uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Brijesh Kadri Mohandas
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.C.S.
GRADE / DEGREE

School of Information Technology and Engineering
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Congestion Management in Vehicular Ad-Hoc Networks

TITRE DE LA THÈSE / TITLE OF THESIS

Oliver Yang
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

Ramico Liscano
CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Wael Gueaieb

Chung Horng-Lung

Gary W. Slater
Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Congestion Management in Vehicular Ad-Hoc Networks

Brijesh Kadri Mohandas

A thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

MASTER OF COMPUTER SCIENCE

Ottawa-Carleton Institute for Computer Science
School of Information Technology and Engineering
University of Ottawa
Ottawa, Canada

Your file Votre référence
ISBN: 978-0-494-59898-6
Our file Notre référence
ISBN: 978-0-494-59898-6

Dedicated to my parents,

Mrs. Dhanavanthi Mohandas and (Late) Mr. Kadri Mohandas…

Abstract

We study the performance of the adaptive PI (Proportional Integral) rate controller [HoYa07] to deal with the problem of vehicle traffic congestion and data traffic congestion in vehicular ad-hoc networks. Derived from classical control theory, the adaptive PI rate controller performs active queue management by controlling the source rate based on the instantaneous queue length of the congested node. Using simulations we have demonstrated the capability of the adaptive PI rate controller to deal with vehicle traffic and data traffic congestion in vehicular ad-hoc networks. For data traffic congestion management, we have evaluated the performance of the algorithm in both static and mobile network scenarios. The controller dominated the combined performance of TCP NEWRENO and RED (Random Early Detection) queue with a positive influence on source throughput. For vehicle traffic management we have studied the congestion condition that arises when vehicle traffic volume exceeds the capacity of the road. The adaptive PI rate algorithm controlled the traffic volume in the road segment of interest to a preconfigured target threshold additionally improving the vehicle speed. Various other performance measures have been evaluated.

Acknowledgements

Foremost I would like to thank my supervisors Dr. Oliver Yang and Dr. Ramiro Liscano for their constant guidance and support. I would also like to express my appreciation to members of CCNR lab at University of Ottawa for their fruitful discussions.

I reserve my highest gratitude for my mother for her everlasting love and support. I also wish to thank my sisters, Asha and Kavitha, and my brother Dr. Shailesh, for their constant love and support.

The financial support from Auto21, a Network of Centres of Excellence, is acknowledged.

Table of Contents

Title I
Abstract III
Acknowledgements IV
Table of Contents V
List of Figures VIII
List of Tables X
List of Acronyms XI
List of Notations and Symbols XII
Chapter 1 Introduction 1
1.1 Vehicular Ad-Hoc Networks Overview 1
1.2 Congestion in Vehicular Ad-Hoc Networks 1
1.3 Dedicated Short Range Communications (DSRC) 2
1.4 Motivations 3
1.5 Objectives 5
1.6 Methodologies 6
1.7 Thesis Contributions 8
1.8 Thesis Organization 8
1.9 Publications 9
Chapter 2 Literature Review 10
2.1 Data Traffic Congestion Management in VANET 10
2.1.1 Window-based Congestion Control Algorithms 12
2.1.2 Rate-based Congestion Control Algorithms 14
2.2 Vehicle Traffic Congestion Management in VANET 16
2.2.1 Traffic Signal Management 16
2.2.2 Traffic Congestion Detection and Reporting 17
2.2.3 Intelligent Traffic Control 18
2.3 Vehicle traffic simulator 19
2.4 Routing in VANET 20
2.5 Concluding Remarks 20
Chapter 3 Network Operations, Models and Assumptions 21
3.1 Network Operations 21
3.2 Network Model 22
3.2.1 Congestion Control Model for Vehicle Traffic 22
3.2.2 Congestion Control Model for Data Traffic 24
3.2.3 Assumptions 25
3.3 Simulation Model 25
3.3.1 Vehicle Traffic Management Model 25
3.3.1.1 Implementation in Qualnet 26
3.3.1.2 Vehicle Behavior 27
3.3.1.3 Mobility Characteristics 28
3.3.1.4 Roadside Unit Behavior 29
3.3.1.5 Assumptions 31
3.3.2 Data Traffic Management Model 32

3.3.2.1 Implementation in Qualnet 32
3.3.2.2 Source Behavior 33
3.3.2.3 Intermediate Node Behavior 35
3.3.2.4 Destination Node Behavior 35
3.3.2.5 Mobility Characteristics 36
3.3.2.6 Assumptions 37
3.4 Simulation Scenarios 37
3.4.1 Vehicle Traffic Congestion Control Scenario 38
3.4.2 Data Traffic Congestion Control Scenario in a Static VANET 39
3.4.2.1 Small Static Network 39
3.4.2.2 Large Static Network 40
3.4.3 Data Traffic Congestion Control Scenario in Mobile VANET 41
3.4.3.1 City Traffic Scenario 42
3.4.3.2 Rural Scenario 43
3.5 Performance Measures 44
3.5.1 Performance Measures for Vehicle Traffic Congestion Management 44
3.5.2 Performance Measures for Data Traffic Congestion Management 45
3.6 Concluding Remarks 45
Chapter 4 Vehicle Traffic Management in VANET 47
4.1 Simulation Environment 47
4.2 Simulation Configuration Setting 47
4.3 Performance Evaluation 49
4.3.1 Vehicle Traffic Congestion Management 49
4.3.1.1 Scenario 1: Desired Speed of 20 m/s (72 KM/hr) and Target Queue Length of 60 vehicles in a 2000 meter road segment 50
4.3.1.2 Scenario 2: Desired Speed of 25 m/s (90 KM/hr) and Target Queue Length of 50 vehicles in a 2000 meter road segment 52
4.3.1.3 Scenario 3: Desired Speed of 30 m/s (108 KM/hr) and Target Queue Length of 40 vehicles in a 2000 meter road segment 54
4.4 Concluding Remarks 56
Chapter 5 Data Traffic Management in VANET 57
5.1 Simulation Environment 57
5.2 Simulation Configuration Setting 58
5.3 Performance Evaluation & Comparison 60
5.3.1 Data Traffic Congestion Control in Static VANET 60
5.3.1.1 Small Static Network 60
5.3.1.2 Large Static Network 65
5.3.2 Data Traffic Congestion Control in Mobile VANET 71
5.3.2.1 Large City Network 71
5.3.2.2 Large Rural Network 76
5.4 Concluding Remarks 77
Chapter 6 Design Guideline 78
6.1 Vehicle Traffic Congestion Management 78
6.1.1 Controller Design 78
6.1.2 Guidelines for Designing the System 79
6.2 Data Traffic Congestion Management 80
6.2.1 Controller Design 80

6.2.2 Guidelines for Designing the System 82
Chapter 7 Conclusions 83
7.1 Future Work 84
References 86
Appendix A Adaptive PI Rate Controller [HoYa07] 91
Appendix B Single-Lane Traffic Simulation with Multi Agent System (Car following model) [ZhLi05] 94
Appendix C Designing Scenarios in Qualnet [Qual08] 97
Appendix D Large Mobile Network Results for Rural Traffic Scenario 99
Appendix E Small Mobile Network Results for City Traffic and Rural Traffic Scenarios 103

List of Figures

Figure 2.1: Literature classification for congestion management in VANET 10
Figure 3.1: Network model depicting vehicle traffic congestion in VANET 22
Figure 3.2: Network model depicting data traffic congestion in VANET 24
Figure 3.3: Vehicle behaviors while using PI Rate control 27
Figure 3.4: Source Roadside unit behaviors while using PI Rate control 29
Figure 3.5: Destination Roadside unit behaviors while using PI Rate control 30
Figure 3.6: Adaptive PI Rate algorithm operations for vehicle traffic management 30
Figure 3.7: Vehicle behaviors while using PI Rate control 33
Figure 3.8: Roadside unit behaviors while using PI Rate control 34
Figure 3.9: Destination node behaviors while using PI Rate control 35
Figure 3.10: Vehicle traffic congestion management scenario. 38
Figure 3.11: Small static vehicular network 39
Figure 3.12: Large static vehicular network 41
Figure 3.13: A City road network Example 42
Figure 3.14: City traffic congestion scenario 42
Figure 3.15: A Rural road network Example 43
Figure 3.16: Rural traffic congestion scenario 43
Figure 4.1: Scenario 1 – Queue length behavior of road segment 50
Figure 4.2: Scenario 1 – Throughput at the source roadside unit 51
Figure 4.3: Scenario 1 – Average speed of the vehicle 51
Figure 4.4: Scenario 1 – Average trip time of the vehicle 52
Figure 4.5: Scenario 2 – Queue length behavior of road segment 52
Figure 4.6: Scenario 2 – Throughput at the source roadside unit 53
Figure 4.7: Scenario 2 – Average speed of the vehicle 53
Figure 4.8: Scenario 2 – Average trip time of the vehicle 54
Figure 4.9: Scenario 3 – Queue length behavior of road segment 54
Figure 4.10: Scenario 3 – Throughput behavior at the source roadside unit 55
Figure 5.1: Application throughput for a small static network 61
Figure 5.2: Mean throughput by node ID for a small static network 62
Figure 5.3: Instantaneous queue length behavior for a small static network 62
Figure 5.4: Average queue length vs. uncontrolled traffic rate in a small static network 63
Figure 5.5: Source TCP window behavior for a small static network 63
Figure 5.6: RTT behavior for a small static network 64
Figure 5.7: Average RTT vs. uncontrolled traffic rate in a small static network 64
Figure 5.8: Application throughput for a large static network 66
Figure 5.9: Mean throughput by node ID for a large static network 67
Figure 5.10: Instantaneous queue length behavior for a large static network 67
Figure 5.11: Average queue length vs. uncontrolled traffic rate in a large static network 68
Figure 5.12: Source TCP window behavior for a large static network 68
Figure 5.13: RTT behavior for a large static network 69
Figure 5.14: Average RTT vs. uncontrolled traffic rate in a large static network 69
Figure 5.15: Application throughput for a large city network 71
Figure 5.16: Mean throughput by node ID for a large city network 72
Figure 5.17: Instantaneous queue length behavior for a large city network 72
Figure 5.18: Average queue length vs. uncontrolled traffic rate for a large city network 73

Figure 5.19: Source TCP window behavior for a large city network 73
Figure 5.20: RTT behavior for a large city network 74
Figure 5.21: Average RTT vs. uncontrolled traffic rate for a large city network 74
Figure A.1: System model of the intermediate node..91
Figure A.2: Control System 92
Figure B.1: Decision tree for longitudinal movement 95
Figure D.1: Application throughput for a large rural network..99
Figure D.2: Instantaneous queue length behavior for a large rural network.........................100
Figure D.3: Average queue length vs. uncontrolled traffic rate for a large rural network...100
Figure D.4: Source TCP window behavior for a large rural network..................................101
Figure D.5: RTT behavior for a large rural network..101
Figure D.6: Average RTT vs. uncontrolled traffic for a large rural network........................102

List of Tables

Table 4.1: General Simulation Configuration Settings 48
Table 4.2: Radio/Physical Layer Configuration Settings 48
Table 4.3: Network Protocol Configuration Settings 49
Table 4.4: Performance summary for vehicle traffic congestion management 55
Table 5.1: General Configuration Settings 58
Table 5.2: Wireless Configuration Settings 59
Table 5.3: MAC, Network & Transport layer Configuration Settings 59
Table 5.4: Performance summary for a static network with 20 vehicles 65
Table 5.5: Performance summary for a static network with 100 vehicles 70
Table 5.6: Performance summary for a city network with 100 vehicles 75
Table 5.7: Performance summary for a rural network with 100 vehicles 76
Table 6.1: TCP header structure 81
Table E.1: Performance summary for a city network with 20 vehicles 103
Table E.2: Performance summary for a rural network with 20 vehicles 104

List of Acronyms

		Section of 1st appearance
ACK	Acknowledgement	3.2.2
AIMD	Additive Increase Multiplicative Decrease	2.1.1
AODV	Ad-hoc On Demand Distance Vector	2.4
API	Application Program Interface	1.6
APIRATE	Adaptive Proportional Integral Rate Controller	1.4
AQM	Active Queue Management	1.4
ASTM	American Society for Testing and Materials	1.3
ATSAC	Automated Traffic Surveillance and Control System	2.2.2
AVQ	Adaptive Virtual Queue	2.1.1
CBR	Constant Bit Rate	5.1
DSRC	Dedicated Short Range Communications	1.1
EXACT	Explicit rate based flow control	2.1.2
FCC	Federal Communications Commission	1.3
FIFO	First In First Out	3.2.2
FTP	File Transfer Protocol	3.4.2.1
GMSF	Generic Mobility Simulation Framework	3.3.2.5
GPS	Global Positioning System	2.2.3
IEEE	Institute of Electrical and Electronics Engineers	1.3
IP	Internet Protocol	1.6
MAC	Medium Access Control	2.1
MANET	Mobile ad-hoc network	1.1
NRED	Neighborhood Random Early Detection	2.1.1
PI	Proportional Integral	3.3
RED	Random Early Detection	1.4
REM	Random Exponential Marking	2.1.1
RSU	Roadside Unit	1.6
RTT	Round Trip Time	1.6
SUMO	Simulation of Urban Mobility	1.6
TCP	Transmission Control Protocol	1.2
TFRC	TCP Friendly Rate Control	1.4
TV	Television	2
UBPFCC	Utility based packet forwarding and congestion control	2.1.2
UDP	User Datagram Protocol	3.4.2.1
VANET	Vehicular ad-hoc network	1.1

List of Notations and Symbols

Symbol	Description	Section of 1st appearance
F	Destination roadside unit installed at the end of the road segment of interest	3.2.1
K	Target threshold of vehicles that can use the road segment of interest	3.2.1
K_i	Target threshold of vehicles that can use the road segment in area i	3.2.1
K_I	Integral gain	3.3.1.4
K_P	Proportional gain	3.3.1.4
N	Number of flows	3.2.2
$q_i(t)$	Initial source transmission rate	3.2.2
$q'_i(t)$	Newly calculated source transmission rate	3.2.2
I	Source roadside unit installed at the entrance of the road segment of interest	3.2.1
S_i	Source roadside unit installed in area i	3.2.1
$v(t)$	Uncontrolled traffic	3.2.2
$x(t)$	Instantaneous queue length	3.2.1
Z	Road segment of interest	3.2.1
τ_{fi}	Forward time delay	3.2.2
τ_{bi}	Feedback time delay	3.2.2
τ_i	Round trip time	3.2.2
μ	Service rate	3.2.2
λ	Aggregate arrival rate	3.2.2

Chapter 1
Introduction

1.1 Vehicular Ad-Hoc Networks Overview

One of the latest developments in the automobile industry is to embed intelligence into automobiles so that they can work together to provide a safe and enjoyable ride. The scientific term for this technology is vehicular ad-hoc network. A VANET (Vehicular Ad-Hoc Network) is a network of intelligent vehicles capable of communicating with each other and a roadside infrastructure consisting of RSUs (Roadside Units). A vehicular network exhibits similarities with the existing MANET (Mobile ad-hoc network). The phrase ad-hoc network is used for a network of distributed nodes capable of communicating with each other without relying on a backbone network. In order to facilitate communication in vehicular networks, each automobile is equipped with a radio device. A new standard called DSRC (Dedicated Short Range Communications) is being developed for vehicle to vehicle and vehicle to infrastructure communication. Existing wireless standards like WiFi and WiMAX are also being considered for vehicular networks. The main aim of introducing vehicular networks is to enhance safety and comfort for passengers. This is done by informing the drivers about events that might influence their driving behavior. In addition to this, various application services could be provided over VANET. The capabilities of VANET can also be used to provide efficient and innovative solutions to some of the existing problems in vehicular networks.

1.2 Congestion in Vehicular Ad-Hoc Networks

Vehicular ad-hoc networks have revolutionized the automobile industry. Using this technology, automobile manufactures and service providers are able to provide various services for drivers and passengers. On demand traffic updates, collision warnings, warnings about road conditions, Internet access, and streaming videos are some examples of the services that can be provided over vehicular networks. These services provide assistance and entertainment for drivers and passengers.

The service providing capability of VANET introduces the problem of data traffic congestion in vehicular networks. Service providers are usually allotted with fixed bandwidth to provide their service. Congestion arises when there is a demand for more bandwidth than the available capacity. One way to address this problem is to increase the bandwidth. However, this approach is not cost effective for a wireless network. Another approach is to share the available bandwidth among the users without causing congestion, i.e. congestion avoidance with efficient bandwidth management. Some of the applications, like streaming video, require a steady supply of bandwidth to provide un-interrupted service. This is a challenging task to achieve in the presence of congestion. The transport layer protocol like TCP (Transmission Control Protocol) plays a significant role in addressing the problem of congestion. Some researchers have also devised new protocols to address the problem of data traffic congestion.

Another nature of congestion observed in vehicular networks is the vehicle traffic congestion. Accidents, construction work, weather conditions, increase in traffic volume, or poor signal timing are some causes for vehicle traffic congestion. In order to address the problem of vehicle traffic congestion, each problem causing congestion has to be identified and addressed. One of the significant causes for traffic congestion is increase in vehicle traffic volume above the road capacity. Traffic engineers have devised different techniques to address the problem of vehicle traffic congestion. One significant approach is to monitor the traffic and take necessary measures when there is congestion. Traffic lights and message signs also play an important role in dealing with this problem. Information about traffic congestion can also be obtained from radio or television programs. The advent of VANETs has made it possible to use new techniques while dealing with problems like vehicle traffic congestion.

1.3 Dedicated Short Range Communications (DSRC)

DSRC is a medium range communication service that supports inter-vehicle and vehicle-to-roadside communication. The FCC (Federal Communications Commission) has allotted a block of spectrum in the range of 5.850 to 5.925 GHz to provide this service. Since then, The IEEE (Institute of Electrical and Electronics Engineers) working group and ASTM (American Society for Testing and Materials) are working on standards for this service. The standard extends IEEE 802.11a for vehicular networks. The IEEE version for DSRC is IEEE 802.11p. DSRC supports a very high data rate (6 – 54 Mbps) with a communication range of

1000 meters. Current DSRC applications operate in 915 MHz band and provide 12 MHz of spectrum shared by DSRC applications, cordless telephone, door openers and many other non-licensed wireless applications. The new 5.9 MHz frequency supports much higher data rate and provides 75 MHz of spectrum for DSRC applications.

At this time, DSRC is mainly used in electronic toll collection. Potential applications of DSRC are emergency warning systems for vehicles, adaptive cruise control, forward collision warning, electronic parking payments, approaching emergency vehicle warning, transit or emergency vehicle signal priority, and in-vehicle signing. Vehicles equipped with DSRC can communicate directly with each other, making it possible to send warning messages over to neighboring vehicles. Using the ad-hoc networking concept, this information can further penetrate into the network. Each vehicle in vehicular network can act as a sender, receiver or a router. DSRC can also be used to provide in-vehicle entertainment for drivers and passengers.

1.4 Motivations

We are interested in the problem of data traffic congestion and vehicle traffic congestion in VANET in its simple and common form. The ability to exchange data / information over the VANET gives way to data traffic congestion. Data traffic congestion leads to formation of queue at the congested node. As the buffer size is finite, any newly arrived packet will be dropped once the buffer is full. If an efficient congestion control algorithm is not used, such an event drastically affects the service provided.

A transport layer protocol like TCP provides end-to-end congestion control. It performs an initial handshake where the source and destination node exchange their window size to learn their transmission capabilities. This is an efficient approach only if the intermediate nodes can support the negotiated rate. Various TCP variants have also been devised to provide better congestion control mechanisms. It has also been observed that TCP variants with active queue management are more efficient than just using the TCP protocol [FlJa93].

In vehicular networks a wide variety of application services can be provided to vehicles using the external interface provided by RSUs, often called base stations. The ability to provide these services introduces more traffic, and therefore the problem of data traffic congestion in VANET. To address such a problem, we need a congestion control algorithm that can act upon congestion at any of the nodes along the transmission path. In the event of congestion, most of the congestion control algorithms are not able to distribute the bandwidth fairly among all the users. In some cases, packet drops are also observed at intermediate nodes. Furthermore, the throughput at the source node fluctuates resulting in poor quality of service. These factors have driven us to look for a congestion control algorithm that can address all these issues.

The algorithm we are looking for should be able to share the fixed bandwidth fairly among all the users and it should react to congestion without dropping packets. An important resource at each node is the buffer. The size of this buffer is often finite and it should be shared equally among all the users. For doing this, the algorithm should be able to tune the source sending rate based on instantaneous queue length at the intermediate node. Even in the event of congestion, the algorithm should be able to provide a constant throughput at the source node without reducing the quality of service. The APIRATE (Adaptive Proportional Integral rate) controller [HoYa07] is such an algorithm designed for the Internet. We are interested in extending and evaluating the performance of this algorithm in vehicular networks.

The significance of this algorithm is that it relies only on transport and network layers. It is independent of any other layers in the protocol stack. It is based on control theory and uses the proportional and integral value to address the deviations in queue length around the target value. Unlike most of the congestion control algorithms based on control theory, the APIRATE controller calculates the control parameters only when there is a dramatic change in the network traffic. To our knowledge, the performance of the APIRATE controller has not been evaluated to control data traffic congestion in vehicular networks.

