the scenario. As observed, when $w_i = 450$, vehicles reduce both their power and rate to control the congestion, whereas with $w_i = 650$, vehicles contribute to congestion control simply by adapting their rates. Therefore, with $w_i = 650$, the vehicles use less rate in comparison to the case with $w_i = 450$ to control the CBR almost at the same level (approximately 0.5).

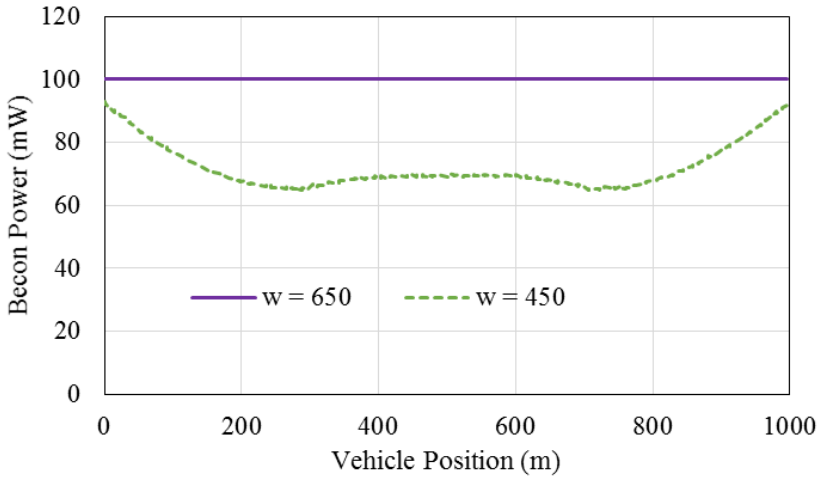

Figure 6.3 Beacon power over the 1000 m track with 396 vehicles

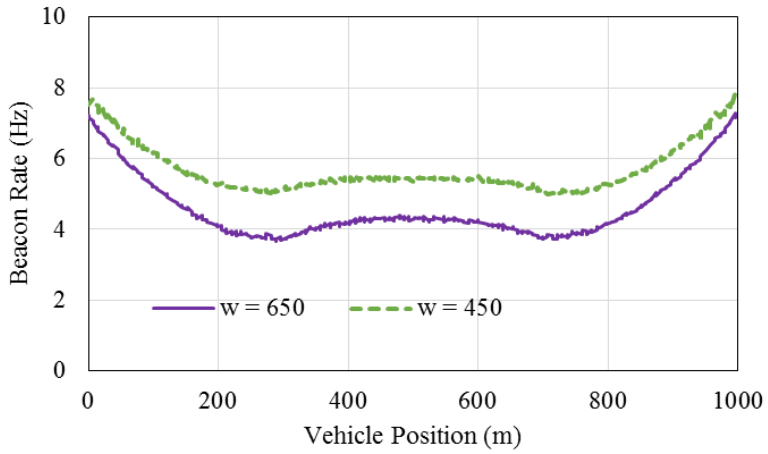

Figure 6.4 Beacon rate over the 1000 m track with 396 vehicles

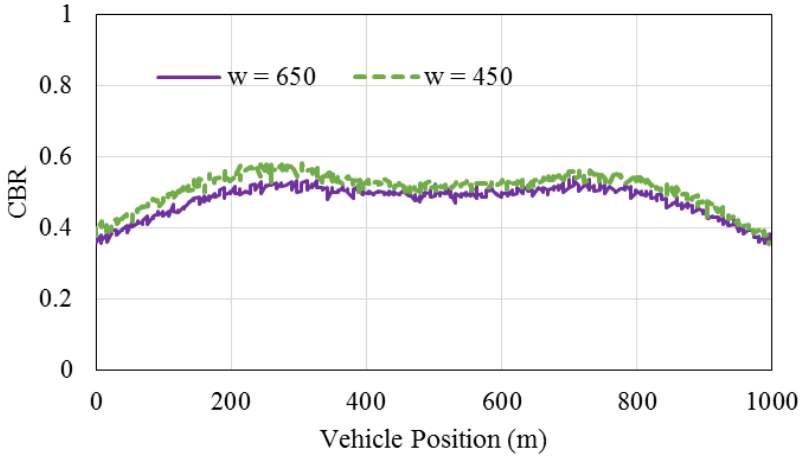

Figure 6.5 CBR over the 1000 m track with 396 vehicles

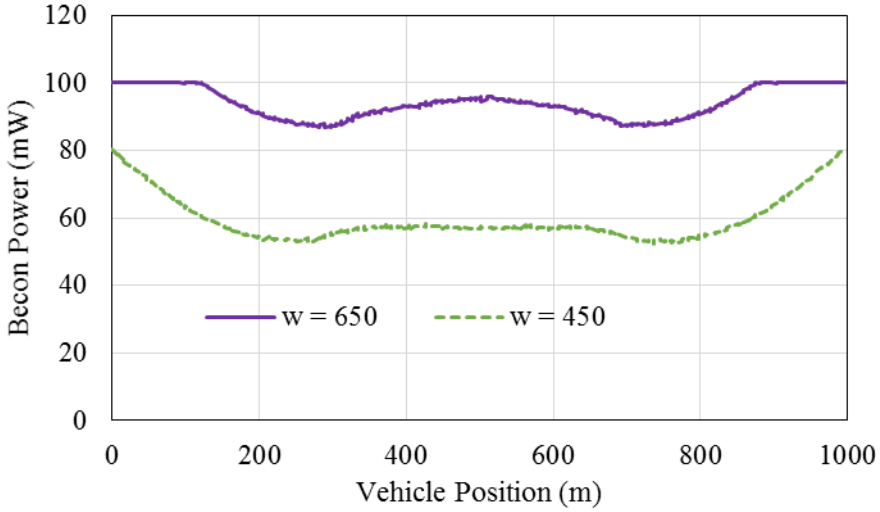

Figure 6.6 Beacon Power over the 1000 m track with 660 vehicles

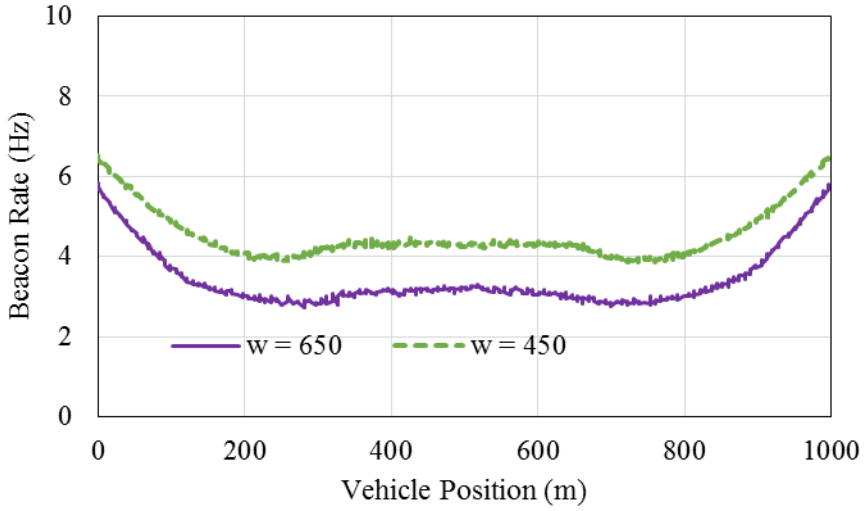

Figure 6.7 Beacon Rate over the 1000 m track with 660 vehicles

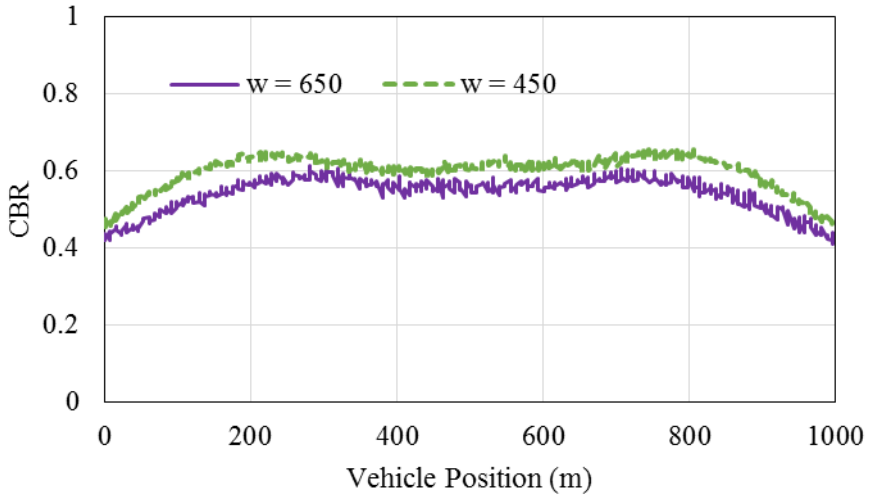

Figure 6.8 CBR over the 1000 m track with 660 vehicles

Then, the number of lanes was increased to five lanes (660 vehicles). Figures 6.6, 6.7 and 6.8 indicates the results for this condition. Figure 6.7 shows when the number of nodes increased, even in with $w_i = 650$, the vehicles reduce their power. However, in comparison to the state with $w_i = 450$, they still use higher power and lower rate.

In all the presented configurations, it is observed that CBR is controlled at an acceptable level, and good fairness in rate and power is seen for the nodes that are far enough from the edges. In addition, considering rate and power over the entire track (including positions close to the edges), the protocol is fairer than other protocols used for comparison in previous chapters. In other words, the edge effect that is observed in most beaconing congestion control mechanisms is more moderate in PARC.