Another nature of congestion in vehicular networks is due to automobile traffic. There are various causes for vehicular traffic congestion. One of the main causes is the increase in traffic volume above the road capacity. Interestingly, these terminologies exhibit similarities with the terminologies used in the computer communication network. In the computer communication network, one reason for congestion is an increase in demand for bandwidth above the available capacity. Due to congestion the queue starts building up, and if the problem is not addressed the node will have no room for new packets. A similar problem exists in vehicular networks. If we consider the road as a queue and each vehicle as a packet in the queue, then once the road capacity is reached there will be no room for any more vehicles. Congestion in vehicular network arises when the traffic volume increases above the road capacity. This similarity has motivated us to control vehicle traffic congestion using the congestion control algorithm designed for Internet.

In order to use a data traffic congestion control algorithm, we should be able to map each parameter in the algorithm to suitable parameter in vehicular network. Furthermore, the algorithm operation should be suitable for congestion management in vehicular network. The algorithm we are looking for should be able to control the queue length, i.e. the number of vehicles on road, to a target threshold by controlling the number of vehicles that are entering the road. The window-based congestion control algorithms used in the Internet modifies the size of the TCP window at the source to control congestion at intermediate nodes, e.g. RED (Random Early Detection). Whereas the rate-based congestion control algorithm calculates the source sending rate to avoid congestion at intermediate nodes. Window size is derived from the source sending rate. The concept of modifying the TCP window to control data traffic used in window-based congestion control algorithm cannot be reasonably applied to control vehicular traffic, because TCP is a concept for the reliable management of packet re-transmissions, which is not akin to vehicular networks. Therefore, we would like to try a rate-based controller instead, and there are contenders like TFRC (TCP Friendly Rate Control). However, the source-based TFRC makes use of packet loss as a parameter to calculate the source sending rate, and the packet loss notion does not have any significant meaning in vehicular networks. Consequently, we would like to try out the APIRATE controller to control vehicle traffic in VANET. The APIRATE controller employs control theory and AQM (Active Queue Management) in order to deal with congestion. This algorithm does not interpret packet loss as the reason for congestion. Furthermore, the rate information calculated by the algorithm can be used to control the inter-departure time between the vehicles. With certain assumptions, we have successfully mapped the algorithm parameters to suitable parameters in vehicular networks. To our best knowledge, we have not seen any research work where the rate-based controller has been used to control vehicle traffic congestion. So our APIRATE controller is the first attempt.

1.5 Objectives

Our general interest is to address the problem of data traffic and vehicle traffic congestion in VANET. Specifically we are interested in the following:

1. Demonstrate the capabilities of using the APIRATE (Adaptive Proportional Integral rate) controller in static and mobile scenarios of data traffic congestion.
2. Demonstrate the feasibility of using the APIRATE controller to control vehicular traffic congestion.
3. Evaluate and analyze the performance of the APIRATE controller to control data traffic congestion in VANETs.
4. Evaluate and analyze the performance of the APIRATE controller to control vehicle traffic congestion in VANETs.

1.6 Methodologies

We would like to perform congestion management in VANETs (Vehicular Ad-Hoc Networks) in two tasks: data traffic congestion management and vehicle traffic congestion management. To begin with, we need to make changes in the network and transport layers of the communication stack in order to use the APIRATE controller for data traffic congestion management. Data traffic congestion occurs when information arriving at a node demands more bandwidth than available capacity. As the buffer size is finite, an arrival to a buffer which is full will be dropped. In order to overcome this problem, we shall use the APIRATE controller to control the queue length to a target threshold by controlling the source sending rate.

The APIRATE controller has been specifically designed to deal with data traffic congestion. In order to demonstrate the feasibility of using this algorithm for vehicle traffic congestion management, we shall map the algorithm parameters to suitable parameters in vehicular networks. The congestion controller is then implemented as a new application at each node where the operations of the vehicle and the RSU (Roadside Unit) are defined. This application dynamically controls the behavior of vehicle based on the information received from RSUs. We shall use this algorithm to control the number of vehicles on the road segment to a target threshold by controlling the rate at which vehicle enters the road segment.

To demonstrate the congestion management capability of the APIRATE controller in VANET, we would like to study and evaluate its performance in different network scenarios. One of the significant capabilities of the controller is to control the queue length to a target threshold. To achieve this, the queuing and network models are established for analysis and simulation.

For data traffic congestion management, the performance of the APIRATE controller is compared with TCP NEWRENO along with RED queue in both static and mobile scenarios of data traffic congestion. In each scenario, information is transmitted to the destination node over a bottleneck link. When the bottleneck link is saturated, any increase in the demand for bandwidth leads to congestion. For mobile scenarios, we have considered city traffic and rural traffic mobility patterns for our simulations. These mobility patterns are obtained from the traffic modeling software which is discussed in detail later. The behavior of the performance metrics are studied by varying the uncontrolled traffic. We shall study the performance measures like throughput, fairness, window size, queue length and RTT (Round Trip Time). Since throughput is an important measure to provide uninterrupted video streaming service, we shall emphasize the study of average and instantaneous throughput over time for all the scenarios. We would also like to study the behavior of RTT and queue length by varying the uncontrolled traffic.

For vehicle traffic congestion management, the controller performance is evaluated in a scenario where all the vehicles move in one direction in a road segment of interest. The entry of the vehicles to the road segment is controlled by the RSUs installed at the entrance of the road segment. We will be addressing the problem of traffic congestion that is seen due to an increase in the vehicle traffic volume using the road. To introduce congestion, we shall allow all the vehicles into this road segment. Different scenarios will be considered by varying the desired speed of the vehicle and the target queue thresholds accordingly. In order to achieve a realistic mobility pattern, we shall use a single lane car following model. We shall analyze the application and evaluate the performance of a data traffic congestion control algorithm (APIRATE controller) to control vehicular traffic. We shall study the performance measures like queue length (number of vehicles on road segment), throughput (rate at which vehicle enter the road segment) and trip time. Using Microsoft Excel, we shall display our results in the form of graphs.

The performance of the APIRATE algorithm to control data traffic and vehicle traffic cannot be evaluated using mathematical equations alone. For this reason, we shall use simulation. For data traffic congestion management, we would use the Qualnet simulation tool as it provides a complete set of tools for modeling, simulation and analysis. Qualnet provides a set of modules for each layer in the protocol stack. This makes it easy when modifications are needed in different layers (TCP/IP protocol stack). Each layer would then interact with each other and provide the necessary statistics. For vehicle traffic management, we needed a simulation tool that supports vehicular mobility model that can be changed dynamically during the simulation. Furthermore, each node should have the capability to communicate with other nodes in the network (wireless and wired communication). Traffic modeling software like SUMO (Simulation of Urban Mobility) [Sumo08] does not support inter-vehicle communication over the wireless medium. Therefore, we shall use the Qualnet simulation tool for vehicle traffic congestion management. Qualnet being a network modeling software supports node mobility to some extent. A simple dynamic vehicular mobility model is achieved using the API (Application Program Interface) available in Qualnet.

1.7 Thesis Contributions

The following are the contributions of this thesis:

1. Extend the application of the APIRATE data traffic congestion control algorithm for vehicle traffic management.
2. Present the simulation based performance analysis of the controller to control vehicle traffic congestion.
3. Design the APIRATE controller in Qualnet for data traffic congestion management.
4. Present the simulation based performance analysis of the controller to control data traffic congestion.

1.8 Thesis Organization

The rest of the thesis is organized as follows. In Chapter 2, we have reviewed some of the existing congestion control algorithms. The network operations, models and assumptions used in this thesis are discussed in Chapter 3. Analysis of APIRATE controller to control vehicle traffic congestion is presented in Chapter 4. In Chapter 5, we have presented the simulation based performance evaluation of APIRATE controller to control data traffic congestion in vehicular networks. In Chapter 6, we have discussed the design guideline. Finally, in Chapter 7, we present the future work and conclude.

1.9 Publications

1. Brijesh Kadri Mohandas, Ramiro Liscano and Oliver W. W. Yang, "Vehicle Traffic Congestion Management in Vehicular Ad-Hoc networks", under review.
2. Brijesh Kadri Mohandas, Oliver W. W. Yang and Ramiro Liscano, "Data Traffic Congestion Management in Vehicular Ad-Hoc Networks", under review.
3. Brijesh Kadri Mohandas, Ramiro Liscano & Oliver W. W. Yang, "Vehicle traffic congestion management using adaptive PI rate controller", AUTO21 2008 Conference poster presentation, London, ON, June 2 - 4, 2008.

Chapter 2
Literature Review

Our research work involves addressing vehicular traffic congestion and data traffic congestion in VANET. In this section, we will discuss some of the existing congestion management techniques in this area. We will then select a new and efficient technique to deal with the problem of data traffic and vehicle traffic congestion in VANET. Our work is innovative in the field of VANETs.

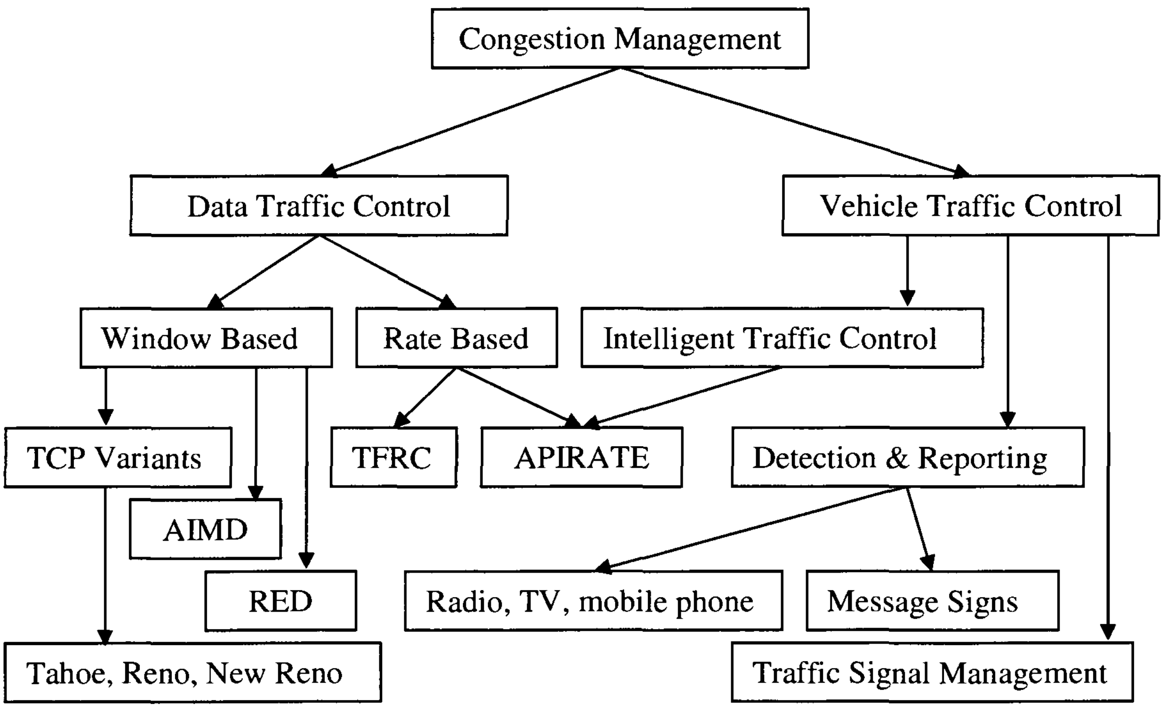


Figure 2.1: Literature classification for congestion management in VANET

The literature classification associated with this thesis is shown in Figure 2.1. The hierarchical classification in the figure gives a brief overview of the techniques involved and proposed to deal with data traffic congestion and vehicle traffic congestion in VANETs.

2.1 Data Traffic Congestion Management in VANET

Communication networks have evolved over the period of time. Wireless networks have attracted a lot of attention because of its pervasive nature and many more significant advantages. It has been observed that TCP (Transmission Control Protocol) [Post81, Stev94, StWr95, Tanen02, Stal03] performs less efficiently in wireless networks compared to wired networks [BaKr97, BaPa97, WeAl04]. TCP as a transport layer protocol provides connection oriented reliable service to the application layer. In a wired network, TCP interprets data loss as the main reason for congestion [JaKa88]. This assumption holds well in reliable wired networks. In wireless networks packet loss may be observed due to link failure, high bit error rate, handover or any other reason [CaIf94, CoAg95, WaZh02, FuLu05]. When packet loss is observed due to such an event, triggering the congestion control algorithm may result in unnecessary reduction in source sending rate. Therefore, considering packet drops as the major cause for congestion is not efficient in wireless networks. In wireless ad-hoc networks, any lower layer problems should be hidden from the TCP sources [YiSh04]. In [ZaSt07], the performance of Emergency Electronic Brake Light with Forwarding (EEBL-F) application has been studied in congested scenarios. The proposed technique manipulates MAC (Medium Access Control) queue to deal with channel congestion in a contention based channel access scheme. In order to address the problem of data traffic congestion in VANET, an example application of wireless networks, we need an algorithm that can avoid congestion considering the factors for congestion seen in wireless networks. Furthermore, we are interested in services like streaming multimedia, where there is a requirement for a steady bandwidth to provide an un-interrupted service. In this section, we will discuss some of the congestion control techniques and choose an efficient algorithm for vehicular networks without sacrificing the quality of service.

Addressing the problem of data traffic congestion is an interesting but challenging research topic. Significant research still needs to be done in this area. The existing congestion control algorithms can be classified into two main types:

1) Window-based congestion control algorithms
2) Rate-based congestion control algorithms.

These algorithms can be further classified based on whether it is located in source node or the intermediate node [MaSe97]. There are also many implementations of TCP, often called TCP variants, designed to address the problem of congestion. We have classified them under the window-based congestion control algorithms. The utility based scheme is classified as a variation under rate-based controller.

2.1.1 Window-based Congestion Control Algorithms

TCP supports end-to-end flow control [Tanen02, Stal03]. During the connection setup, the source node negotiates the source sending rate with the destination node. This is the rate at which the destination can accept information from the source. This is done by negotiating the size of the TCP window [Post81]. TCP creates message segments by packaging the information sent by application layer. These segments are often referred as TCP message segments. During transmission, TCP transmits a fixed number of such segments to the destination node before waiting for a response. The number of segments to be transmitted before waiting for a response is specified by the TCP window. Consequently, the source sending rate can be controlled by controlling the size of TCP window. During the three way handshake, the source and the destination node negotiate their sending rate by specifying the window size. The sending rate negotiated through this process may not be supported by the intermediate node. This may lead to congestion at the intermediate node. This is a major setback of the end-to-end flow control technique.

TCP window plays an important role in order to address the problem of congestion [MoWa00]. TCP interprets packet loss as the major reason for congestion. It relies on AIMD (Additive Increase Multiplicative Decrease) [JaKa88, ZhJi03] technique to address the congestion problem. This technique, increases the window size 1 TCP segment every RTT until loss is detected, in the event of packet loss the window size is halved. When such a technique is used, the TCP window fluctuates resulting in fluctuation of the source sending rate. As a result, such a practice is not efficient for streaming multimedia transmission [Kaha02, LoPa02, HuXu03].

In order to improve the performance of TCP, new features were added to the existing protocol. The resulting protocols were termed TCP variants [JaKa88, AlPa99, FlHe99, Floy03, KuKn06, WeJi06, LiLe07,]. Some of the well known TCP variants are Tahoe, Reno and NewReno. TCP Tahoe [JaKa88, Stev97, HaBe07] includes slow start and congestion avoidance techniques. The purpose of adding these features is to avoid congestion. During slow start, the initial size of the TCP window is set to 2 TCP segments (a maximum of 4 TCP segments). The TCP window is then increased by one segment for every acknowledgement received. As a result the TCP window is doubled for every RTT. Slow start phase is continued until a timeout occurs or a threshold value (*ssthresh*) is reached. In the event of timeout, the *ssthresh* value is reduced to half the amount of outstanding data and the TCP window is set to one TCP segment. This is followed by another slow start phase. If *ssthresh* is reached, slow start phase ends and congestion avoidance phase is entered. In this phase, the congestion window is increased by one segment every RTT.

TCP Reno [AlPa99, HaMu99] was developed by adding two additional features to TCP Tahoe, fast retransmit and fast recovery. This technique, at times, detects packet loss and performs error recovery before the transmission timer expires. When a packet arrives out of order at the destination, it transmits an acknowledgement to the last segment received in sequence. As the segment is already acknowledged earlier, any further acknowledgement sent will be duplicates. Once the source receives three such duplicate acknowledgements, it concludes that unacknowledged segments have been lost. It then transmits the unacknowledged segments without waiting for the retransmission timer to timeout. The *ssthresh* is reduced to half the amount of outstanding data, as after timeout. This is followed by fast recovery until all lost segments are recovered. The TCP window is set to three segments more than *ssthresh.* The fast recovery phase ends when an acknowledgement is received for the new data sent. In order to avoid slow start, the TCP window is now set to same values as *ssthresh.* TCP New Reno [FlHe99, KiCh00, PaMa06] is an improved version of TCP Reno. TCP New Reno provides better thresholds for fast recovery and improves retransmissions. TCP New Reno is the most commonly implemented algorithm. For all the TCP variants discussed so far, the TCP window size exhibits a saw-tooth behavior resulting in source sending rate fluctuations. Furthermore, TCP along with its variants assumes packet loss as the main reason for congestion. As discussed earlier, this assumption is not appropriate for the wireless networks.

In order to improve the performance of congestion management, the source-based techniques discussed so far often co-exist with router based techniques. AQM (Active Queue Management) is a router based congestion avoidance mechanism to improve the performance of TCP congestion control. A well know AQM example is RED (Random Early Detection) [FlJa93]. Traditional tail drop techniques buffer packets until the buffer is full. Once full, the node drops any new packets leading to congestion. As a result, there is an unfair distribution of buffer among the traffic flows. Buffering may also lead to an increase in RTT, thus degrading the quality of service. RED addresses these problems by dropping packets with a probability and maintaining the queue size to an average between the configured minimum and maximum thresholds. When the queue is empty, all incoming packets that cannot be serviced immediately will be queued. As the queue grows, RED will start dropping packets with a probability that increases with the queue. It has been observed that the throughput of RED is sensitive to traffic load and decreases when the queue size is greater than the configured maximum threshold [MaBo99, FlGu01]. Furthermore, RED does not perform well in wireless ad-hoc networks [XuGe03]. NRED (Neighborhood Random Early Detection) was proposed in [XuGe03] to deal with the problems of RED in wireless ad-hoc networks. NRED exhibits source sending rate fluctuations which is not suitable for streaming multimedia transmission.

AQM techniques like BLUE [FeSh02], FRED [LiMo97], SRED [OtLa99] and CHOKe [PaPr99] controls congestion by adjusting the packet drop probability. REM (Random Exponential Marking) [AtLi01] is different from RED by using a different congestion measure and marking probability. Each queue maintains a parameter called price, which is used to determine the marking probability. REM stabilizes the rate and the queue length around a target threshold independent of number of users sharing the link. AVQ (Adaptive Virtual Queue) [KuSr04] makes use of token bucket algorithm to provide AQM. Most of these protocols are based on heuristics without strong theoretical backing. RED, BLUE and other AQM mechanisms are judged to an on-off control strategy which often leads to oscillations and exhibits complex behavior [AsGo95]. Furthermore, in many cases window size exhibits a saw-tooth behavior which leads to sending rate fluctuations.

2.1.2 Rate-based Congestion Control Algorithms

Rate-based congestion controllers adjust the source transmission rate by controlling the size of the TCP window at the source. Source transmission rate is directly proportional to the TCP window and inversely proportional to RTT. As a result, the size of TCP window can be derived from the source transmission rate. Rate-based controllers are well suitable for streaming multimedia applications [HuXu03]. In this section, we will discuss some of the well known rate-based congestion control algorithms and choose an efficient algorithm to perform data traffic congestion management in VANET.

TFRC [FlKa00, HaFl03] is a well known example for a rate-based controller. As in the traditional AIMD, TFRC does not halve the sending rate in response to congestion notification. This makes TFRC a candidate protocol for streaming multimedia transmission in wired network. TFRC resides at the source and provides an equation-based congestion control mechanism for unicast traffic. The proposed equation is a function of packet loss observed during the transmission. As discussed earlier, considering packet loss as a reason for congestion is not reasonable for wireless networks.

Rate-based controllers also use AQM at the intermediate nodes to participate in congestion management [GeLo02, KaHa02, HoYa05]. The algorithm residing at the intermediate node calculates the source sending rate based on congestion and converts the rate to TCP window in order to manage traffic. The TCP rate control [KaKa00] improves the TCP performance by controlling the source sending rate. The sending rate of the TCP source is derived from window size, RTT and the rate of acknowledgements. TCP rate control maintains states per TCP flow and provides protection against misbehaving TCP sources. In [AwOu02], the window size is calculated as a function of input-output rate mismatch in order to match the servicing capability or bandwidth of a node. In another approach, the network layer conveys available bandwidth and propagation delay measurements to TCP sources, which computes the source window using bandwidth delay product [GeWe00]. EXACT (Explicit rate-based flow control) [ChNa05] presents an end-to-end flow control scheme where the intermediate node conveys the source rate to the end host in the control header of each packet. It relies on the MAC and the transport layers to address the problem of congestion. The source sending rate is calculated based on current measured bandwidth of the outgoing links and the current rate of information flow going through the router. Most of the techniques discussed so far are based on heuristics without theoretical backing of their performance claim.

The utility-based control is a variation of rate-based control method. The source node adjusts its transmission rate based on feedback network congestion information and utility function. UBPFCC (Utility based packet forwarding and congestion control) [WiRo05] assumes broadcast to be the dominant form of communication in VANET and addresses congestion caused by broadcast applications. This is a significant setback of this approach as several point-to-point application services can also co-exist in VANET and either of these applications can be the reason for congestion. The algorithm utilizes a utility function and encodes the utility information in each data packet. The average utility value is then calculated for each individual node and based on this the available data rate is shared.

A congestion control algorithm should be able to address the problem of congestion without sacrificing the Quality of Service. The APIRATE (Adaptive Proportional Integral Rate) Controller [HoYa07] is one such rate-based controller that supports best-effort traffic in the Internet. It employs control theory and AQM in order to deal with congestion. The proposed controller calculates the source sending rate based on instantaneous queue length of the intermediate routers and advertises it to the sources using the acknowledgement message. The adaptive feature of the controller allows it to self-tune only when there is a dramatic change in the network traffic. This is achieved using a simple heuristics monitoring parameter. The APIRATE controller does not interpret packet loss as the reason for congestion. Furthermore, it is capable of fairly distributing the available bandwidth among all the users without scarifying the quality of service. The controller is capable of providing a steady throughput at the source which is well suitable for multimedia streaming.

2.2 Vehicle Traffic Congestion Management in VANET

Vehicle traffic congestion is a condition where driver's experience trip time delays due to slow moving traffic or a complete traffic jam. The main reason for congestion is an increase in traffic volume above the road capacity. Traffic incidents, weather conditions, work zones, poorly timed traffic signals can also lead to traffic congestion. Traffic congestion has a number of negative effects influencing driver behavior. The vehicle traffic congestion control mechanisms used should be sensitive, dynamic and adaptive in order to avoid affecting driver's behavior. There are various techniques proposed to deal with vehicle traffic congestion. Some of these techniques are designed for specific scenarios that cause congestion. In this section we shall discuss some of these techniques and then propose an innovative approach to deal with vehicle traffic congestion.

2.2.1 Traffic Signal Management

Traffic lights provide assistance for drivers and pedestrians when using the road network in order to avoid collision and other hazards. A well timed traffic signal can also be used to control the flow of vehicles through a given road network. It has been observed [KaHa94] that traffic signal timing can be a cause for traffic congestion. When the traffic signal timing was reviewed considering cycle length, green signal time and reference offset, traffic congestion was alleviated without sacrificing traffic demand. The traffic signal controller proposed in [SmCh95] operates in real time and adjusts signal timing for multiple signals in a traffic network as a response to short term conditions such as congestion, accidents, construction blockages, weather etc. It has also the ability to automatically adjust to long term system changes without remodeling. The approach is based on a neural network approach where internal neural network weights are estimated in real-time via a simultaneous perturbation stochastic approximation algorithm. Traffic signals can improve the traffic conditions on urban freeway systems by controlling the rate at which vehicles enter the roadway [PaKo02].

Wireless sensors have also been used to control traffic lights [TuQi08]. Traffic behavior was studied using one and two sensors. The one-sensor scheme detects only the vehicles that pass through it and the two-sensor scheme detects vehicles that pass through it and the vehicles that stop near the sensor. It was observed that two-sensor scheme performed better than the one-sensor scheme and was able to calculate the exact number of waiting vehicles. The performance of sensor-based traffic light controllers was compared with one of the existing best first traffic light controllers. A reduction in average trip waiting time was observed using the sensor based traffic light controllers.

2.2.2 Traffic Congestion Detection and Reporting

Traffic reporting is a universal approach to deal with congestion. The volume of vehicle traffic is monitored using fixed cameras or traffic reporters, and delivered to drivers using radio, television, mobile phone or using message signs. In some major highways, drivers even have the option to tune the radio to a particular frequency in order to receive information about vehicle traffic. In order to provide accurate information associated with traffic congestion, it is necessary to follow an efficient technique in order to detect vehicle traffic congestion.

The automated incident detection scheme proposed in [KaKo05] makes use of the data sent by surveillance cameras to detect slow moving vehicles, congestion and traffic incidents. This technique makes use of the Spatio-Temporal Markov Random Field algorithm for tracking along with incident detection algorithms. This system has proven to be effective for traffic monitoring with few false alarms. Traffic management systems have also relied on information sent by satellite to interpret congestion [KiLe97]. Using satellite remote sensing imagery, the information can be collected easily and widely, hence addressing previous limitations of accuracy and spatial coverage range.

The information associated with vehicle traffic congestion can also be obtained from the telecommunication network using cell dwell time [ThPa06]. Cell dwell time is the duration for which a mobile phone remains registered to a base station until it switches to another base station. Based on measurement done near freeways, it was observed that large cell dwell times were usually associated with incidents that interrupted normal road traffic flow. The use of neural networks along with cell dwell time [PaPe07] showed a better performance associated with congestion estimation.

A knowledge-based traffic management system called TERMINUS [GiEl94] was proposed in order to respond to real time traffic problems. The main idea is to have a blackboard architecture using distributed knowledge sources to manage traffic. Each knowledge source provides information related to an area of interest like traffic congestion, traffic accidents or traffic signal optimization. The traffic engineering expert, associated with each knowledge source makes use of the information in the blackboard to solve their problem or alter their solutions with regard to the new requirement. An evaluation study [Rowe91] shows a significant improvement in traffic management using ATSAC (Automated Traffic Surveillance and Control system). ATSAC employs both traffic monitoring and signal time management in order to address the problem of congestion.

Variable messages signs, as a part of route guidance system, play an important role in order to convey useful information to drivers. In [BrVl91], the authors have studied the influence of variable message signs on driving behavior and route choice. In case of traffic incidents it was observed variable message signs played a significant role in diverting vehicle traffic. Variable message sings, as a dynamic information provider, assists drivers in selecting a suitable route in the event of congestion [HeGu06].

2.2.3 Intelligent Traffic Control

VANETs have revolutionized the automobile industry making it possible to use different techniques to deal with some of the existing problems. The communication capability of each vehicle creates a virtual medium to transmit information over VANET in an ad-hoc fashion. This allows drivers to have up-to-date information about events that effect driver behavior. One such event is traffic congestion. In this section we will discuss some of the techniques proposed to address the problem of congestion using the intelligence of VANETs.

The Proactive traffic merging strategy [WaKu07] addresses the problem of congestion seen in intersections where a ramp leads on to a highway. It is assumed that vehicles are equipped with sensors and can communicate their velocity and acceleration with other vehicles in the network. Using the information sent by other vehicles, each vehicle decides where and when it can merge onto the highway before arriving at the merging point. The Proactive traffic merging algorithm performed better than traditional priority-based merging algorithms in terms of increased traffic throughput. The Street Smart Scheme [DoJo07] proposes a new technique to collect real-time congestion information. Each vehicle builds a speed map based on the speed of other vehicles in vicinity and transmits it to the neighboring vehicles. As a result, each vehicle is able to build a map of the expected speed on every road. The multi-agent based traffic management system [BaSa07] makes use of an evolutionary technique to manage urban traffic. Using this strategy, the total mean delay in a six junction network was reduced. Evolutionary algorithm was found to be a good candidate for road traffic management and congestion avoidance.

A routing scheme called BeejamA [WeLe07] derived from multi-agent Bee Hive routing algorithm was proposed to control vehicle traffic. In this scheme vehicles are directed under decentralized control at each road intersection. The route decision method in [InSh07] follows two important procedure, traffic information sharing procedure and route selection procedure. Traffic information is shared dynamically over VANETs. Based on this information, a congestion free route is selected using the route decision method. The route decision method is performed repeatedly in order to avoid traffic jam in selected route. Congestion can be interpreted by monitoring the velocity of the moving vehicle. Based on vehicle velocity, congestion is classified into three different thresholds [AtPo06]: free flow, slow moving or heavily congested. Vehicle velocity is monitored using GPS (Global Positioning System) or video cameras and based on the received information a threshold is selected and reported to public.

2.3 Vehicle traffic simulator

Vehicle traffic simulators like SUMO (Simulation of Urban Mobility) [Sumo08] provides vehicular mobility models suitable for vehicle traffic research community. SUMO is an open source, microscopic road traffic simulation package developed by employees of the Centre for Applied Informatics (ZAIK) and the Institute of Transport Research at the German Aerospace Centre. SUMO provides a common platform to study vehicle behavior, traffic light optimization, routing etc. A vehicle or node in SUMO lacks the ability to communicate as wired or wireless communication features are not embedded in these tools. This is a significant drawback of these tools when considered for simulations where vehicles need to communicate with each other.

2.4 Routing in VANET

Routing is an important aspect for communication. It is a process of selecting the network route towards the destination node during or before transmitting the information. This process builds the routing table which will be used during the transmission. Routing in VANET has attracted a lot of interest. The performance of some of the existing MANET routing protocols has been evaluated in VANET. It has been observed that AODV (Ad-hoc On Demand Distance Vector) protocol dominates some of the existing MANET protocols when used in VANET [JaBe05, SaMa05].

2.5 Concluding Remarks

In this chapter, we have presented a literature classification showing a brief overview of different congestion management techniques. We have analyzed some of the prominent techniques to deal with the problem of data traffic congestion and vehicle traffic congestion in VANET. Based on the quality of service requirements of VANET, we have chosen the APIRATE controller as the candidate algorithm to perform congestion management in VANET. In the subsequent chapters, we will discuss our approach and then present our results.

Chapter 3
Network Operations, Models and Assumptions

This chapter provides information on the essential operations of different network elements in VANET. The network models for vehicle traffic and data traffic congestion in VANET are presented. Simulation models along with traffic and vehicular characteristics are specified for the simulation platform to facilitate the performance evaluation of the APIRATE controller. We have also described the simulation scenarios and defined the performance measures used in the performance evaluation of the controller.

3.1 Network Operations

A VANET usually consists of a number of vehicles equipped with DSRC capable wireless device that supports inter-vehicle and vehicle-to-roadside communication. The VANET also has a number of RSUs (Roadside Units) that are connected to each other over the high speed links. They provide external interfaces to the vehicle to deliver various application services for the subscribers (drivers or passengers). These application services utilize the TCP/IP stack for communication. To provide a reliable communication service, TCP is used as the Transport Layer protocol. In order to facilitate routing, IP (Internet Protocol) at the network layer is supported by the AODV (Ad-hoc On Demand Distance Vector) routing protocol as it dominates some of the existing MANET routing protocols when used in VANET. AODV is used to determine the route to destination vehicle or RSU. The 802.11 MAC is used at the MAC layer. In our research, the RSUs in the network are configured to deal with vehicle traffic congestion or data traffic congestion in VANET.

During vehicle traffic congestion, the congestion control algorithm implemented as an application in the RSU (destination) calculates the desired rate and transmits it to the prospective RSU (source). The prospective RSU controls the rate at which vehicles enters the road segment based on the newly calculated rate obtained from the APIRATE algorithm. While dealing with data traffic congestion, the APIRATE algorithm is implemented in the Network Layer of the congested RSU. Based on the instantaneous queue length, the APIRATE algorithm calculates the window size and advertises it to the source. The newly calculated TCP window is upper-bounded by the window size negotiated during connection setup. The APIRATE algorithm performs AQM by controlling the queue length to a pre-configured target threshold. Implementation of the controller along with behaviors of each network component is discussed in later sections.

3.2 Network Model

We consider a VANET system with many vehicles distributed across a road network equipped with RSUs. The network models as seen by the controller at each RSU are used to study the problem and to assess the competence of the proposed solution. They are presented below. Note that the parameter N used in these network models in general represents the number of flows. It adopts a suitable meaning depending on the network model in which it is used.

3.2.1 Congestion Control Model for Vehicle Traffic

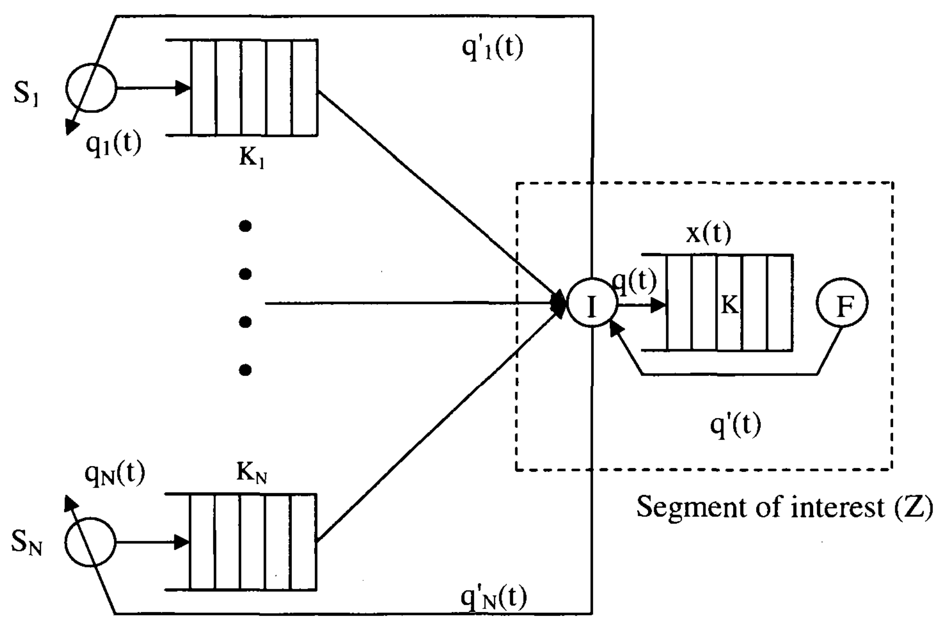


Figure 3.1: Network model depicting vehicle traffic congestion in VANET

Figure 3.1 models vehicle traffic congestion seen in a segment of road (Z) that is part of a complex road network. The figure reflects the adoption of the rate controller (data traffic congestion controller) for vehicle traffic control. We have considered the road as a queue (buffer) and each vehicle as a packet in the queue. Our system has three major components, source RSU (*I*) installed at the entrance of the road segment, destination RSU (*F*) installed at the exit of the road segment and the road segment (Z) itself which is shown as a queue. The parameter $x(t)$ represents the instantaneous queue length i.e. number of vehicles using the road at any given time. The target queue size K is the maximum number of vehicles that can use the road without influencing the behavior of other vehicles and without causing congestion. The RSU controls the rate at which vehicles enter the road segment by controlling the inter-departure time for vehicles at the entrance. In this system model, we have shown vehicles from different areas using a common segment of the road to reach the destination. The parameter N in this model represents the number of vehicle traffic flows. The notations S_1, $S_2 ... S_N$ represents the source RSUs installed in different areas that control the number of vehicles that enter the road segments connected to segment of interest. The segment of interest can be mapped to a highway or a cross-section of a highway. K_1, $K_2 ... K_N$ represents target queue length of the road segments connected to the segment of interest. If the vehicle traffic volume using the road segment of interest in greater than K then such a condition will lead to vehicle traffic congestion in the segment of interest.

We have used the APIRATE controller in order to deal with the problem of vehicle traffic congestion in VANET for the reasons discussed in Section 1.4. The controller is installed in destination RSU F. The APIRATE controller controls the queue length in the segment of interest to a target threshold (K) by controlling the rate at which the vehicles enter the segment of interest. The destination RSU transmits the newly calculated rate to the source RSU (I). The initial rate $q(t)$ at the source RSU is then changed to the newly calculated rate $q'(t)$. The rate information is used to control the inter-departure time at the entrance of the road segment of interest. The source RSU (I) that services traffic from S_1, $S_2 \, . \, . \, S_N$, transmits the newly calculated rate to all the different sources so that all the sources can control the volume of traffic entering the road segment of interest. The rate information can also be transmitted to prospective drivers in order to assist them while making route decisions. In a complex network, the APIRATE algorithm can be configured in many RSUs and based on the rate calculated by the algorithm the queue length in one or more segments of interest can be controlled to a target threshold.

3.2.2 Congestion Control Model for Data Traffic

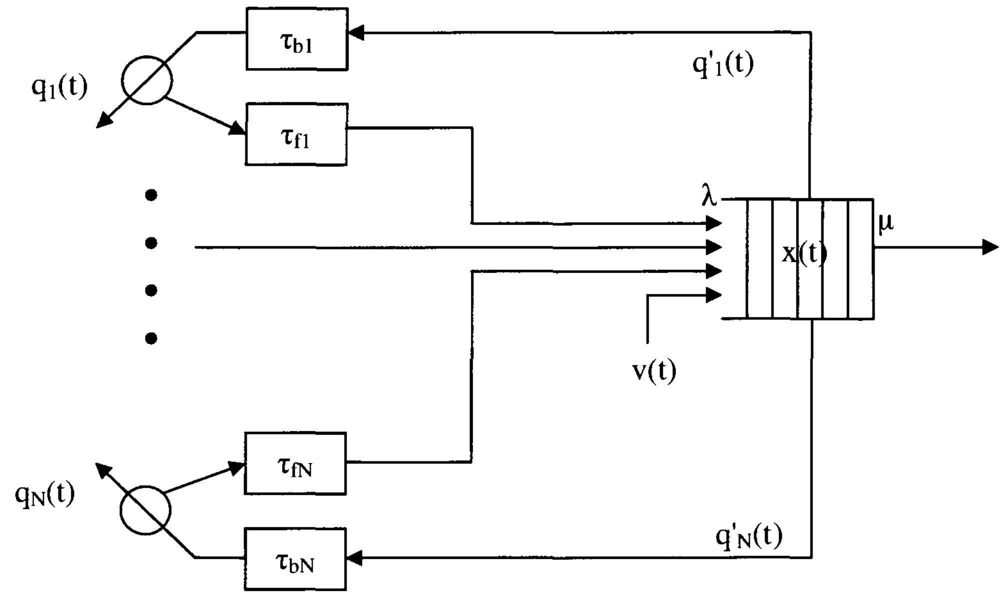


Figure 3.2: Network model depicting data traffic congestion in VANET

As most of the multimedia services are provided by RSUs, one of the candidate nodes to experience congestion in VANET is the RSU. Figure 3.2 models data traffic congestion in one of the RSU. In this network model, N represents the number of TCP flows to an RSU capable of servicing information at μ data packets per second. There are two kinds of delays experienced during transmission: 1) Forward time delay (τ_{fi}) which is the delay between the source node and the intermediate node 2) Feedback time delay (τ_{bi}) which is the delay from intermediate node to destination node and from destination node to the source node. The roundtrip time is $\tau_i = \tau_{fi} + \tau_{bi}$. The aggregate arrival rate at the RSU is λ data packets per second. We also consider an uncontrolled traffic with an input rate $v(t)$ into the congested node. When the demand for bandwidth is greater than μ, the RSU starts queuing packets and services it at later stage in FIFO (First In First Out) manner. As the queue size is finite, once the queue is full, the RSU starts dropping packets. Over the period of time, if this problem is not addressed, there will be severe reduction in quality of service.

The APIRATE controller attends this problem by controlling the source sending rate ($q_i(t)$) based on instantaneous queue length ($x(t)$) of the congested node. The newly calculated source rate ($q'_i(t)$) is advertised as window size to the source using TCP ACK (Acknowledgement) message. The APIRATE controller employs classical control theory and adaptive control in order to perform AQM (Active Queue Management). The adaptive nature of the algorithm monitors the number of flows (*N*) and the RTT in order to self-tune the controller only when there is dramatic change in network traffic.

3.2.3 Assumptions

Following are the common assumptions for congestion management in VANETs:

1. Enough RSUs are deployed across the road network: These units play an important role to deliver the requested service to the vehicles and also to control the traffic light or gates in order to perform vehicle traffic management. We need an adequate number of them.
2. Vehicles are equipped with 802.11a radio instead of DSRC: Qualnet does not provide the implementation of DSRC. Since DSRC has been inherited from IEEE 802.11a standard, our simulation shall implement 802.11a radio instead.
3. Vehicles are equipped with necessary software and hardware: Necessary software and hardware are installed in the vehicle to facilitate communication and to use other services. A vehicle may also be equipped with onboard entertainment devices to use services like video streaming in real-time.
4. The electric energy supply for the vehicle is unlimited: Nodes in MANET have finite supply of energy, whereas vehicles have unlimited supply of electric energy from the rechargeable battery installed in the vehicle.

3.3 Simulation Model

Congestion management concerns with the handling of different facets of congestion in VANETs. This section presents the simulation model of vehicle traffic and data traffic congestion management used in our research. More details about the functionalities of the adaptive PI (Proportional Integral) rate controller are presented in Appendix A.

3.3.1 Vehicle Traffic Management Model

In this section, we briefly discuss the implementation of the controller to control vehicular traffic. Functionality of RSUs and the vehicles are discussed in detail. In order to control the vehicular traffic, it is necessary to dynamically control the vehicular mobility. A detailed description of how dynamic mobility is achieved in Qualnet is presented. This section also includes the assumptions made while using the APIRATE controller for vehicle traffic management in VANET.

3.3.1.1 Implementation in Qualnet

For data traffic congestion management, the APIRATE (Adaptive Proportional Integral rate) controller is implemented in the network layer to perform active queue management by controlling the source sending rate. In order to use the controller to control vehicular traffic, we have added a new application at each node. Using the application, a node can be configured as a vehicle or as a RSU. As mentioned in Section 3.2.1, RSUs are installed at the entrance and the exit of a road segment. These intelligent devices control the rate at which vehicles enter and leave the road segment by controlling the gates or by timing traffic signals. The RSU installed at the exit (destination RSU) of the road segment runs the APIRATE control algorithm and calculates the rate at which vehicles can enter the road segment without causing congestion by controlling the queue length (i.e. number of vehicles on the road segment) to a target threshold. The target threshold is determined by thorough analysis of the traffic volume. It is the maximum number of vehicles that can use the road segment without influencing the behavior of other vehicles and without causing congestion. The new rate information is transmitted to the RSU installed at the entrance (source RSU) over the backbone network. The source RSU makes use of this information to control the number of vehicles entering the segment of interest. In a complex network, the source RSU can transmit the new rate information to other sources in the network sharing the segment of interest or to prospective drivers interested in using the road segment of interest. The two important parameters associated with the APIRATE algorithm are the number of flows and the RTT. In a complex road network, vehicles from different areas often use a common road segment to reach the desired destination. Each such area represents a flow or a source. The number of flows can be calculated based on the source and destination RSU identifiers. The bottleneck road segment will be equally shared among all the sources. Due to limitations of Qualnet with respect to mobility, we have considered only one flow in this thesis. The RTT parameter used in data network is mapped to trip time in vehicular network. Trip time is the time taken to travel from one point to the other (from source to destination RSU). In vehicular network, queue length is the number of vehicles using the road segment at any given instant of time. The source and the destination RSU, together calculate the queue length. In the following section, we have discussed the step-by-step algorithm for the vehicular traffic management.