6.3.2 Dynamic Scenarios

For this section, the same scenarios that were used in the previous chapters are simulated: a 1200 m track with two or six lanes of stationary vehicles and also moving vehicles with speeds of 10 m/s, 15 m/s, and 20 m/s. The parameters of the PARC protocol are $w_i = 650$, $c_i = 3.0$, and $u_i = [v_i]_4$, where v_i is the speed of the ith vehicle. Figures 6.9-6.11 show beacon rate, beacon power, and CBR for the case with two lanes of stationary vehicles (316 stationary vehicles). Only the vehicles with speeds of 0 m/s and 10 m/s contribute to congestion control by adapting their rates. In addition, because w_i has a high value, the vehicles do not decrease their power. All the vehicles use the same w_i; thus, there is fairness in power. The values for u_i are proportional to the speed; therefore, the achieved rates are proportional to the speeds of the vehicles.

Figures 6.12-6.14 show the results for the case with six lanes of stationary vehicles (948 stationary vehicles). In this condition, all the vehicles have decreased their power. Furthermore, vehicles with speed of 15 m/s also contribute in congestion control by reducing their rate. As observed, again there is fairness in beacon power and weighted fairness in rate.

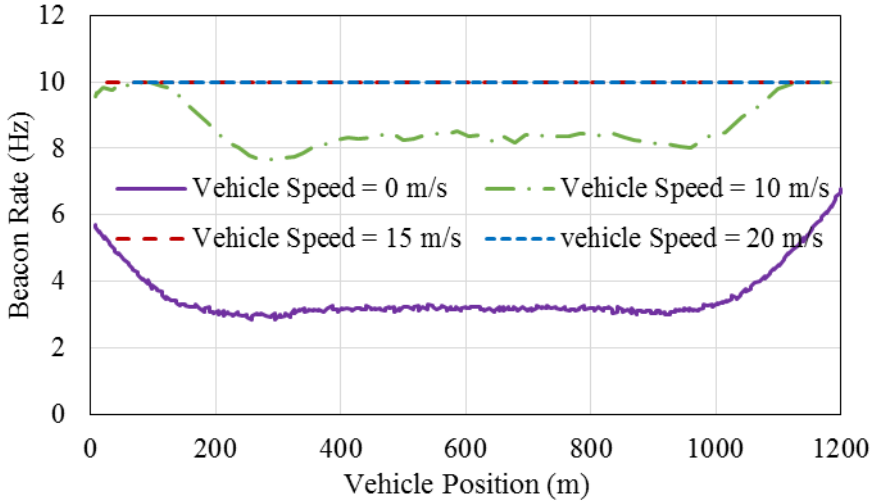

Figure 6.9 Beacon rate for a track of length 1200 m with two lanes of stationary vehicles - vehicles in the other lanes have speeds of 10, 15, and 20 m/s