3.3.1.2 Vehicle Behavior

One main task of a vehicle is to provide a smooth and comfortable ride for passengers. A vehicle exhibits similarities to a data packet in communication network. In the Internet, data packets carry information and travel from one point to another. While doing so, it has to combat various problems in the backbone network. Successful transmission of a data packet depends mainly on the network and the transmission medium. In the case of a vehicle, successful transmission from one point to another depends on the road network and the driver who is the main controller of the vehicle.

Vehicle Behavior:

Near source roadside unit:

a. Waits for the signal / message from the source roadside unit.
b. On receiving the message, records the time and travels towards the destination at a desired speed

Near destination roadside unit:

On arriving near the destination roadside unit, calculates the trip time and transmits it to the destination roadside unit.

Figure 3.3: Vehicle behaviors while using PI Rate control

Figure 3.3 describes the vehicle behavior while using the APIRATE control algorithm. Vehicles enter the road segment from the entrance near source RSU and leave from the exit near destination RSU. While using the controller to control vehicular traffic, we consider a pool of vehicles waiting to use a road segment. This is a virtual poll created mainly to provide a continuous flow of traffic into the road segment. In practice, the APIRATE controller can calculate the tolerable rate and broadcast to other RSUs or prospective vehicles. Based on this, a driver can decide if they want to use the road segment. The RSU that receives the rate can use this information to calculate the inter-departure delay and control the flow of vehicles into the segment of interest by controlling traffic lights or gates. In the current implementation, a vehicle in the front of the pool waits for a message from the RSU before using the road segment. Once the vehicle receives the message, it travels to the destination at a desired speed. Based on the rate, the source RSU informs the vehicle in front of the pool to use the road segment. Initially, the source RSU will make use of the preconfigured rate. As the APIRATE algorithm starts the congestion avoidance procedure, the source RSU will utilize the new rate to control the number of vehicles entering the road segment.

A vehicle plays an important role while calculating the trip time. While starting from the source RSU, a vehicle makes note of the time and on reaching the destination it calculates the trip time and transmits it to the destination RSU. The destination RSU will make use of this information in the APIRATE algorithm in order to calculate the rate at which vehicles can enter the road segment. A vehicle moves along the road network according to a vehicular mobility model. In Qualnet, the number of nodes/vehicles that can be used in the simulation is finite. In order to have a continuous flow of vehicles at the entrance we have reused the vehicle once it has crossed the exit. In order to do so, once the vehicle has crossed the exit, it is moved back to the location near the entrance and is added to the virtual pool of vehicles at the entrance.

3.3.1.3 Mobility Characteristics

The mobility characteristic of a vehicle depends on various factors like driver behavior, traffic lights, weather conditions, gap between two vehicles etc. Based on these factors drivers may accelerate, decelerate or cruise while driving towards their desired destination. Predetermined mobility characteristics provided by the traffic simulation software to be used in the network modeling software is not suitable for our research. In such a scenario, the mobility pattern of the vehicle is listed before the simulation and remains unchanged during the simulation. Such a mobility pattern is suitable for simulations where the mobility pattern does not change during simulation based on the events seen during the simulation.

For vehicular traffic congestion management, we need to dynamically control the vehicle movement based on congestion event or as directed by events seen during simulation. Furthermore, vehicles should be able to communicate with RSUs and also the RSU should be able to communicate with each other over the back bone networks. For this reason, we could not completely rely on vehicle traffic modeling software like SUMO as wired or wireless communication features are not embedded in these tools. Qualnet network modeling software supports mobility to some extent. To achieve dynamic mobility in Qualnet, we have used an API provided by Qualnet. This API (MOBILITY_InsertANewEvent) allows us to dynamically input the node positions during the simulation. In this thesis, we have considered a mobility pattern where vehicles move along increasing x-axis, i.e. all vehicles move in one direction as in a highway scenario. In such a scenario, future node positions are obtained by increasing one axis (x-axis) and keeping the other two axis constant (y & z axis). The application installed in each vehicle, makes use of this API to move the vehicle along the road network.

Another important aspect of vehicular mobility is the car following model. While dynamically controlling the vehicular behavior it is necessary to avoid inter-vehicle collisions. In a single lane scenario where a vehicle follows a leading vehicle, the speed of the follower depends on the leader behavior. The driver selects a basic maneuver to cruise, accelerate or decelerate based on the behavior of the leading vehicle. All these maneuvers are carried out safely without collisions by using a car following model. In this thesis, we have used the "Single-Lane Traffic Simulation with Multi-Agent System", [ZhLi05], car-following model. It is an agent based single-lane traffic simulation model which presents a decision tree for longitudinal movement and simplified formulas for acceleration and deceleration. More details about the car-following model are presented in Appendix B.

Source roadside unit behavior:

a. In order to allow a vehicle to use the road segment the source roadside unit transmits a go-ahead message to the vehicle in the front of the pool.
b. Initially allows all the vehicles to use the road segment at a highest possible rate.
c. On receiving the rate from the destination road side unit, tunes the inter-departure time according to the newly received rate.
d. Updates the queue length when a vehicle enters the road segment.

Figure 3.4: Source Roadside unit behaviors while using PI Rate control

3.3.1.4 Roadside Unit Behavior

Figure 3.4 describes the source RSU behavior while using the APIRATE control algorithm. One of the main causes for congestion is increase in traffic volume above the road capacity. We have used the APIRATE controller to control the queue length, i.e. number of vehicles on the road, to a target threshold by controlling the in-flow of vehicles to the road segment of interest. The RSU installed at the exit of the road segment calculates the desirable rate using the APIRATE control algorithm. It then transmits the newly calculated rate to the RSU installed at the source. The source RSU, on receiving the advertised rate, controls the inflow of vehicles into the road segment by controlling the inter-departure time based on the rate.

Destination roadside unit behavior:

a. Updates the queue length when a vehicle leaves the road segment at the exit.
b. Calculates the average trip time based on the trip time information received from the vehicle.
c. Calculates the source rate based on instantaneous queue length and algorithm specific parameters.
d. Transmits the newly calculated rate to the source roadside unit over the back bone network.

Figure 3.5: Destination Roadside unit behaviors while using PI Rate control

Figure 3.5 describes the destination RSU behavior while using the APIRATE control algorithm. The APIRATE controller was initially designed to control data traffic congestion in the Internet. In order to use the APIRATE controller for vehicle traffic management, we have mapped the algorithm parameters to suitable parameters in vehicular network. Two important parameters are number of flows and RTT. The round trip time factor in a data communication network is mapped to trip time in vehicular network. Section 3.3.1.1 describes how number of flows is mapped to a suitable parameter in vehicular network.

Adaptive PI rate algorithm for vehicle traffic management:

a. Calculate the monitoring parameter based on number of flows and average round trip time
b. If control parameters, K_P and K_I has not been calculated before, calculate K_P and K_I
c. else if the monitoring parameter is outside the prescribed limits
   Recalculate K_P and K_I
d. Calculate the source sending rate based on K_P, K_I, instantaneous queue length and target buffer.
e. Update the rate with newly calculated rate and use it for further transmissions to the source RSUs.

Figure 3.6: Adaptive PI Rate algorithm operations for vehicle traffic management

Figure 3.6 describes the APIRATE[*] algorithm used for vehicle traffic management in VANET. We assume that RSUs are capable of communicating with each other over high speed backbone network. The RSU does also have a wireless interface through which it communicates with other vehicles in the network. Although the distribution of RSU has not been studied, it can be assumed that RSUs are deployed at natural control points like traffic junctions, highways, etc. In this thesis, we have utilized the RSUs installed at the entrance and exit (traffic junctions) of the road segment to facilitate vehicle traffic control.

In a complex road network, vehicles from many areas tend to use a common road segment to travel to a desired destination. The RSUs or traffic lights installed in these areas control inflow of vehicles to the shared road segment. If the APIRATE controller is used to avoid congestion, rate information calculated by the destination RSU installed at the shared road segment can be transmitted to all the sources. In such a case, the common road segment will be equally shared among all the sources.

3.3.1.5 Assumptions

Following assumptions are associated with vehicle traffic congestion management in VANETs:

1. Vehicles can communicate with RSUs: All the vehicles can communicate with both source and destination RSU installed in the entrance and exit of the road segment to facilitate congestion management.
2. The road is a buffer and each vehicle is a packet in the buffer: This mapping allows us to use the data traffic congestion control algorithm to manage vehicle traffic. We have mapped the algorithm parameters to suitable parameters in vehicular network.
3. Queue length is the number of vehicles on the road: The queue length parameter used in the algorithm is mapped to number of vehicles on the road to facilitate the algorithm operation.
4. Target queue threshold for vehicle traffic management: The target queue threshold used by the APIRATE controller is mapped to the maximum number of vehicles that can use the road segment without influencing the behavior of other vehicles and without causing congestion.

[*] A detailed description of the APIRATE controller is presented in Appendix A.

5. RTT (Round Trip Time) parameter for vehicle traffic management: The RTT factor used in the APIRATE controller is equivalent to trip time in vehicular network. This is the time taken to travel from one point of the road network to another.

3.3.2 Data Traffic Management Model

This section briefly discusses the implementation of APIRATE controller as well as the behavior of each node in the system. The distinctive feature of VANETs is the mobility pattern of vehicles. Since Qualnet does not support vehicular mobility patterns, a detailed description of how vehicular mobility can be achieved in Qualnet is presented.

3.3.2.1 Implementation in Qualnet

The APIRATE algorithm is implemented in the network layer of each node. While servicing a packet in the network layer (transmitting a packet to the next hop), the APIRATE algorithm calculates the rate at which the intermediate nodes can process data while maintaining the queue length to a target threshold. The APIRATE congestion control operation is discussed briefly in Section 3.1. The steps associated with the controller are executed only while processing the TCP messages.

The source node initially transmits at highest possible rate. In order to facilitate the congestion avoidance procedure carried by the APIRATE algorithm, two new fields are added in the header of each packet, RTT and window size. The intermediate node, calculates the source sending rate based on instantaneous queue length of the buffer. The algorithm employs control theory to calculate the rate and makes use of two important parameters, number of active flows and the average RTT. The number of flows is calculated at the intermediate node by monitoring the source and destination address of each IP packet. The average roundtrip time is calculated using the RTT in the header of each packet. The newly calculated rate is then converted into window size and updated in the header of each packet. Normally, an IP packet travels through multiple intermediate nodes before reaching the destination. Each intermediate node compares the newly calculated window size with the window size in the header of the packet and then records the smaller value in the header. When this packet arrives at the destination, the destination node extracts the window size information from the header and transmits it to the source in the next ACK message. On receiving the ACK message, the source node tunes the sending rate based on the newly calculated window size. To achieve self-tuning, the algorithm monitors two parameters, number of flows and average RTT. When these parameters change beyond the monitoring interval, the PI rate controller will self-tune by recalculating the control parameters.

RTT samples provided by Qualnet 4.0 simulation tool are in 500 millisecond units. In a practical scenario, RTT can be less than 500 milliseconds. In order to measure RTT that is less than 500 milliseconds, we have made some changes in the transport layer. At the transport layer, we use and modify TCP New Reno, a TCP variant, for end-to-end communication. We have added a timestamp field in the TCP header of each packet to facilitate RTT calculation. Initially the delay from source node to the destination node is determined. While transmitting the ACK message, the destination node updates the timestamp field considering the delay from the source node to the destination node. When the source node receives the ACK message, it calculates the RTT based on the timestamp field. The newly calculated RTT is then used to update the RTT field in each packet.

3.3.2.2 Source Behavior

A vehicle in VANET is an intelligent device capable of communicating with other network elements. We have configured an application in each vehicle which makes use of TCP for end-to-end communication. The source sending rate of the application is directly proportional to the TCP window and inversely proportional to the RTT. As a result, by controlling the window size at the source, the source sending rate can be controlled.

<u>*Source node transport layer functionalities:*</u>

Initial State:

a. Transmits at highest possible rate
b. Estimates the round trip time using the timestamp information in the TCP header.

During Congestion Avoidance:

a. Extracts the advertised window size from the ACK packet
b. Adjust the source sending rate based on the newly received window size from the destination node.

Figure 3.7: Vehicle behaviors while using PI Rate control

Figure 3.7 describes the transport layer functionalities of a vehicle while using the APIRATE control algorithm. Initially, the source node (vehicle) negotiates the window size with the destination node during the connection setup and then starts transmitting information packets at the highest possible rate. For each data packet transmitted, the source node encapsulates the timestamp in the TCP header in order to calculate the RTT. On receiving the ACK message from the destination node, the source node calculates the RTT and records it for later use while updating the header. The ACK message received from the destination node also carries the window size calculated based on congestion at intermediate nodes. On receiving this message, the source node adjusts its sending rate based on newly determined window size. Another significant characteristic of the vehicle is the mobility pattern. Section 3.3.2.5, describes the vehicular mobility model used for data traffic congestion management in this thesis.

Intermediate Node Network layer functionalities:

a. Calculate the number of flows based on the source and destination address in each packet.
b. Calculate the average round trip time using the round trip time information available in the header of each packet.
c. Calculate the monitoring parameter
d. If K_P and K_I has not been calculated before, calculate K_P and K_I
e. else if the monitoring parameter is outside the prescribed limits
   Recalculate K_P and K_I
f. Calculate the source sending rate using the algorithm specific parameters
g. Derive the window size from the source sending rate
h. If the newly calculated window size is less than the window size in the TCP header, then update the window size in TCP header.
i. Send the packet to lower layers (MAC layer) for transmission to the next hop.

Figure 3.8: Roadside unit behaviors while using PI Rate control

3.3.2.3 Intermediate Node Behavior

As discussed earlier, one of the candidate nodes to experience congestion in VANET is the RSU. These are intelligent devices distributed across the vehicular network that can provide an external interface to the vehicles for Internet access and various on-demand and real-time services. RSUs often have a fixed bandwidth which has to be equally shared among all the vehicles without causing congestion.

Figure 3.8 describes the network layer functionalities of a RSU while using the APIRATE control algorithm. The RSU acts as an intermediate node for most of the application services provided in VANETs. Once the packet is removed from the queue, the APIRATE control algorithm executes congestion avoidance procedure discussed in Figure 3.8. This operation is performed for only TCP messages. Processing of routing layer messages and uncontrolled traffic is left unchanged. The algorithm advertises the source to reduce its sending rate based on the instantaneous queue length of the intermediate node (RSU).

The adaptive nature of the algorithm introduces a monitoring parameter which is used to self tune the controller only when there is dramatic change in network traffic. Monitoring parameter is a factor of RTT and number of flows. If any of these parameters changes drastically causing the monitoring parameter to fall outside the prescribed interval, then the APIRATE controller will self tune by recalculating the control parameters. The monitoring interval was determined by carrying out number of simulations.

3.3.2.4 Destination Node Behavior

The transport layer of the destination node plays an important role in controlling the source sending rate during the congestion avoidance procedure carried out by the APIRATE control algorithm. The destination node transmits the information received from the intermediate nodes to the source node using the ACK message.

Destination node transport layer functionalities:

a. Extracts the newly calculated window size from the TCP header
b. Copies the advertised window size into the ACK message
c. Transmits the message to the lower layers for transmission to the next hop.

Figure 3.9: Destination node behaviors while using PI Rate control

Figure 3.9 describes the transport layer functionalities of a destination node while using the APIRATE control algorithm. In addition to this, the transport layer at the destination node also assists in calculating the RTT. For each data packet received, it calculates the delay from source node to the destination node using the timestamp field in the header of each packet. While transmitting an ACK packet, the timestamp field is updated considering the delay from the source node to the destination node. This information is used by the source node to calculate the RTT.

3.3.2.5 Mobility Characteristics

The distinctive feature of VANET is the mobility characteristics of the vehicles. A vehicle in VANET can travel along the road in a vehicular speed ranging from 10 to 120 Km/Hr. The mobility characteristic of a vehicle depends on various factors like traffic signals, weather conditions, speed and distance of the vehicle travel in front, vehicle traffic congestion etc. Since the mobility pattern of a node in a general MANET is random, a random waypoint model [Qual08] has been designed and extensively used in network modeling software for MANET simulations. Unfortunately a random waypoint model selects random destination and speed for each node, and is not suitable for vehicular networks as the vehicles follow strict mobility patterns in the direction towards the destination. Furthermore, a random waypoint model does not give any importance to inter-node collisions. The use of traffic modeling software like SUMO is ruled out as it does not support vehicle to vehicle and vehicle to roadside unit communication.

The Qualnet simulation tool supports two types of mobility, file-based mobility and mobility based on the mobility model designed in Qualnet. In file-based mobility the node mobility information is specified in a user-readable text file. File-based mobility comes in handy while mobility information is available in logs or when third party mobility generators are used. Random waypoint model is an example for mobility model available in Qualnet. Another mobility model available in Qualnet is the group-based mobility model. In this model a group of nodes are assigned the same mobility vector. Within each group, a node moves using random waypoint model within the area of the group. The group based mobility option does not have a significant use in VANETs. In order to achieve vehicular mobility pattern in Qualnet, we have made use of the GMSF tool (Generic Mobility Simulation Framework) [BaLe08]. GMSF was developed as a part of master thesis at Swiss Federal

Institute of Technology, Zurich. GMSF provides mobility information compatible with Qualnet in a text file. Using file-based mobility option in Qualnet and the mobility information supplied by GMSF, vehicular mobility can be achieved in Qualnet. We have utilized the geographic information system based mobility model designed in GMSF with both car following and traffic light features. It is important to note that the speed of the vehicle does not remain fixed during the course of simulation. It changes based on different factors like traffic signals, the speed of the vehicle traveling in front etc. The mobility file provided by GMSF is well suitable for our simulations as it considers all these factors as in a real world scenario.

3.3.2.6 Assumptions

Following assumptions are associated with data traffic congestion management in VANETs:

1. Each vehicle is configured with a fixed wireless bandwidth: Due to the problems in a wireless network, the achieved throughput between two wireless nodes is usually much less than the supported bandwidth. This problem is resolved by using wireless links with fixed bandwidth between the RSU and the vehicle as each vehicle is capable of leasing a desired bandwidth from the service provider.
2. A greedy source is used: A vehicle always has some information to transmit to the destination node.
3. TCP window limitation: The window size advertised by the APIRATE algorithm is upper-bounded by the window size negotiated during initial connection setup to avoid complications.
4. Influence of other communication layers: No lower layer approaches are used or relied-on for further improvement of TCP over wireless networks.
5. Mobility characteristics across the road network: The road network information and the mobility pattern of the vehicles are predefined and do not change during the course of simulation.

3.4 Simulation Scenarios

This section discusses the simulation scenarios used for data traffic and vehicle traffic congestion management in this thesis. Each simulation scenario has been designed using the network models discussed in Section 3.2. The approach for designing and executing experiments using network modeling software is discussed in detail.

Realistic vehicular mobility is an important aspect for VANET simulations. Based on the requirement of our research, we have used different techniques to achieve vehicular mobility. In order to achieve dynamic vehicular mobility for vehicle traffic congestion management, the mobility characteristics and the car following model is defined in the application running in each node. The newly created application for vehicle traffic congestion management is configured in each node using the application file. The vehicle and RSU parameters are defined in the application file. For mobility scenarios in data traffic congestion management, the mobility characteristics are derived from GMSF (Generic Mobility Simulation Framework tool). The mobility trace generated by GMSF is fed to Qualnet using the node positions file. For Qualnet to read the mobility characteristics from node position file, the file-based mobility option is enabled in configuration settings. More details about simulation settings are discussed in later chapters.

3.4.1 Vehicle Traffic Congestion Control Scenario

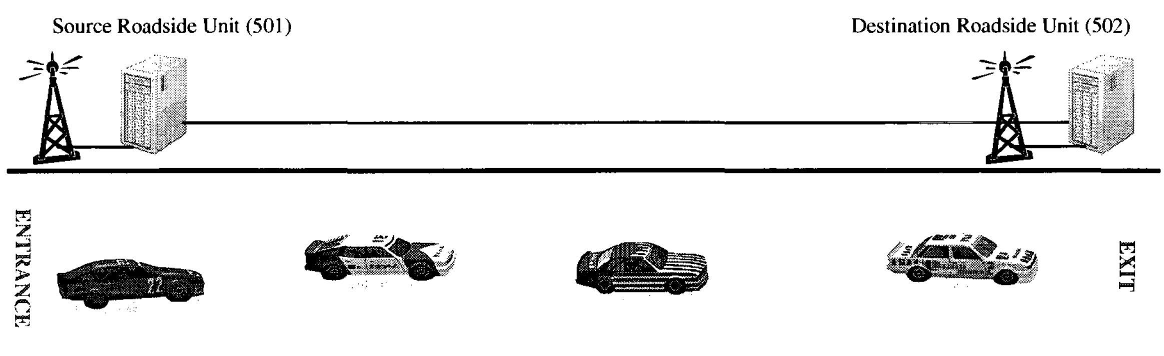


Figure 3.10: Vehicle traffic congestion management scenario.