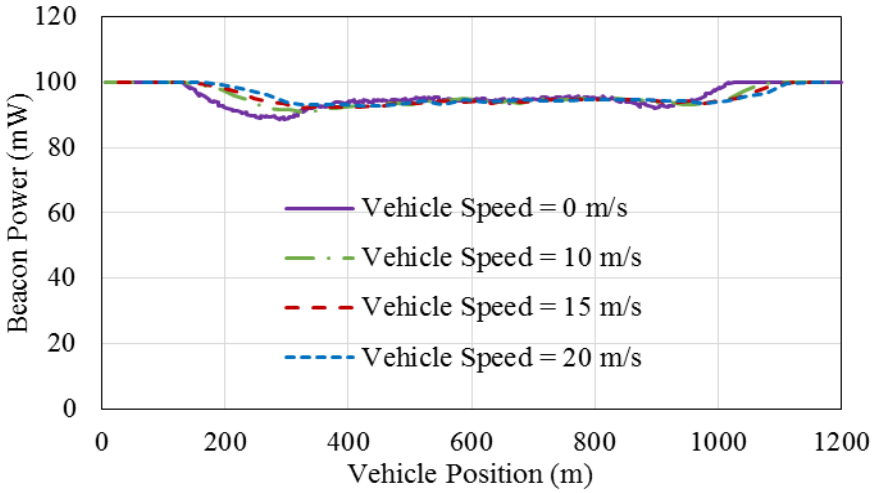

Figure 6.10 Beacon power for Figure 6.9

Figure 6.11 CBR for Figure 6.9

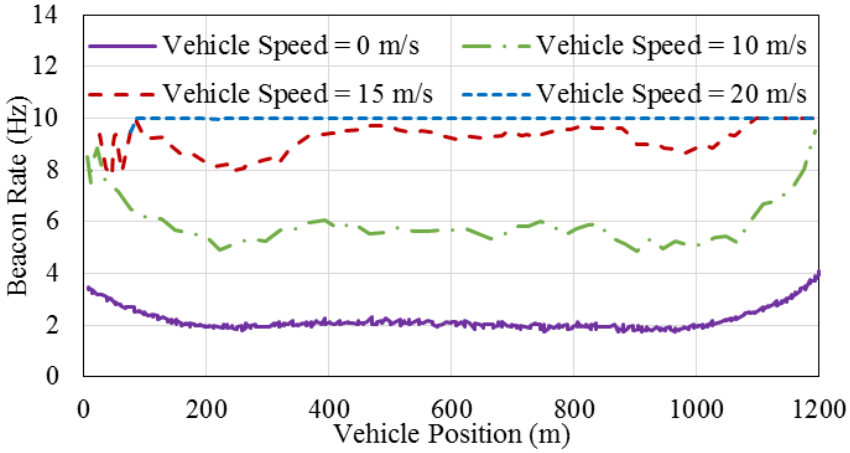

Figure 6.12 Beacon rate for a track of length 1200 m with six lanes of stationary vehicles - vehicles in the other lanes have speeds of 10, 15, and 20 m/s

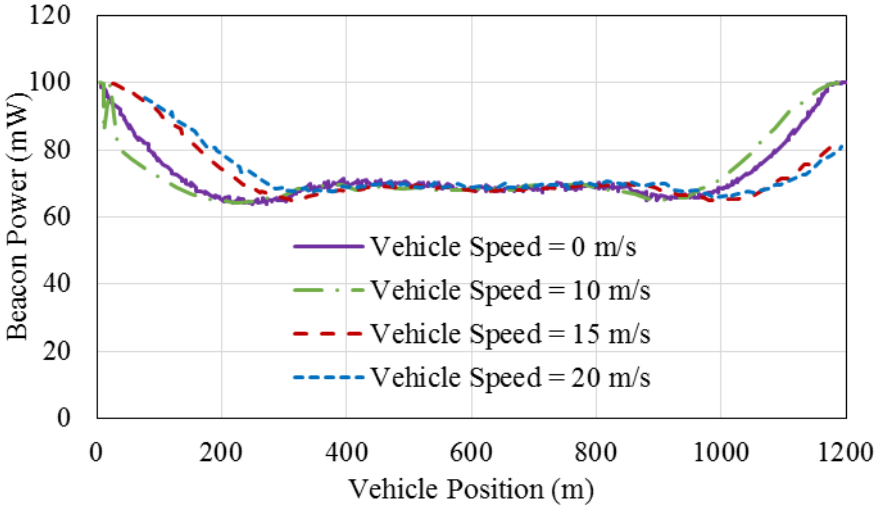

Figure 6.13 Beacon power for Figure 6.12

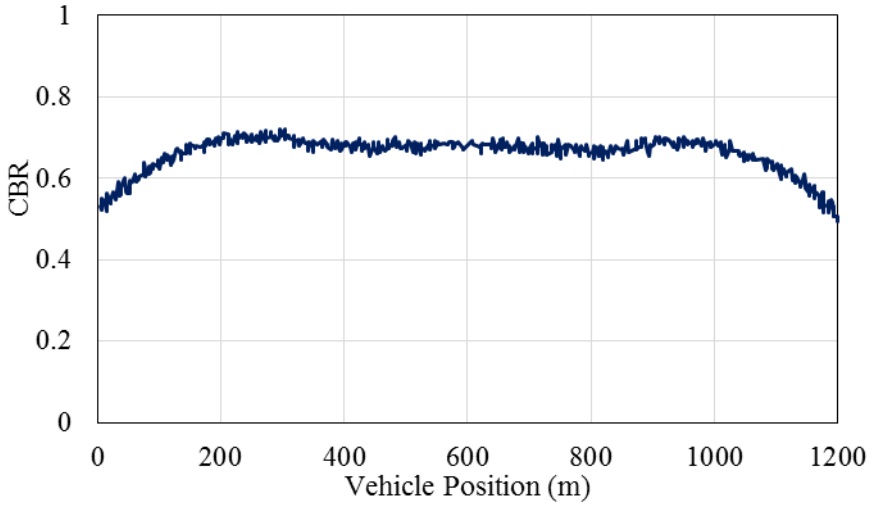

Figure 6.14 CBR for Figure 6.12

6.4 Summary

The problem of beaconing congestion was addressed with using joint adaption of beacon rate and power that was formulated as a non-cooperative game, in which the strategy spaces of the players are two-dimensional. The existence and uniqueness of the NE was proven mathematically, and an algorithm based on the gradient projection method for solving the game was proposed. The stability and convergence of the algorithm was demonstrated by simulation. Simulation results indicate that the algorithm converges to the NE from any initial point. It is seen that by selecting appropriate values for the parameters of the algorithm, fairness in beacon power and weighted fairness in beacon rate is achieved, and CBR is controlled at an appropriate level.

Chapter 7

Conclusions and Future Work

This thesis is commenced by a brief introduction to VANETs technology. As simulators are necessary tools for the study of VANET, a review on features of four widely used open source simulators in the simulation of VANETs is presented. As a result of the study, OMNeT++ was selected for making experiments in this thesis.

Next, the requirement of adaptive beaconing has been indicated by studying the performance of a proposed position-based routing protocol with a constant beaconing rate. In particular, we looked at a decrease of performance of the routing protocol by increasing the speed of vehicles. The proposed protocol is a bio-inspired routing protocol called EGSR based on the ACO. EGSR adds traffic awareness to the well-known GSR protocol without using any infrastructure or information obtained from outside the network.

In dense vehicular traffic conditions, many of the BSMs are lost due to packet collision. To increase the number of successfully disseminated packets, the channel load should be controlled under a specified level. Moreover, the scarce channel capacity should be allocated to vehicles based on their requirements or dynamics to have an efficient beaconing mechanism. This goal was addressed in Chapter 4 by BSM rate adjustment [131]. The problem of rate adjustment was modelled as a non-cooperative game; then, an algorithm was proposed so that the network nodes could solve the game.

Then, in the study, we turned our attention to BSM congestion control by beacon power adaptation. An algorithm for BSM power adjustment was proposed that can control channel usage at a desired level in a non-cooperative way. In addition, the algorithm has per-vehicle parameters so that every vehicle can control its share of the channel and weighted fairness in power is achieved by adjusting these parameters.

Afterwards, an algorithm for joint BSM power and rate control is proposed. This approach is useful in very dense traffic conditions when a decrease in beacon power

or rate alone cannot reduce the channel usage to an appropriate level. As shown, the algorithm can achieve fairness in power and weighted fairness in rate in a non-cooperative manner. In addition, the stability and convergence of the algorithm from any initial point to the unique equilibrium point of the system was verified in simulation scenarios.

7.1 Conclusions

In Chapter 3, it was indicated that in dense vehicular conditions, BSMs consume a lot of channel capacity. In addition, it was shown that a constant low beaconing rate cannot create a good awareness of neighbouring vehicles in VANETs when the speed of the vehicles is high. Therefore, an adaptive beaconing that is capable of decreasing the beaconing load in dense networks and increasing the beaconing rate based on the dynamics of a vehicle will result in higher performance for VANETs protocols.

Then, in Chapters 4 to 6, algorithms for beacon rate and power adaptation were proposed. The algorithms were compared with other state-of-the-art algorithms in several stationary and dynamic scenarios. The algorithms can efficiently control channel load at a desired level with reasonable speed. One important feature of the proposed congestion control algorithms is that their functionality does not rely on excess information exchange in beacons between vehicles, which results in bandwidth saving and more robustness to errors. In contrast to other congestion control algorithms that obtain fairness by exchanging algorithms information between nodes, fairness in the proposed algorithm is achieved based on the fairness concept of NE, which results in overhead-free algorithms. Furthermore, using simulation it was indicated that algorithms that obtain fairness by exchanging local information fail to provide fairness when the length of the scenario exceeds the range of information exchange. Being overhead free also makes the algorithms more scalable. Actually, the proposed algorithms just require local information, while, they achieved good fairness globally.

The proposed algorithms also have the capability to meet application requirements and provide weighted fairness in rate or power. This achievement is useful for situations in which vehicles with different application requirements contribute to

congestion control. The proposed algorithms also converge in a few seconds, which makes them suitable for high-speed scenarios of VANETs.

7.2 Future Work

For future work, we are interested in security issues in VANETs. As with any other computer network, a malicious attacker can inject incorrect information into VANETs or disable the services, which endangers the safety of driving. In VANETs, every node should be able to detect intruders and make the best decision in response.

Traditional network security solutions solve particular problems for which they are designed. In general, they fail to respond well in a dynamically changing scenario. This trend cannot be acceptable in the future world, and more sophisticated approaches will be required to manage such scenarios. Game theory has the potential of presenting improved security mechanisms. Network security shares many concerns with decision-making in game theory [132]. Securing networks requires detecting vulnerabilities and attacks, controlling access to networks, developing protocols for safe access to services, etc. and all these involve decision-making in different network layers. The security game framework has applications in security problems in a variety of areas ranging from intrusion detection to social, wireless and vehicular networks [132]. By applying game theory, instead of designing a defence against a specific attack, it is possible to design a defence against a sophisticated attacker who dynamically and strategically changes her attack methods and targets to fulfil her aims [18].

Game theory provides a rich mathematical tool for multi-person strategic decision-making to model the interactions of agents in security problems and to deploy limited security resources to maximize their effectiveness [18], [133]. This feature of game theory has many advantages over traditional security measures [132]. First, the process of decision-making is expressed through mathematical models in a transparent and persistent manner that enables a meticulous analysis of the problem. Second, the decision-making can now be generalized and deployed on a large scale as opposed to traditional methods, which are suitable just for specific problems. Third, it creates the opportunity to implement them numerically, so machines can run them as fast as the

machine speed allows. Finally, the decision-making process implemented by the analytical model can be checked experimentally and improved on a daily basis, providing a way of aggregating the knowledge of security experts.

Appendix A

A.1 Mathematical Model for CBR [121]

If there are N transmitter then the Channel load sensed by node i at position x is the sum of the load created by each one of those transmitters at point x so we can write:

$$load_i(x) = \sum_{n \in N} load_n(x) = \sum_{n \in N} sensible\ transmisions\ at\ x\ created\ by\ node\ n \times T_{frame} \quad \text{(A.1.1)}$$

Where T_{frame} is the duration of receiving a packet and $load_n(x)$ is the load created by node n at x and can be written as:

$$load_n(x) = r_n\ Prob(d_n)\ T_{frame} \quad \text{(A.1.2)}$$

Where, d_n is the distance between sender and receiver, $Prob(d_n)$ is the probability of reception of a packet at distance d_n and r_n is the packet rate of node n. By considering a Nakagami-m distribution for the received power p:

$$PDF\left(p\right) = \left(\frac{m}{\Omega_n}\right)^m \left(\frac{p^{m-1}}{\Gamma(m)}\right) e^{-\frac{m}{\Omega_n}p} \quad \text{(A.1.3)}$$

Where

$$\Omega_n = \frac{p_t\ \lambda^2}{(4\pi)^2\ d_n^{\gamma}} \quad \text{(A.1.4)}$$

p_t is the transmitter power, m is shape factor in Nakagami model and λ is the wavelength, then we can write:

$$prob\left(d_n\right) = \int_{C_{th}}^{\infty} PDF\left(p\right)\ dp = \int_{C_{th}}^{\infty} \left(\frac{m}{\Omega_n}\right)^m \left(\frac{p^{m-1}}{\Gamma(m)}\right) e^{-\frac{m}{\Omega_n}p}\ dp \quad \text{(A.1.5)}$$

By considering $t = \frac{m}{\Omega_n} p$ and $dp = \frac{\Omega_n}{m} dt$;

$$prob\left(d_n\right) = \int_{\frac{mC_{th}}{\Omega_n}}^{\infty} \left(\frac{m}{\Omega_n}\right)^{m-1} \left(\frac{1}{\Gamma(m)}\right) \left(\frac{\Omega_n\, t}{m}\right)^{m-1} e^{-t} dt = \frac{1}{\Gamma(m)} \int_{\frac{mC_{th}}{\Omega_n}}^{\infty} t^{m-1}\, e^{-t}\, dt$$

$$= \frac{\Gamma\left(m, \frac{m C_{th}}{\Omega_n}\right)}{\Gamma(m)} \tag{A.1.6}$$

Thus

$$load_n(x) = r_n T_{frame} \frac{\Gamma\left(m, \frac{m C_{th}}{\Omega_n}\right)}{\Gamma(m)} \tag{A.1.7}$$

And therefore, the sensed CBR by node i is:

$$CBR_i(x) = load_i(x) = \sum_{n \in N} r_n T_{frame} \frac{\Gamma\left(m, \frac{m C_{th}}{\Omega_n}\right)}{\Gamma(m)} \tag{A.1.8}$$

In the above formulas the effect of packet collision was not considered. The formula can be corrected by multiplying it to a coefficient less than one in order to include the effect of collisions.