To evaluate the performance of the APIRATE controller for vehicle traffic management, we have studied the vehicle traffic in a road segment of a complex network. Figure 3.10 shows the topology of the scenario used for vehicular traffic congestion management. A terrain dimension of 2000 x 400 meters was used in this scenario. The length of the road segment is 2000 meters. Nodes 1 to 500 are configured as vehicles and reused till the end of the simulation. All the vehicles are equipped with a 802.11a radio device with a data rate of 6 Mbps. Node 501 at the entrance is configured as the source RSU, and Node 502 at the exit is configured as the destination RSU. Nodes 501 and 502 communicate with each other over the backbone network. The scenario has been designed using the network model discussed in Section 3.2.1. All the vehicles start at (0, 160) and travel towards the destination along the horizontal x-axis at a configured speed. The car-following model defined in the application avoids any inter-vehicle collisions and controls the vehicle speed. Congestion arises when the number of vehicles using the road segment increases above the road capacity. We are addressing this problem by using the APIRATE controller to control the number of vehicles using the road to a target threshold. The performance of the controller is evaluated at different vehicle speeds and target thresholds.

3.4.2 Data Traffic Congestion Control Scenario in a Static VANET

A vehicle in VANET may be static or mobile. Vehicles parked in a parking lot are an example for a static scenario in VANET. We consider two network sizes, a small network and a large network. The static network scenarios will be used as a benchmark for the mobile scenarios.

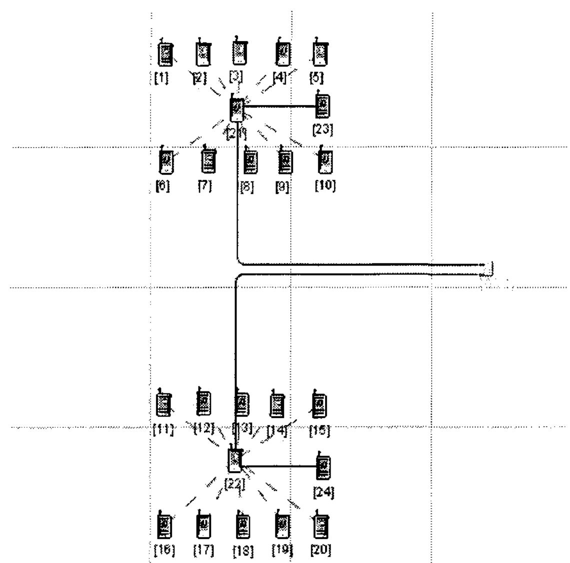


Figure 3.11: Small static vehicular network

3.4.2.1 Small Static Network

Figure 3.11 shows the topology of small static VANET with 20 vehicles. Ten vehicles parked in a parking lot are communicating with ten other vehicles in another area over the bottleneck link. Node#1 to Node#20 are vehicles and Node#21 and Node#22 are RSUs. The two RSUs are connected to each other via a backbone network with a link capacity of 20 Mbps. Each vehicle has leased a fixed bandwidth of 6Mbps (DSRC supports a data rate of 6 to 54 Mbps) from the respective RSUs in their area. Node#23 transmits uncontrolled UDP (User Datagram Protocol) traffic at a rate of $v(t)$ to Node#24 using the bottleneck link between RSU#21 and RSU#22. The initial rate of UDP traffic is set to 4 Mbps. Ten FTP (File Transfer Protocol) applications traffic streams are generated and sent to 10 vehicles in another area. We have used FTP/GENERIC [Qual08] version of the FTP application available in Qualnet. An FTP/GENERIC application is an FTP application available in Qualnet with more configurable parameters.

The initial aggregate arrival rate to RSU#21 is 64 Mbps (10 TCP flows, each transmitting at 6 Mbps and the uncontrolled UDP traffic with a data rate of 4 Mbps). Since the maximum achievable throughput between the RSUs is 20 Mbps, queue starts building up at RSU#21 connected to the TCP sources. The APIRATE controller is expected to handle the congestion and to share the bottleneck links fairly among all the TCP sources. The UDP traffic will be varied at different intervals in order to evaluate the performance of the algorithm at different controlled and uncontrolled traffic combination.

The definition of an FTP/GENERIC application for each vehicle is given below:

FTP/GENERIC <src> <dest> <items to send> <item size> <start time> <end time>

where:

FTP/GENERIC	Refers to the application type
<src>	Refers to source node ID
<dest>	Refers to destination node ID
<items to send>	Specifies the number of items to send. This parameter is set to zero in order to achieve a continuous flow
<item size>	Specifies size of each item in bytes. Set to 512 bytes in our study
<start time>	Indicates when transmission should begin. Set to 1 second in our study
<end time>	Indicates when transmission should end. Set to 0 in our study. (When set to zero, application will persist till the end of the simulation)

3.4.2.2 Large Static Network

Figure 3.12 depicts a large static network. The setup is similar to the small network where two RSUs are installed in two different areas except that the number of vehicles in each area is increased to 50 and the two RSUs are connected to each other by a link with a higher bandwidth capacity at 100 Mbps. Node#1 to Node#100 are configured as vehicles. Node#101 and Node#102 are configured as RSUs. As in the small network scenario, each vehicle has leased a fixed bandwidth of 6Mbps from the respective RSUs in their area. Node#103 transmits uncontrolled UDP traffic at a rate of $v(t)$ to Node#104 using the bottleneck link between RSU#101 and RSU#102. The initial rate of UDP traffic is set to 20 Mbps. Fifty FTP applications traffic streams are generated and sent to 50 vehicles in another area.

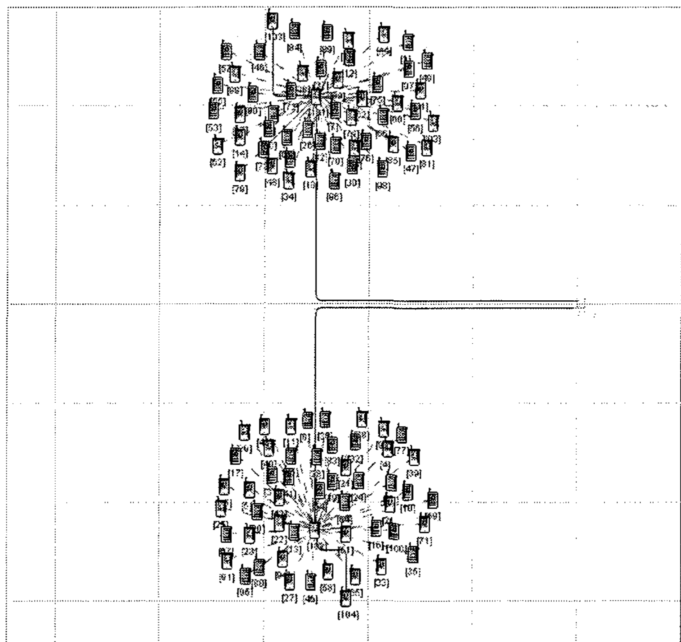


Figure 3.12: Large static vehicular network

The initial aggregate arrival rate at RSU#101 is 320 Mbps (50 TCP flows, each transmitting at 6 Mbps and the uncontrolled UDP traffic with a data rate of 20 Mbps). As the maximum achievable throughput between the RSUs is 100 Mbps, queue starts building up at the RSU connected to the TCP sources (RSU#101).

3.4.3 Data Traffic Congestion Control Scenario in Mobile VANET

The mobile scenarios will reflect the vehicular mobility pattern which is the main characteristics of vehicles as described in section 3.3.2.5. Two scenarios in urban (city traffic scenario) and rural (highway scenario) areas are used. The mobility patterns for these scenarios are derived from GMSF [BaLe08].

3.4.3.1 City Traffic Scenario

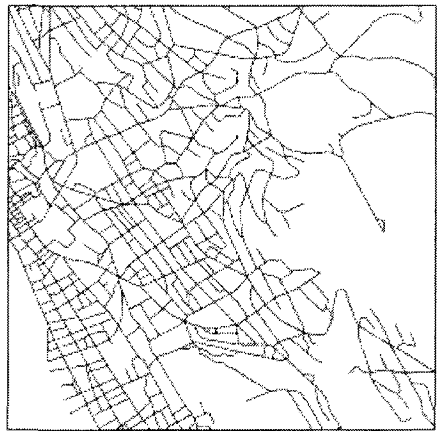


Figure 3.13: A City road network Example

The city road network connects different areas and comprises of number of intersections with traffic lights. The speed of the vehicle in a city traffic scenario is low compared to a rural or highway scenario. Figure 3.13 shows a city road network with a terrain dimension of 3000 x 3000 meter. This is used by GMSF [BaLe08] to generate the city traffic mobility pattern for the vehicles.

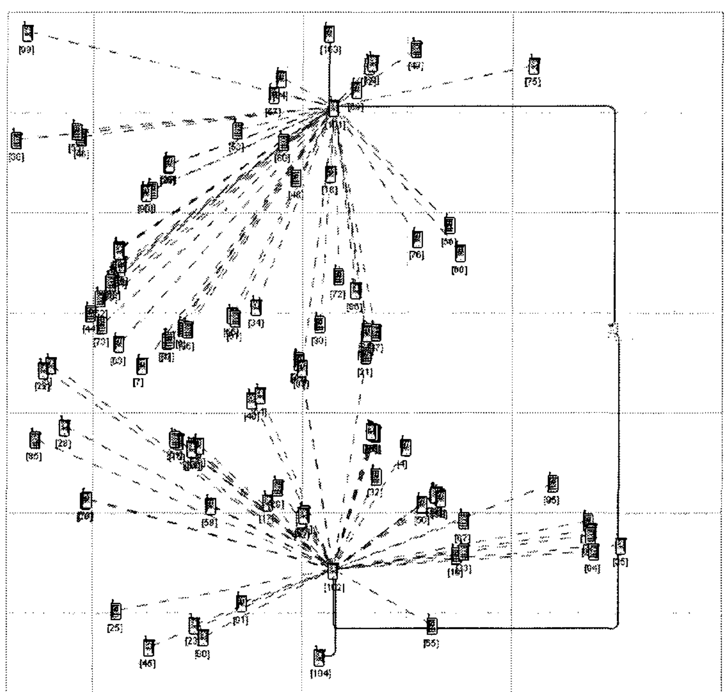


Figure 3.14: City traffic congestion scenario

Figure 3.14 shows the initial positions of the vehicles in the network example of Figure 3.13. The network architecture and application configuration is same as the one used in

Section 3.4.2. The impact of city traffic mobility model on the performance metrics in both large network (100 vehicles) and small network (20 vehicles) are studied.

3.4.3.2 Rural Scenario (Highway Scenario)

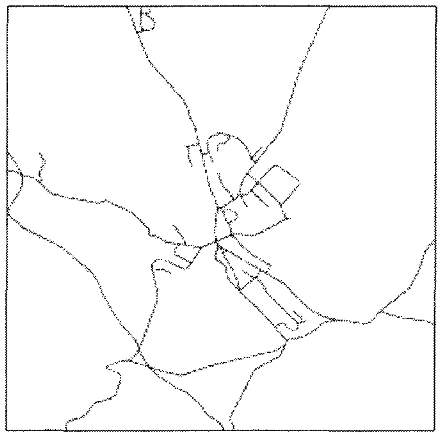


Figure 3.15: A Rural road network Example

A rural road network comprises of a number of highways where vehicles can travel at much higher speed compared to the city traffic scenario. Figure 3.15, shows rural network with a terrain dimension of 3000 x 3000 meters. This is used by GMSF to generate the rural traffic mobility pattern for the vehicles [Bale08].

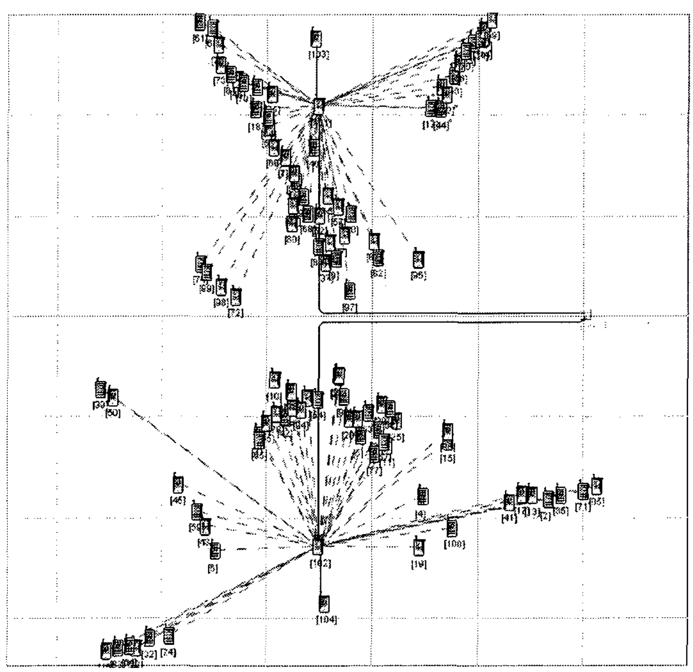


Figure 3.16: Rural traffic congestion scenario

Figure 3.16, shows the initial positions of the vehicles in the network example of Figure 3.15. The impact of rural traffic mobility model on the performance metrics in both large network (100 vehicles) and small network (20 vehicles) are studied. The network architecture and application configuration is same as the one used in Section 3.4.2.

3.5 Performance Measures

In this section we have discussed the performance metrics used for both vehicle traffic congestion management and data traffic congestion management in this thesis. The average values presented in this thesis are obtained by taking an average of the results obtained from four simulation runs, 95% confidence intervals are also computed. This is different from the time average measurement such as average trip time below.

3.5.1 Performance Measures for Vehicle Traffic Congestion Management

Following is a list of performance measures along with their definitions:

1. Queue Length: It is defined as the number of vehicles that have occupied the road segment at any given instant of time. Average queue length is the average value of the queue length over the simulation period. Instantaneous queue length is the queue length computed at fixed time intervals.
2. Throughput: It is defined as the number of vehicles allowed into the road segment from the entrance per time unit. Average throughput is the average number of vehicles per unit time allowed to use the road segment over the simulation period. Instantaneous throughput is the number of vehicles per time unit allowed to use the road segment which are computed at fixed time intervals.
3. Average trip time: This is defined as the average time taken by a vehicle to travel from beginning of the road segment till the end of the road segment.
4. Average speed: This is the average speed of the vehicle while traveling from beginning of the road segment till the end of the road segment.

Queue length is the main performance measure used to evaluate the performance of APIRATE (Adaptive Proportional Integral rate) controller while dealing with the problem of vehicle traffic congestion. One of the main objectives of our work is to address the problem of congestion by controlling queue length to a target threshold. By doing so, we will be able to control the vehicle traffic volume below the capacity of the road.

3.5.2 Performance Measures for Data Traffic Congestion Management

Following are the list of performance measures along with their definitions:

1. Throughput: Average throughput is defined as average number of bits transmitted per unit time over the simulation period. Instantaneous throughput is defined as number of bits transmitted per time unit which are computed at fixed time intervals.
2. Round Trip Time (RTT): This is time interval between the transmission of first bit of a data packet and the reception of last bit of an ACK packet. Average RTT is the average value of RTT over the simulation period. Instantaneous RTT is value of RTT per time unit which are computed at fixed time intervals.
3. Queue Length: It is defined as the number of packets that have occupied the buffer at any given instant of time. Average queue length is the average value of the queue length over the simulation period. Instantaneous queue length is the queue length computed over fixed time intervals.
4. Fairness: This performance metric compares the actual individual allocations with ideal allocations. We have used Jain's fairness index [JaCh84] defined as follows:

$$F(x) = \left[\sum_{i=1}^{n} x_i\right]^2 \Big/ n\sum_{i=1}^{n} x_i^2$$

   In the equation, n represents the number of flows and x_i denotes the throughput achieved by flow i. This metric evaluates the performance of bandwidth allocations to users in a shared medium or channel.
5. Loss percentage: It is defined as the percentage of packets lost over the simulation period. It is calculated based on total number of packets transmitted by the source and the total number of packets dropped.

In order to provide real-time multimedia service like streaming multimedia there is a need for a constant throughput. The throughput parameter directly influences the quality of service of these applications. Therefore, throughput has been the main performance measure.

3.6 Concluding Remarks

In this chapter, we have discussed the network models used for data traffic and vehicle traffic congestion management. The PI rate control mechanism to deal with data traffic and vehicle traffic congestion has been presented. The implementation of the PI rate algorithm along with the main operations of different network components is discussed in detail. The simulation scenarios derived from network models are listed. We have also discussed the mobility characteristics of vehicles used in the simulations. This chapter additionally includes the assumptions made in this thesis and the performance measures used in the simulation.

Chapter 4
Vehicle Traffic Management in VANET

This section evaluates the performance of the APIRATE controller using the Qualnet simulation model depicted in Section 3.3.1. The simulation environment and configuration settings associated with the simulation are discussed. We shall evaluate the performance of the algorithm in terms of queue length, throughput, speed and delay under the scenario depicted in Section 3.4.1.

4.1 Simulation Environment

The performance of the APIRATE controller, to perform vehicle traffic management has been done by using Qualnet simulation tool [Qual08]. Mobility pattern supported by most of the network modeling software is predefined and does not change during the course of the simulation. Qualnet as a network modeling software supports dynamic mobility using inbuilt APIs. Due to the limitations of Qualnet with respect to dynamic mobility along a realistic road network, we have considered a simple single lane scenario where vehicle moves in one direction along increasing x-axis. RSUs (Roadside Units) installed at the entrance and exit of the road segment perform vehicle traffic management by controlling the number of vehicle that enter the road segment. The APIRATE controller installed in the destination RSU calculates the desired rate and transmits it to the source RSU. The dynamic mobility for the vehicles is controlled by the application installed in each vehicle. We have implemented the "Single-Lane Traffic Simulation with Multi-Agent System", car-following model to achieve realistic vehicular mobility.

The Qualnet 4.0 network modeling software is installed in Windows XP machine with Pentium IV 6.6 GHz processor and 2 GB of memory. The results are obtained by taking an average of 4 different seeds, from which 95% confidence intervals are computed and shown in brackets later. Each simulation run takes about 7 minutes to complete.

4.2 Simulation Configuration Setting

This section presents and discusses the configuration parameters and their settings associated with vehicle traffic congestion management simulation.

Table 4.1: General Simulation Configuration Settings

Parameter	Value
SIMULATION-TIME	18000 seconds
SEED	31, 97, 151, 191
TERRAIN-DIMENSIONS	2000 x 400 meters
DUMMY-NUMBER-OF-NODES	500
NODE-PLACEMENT	FILE
NODE-POSITION-FILE	file_name.nodes
MOBILITY	None

Table 4.1 presents the general configuration setting associated with the simulation. Each scenario is simulated for 18000 seconds. In order to analyze the consistency of our results, we have taken average of the required statistics from different simulation runs. For each simulation run, a different seed (set by the SEED parameter) is used in order to generate different sequences of random numbers as needed by a new simulation. We have used a 2000 x 400 meters terrain for our simulations configured using the TERRAIN-DIMENSIONS parameter. TERRAIN-DIMENSIONS is used to configure the dimensions (x, y) of the terrain. Parameters, x and y are dimensions in meters. We have configured 500 nodes in the simulated environment. The initial positions of these nodes are configured using the NODE-PLACEMENT and NODE-POSITION-FILE parameter. The MOBILITY parameter is set to none, for we are not using any of the in-built mobility models in Qualnet. We have achieved dynamic mobility by modifying the application layer protocol and implementing the car following model.

Table 4.2: Radio/Physical Layer Configuration Settings

Parameter	Value
PHY-MODEL	PHY802.11a
PHY802.11-DATA-RATE	6 Mbps
PHY802.11a-TX-POWER--6MBPS	20.0 dBm
PHY802.11a-RX-SENSITIVITY--6MBPS	-85.0 dBm
ANTENNA-MODEL	OMNIDIRECTIONAL
ANTENNA-GAIN	0.0 dBi

Table 4.2 presents the configuration settings associated with the Radio/Physical layer. Each node in the network is configured with an 802.11a radio with a data rate of 6Mbps. The transmission power and receiver sensitivity for the radio is set to 20.0 dBm and -85.0 dBm

respectively. An omni-directional antenna with a gain of 0.0dBi has been configured at each node.

Table 4.3: Network Protocol Configuration Settings

Parameter	Value
NETWORK-PROTOCOL	IP
IP-QUEUE-TYPE	FIFO
ROUTING-PROTOCOL	AODV
AODV-NET-DIAMETER	35
IP-FORWARDING	YES

The configuration settings associated with the network layer are presented in Table 4.3. The network layer protocol at each node is configured as IP. Each node is configured with a FIFO queue to store the incoming packets that cannot be serviced immediately. We have used the AODV (Ad hoc On Demand Distance Vector) routing protocol as it dominates some of the existing MANET protocols when used for VANETs. AODV is configured to support a maximum number of 35 hops between two nodes in the network. Each node in the network is capable of forwarding packets.

4.3 Performance Evaluation

In this section, we shall analyze and evaluate the performance of APIRATE controller to perform vehicle traffic congestion management in VANETs (Vehicular Ad-Hoc Networks).

4.3.1 Vehicle Traffic Congestion Management

The performance of the APIRATE controller is evaluated in a single lane road segment in terms of queue length, throughput, speed, and trip time. Details of the scenario have been explained in Section 3.4.1. Our aim is to observe how the APIRATE controller performs vehicle traffic management in a linear road segment. We have derived different scenarios by varying the speed of the vehicle and by using different target queue thresholds. The target queue threshold is obtained by considering the speed of the vehicle, length of the road segment and the length of the vehicle.