A.2 Upper Incomplete Gamma Function [134]

$$\Gamma(a,z) = \int_z^\infty t^{a-1} e^{-t}\, dt \tag{A.2.1}$$

$$\frac{d}{dz}\Gamma(a,z) = -z^{a-1} e^{-z} \tag{A.2.2}$$

A.3 Gamma Function [134]

$$\Gamma(a) = \int_0^\infty t^{a-1} e^{-t} dt \tag{A.3.1}$$

References

[1] K. Katsaros and M. Dianati, "A conceptual 5G vehicular networking architecture," in *5G Mobile Communications* Springer, 2017, pp. 595-623.

[2] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, *(1)*, pp. 41-50, 2003.

[3] F. Dressler and O. B. Akan, "Bio-inspired networking: from theory to practice," *IEEE Communications Magazine*, vol. 48, *(11)*, pp. 176-183, 2010.

[4] G. Karagiannis *et al*, "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions," *IEEE Communications Surveys & Tutorials*, vol. 13, *(4)*, pp. 584-616, 2011.

[5] A. M. Vegni, M. Biagi and R. Cusani, *Smart Vehicles, Technologies and Main Applications in Vehicular Ad Hoc Networks.* INTECH Open Access Publisher, 2013.

[6] P. Papadimitratos *et al*, "Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation," *IEEE Communications Magazine*, vol. 47, *(11)*, pp. 84-95, 2009.

[7] C. Lochert *et al*, "A routing strategy for vehicular ad hoc networks in city environments," in *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, 2003, pp. 156-161.

[8] B. Karp and H. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, 2000, pp. 243-254.

[9] A. Mackenzie, L. DaSilva and W. Tranter, *Game Theory for Wireless Engineers.* (1st ed.) 2006*1*.

[10] D. Fudenberg and J. Tirole, "Game theory, 1991," *Cambridge, Massachusetts*, vol. 393, 1991.

[11] C. Zheng and D. C. Sicker, "A survey on biologically inspired algorithms for computer networking," *IEEE Communications Surveys & Tutorials*, vol. 15, *(3)*, pp. 1160-1191, 2013.

[12] C. U. Saraydar, N. B. Mandayam and D. J. Goodman, "Efficient power control via pricing in wireless data networks," *IEEE Trans. Commun.*, vol. 50, *(2)*, pp. 291-303, 2002.

[13] C. Wu and D. P. Bertsekas, "Distributed power control algorithms for wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 50, *(2)*, pp. 504-514, 2001.

[14] W. Yu, G. Ginis and J. M. Cioffi, "Distributed multiuser power control for digital subscriber lines," *IEEE J. Select. Areas Commun.,* vol. 20, *(5),* pp. 1105-1115, 2002.

[15] C. Doubligeris and R. Mazumdar, "A game theoretic approach to flow control in an integrated environment with two classes of users," in 1988.

[16] Z. Fang and B. Bensaou, "Fair bandwidth sharing algorithms based on game theory frameworks for wireless ad-hoc networks," in 2004.

[17] T. Alpcan, T. Basar and R. Tempo, "Randomized algorithms for stability and robustness analysis of high-speed communication networks," *IEEE Trans. Neural Networks,* vol. 16, *(5),* pp. 1229-1241, 2005.

[18] X. Liang and Y. Xiao, "Game theory for network security," *IEEE Communications Surveys & Tutorials,* vol. 15, *(1),* pp. 472-486, 2013.

[19] M. Sheng *et al*, "Zone-based load balancing in LTE self-optimizing networks: a game-theoretic approach," *IEEE Transactions on Vehicular Technology,* vol. 63, *(6),* pp. 2916-2925, 2014.

[20] A. B. Mac Kenzie and S. B. Wicker, "Game theory and the design of self-configuring, adaptive wireless networks," *IEEE Communications Magazine,* vol. 39, *(11),* pp. 126-131, 2001.

[21] S. Al-Sultan *et al*, "A comprehensive survey on vehicular Ad Hoc network," *Journal of Network and Computer Applications,* vol. 37, pp. 380-392, 2014.

[22] IEEE Std, "IEEE guide for wireless access in vehicular environments (WAVE) - architecture, IEEE std. 1609.0-2013," 2013.

[23] H. Hartenstein and K. Laberteaux, *VANET: Vehicular Applications and Inter-Networking Technologies.* Wiley Online Library, 2010*1*.

[24] IEEE Std, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," IEEE Std. 802.11, 2012.

[25] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proc IEEE,* vol. 99, *(7),* pp. 1162-1182, 2011.

[26] IEEE Std, "IEEE standard for wireless access in vehicular environments (WAVE) - multi-channel operation, IEEE std 1609.4-2016," 2016.

[27] IEEE Std, "IEEE Standard for Wireless Access in Vehicular Environments (WAVE)-Networking Services, IEEE Std. 1609.3-2010," *IEEE Std*.

[28] IEEE Std, "IEEE Standard for Wireless Access in Vehicular Environments (WAVE)—Over-the-Air Electronic Payment Data Exchange Protocol for Intelligent Transportation Systems (ITS), IEEE Std. 1609.11-2010," *IEEE Std*.

[29] IEEE Std, "IEEE Standard for Wireless Access in Vehicular Environments — Security Services for Applications and Management Messages, IEEE Std. 1609.2-2013," *IEEE Std*.

[30] B. Jerry, *Discrete-Event System Simulation.* Pearson Education India, 1984.

[31] D. Eckhoff, C. Sommer and F. Dressler, "On the necessity of accurate IEEE 802.11 p models for IVC protocol simulation," in *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th,* 2012, pp. 1-5.

[32] J. Gozalvez, M. Sepulcre and R. Bauza, "Impact of the radio channel modelling on the performance of VANET communication protocols," *Telecommunication Systems,* vol. 50, *(3),* pp. 149-167, 2012.

[33] *OMNeT++ Discrete Event Simulator*. Available: https://omnetpp.org/.

[34] *"SWANS user guide" jist.ece.cornell.edu* [Accessed 1/10/2013].

[35] S. Joerer, F. Dressler and C. Sommer, "Comparing apples and oranges?: Trends in IVC simulations," in *Proceedings of the Ninth ACM International Workshop on Vehicular Inter-Networking, Systems, and Applications,* 2012, pp. 27-32.

[36] R. Stanica, E. Chaput and A. Beylot, "Simulation of vehicular ad-hoc networks: Challenges, review of tools and recommendations," *Computer Networks,* vol. 55, *(14),* pp. 3179-3188, 2011.

[37] *www.nsnam.org* [Accessed 1/10/2013].

[38] P. K. Singh and K. Lego, "Comparative study of radio propagation and mobility models in vehicular adhoc network," *International Journal of Computer Applications (0975-8887),* vol. 16, *(8),* 2011.

[39] T. K. Sarkar *et al*, "A survey of various propagation models for mobile communication," *IEEE Antennas and Propagation Magazine,* vol. 45, *(3),* pp. 51-82, 2003.

[40] E. Weingartner, H. Vom Lehn and K. Wehrle, "A performance comparison of recent network simulators," in *2009 IEEE International Conference on Communications,* 2009, pp. 1-5.

[41] S. M. Bilalb and M. Othmana, "A Performance Comparison of Network Simulators for Wireless Networks," *arXiv Preprint arXiv:1307.4129,* 2013.

[42] F. J. Martinez *et al*, "A survey and comparative study of simulators for vehicular ad hoc networks (VANETs)," *Wireless Communications and Mobile Computing,* vol. 11, *(7),* pp. 813-828, 2011.

[43] *SUMO - Simulation of Urban MObility*. Available: https://sourceforge.net/.

[44] Y. P. Fallah *et al*, "Analysis of Information Dissemination in Vehicular Ad-Hoc Networks with Application to Cooperative Vehicle Safety Systems," *IEEE Transactions on Vehicular Technology,* vol. 60, *(1),* pp. 233-247, 2011.

[45] E. Egea-Lopez *et al*, "Statistical beaconing congestion control for vehicular networks," *IEEE Transactions on Vehicular Technology,* vol. 62, *(9),* pp. 4162-4181, 2013.

[46] M. Torrent-Moreno, P. Santi and H. Hartenstein, "Fair sharing of bandwidth in VANETs," in *Proceedings of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks,* 2005, pp. 49-58.

[47] J. Mittag *et al*, "Analysis and design of effective and low-overhead transmission power control for VANETs," in *Proceedings of the Fifth ACM International Workshop on VehiculAr Inter-NETworking,* 2008, pp. 39-48.

[48] M. Torrent-Moreno *et al*, "Vehicle-to-vehicle communication: fair transmit power control for safety-critical information," *IEEE Transactions on Vehicular Technology,* vol. 58, *(7),* pp. 3684-3703, 2009.

[49] G. Bansal, J. B. Kenney and C. E. Rohrs, "LIMERIC: A Linear Adaptive Message Rate Algorithm for DSRC Congestion Control," *IEEE Transactions on Vehicular Technology,* vol. 62, *(9),* pp. 4182-4197, 2013.

[50] E. Egea-Lopez and P. Pavon-Marino, "Distributed and Fair Beaconing Rate Adaptation for Congestion Control in Vehicular Networks," *IEEE Transactions on Mobile Computing, (1),* pp. 1-14, 2016.

[51] C. Huang *et al*, "Information dissemination control for cooperative active safety applications in vehicular ad-hoc networks," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE,* 2009, pp. 1-6.

[52] C. Huang *et al*, "Adaptive inter-vehicle communication control for cooperative safety systems," *IEEE Network,* vol. 24, *(1),* pp. 6-13, 2010.

[53] B. Kim, I. Kang and H. Kim, "Resolving the Unfairness of Distributed Rate Control in the IEEE WAVE Safety Messaging," *IEEE Transactions on Vehicular Technology,* vol. 63, *(5),* pp. 2284-2297, 2014.

[54] S. Kuk and H. Kim, "Preventing Unfairness in the ETSI Distributed Congestion Control," *IEEE Communications Letters,* vol. 18, *(7),* pp. 1222-1225, 2014.

[55] T. Tielert *et al*, "Joint power/rate congestion control optimizing packet reception in vehicle safety communications," in *Proceeding of the Tenth ACM International Workshop on Vehicular Inter-Networking, Systems, and Applications,* 2013, pp. 51-60.

[56] F. Goudarzi and H. Al-Raweshidy, "Distributed transmit power control for beacons in VANET," in *Proceedings of the 3rd International Conference on Vehicle Technology and Intelligent Transport Systems*, 2017, pp. 181-187.

[57] D. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Comput. Networks ISDN Syst.*, vol. 17, *(1)*, pp. 1-14, 1989.

[58] H. Shi *et al*, "Fairness in Wireless Networks: Issues, Measures and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 16, *(1)*, pp. 5-24, 2014.

[59] R. Jain, D. Chiu and W. R. Hawe, *A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System*. Eastern Research Laboratory, Digital Equipment Corporation Hudson, MA, 198*38*.

[60] M. Sepulcre *et al*, "Congestion and Awareness Control in Cooperative Vehicular Systems," *Proc IEEE*, vol. 99, *(7)*, pp. 1260-1279, 2011.

[61] M. Sepulcre *et al*, "Application-Based Congestion Control Policy for the Communication Channel in VANETs," *IEEE Communications Letters*, vol. 14, *(10)*, pp. 951-953, 2010.

[62] T. Tielert *et al*, "Design methodology and evaluation of rate adaptation based congestion control for vehicle safety communications," in *Vehicular Networking Conference (VNC), 2011 IEEE*, 2011, pp. 116-123.

[63] M. Sepulcre *et al*, "Adaptive beaconing for congestion and awareness control in vehicular networks," in *Vehicular Networking Conference (VNC), 2014 IEEE*, 2014, pp. 81-88.

[64] C. Sergiou, P. Antoniou and V. Vassiliou, "A comprehensive survey of congestion control protocols in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 16, *(4)*, pp. 1839-1859, 2014.

[65] R. K. Schmidt *et al*, "Exploration of adaptive beaconing for efficient inter-vehicle safety communication," *IEEE Network*, vol. 24, *(1)*, pp. 14-19, 2010.

[66] T. Hwang and P. Huang, "On new moment estimation of parameters of the gamma distribution using its characterization," *Annals of the Institute of Statistical Mathematics*, vol. 54, *(4)*, pp. 840-847, 2002.