4.3.1.1 Scenario 1: Desired Speed of 20 m/s (72 KM/hr) and Target Queue Length of 60 vehicles in a 2000 meter road segment

In this scenario, we have set the desired speed of the vehicle to 20 m/s. The desired speed is the minimum speed desired by the driver while driving. The target queue length is set to 60 vehicles, considering the size of the road segment, the desired gap between two vehicles, length of the vehicle (5meters) and the vehicle speed.

a) Queue Length:

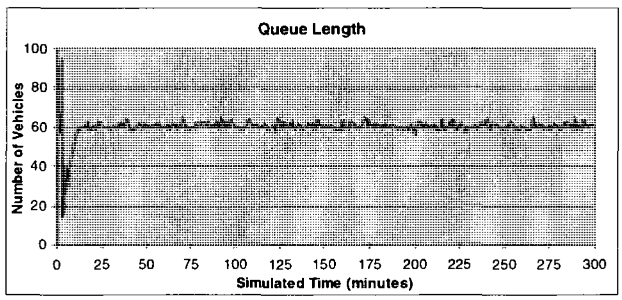


Figure 4.1: Scenario 1 – Queue length behavior of road segment

Figure 4.1 shows the instantaneous queue length (number of vehicle on the road segment of interest) behavior at the road segment. The APIRATE controller is able to control the queue length to the target threshold in less than 10 minutes with an average queue length of 60 (±0.045) vehicles[†]. Initially, when all the vehicles are allowed to use the road segment, the traffic volume using the road segment increases, which results in a large queue length (congestion condition). When the APIRATE controller starts the congestion avoidance procedure, the target queue length reduces to the desired target queue threshold. The PI rate controller calculates the desired rate based on the instantaneous queue length and transmits it to the source RSU. Based on the newly received rate, the source RSU controls the traffic volume entering the road segment. Traffic volume being one of the reasons for congestion, we can avoid congestion by controlling the queue length to a desired target threshold.

b) Throughput:

Figure 4.2 shows the instantaneous throughput at the source RSU. Initially all the vehicles

[†] The figure in the brackets are 95% confidence interval as discussed in Section 4.1

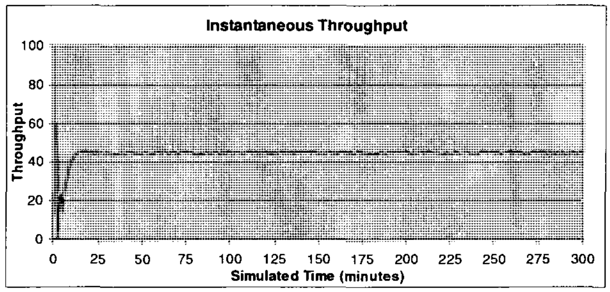


Figure 4.2: Scenario 1 – Throughput at the source roadside unit

are allowed to use the road segment. When the newly calculated rate is received from the destination RSU, the source RSUs controls the traffic volume entering the road segment based on newly calculated rate. The average throughput was observed to be around 44.16 (± 0.08) vehicles/minute. The APIRATE algorithm provides a steady throughput at the source without causing congestion.

c) Average Speed:

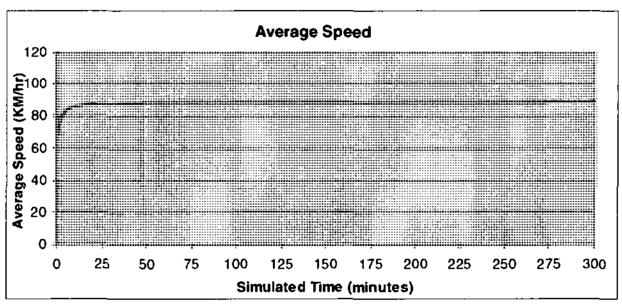


Figure 4.3: Scenario 1 – Average speed of the vehicle

Figure 4.3 shows the average speed of the vehicle. Initially, the average speed was observed to be around 67 KM/hr. As the APIRATE controller initiated the congestion avoidance procedure, the average speed of the vehicle increased from 67 KM/hr to 88.129 (±0.065) KM/hr. The observed speed of the vehicle, i.e. 88 KM/hr (24.48 m/s) was higher than the configured desired speed (20 m/s).

d) Average Trip Time:

The average time taken to travel the entire road segment was nearly 81.72 (±0.059) seconds. Figure 4.4 shows the behavior of average trip time during the simulation. The time taken to travel from the source to destination reduces from 107.06 seconds to 81.72 seconds.

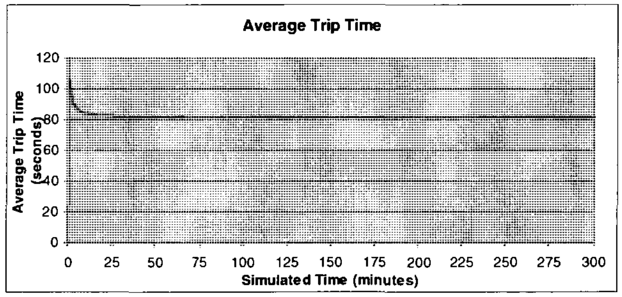


Figure 4.4: Scenario 1 – Average trip time of the vehicle

From the results discussed so far, we have observed that the APIRATE algorithm is capable of addressing the problem of congestion by controlling the queue length to a per-configured target threshold. We have also observed an increase in average speed of the vehicle while using this algorithm. We shall now vary the speed and queue length parameter and analyze the performance of the algorithm in the same scenario.

4.3.1.2 Scenario 2: Desired Speed of 25 m/s (90 KM/hr) and Target Queue Length of 50 vehicles in a 2000 meter road segment

In this scenario we have set the desired speed of the vehicle to 25 m/s and the target queue threshold of the road segment to 50 vehicles. The target queue threshold is determined based on the conditions discussed in Section 4.3.1.1.

a) Queue Length:

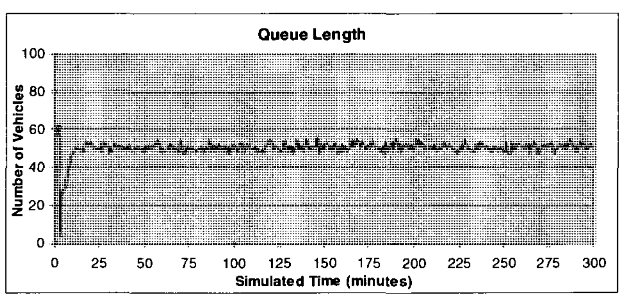


Figure 4.5: Scenario 2 – Queue length behavior of road segment

Figure 4.5 shows the instantaneous queue length behavior at the road segment. The APIRATE controller is able to tune the queue length to target threshold in less than 10 minutes. The average queue length was observed to be around 50 (± 0.067) vehicles same as the configured target threshold. Comparing to Scenario 1, the APIRATE (Adaptive Proportional Integral rate) controller is able to control the queue length to a target threshold even when the desired speed is increased to 25 m/s.

b) Throughput:

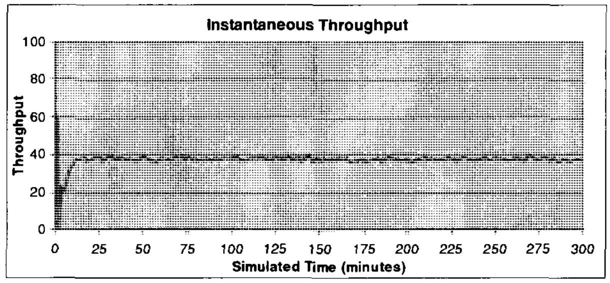


Figure 4.6: Scenario 2 – Throughput at the source roadside unit

Figure 4.6 shows the throughput behavior at the RSU. The average throughput was observed to be around 37.15 (±0.077) vehicles/minute. When the queue length is above the target threshold, the APIRATE controller reduces the rate to reduce the queue length. The algorithm then increases the rate based on the desired queue length, and consequently increasing the throughput. The APIRATE algorithm provides a steady throughput at the source without causing congestion.

c) Average Speed:

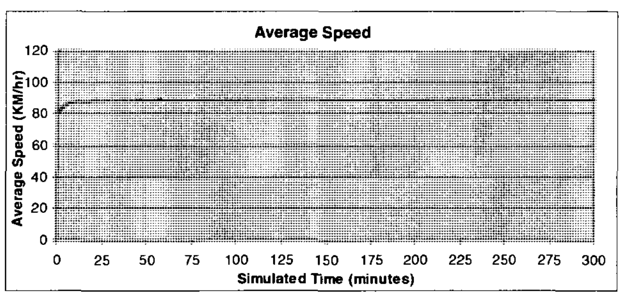


Figure 4.7: Scenario 2 – Average speed of the vehicle

Figure 4.7 shows the average speed of vehicles. The average speed of the vehicle increased from 81 KM/hr to 88.74 (± 0.052) KM/hr. The improvement is observed as a result of the congestion avoidance procedure carried out by the APIRATE algorithm. As the leading vehicle travels slowly, all the followers cannot travel faster than the leading vehicle, this results in congestion. When the speed of the leading vehicle is increased, the speed of followers increases.

d) Average Trip Time:

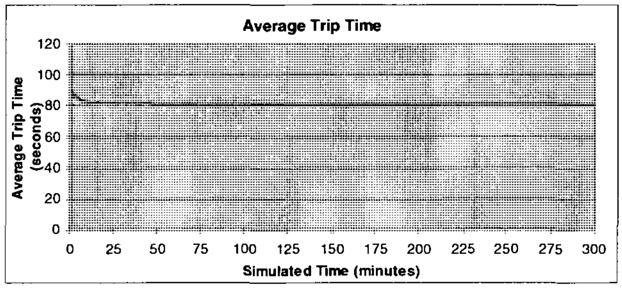


Figure 4.8: Scenario 2 – Average trip time of the vehicle

Figure 4.8 shows the average trip time behavior for the vehicles. The average trip time delay was observed to be around 81 seconds. The average trip time reduced from 88 seconds to 81.13 (± 0.061) seconds. Initially during congestion, the trip time is large. When the controller executes the congestion avoidance procedure, the average trip time reduces.

4.3.1.3 Scenario 3: Desired Speed of 30 m/s (108 KM/hr) and Target Queue Length of 40 vehicles in a 2000 meter road segment

In this scenario we have set the desired speed of the vehicle to 30 m/s and the target queue threshold of the road segment to 40 vehicles. The target queue threshold is determined based on the conditions discussed in Section 4.3.1.1.

a) Queue Length:

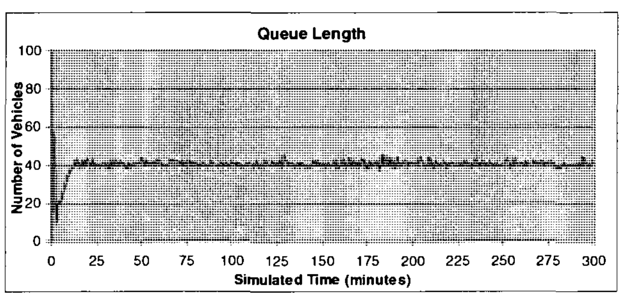


Figure 4.9: Scenario 3 – Queue length behavior of road segment

Figure 4.9 shows the queue length behavior at the road segment. The APIRATE controller is able to control the queue length to target threshold as in other scenarios. Traffic volume being one of the reasons for congestion, we can avoid congestion by controlling the queue length to a desired target threshold.

b) Throughput:

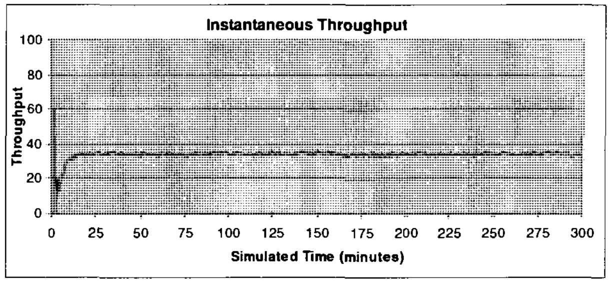


Figure 4.10: Scenario 3 – Throughput behavior at the source roadside unit

Figure 4.10 shows the instantaneous throughput behavior at the RSU. The average throughput was observed to be around 33.84 (± 0.055) vehicles/minute. The average speed of the vehicle was observed to be around 101.32 (± 0.061) KM/hr and the average trip time was nearly 71.06 (± 0.089) seconds.

One of the factors influencing congestion is the speed of the vehicle. In this scenario, all the vehicles are able to travel at a speed close to the desired speed without causing congestion. As in other scenarios, the APIRATE controller is able to control the queue length to target threshold even when the desired speed of the vehicle is set to 30 m/s. In this thesis, we have determined the target queue length by considering the speed of the vehicle, the length of the road segment, the desired gap between two vehicles and the size of the vehicle. In reality, in addition to these factors, a thorough analysis of vehicle traffic in a given area has to be done to determine this threshold.

Table 4.4: Performance summary for vehicle traffic congestion management

Linear Road Segment			
Parameter	Scenario 1	Scenario 2	Scenario 3
Desired Speed	20 m/s	25 m/s	30 m/s
Average Queue Length	60 vehicles	50 vehicles	40 vehicles
Throughput	44 vehicles/minute	37 vehicles/minute	33 vehicles/minute
Average trip time	81 seconds	81 seconds	71 seconds
Achieved speed	24.48 m/s	24.65 m/s	28.14 m/s

Table 4.4 shows the summary of results for the vehicle traffic congestion management scenarios. The APIRATE controller has successfully managed to control the queue length to a target threshold at different vehicular speeds. Furthermore, we also observed an increase in vehicle speed in some scenarios while using this controller.

4.4 Concluding Remarks

In this chapter, the simulation environment and configuration settings associated with vehicle traffic congestion management are discussed. We have analyzed and evaluated the performance of the APIRATE controller to perform vehicle traffic management at different vehicle speeds and different queue thresholds. We have observed that the APIRATE controller is able to control the queue length to a target threshold, while running the congestion avoidance procedure. Furthermore, we also observed an increase in vehicle speed while using this controller. From our results, we would like to conclude that the APIRATE controller is a potential algorithm to deal with the problem of vehicle traffic congestion seen when the traffic volume exceeds the road capacity. By avoiding congestion in regions that are subject to vehicle traffic congestion, end-to-end delay can be reduced.

Chapter 5
Data Traffic Management in VANET

This section evaluates the performance of the APIRATE controller using the Qualnet simulation model depicted in Section 3.3.2. The performance of the controller is compared with the combined performance of TCP NEWRENO, which performs source-based congestion management, and RED queue which performs AQM in terms of throughput, window size, queue length, fairness and RTT under different scenarios depicted in Section 3.4. The simulation environment and configuration settings associated with the simulation are discussed.

5.1 Simulation Environment

We have implemented the data traffic congestion management using the APIRATE controller in a Qualnet simulator. The static and mobile scenarios discussed in Section 3.4 are designed using the scenario designer tool available in Qualnet. Details about designing scenarios are discussed in Appendix C. In order to generate TCP (Transmission Control Protocol) traffic, FTP/GENERIC applications are defined from source to destination via the congested node. FTP/GENERIC applications make use of TCP at the transport layer. Furthermore, it is a more configurable model where packet size and total items to send can be controlled. More details of this model have been discussed in Section 3.4.2. The uncontrolled traffic is generated by defining a CBR (Constant Bit Rate) application between two nodes which utilizes the bottleneck link to communicate. The CBR application makes use of UDP at the transport layer for transmission. The vehicular mobility model for the mobile scenario is obtained using the GMSF tool. The impact of city traffic and rural traffic mobility scenario is evaluated.

We have used the Qualnet 4.0 [Qual08] network modeling software installed in a Windows XP PC with a Pentium IV 6.6 GHz processor and 2 GB of memory. The results are obtained by taking an average of 4 different seeds, from which 95% confidence intervals are computed and shown in brackets later.

5.2 Simulation Configuration Setting

This section presents and discusses the configuration parameters and their settings associated with data traffic congestion management simulation.

Table 5.1: General Configuration Settings

Parameter	Value
SIMULATION-TIME	1000s
SEED	31, 97, 151, 191
TERRAIN-DIMENSIONS	3000 x 3000 meters
DUMMY-NUMBER-OF-NODES	- 26 for small network scenarios - 106 for large network scenarios
NODE-PLACEMENT	FILE
NODE-POSITION-FILE	file_name.nodes
MOBILITY	- None for Static scenario - FILE for Mobile scenarios

Table 5.1 presents the general configuration setting of our simulations. Each scenario is simulated for 1000 seconds with four different seeds to collect the average of desired statistics. A terrain dimension of 3000 x 3000 meters is used. We have used 26 nodes in small network scenarios and 106 nodes in large network scenarios. The positions of these nodes are specified in a file supplied via the NODE-PLACEMENT (must be configured as FILE) and the NODE-POSITION-FILE parameters. In addition to the initial node positions, the NODE-POSITION-FILE is used to define the mobility characteristics for the nodes. In order to read the mobility characteristics from the NODE-POSITION-FILE, the MOBILITY parameter should be configured as FILE based mobility. For the static scenario, the MOBILITY parameter is set to NONE.

Table 5.2 presents the configuration settings associated with the wireless device. We have used an 802.11a physical layer model as it is closely related to DSRC. Note that the 6 Mbps version of DSRC is emulated here. The transmission power and receiver sensitivity for the radio is set to 20.0 dBm and -85.0 dBm respectively. We have used and omni-directional antenna with a gain of 0.0dBi. In order to provide a fixed bandwidth to the subscribers we have defined wireless links between the nodes and the infrastructure with a bandwidth of 6 Mbps.

Table 5.2: Wireless Configuration Settings

************ Radio/Physical Layer Settings ************	
Parameter	**Value**
PHY-MODEL	PHY802.11a
PHY802.11-DATA-RATE	6 Mbps
PHY802.11a-TX-POWER--6MBPS	20.0 dBm
PHY802.11a-RX-SENSITIVITY--6MBPS	-85.0 dBm
ANTENNA-MODEL	OMNIDIRECTIONAL
ANTENNA-GAIN	0.0 dBi
************ Wireless Link Settings ************	
Parameter	**Value**
LINK-PHY-TYPE	WIRELESS
LINK-BANDWIDTH	6 Mbps

Table 5.3: MAC, Network & Transport layer Configuration Settings

************ MAC Settings ************	
Parameter	**Value**
MAC-PROTOCOL	MAC802.11
MAC-DOT11-DIRECTIONAL-ANTENNAMODE	NO
************ Network Protocol Settings ************	
Parameter	**Value**
NETWORK-PROTOCOL	IP
IP-QUEUE-NUM-PRIORITIES	1
IP-QUEUE-PRIORITY-INPUT-QUEUE-SIZE	- 500 packets for large network - 100 packets for small network
IP-QUEUE-TYPE	- FIFO for PIRATE - RED for TCP NEWRENO
ROUTING-PROTOCOL	AODV
AODV-NET-DIAMETER	35
IP-FORWARDING	YES
************ Transport Layer Settings ************	
Parameter	**Value**
TCP	NEWRENO
TCP-MSS	512 bytes
TCP-SEND-BUFFER	16384 bytes
TCP-RECEIVE-BUFFER	16384 bytes

Table 5.3 presents the configuration settings associated with the MAC (Media Access Control), the Network and the Transport layers. We have used the 802.11 MAC for our simulation. We have disabled the directional antenna mode feature in the MAC layer for our

simulations. The dominant IP protocol is used at the Network layer to facilitate communication. Each node is configured with one buffer (as indicated in IP-QUEUE-NUM-PRIORITIES parameter) with a size (specified in IP-QUEUE-PRIORITY-INPUT-QUEUE-SIZE parameter) based on bandwidth delay product. The queue is configured as FIFO for PI rate controller and RED when using TCP NEWRENO. In order to facilitate routing, we have used the AODV protocol with a maximum of 35 hops between two nodes in the network. Each node in the network is capable of forwarding packets. At the Transport layer, we have used the TCP NEWRENO variant for our simulations. The maximum TCP message segment size is configured to 512 bytes. The initial size of send and receive buffer are configured to 16384 bytes respectively.

5.3 Performance Evaluation & Comparison

This section first evaluates the performance of the controller in a static network. We then introduce mobility and study the impact of mobility on performance metrics. All the scenarios have been described in Chapter 3.

5.3.1 Data Traffic Congestion Control in Static VANET

The small and large network scenarios in Section 3.4.2 are used in our study. In addition to small and large network, we also study the impact of uncontrolled traffic on the performance metrics.

5.3.1.1 Small Static Network

The small static network has been evaluated in terms of throughput, queue length, window size, RTT, and loss percentage.

a) Throughput:

Figure 5.1 shows the instantaneous application throughput for one of the TCP flows along with the throughput of uncontrolled UDP traffic. Both the PI rate controller (Figure 4.2b) and the NEWRENO (Figure 4.2c) are compared. While using the PI rate controller, the application throughput fluctuates with a very small amplitude. As seen in the figure, when the uncontrolled UDP traffic is increased from 4 Mbps to 8 Mbps, the PI rate controller is able to reduce the source transmission rate of the controlled TCP traffic. When the UDP traffic is increased to 12 Mbps, the TCP traffic is reduce further. When the UDP traffic is reduce to 10 Mbps, the source transmission rate of TCP application is increased. The PI rate controller is able to dynamically adapt to network conditions. In the case of TCP NEWRENO with RED queue, the application throughput fluctuates with a very high amplitude. TCP NEWRENO employs AIMD technique to modify the TCP window this is the main reason for throughput fluctuations.