[67] X. Shen *et al*, "Distributed congestion control approaches for the IEEE 802.11 p vehicular networks," *IEEE Intelligent Transportation Systems Magazine*, vol. 5, *(4)*, pp. 50-61, 2013.

[68] J. Sahoo *et al*, "Congestion-controlled-coordinator-based MAC for safety-critical message transmission in VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, *(3)*, pp. 1423-1437, 2013.

[69] ETSI TS 103 175 V1.1.1, "Intelligent Transport Systems (ITS); Cross Layer DCC Management Entity for operation in the ITS G5A and ITS G5B medium," 2015.

[70] ETSI TS 102 687 V1.1.1, "Intelligent Transport Systems (ITS); Decentralized Congestion Control Mechanisms for Intelligent Transport Systems operating in the 5 GHz range; Access layer part," *ETSI,* 2011.

[71] A. Autolitano *et al,* "An insight into decentralized congestion control techniques for VANETs from ETSI TS 102 687 V1. 1.1," in *Wireless Days (WD), 2013 IFIP,* 2013, pp. 1-6.

[72] A. Rostami *et al,* "Stability challenges and enhancements for vehicular channel congestion control approaches," *IEEE Transactions on Intelligent Transportation Systems,* vol. 17, *(10),* pp. 2935-2948, 2016.

[73] C. Perkins, E. Belding-Royer and S. Das, *Ad Hoc on-Demand Distance Vector (AODV) Routing,* RFC 3561, 2003.

[74] P. Jacquet *et al,* "Optimized link state routing protocol for ad hoc networks," in *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International,* 2001, pp. 62-68.

[75] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing* Springer, 1996, pp. 153-181.

[76] A. Husain *et al,* "Performance Comparison of Topology and Position Based Routing Protocols in Vehicular Network Environments," *International Journal of Wireless & Mobile Networks (IJWMN) Vol,* vol. 3, 2011.

[77] C. Wu, S. Ohzahata and T. Kato, "Flexible, portable, and practicable solution for routing in VANETs: a fuzzy constraint Q-learning approach," *IEEE Transactions on Vehicular Technology,* vol. 62, *(9),* pp. 4251-4263, 2013.

[78] M. Al-Rabayah and R. Malaney, "A new scalable hybrid routing protocol for VANETs," *IEEE Transactions on Vehicular Technology,* vol. 61, *(6),* pp. 2625-2635, 2012.

[79] J. Toutouh, J. García-Nieto and E. Alba, "Intelligent OLSR routing protocol optimization for VANETs," *IEEE Transactions on Vehicular Technology,* vol. 61, *(4),* pp. 1884-1894, 2012.

[80] J. F. Bravo-Torres *et al,* "Optimizing Reactive Routing Over Virtual Nodes in VANETs," *IEEE Transactions on Vehicular Technology,* vol. 65, *(4),* pp. 2274-2294, 2016.

[81] D. S. De Couto *et al,* "A high-throughput path metric for multi-hop wireless routing," *Wireless Networks,* vol. 11, *(4),* pp. 419-434, 2005.

[82] J. Wu, *A Simulation Study on using the Virtual Node Layer to Implement Efficient and Reliable MANET Protocols,* 2011.

[83] A. Gorrieri and G. Ferrari, "Irresponsible AODV routing," *Vehicular Communications,* vol. 2, *(1),* pp. 47-57, 2015.

[84] T. H. Cormen, *Introduction to Algorithms.* MIT press, 2009.

[85] B. Seet *et al*, "A-STAR: A mobile ad hoc routing strategy for metropolis vehicular communications," in *International Conference on Research in Networking,* 2004, pp. 989-999.

[86] F. Giudici and E. Pagani, "Spatial and traffic-aware routing (STAR) for vehicular systems," in *International Conference on High Performance Computing and Communications,* 2005, pp. 77-86.

[87] M. Jerbi *et al*, "An improved vehicular ad hoc routing protocol for city environments," in *2007 IEEE International Conference on Communications,* 2007, pp. 3972-3979.

[88] T. Darwish and K. A. Bakar, "Traffic density estimation in vehicular ad hoc networks: A review," *Ad Hoc Networks,* vol. 24, pp. 337-351, 2015.

[89] J. Nzouonta *et al*, "VANET routing on city roads using real-time vehicular traffic information," *IEEE Transactions on Vehicular Technology,* vol. 58, *(7),* pp. 3609-3626, 2009.

[90] P. K. Sahu *et al*, "BAHG: back-bone-assisted hop greedy routing for VANET's city environments," *IEEE Transactions on Intelligent Transportation Systems,* vol. 14, *(1),* pp. 199-213, 2013.

[91] H. Saleet *et al*, "Intersection-based geographical routing protocol for VANETs: a proposal and analysis," *IEEE Transactions on Vehicular Technology,* vol. 60, *(9),* pp. 4560-4574, 2011.

[92] V. Naumov and T. R. Gross, "Connectivity-aware routing (CAR) in vehicular ad-hoc networks," in *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications,* 2007, pp. 1919-1927.

[93] S. L. O. Correia, J. Celestino and O. Cherkaoui, "Mobility-aware ant colony optimization routing for vehicular ad hoc networks," in *2011 IEEE Wireless Communications and Networking Conference,* 2011, pp. 1125-1130.

[94] I. Chakeres and C. Perkins, "Dynamic MANET on-demand (DYMO) routing," *Draft-Ietf-Manet-Dymo-14,* 2008.

[95] R. R. Sahoo *et al*, "A trust based clustering with ant colony routing in VANET," in *Computing Communication & Networking Technologies (ICCCNT), 2012 Third International Conference On,* 2012, pp. 1-8.

[96] H. Rana, P. Thulasiraman and R. K. Thulasiram, "MAZACORNET: Mobility aware zone based ant colony optimization routing for VANET," in *2013 IEEE Congress on Evolutionary Computation*, 2013, pp. 2948-2955.

[97] G. Li and L. Boukhatem, "Adaptive vehicular routing protocol based on ant colony optimization," in *Proceeding of the Tenth ACM International Workshop on Vehicular Inter-Networking, Systems, and Applications*, 2013, pp. 95-98.

[98] M. Dorigo and G. Di Caro, "Ant colony optimization: A new meta-heuristic," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress On*, 1999, pp. 1470-1477.

[99] S. Lasaulce and H. Tembine, *Game Theory and Learning for Wireless Networks: Fundamentals and Applications.* Academic Press, 2011.

[100] M. Wang and T. Suda, "The bio-networking architecture: A biologically inspired approach to the design of scalable, adaptive, and survivable/available network applications," in *Applications and the Internet, 2001. Proceedings. 2001 Symposium On*, 2001, pp. 43-53.

[101] C. Prehofer and C. Bettstetter, "Self-organization in communication networks: principles and design paradigms," *IEEE Communications Magazine*, vol. 43, *(7)*, pp. 78-85, 2005.

[102] F. Dressler and O. B. Akan, "A survey on bio-inspired networking," *Computer Networks*, vol. 54, *(6)*, pp. 881-900, 2010.

[103] S. Bitam, A. Mellouk and S. Zeadally, "Bio-inspired routing algorithms survey for vehicular ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 17, *(2)*, pp. 843-867, 2015.

[104] M. Farooq and G. A. Di Caro, "Routing protocols for next-generation networks inspired by collective behaviors of insect societies: An overview," in *Swarm Intelligence* Springer, 2008, pp. 101-160.

[105] R. Schoonderwoerd *et al*, "Ant-based load balancing in telecommunications networks," *Adapt. Behav.*, vol. 5, *(2)*, pp. 169-207, 1997.

[106] G. Di Caro and M. Dorigo, "AntNet: Distributed stigmergetic control for communications networks," *Journal of Artificial Intelligence Research*, vol. 9, pp. 317-365, 1998.

[107] G. Di Caro, F. Ducatelle and L. M. Gambardella, "AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks," *Eur. Trans. Telecommun.*, vol. 16, *(5)*, pp. 443-455, 2005.

[108] S. Kamali and J. Opatrny, "Posant: A position based ant colony routing algorithm for mobile ad-hoc networks," in *Wireless and Mobile Communications, 2007. ICWMC'07. Third International Conference On*, 2007, pp. 21-21.

[109] T. Camp, J. Boleng and L. Wilcox, "Location information services in mobile ad hoc networks," in *Communications, 2002. ICC 2002. IEEE International Conference On,* 2002, pp. 3318-3324.

[110] M. Käsemann *et al*, "Analysis of a location service for position-based routing in mobile ad hoc networks." in *Wman,* 2002, pp. 121-133.

[111] K. Katsaros, M. Dianati and L. Le, "Effective implementation of location services for VANETs in hybrid network infrastructures," in *Communications Workshops (ICC), 2013 IEEE International Conference On,* 2013, pp. 521-525.

[112] K. N. Nakorn and K. Rojviboonchai, "POCA: Position-aware reliable broadcasting in vehicular ad-hoc networks," in *Proceedings of 2010 Second Asia-Pacific Conference on Information Processing (APCIP 2010),* 2010.

[113] F. Ducatelle, G. A. Di Caro and L. M. Gambardella, "An analysis of the different components of the AntHocNet routing algorithm," in *International Workshop on Ant Colony Optimization and Swarm Intelligence,* 2006, pp. 37-48.

[114] S. Yousefi, S. Bastani and M. Fathy, "On the performance of safety message dissemination in vehicular ad hoc networks," in *Universal Multiservice Networks, 2007. ECUMN'07. Fourth European Conference On,* 2007, pp. 377-390.

[115] J. Harri, F. Filali and C. Bonnet, "Mobility models for vehicular ad hoc networks: a survey and taxonomy," *IEEE Communications Surveys & Tutorials,* vol. 11, *(4),* 2009.

[116] K. Prasanth *et al*, "Improved packet forwarding approach in Vehicular ad hoc networks using RDGR algorithm," *arXiv Preprint arXiv:1003.5437,* 2010.

[117] A. Mostafa, A. M. Vegni and D. P. Agrawal, "A probabilistic routing by using multi-hop retransmission forecast with packet collision-aware constraints in vehicular networks," *Ad Hoc Networks,* vol. 14, pp. 118-129, 2014.

[118] K. Katsaros *et al*, "An evaluation of routing in vehicular networks using analytic hierarchy process," *Wireless Communications and Mobile Computing,* vol. 16, *(8),* pp. 895-911, 2016.

[119] J. K. MacKie-Mason and H. R. Varian, "Pricing congestible network resources," *IEEE J. Select. Areas Commun.,* vol. 13, *(7),* pp. 1141-1149, 1995.

[120] F. Kelly, "Charging and Rate Control for Elastic Traffic," *Eur. Trans. Telecommun.,* vol. 8, *(1),* pp. 33-37, 1997.

[121] Q. Chen *et al*, "Mathematical modelling of channel load in vehicle safety communications," in *Vehicular Technology Conference (VTC Fall), 2011 IEEE,* 2011, pp. 1-5.

[122] D. M. Topkis, "Equilibrium points in nonzero-sum n-person submodular games," *SIAM Journal on Control and Optimization,* vol. 17, *(6),* pp. 773-787, 1979.

[123] T. Başar and G. J. Olsder, *Dynamic Non-cooperative Game Theory.* 1982*160.*

[124] J. B. Rosen, "Existence and uniqueness of equilibrium points for concave n-person games," *Econometrica: Journal of the Econometric Society,* pp. 520-534, 1965.

[125] A. Hefti, "On the relationship between uniqueness and stability in sum-aggregative, symmetric and general differentiable games," *Mathematical Social Sciences,* vol. 80, pp. 83-96, 2016.

[126] J. B. Rosen, "The Gradient Projection Method for Nonlinear Programming. Part II. Nonlinear Constraints," *Journal of the Society for Industrial and Applied Mathematics,* vol. 9, *(4),* pp. 514-532, 1961.

[127] F. Chatelin, *Eigenvalues of Matrices.* 1993.

[128] J. M. Carnicer, T. N. T. Goodman and J. M. Peña, "Linear conditions for positive determinants," *Linear Algebra and its Applications,* vol. 292, *(1),* pp. 39-59, 1999.

[129] J. M. Peña, "A class of P-matrices with applications to the localization of the eigenvalues of a real matrix," *SIAM Journal on Matrix Analysis and Applications,* vol. 22, *(4),* pp. 1027-1037, 2001.

[130] G. Bansal and J. B. Kenney, "Achieving weighted-fairness in message rate-based congestion control for DSRC systems," in *Wireless Vehicular Communications (WiVeC), 2013 IEEE 5th International Symposium On,* 2013, pp. 1-5.

[131] F. Goudarzi and H. Asgari, "Non-cooperative Beacon Rate and Awareness Control for VANETs," *IEEE Access,* vol. 5, 2017.

[132] T. Alpcan and T. Başar, *Network Security: A Decision and Game-Theoretic Approach.* Cambridge University Press, 2010.

[133] M. H. Manshaei *et al*, "Game theory meets network security and privacy," *ACM Computing Surveys (CSUR),* vol. 45, *(3),* pp. 25, 2013.

[134] F. W. Olver, *NIST Handbook of Mathematical Functions.* Cambridge University Press, 2010.