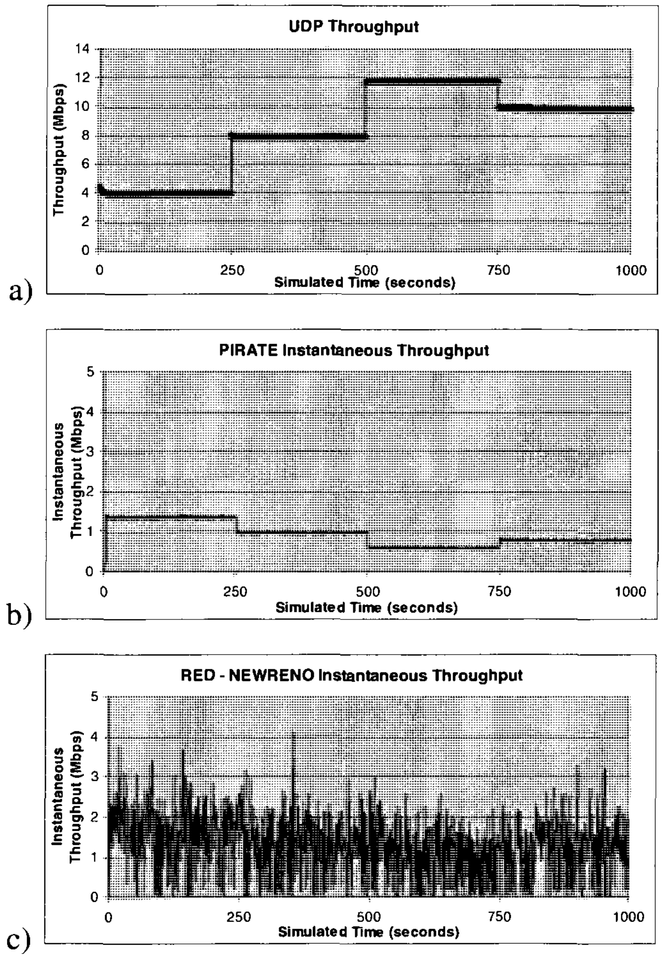


Figure 5.1: Application throughput for a small static network

Figure 5.2 shows the mean node throughput for the source nodes. While using the PI rate controller the throughput among all nodes remains almost same. As seen in the figure, TCP NewReno with RED queue does not succeed in distributing the bottleneck bandwidth equally among all the source nodes.

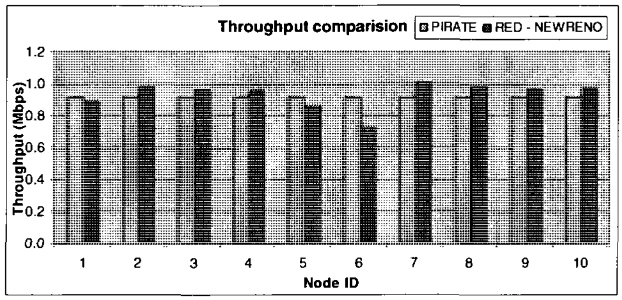


Figure 5.2: Mean throughput by node ID for a small static network

b) Queue Length:

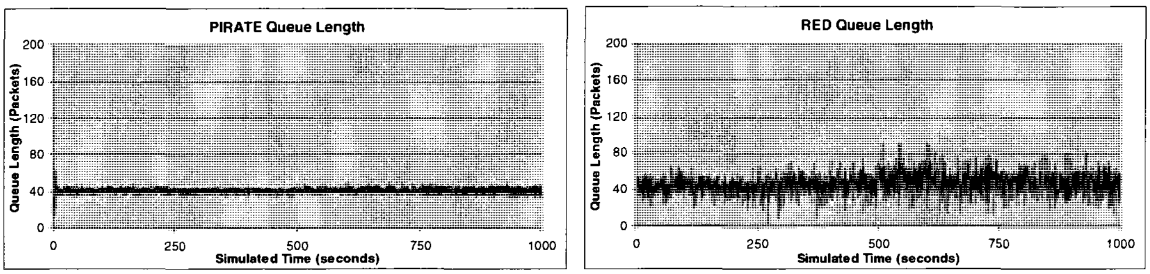


Figure 5.3: Instantaneous queue length behavior for a small static network

Figure 5.3 shows the queue length behavior of congested node (RSU) while using the PI rate controller and the RED queue. The target queue size in the case of PI rate controller is set to 40 packets (40% of queue size). For RED queue, the minimum and maximum queue thresholds are set to 30 and 70 packets respectively. As seen in the figure, the PI rate controller is able to control the queue length under the configured target threshold of 20 (± 0.739) Kbytes‡ even in the presence of uncontrolled UDP traffic. When the UDP traffic is varied, in the presence of the PI rate controller, the average queue length remains close of target threshold. In the case of RED queue, the queue length fluctuates with much higher amplitude between the minimum and maximum threshold with respect to an average of 23 (± 0.835) Kbytes.

Figure 5.4, shows the behavior of queue length when the UDP traffic ($v(t)$ in network model) is varied. In the case of PI rate controller, the average queue length is quite stable and close to the configured target threshold. In the case of TCP NEWRENO with RED queue, the average queue length increases with increase in uncontrolled UDP traffic. The increase in average queue length also results in increase in RTT.

‡ The figure in the brackets are 95% confidence interval as discussed in section 5.1

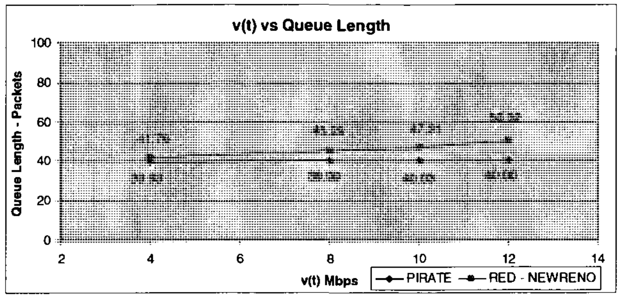


Figure 5.4: Average queue length vs. uncontrolled traffic rate in a small static network

c) TCP Window:

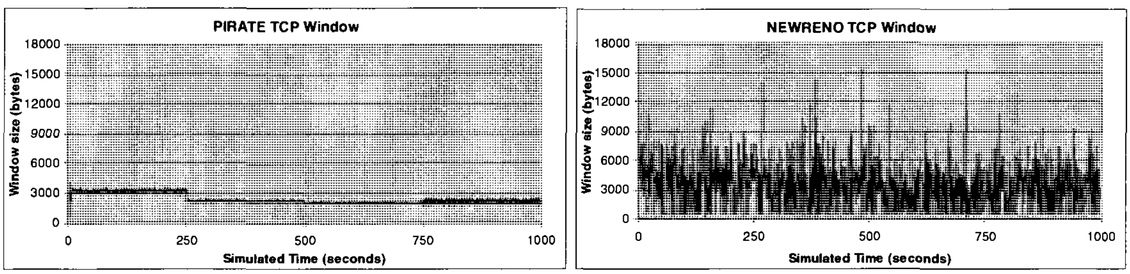


Figure 5.5: Source TCP window behavior for a small static network

Figure 5.5 shows the window size behavior at one of the TCP sources for both the PI rate controller and the TCP NEWRENO with RED queue. In the case of PI rate controller the TCP window fluctuates with a very small amplitude with respect to an average of 2317 (± 12.45) bytes. The APIRATE controller dynamically controls the size of the TCP window based on congestion at the intermediate node. In the case of TCP NewReno the TCP window fluctuates with a very large amplitude with respect to an average of 3368 (±53.91) bytes. The observed fluctuation is mainly due to the AIMD mechanism used by TCP NEWRENO. The window size fluctuation for TCP NewReno continues even while varying uncontrolled traffic.

d) Round Trip Time (RTT):

Figure 5.6 shows the RTT behavior for PI rate and TCP NewReno. While using the APIRATE controller the average RTT was observed to be around 18.297 (± 0.035) milliseconds. In the case of TCP NewReno, the average RTT was around 19.622 (±0.0734) milliseconds. The average RTT in the case of TCP NEWRENO was observed to be higher than PI Rate controller. The increase in RTT is due to the increase in queue length at the congested node.

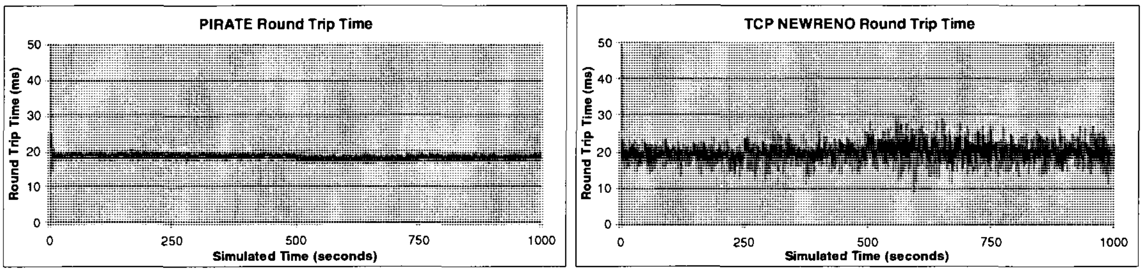


Figure 5.6: RTT behavior for a small static network

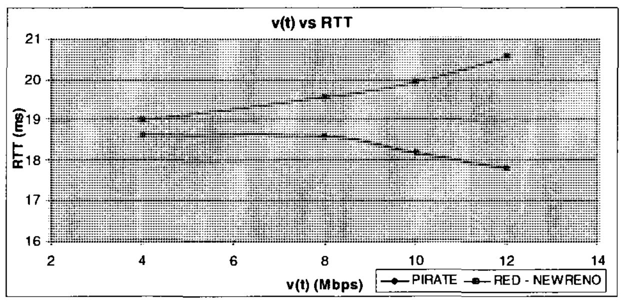


Figure 5.7: Average RTT vs. uncontrolled traffic rate in a small static network

Figure 5.7 shows the impact of uncontrolled traffic on RTT. In the case of TCP NEWRENO with RED queue, as the uncontrolled traffic is increased the average RTT increases. The increase in RTT is mainly due to the increase in average queue length at the congested node. In the case of PI rate controller, the average RTT reduces with the increase in uncontrolled traffic. This is because as the uncontrolled traffic is increased, the APIRATE (Adaptive Proportional Integral rate) controller reduces the source transmission rate by reducing the TCP window at each source. As the window size is reduced, it takes less time to receive an ACK. As a result, the average RTT decreases.

e) <u>Loss percentage:</u>

For PI rate controller, no packets were dropped at the network layer. This is because the PI rate controller does not interpret packet loss as the reason for congestion. While using TCP NEWRENO with RED queue, 1.65% of packets transmitted were dropped at the congested RSU. The operation of RED is discussed in detail in Section 2.1.1.

Table 5.4: Performance summary for a static network with 20 vehicles

Small Static Network			
Parameter	**Node ID**	**PIRATE**	**TCP NEWRENO**
Average Throughput (bps)	1	912909	888446
	2	912791	975890
	3	913296	961767
	4	913176	958253
	5	913153	855411
	6	912922	723202
	7	913120	1006852
	8	913125	974624
	9	912746	962576
	10	913157	968753
Fairness Index		0.9999	0.9925
Average RTT (milliseconds)		18.297	19.622
Average Window (bytes)		2317.554	3368.094
Average Queue Size (packets)		39.990	45.870
Packets drops at network layer		0	72053

Table 5.4 summarizes the performance of the static network with 20 vehicles. In this table we have shown the throughput of each destination node receiving data from the 10 other source nodes from another area. The APIRATE controller performs better than the TCP NEWRENO with RED queue in terms of throughput, fairness index, RTT, window size, Queue size and loss percentage. The APIRATE controller has successfully managed to share the bottleneck link fairly among all the vehicles.

5.3.1.2 Large Static Network

We now carry out the same evaluation as the small network for a large static network (Section 3.4.2.2) and compare the performance of PIRATE with the TCP NEWRENO.

a) Throughput:

Figure 5.8 compares the instantaneous throughput between the PI rate controller (Figure 5.8b) and the NEWRENO (Figure 5.8c) for one of the TCP flows. Also shown in the figure is throughput of uncontrolled UDP traffic. While using the PI rate controller, the application throughput fluctuates with a very small amplitude. Furthermore, the PI algorithm is able to control the source transmission rate based on available bandwidth for controlled traffic. As seen in the figure, when the uncontrolled UDP traffic is increased from 20 Mbps to 40 Mbps, the PI rate controller is able to reduce the source transmission rate of the controlled TCP traffic. When the UDP traffic is increased to 60 Mbps, the TCP traffic is reduce further. When the UDP traffic is reduce to 50 Mbps, the source transmission rate of TCP application is increased. The PI rate controller is able to dynamically adapt to network conditions. In the case of TCP NEWRENO with RED queue, the application throughput fluctuates with a very high amplitude. TCP NEWRENO employs AIMD technique to modify the TCP window this is the main reason for throughput fluctuations.

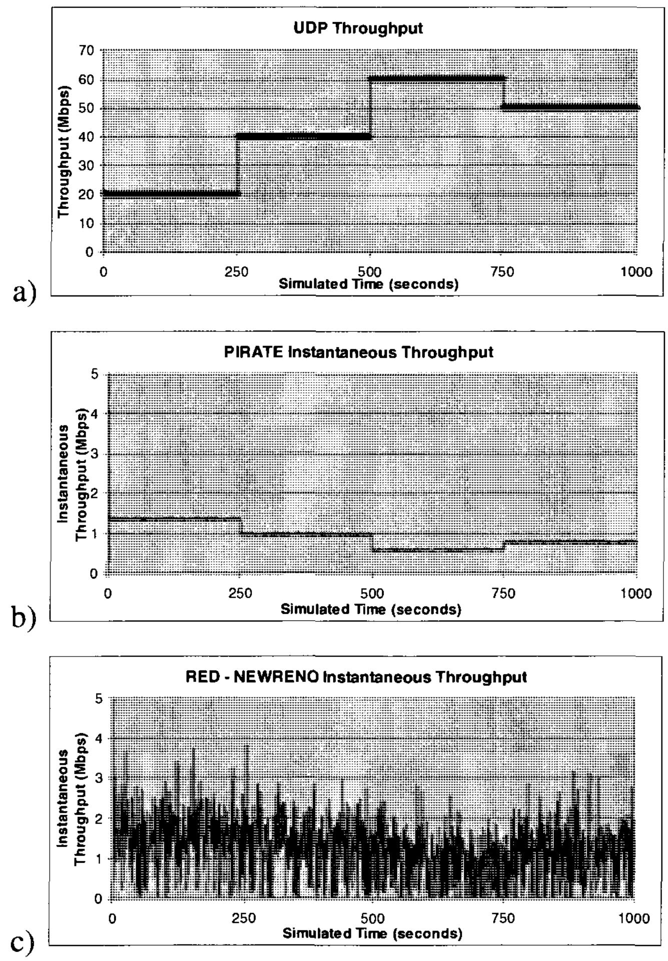


Figure 5.8: Application throughput for a large static network

Figure 5.9 shows the mean node throughput. While using the PI rate controller the throughput among all nodes remains almost same. While using TCP NewReno with RED queue, a throughput difference of 200 Kbps was observed between some nodes.

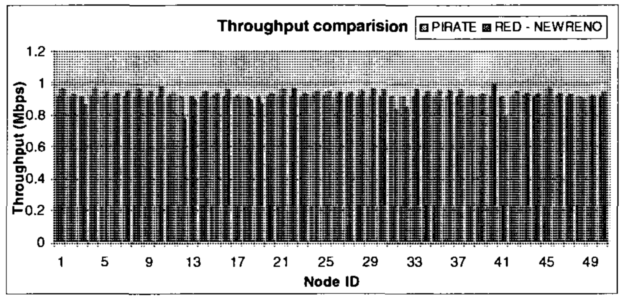


Figure 5.9: Mean throughput by node ID for a large static network

b) Queue Length:

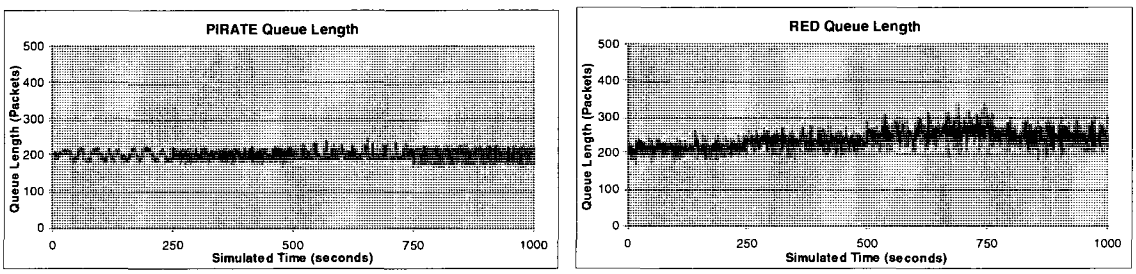


Figure 5.10: Instantaneous queue length behavior for a large static network

Figure 5.10 shows the queue length behavior of congested node (RSU) while using the PI rate controller and the RED queue. The target queue size in the case of PI rate controller is set to 200 packets (40% of queue size). For RED queue, the minimum and maximum queue thresholds are set to 150 and 350 packets respectively. As seen in the figure, the PI rate controller is able to control the queue length under the configured target threshold of 102 (± 0.595) Kbytes even in the presence of uncontrolled UDP traffic. When the UDP traffic is varied, in the presence of the PI rate controller, the average queue length remains close of target threshold. Variations in UDP traffic can be obtained from Figure 5.8. In the case of RED queue, the queue length fluctuates with much higher amplitude between the minimum and maximum threshold with respect to an average of 120 (± 0.704) Kbytes. Another significant observation is that the average queue length increases as the uncontrolled traffic is increased.

Figure 5.11, shows the behavior of queue length when the UDP traffic (*v(t)* in network model) is varied. In the case of PI rate controller, the average queue length is quite stable and close to configured target threshold. In the case of TCP NEWRENO with RED queue, the average queue length increases with increase in uncontrolled UDP traffic. The increase in average queue length also results in increase in RTT.

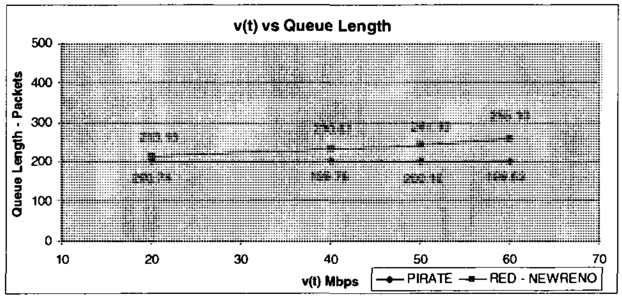


Figure 5.11: Average queue length vs. uncontrolled traffic rate in a large static network

c) <u>TCP Window:</u>

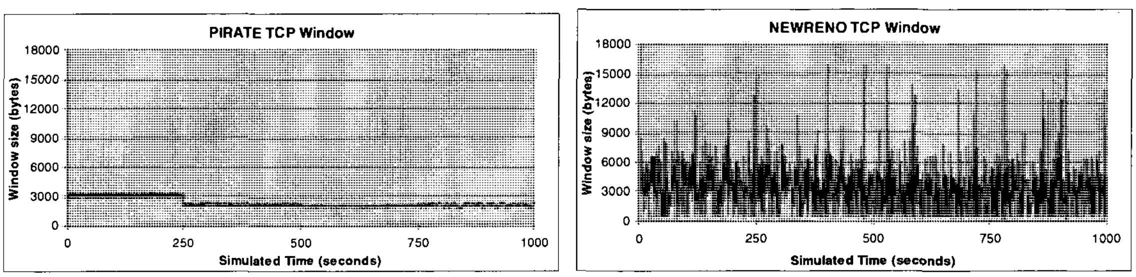


Figure 5.12: Source TCP window behavior for a large static network

Figure 5.12 shows the window size behavior at one of the TCP sources for both the PI rate controller and the TCP NEWRENO with RED queue. In the case of PI rate controller the TCP window fluctuates with a very small amplitude with respect to an average of 2310 (± 10.57) bytes. In the case of TCP NewReno the TCP window fluctuates with a very large amplitude with respect to an average of 3350 (±60.19) bytes. The observed fluctuation is mainly due to the AIMD mechanism adapted by TCP NEWRENO. The APIRATE controller dynamically controls the size of the TCP window based on congestion at the intermediate node. As a result, the average window size is reduced when the uncontrolled traffic is increased. And when the uncontrolled traffic is reduced, the average window size increases. The uncontrolled traffic behavior is shown in Figure 5.8. In the case of TCP NEWRENO with RED queue, TCP window continues to fluctuate while varying the uncontrolled traffic.

d) <u>Round Trip Time (RTT):</u>

Figure 5.13 shows the RTT behavior for PI rate and TCP NewReno. While using the

APIRATE controller the average RTT was observed to be around 17.934 (± 0.0089) milliseconds. In the case of TCP NewReno, the average RTT was around 19.579 (±0.0847) milliseconds. The average RTT in the case of TCP NEWRENO was observed to be higher than PI Rate controller. The increase in RTT is due to the increase in queue length at the congested node.

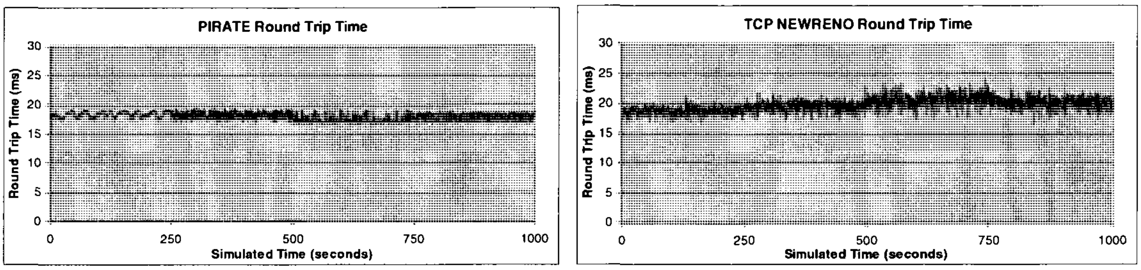


Figure 5.13: RTT behavior for a large static network

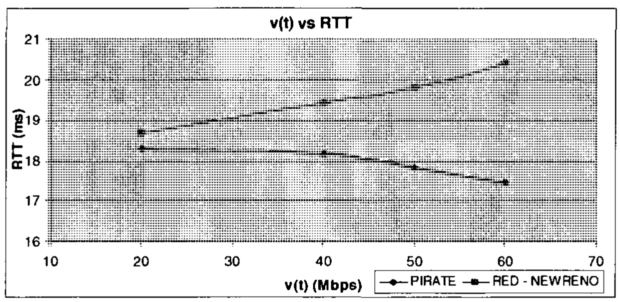


Figure 5.14: Average RTT vs. uncontrolled traffic rate in a large static network

Figure 5.14 shows the RTT behavior at different traffic loads. In the case of TCP NEWRENO with RED queue, as the uncontrolled traffic is increased the average RTT increases. The reason for this is the increase in average queue length at the congested node. In the case of PI rate controller, the average RTT reduces with the increase in uncontrolled traffic. This is because, as the uncontrolled traffic is increased, the APIRATE controller reduces the source transmission rate by reducing the TCP window at the source. As a result, the average RTT reduces.

e) Loss percentage:

For PI rate controller, no packets were dropped at the network layer. While using TCP NEWRENO with RED queue, 1.72% of packets transmitted were dropped.

Table 5.5: Performance summary for a static network with 100 vehicles

Large Static Network						
Parameter	**Node ID**	**PIRATE**	**NEWRENO**	**Node ID**	**PIRATE**	**NEWRENO**
Average Throughput (bps)	1	913041	960576	26	913492	934862
	2	913098	927989	27	913117	937035
	3	913283	865563	28	912929	949752
	4	912989	968957	29	913256	960286
	5	912967	953066	30	913135	953759
	6	913580	931389	31	913089	828328
	7	912649	952841	32	912972	843541
	8	913181	961258	33	912611	954570
	9	913182	942827	34	912954	941521
	10	912793	972311	35	913098	950260
	11	913181	934418	36	913245	952925
	12	913422	776497	37	913117	958827
	13	913291	886538	38	913292	917806
	14	913141	941063	39	912964	930247
	15	913216	930203	40	913072	987702
	16	913696	955844	41	912686	783034
	17	912935	917079	42	913429	941548
	18	913400	892293	43	913082	934281
	19	912632	866253	44	913143	930337
	20	913206	930249	45	913200	967478
	21	913427	964231	46	913367	929734
	22	913309	963983	47	913306	926885
	23	913700	932332	48	913322	904198
	24	913108	941236	49	912925	916149
	25	913090	946267	50	913119	940469

Parameter	**PIRATE**	**NEWRENO**
Fairness Index	0.9999	0.9978
Average RTT (milliseconds)	17.934	19.579
Average Window (bytes)	2310.081	3349.865
Average Queue Size (packets)	199.256	235.739
Packets drops at network layer	0	376924

Table 5.5 summarizes the performance of the static network with 100 vehicles. In this table, we have shown the throughput of each destination node receiving data from the 50 source nodes. The APIRATE controller performs better than the TCP NEWRENO with RED queue in terms of throughput, fairness index, RTT, window size, queue size and loss rate. The APIRATE controller has successfully managed to share the bottleneck link equally among all the vehicles.

The performance of the PI rate controller and its comparison to NEWRENO is similar in small and large networks. Irrespective of the size of the network, the PI rate controller has dominated TCP NEWRENO with RED queue in term of various performance measures.

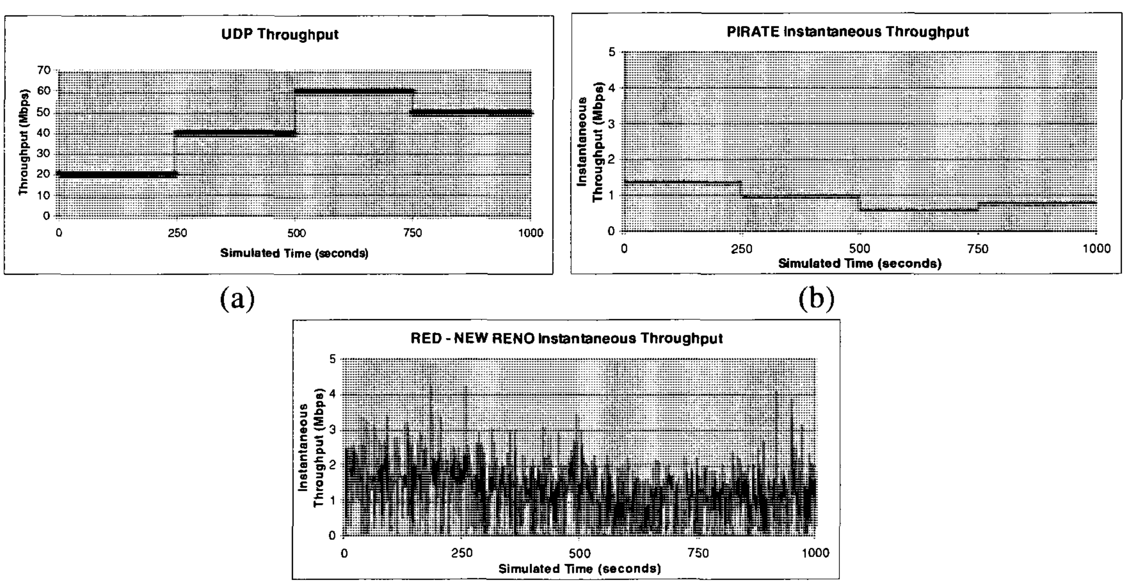
5.3.2 Data Traffic Congestion Control in Mobile VANET

We now study the performance the APIRATE controller when in a mobile VANET in both large city traffic and large rural traffic scenarios. The impact of the mobility characteristics of VANET is of interest to us. We have also simulated a small city traffic and rural traffic scenario with 20 vehicles. Due to similarities of these results with small static network, we have presented the small network mobile scenarios results in the Appendix E.

5.3.2.1 Large City Network

In this scenario, vehicles navigate along the city road network. Details of this scenario are discussed in Section 3.4.3.1. The large city network has been evaluated in terms of throughput, queue length, window size, RTT, and loss percentage.

a) <u>Throughput:</u>



(a) (b) (c)

Figure 5.15: Application throughput for a large city network

Figure 5.15 shows the instantaneous application throughput for one of the TCP flows with city traffic mobility. Also shown in the figure is throughput of UDP traffic. While using the PI rate controller, the application throughput fluctuates with a very small amplitude (Figure

5.15b). The APIRATE controller is able to dynamically adapt to network conditions. In the case of TCP NEWRENO using RED queue (Figure 5.15c), the application throughput fluctuates with a very large amplitude. The main cause of this is that TCP NEWRENO employs the AIMD technique to modify the TCP window at the source. Even with city traffic mobility, one sees that the APIRATE controller is able to provide steady throughout at the source as in the static network scenario.

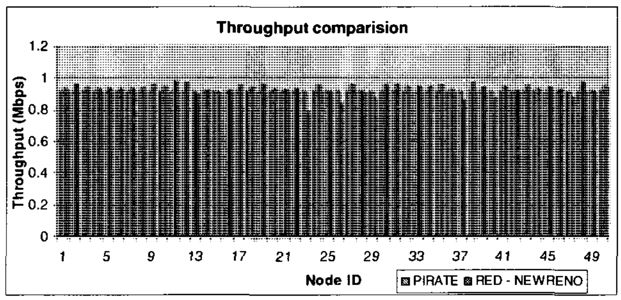


Figure 5.16: Mean throughput by node ID for a large city network

Figure 5.16 shows the mean node throughput. While using the PI rate controller the throughput among all nodes remains almost same. While using TCP NewReno with RED queue, a throughput difference of 200 Kbps was observed between some nodes.

b) Queue Length:

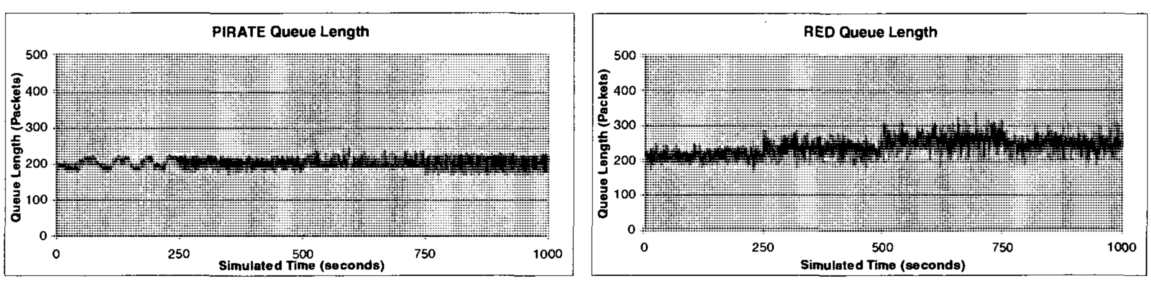


Figure 5.17: Instantaneous queue length behavior for a large city network

Figure 5.17 shows the queue length behavior of congested node (RSU) while using the PI rate controller and the RED queue in the city traffic scenario. The target queue size in the case of PI rate controller is set to 200 packets (40% of queue size). For RED queue, the minimum and maximum queue thresholds are set to 150 and 350 packets respectively. As seen in the figure, with the city traffic mobility and in the presence of uncontrolled traffic, the PI rate controller is able to control the queue length under the target threshold. When the UDP traffic is varied while using PI rate controller, the average queue length remains close of the target threshold as in the static network. Variations in UDP traffic can be obtained from Figure 5.15. In the case of RED queue, the queue length fluctuates with much higher amplitude between the minimum and maximum threshold.

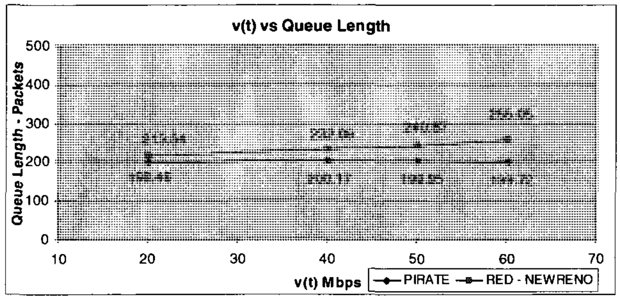


Figure 5.18: Average queue length vs. uncontrolled traffic rate for a large city network

Figure 5.18, shows the behavior of queue length when the UDP traffic (*v(t)* in network model) is varied. In the case of PI rate controller, the average queue length is quite stable and close to configured target threshold (200 packets). In the case of TCP NEWRENO with RED queue, the average queue length increases with increase in uncontrolled UDP traffic.

c) <u>TCP Window:</u>

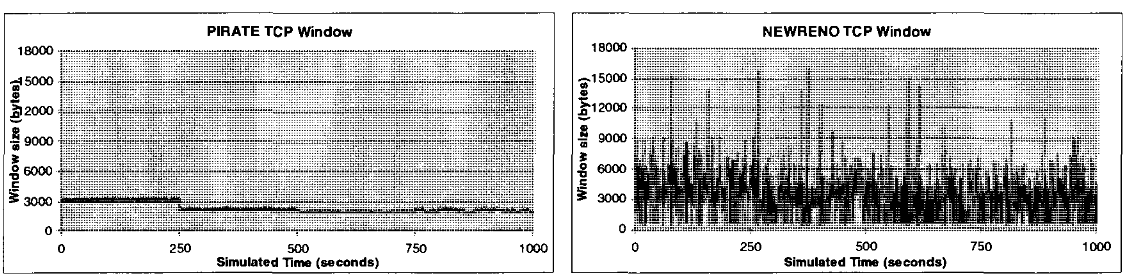


Figure 5.19: Source TCP window behavior for a large city network

Figure 5.19 shows the window size behavior at one of the TCP sources for both the PI rate controller and the TCP NEWRENO with RED queue. In the case of PI rate controller the TCP window fluctuates with a very small amplitude with respect to an average of 2310 (±11.89) bytes. In the case of TCP NewReno the TCP window fluctuates with a very large amplitude with respect to an average of 3352.55 (±139.44) bytes.

d) <u>Round Trip Time (RTT):</u>

Figure 5.20 shows the RTT behavior for the PI rate and the TCP NewReno in a city traffic scenario. The average RTT for PI rate controller was observed to be around 17.96 (±0.0458) milliseconds. In the case of TCP NewReno, the average RTT was around 19.593 (±0.0871) milliseconds. The increase in average queue length RED queue results in increase in RTT. The average RTT in the case of TCP NEWRENO was observed to be higher than PI Rate controller.

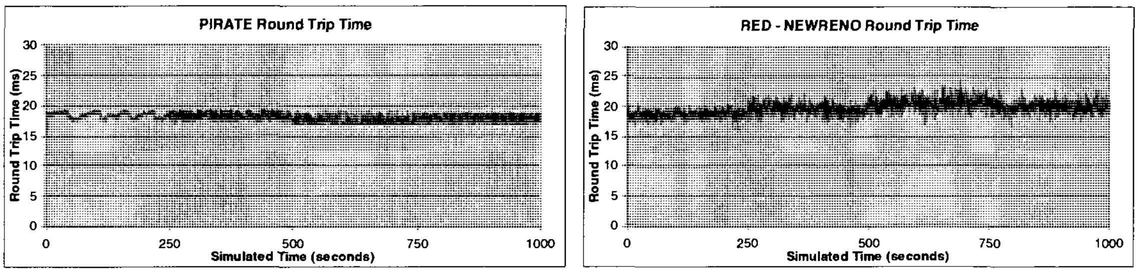


Figure 5.20: RTT behavior for a large city network

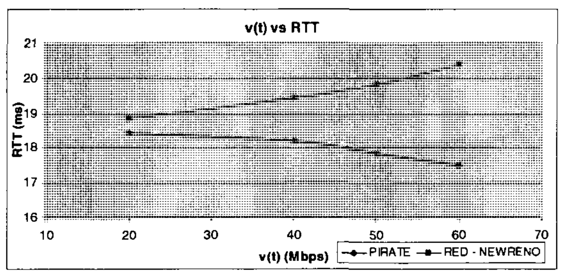


Figure 5.21: Average RTT vs. uncontrolled traffic rate for a large city network

Figure 5.21 shows the impact of uncontrolled traffic on RTT for city traffic scenario. In the case of TCP NEWRENO with RED queue, as the uncontrolled traffic is increased the average RTT increases. For PI rate controller, the average RTT reduces with the increase in uncontrolled traffic, as seen in static network.

e) Loss percentage:

For PI rate controller, no packets were dropped at the network layer. While using TCP NEWRENO with RED queue, 1.71% of packets transmitted were dropped at the congested RSU. The operation of RED is discussed in detail in Section 2.1.1.

Table 5.6 summarizes the performance of the city traffic scenario with 100 vehicles. In this table we have shown the throughput of each destination node receiving data from the 50 source nodes. The APIRATE controller dominates the TCP NEWRENO with RED queue in terms of throughput, fairness index, RTT, window size, queue size, and loss rate. Using the

PI rate controller, the bottleneck link is equally shared among all the source nodes even in the presence of uncontrolled traffic.

Table 5.6: Performance summary for a city network with 100 vehicles

Large City Network						
Parameter	**Node ID**	**PIRATE**	**NEWRENO**	**Node ID**	**PIRATE**	**NEWRENO**
Average Throughput (bps)	1	912852	929246	26	913473	837258
	2	913115	957021	27	912944	957612
	3	913449	940244	28	912974	914099
	4	912865	932262	29	913150	869721
	5	912926	932451	30	913303	951470
	6	913269	929010	31	912805	953172
	7	912907	931561	32	912884	944865
	8	913113	939851	33	912895	945724
	9	913221	953681	34	913097	946411
	10	912497	941327	35	913193	959257
	11	913360	977838	36	913169	927249
	12	913472	969952	37	913105	858280
	13	913634	892702	38	913399	970706
	14	913241	918683	39	913196	938378
	15	912959	907800	40	913172	868252
	16	913557	917745	41	912650	941255
	17	912968	951416	42	913083	916234
	18	913075	944203	43	913167	951405
	19	912560	954120	44	913421	927509
	20	913304	928504	45	913331	935998
	21	913038	924171	46	913159	934211
	22	913257	930779	47	913212	872025
	23	913570	778000	48	913504	968515
	24	913558	950115	49	913408	913236
	25	912972	913044	50	913098	941058

Parameter	**PIRATE**	**NEWRENO**
Fairness Index	0.9999	0.9984
Average RTT (milliseconds)	17.96	19.593
Average Window (bytes)	2310.32	3352.55
Average Queue Size (packets)	200.180	235.074
Packets drops at network layer	0	374118

In this scenario, we have evaluated the performance of PI rate controller is a city traffic scenario. The results obtained in large city traffic scenario are close to the results in large static network scenario. An important parameter that changes with mobility is the RTT. We observed that the increase in RTT from static network to city traffic scenario is very small. As a result, static network and city traffic scenario results are almost same.

Table 5.7: Performance summary for a rural network with 100 vehicles

Large Rural Network						
Parameter	**Node ID**	**PIRATE**	**NEWRENO**	**Node ID**	**PIRATE**	**NEWRENO**
	1	913396	969297	26	913250	943602
	2	913074	966618	27	913099	958637
	3	913490	804483	28	912871	905620
	4	912964	921646	29	912947	875747
	5	913166	930858	30	913334	921023
	6	913667	955702	31	912823	968368
	7	912800	988937	32	912868	909268
	8	913255	950833	33	912865	901445
	9	913090	947431	34	913102	952022
	10	913009	955787	35	913296	940688
	11	913251	946094	36	913458	981486
	12	913274	964629	37	912949	967739
Average Throughput (bps)	13	913233	965586	38	913456	979009
	14	913291	973001	39	912902	934028
	15	913090	899402	40	913201	954543
	16	913641	924259	41	912948	967892
	17	913016	840637	42	913076	680821
	18	913168	909927	43	913239	939072
	19	912943	870829	44	913131	939076
	20	913188	960679	45	912993	950907
	21	913248	948169	46	913278	953092
	22	913118	941083	47	912972	919678
	23	913400	990562	48	913313	946641
	24	913012	676764	49	912898	963343
	25	913365	925437	50	913105	895243

Parameter	**PIRATE**	**NEWRENO**
Fairness Index	0.9999	0.9955
Average RTT (milliseconds)	17.987	19.601
Average Window (bytes)	2323.06	3298.84
Average Queue Size (packets)	199.995	233.780
Packets drops at network layer	0	374118

5.3.2.2 Large Rural Network

In this section, we evaluate the performance of the algorithm in rural traffic scenario. Details for this scenario have been discussed in Section 3.4.3.2. For this scenario, the network architecture and application configuration is same as the city traffic scenario. Table 5.7 summarizes the performance of the rural traffic scenario with 100 vehicles. As observed for other scenarios, the APIRATE controller dominates the TCP NEWRENO with RED queue in terms of throughput, fairness index, RTT, window size, Queue size, and loss rate. For APIRATE controller there were no packets drops observed in the network layer, while in the case of TCP NewReno with RED queue, 1.7 % of the transmitted packets were dropped. The results obtained for large rural traffic scenario is close to the results obtained in large city traffic scenario, as the RTT variations from city traffic to rural traffic is very small. Detailed performance analysis for the large rural network can be found in Appendix D.

5.4 Concluding Remarks

In this chapter, the simulation environment and configuration settings associated with static and mobile scenarios are discussed. We have compared the performance of the APIRATE controller with the combined performance of TCP NEWRENO and RED queue in static and mobile scenarios. The APIRATE controller has dominated TCP NewReno with RED queue in terms of all the performance metrics discussed in this thesis. The APIRATE controller has successfully addressed the congestion problem irrespective of the size (small or large network) and / or mobility model (rural or city traffic). The propagation delay varies based on the distance of the vehicle from the RSU resulting in RTT variations. Comparing the static and mobile scenarios, the variations seen in RTT was very small, as a result, the performance metrics behaved in similar manner.

Chapter 6
Design Guideline

We have used the APIRATE (Adaptive Proportional Integral rate) controller to address the problem of data traffic and vehicle traffic congestion in vehicular networks. In Chapter 3 we have discussed the algorithm implementation and some design aspects. In this chapter, we have summarized our design experience and presented some guidelines for important issues. For data traffic congestion management the algorithm is implemented by modifying the TCP/IP stack. In the case of vehicle traffic management the APIRATE controller is implemented by adding a new application layer protocol. Important design aspects of each approach have been discussed.

6.1 Vehicle Traffic Congestion Management

6.1.1 Controller Design

Vehicle traffic management using APIRATE controller has been done by adding a new application layer protocol. In order to add the new application we have followed the guidelines mentioned in Qualnet documentation (programmer guide). We have defined two instances of the application, the client and the server. The vehicle characteristics and behavior has been defined in the client side of the application and the RSUs (Roadside Units) characteristics are defined in the server side of the application. The APIRATE controller is implemented in destination RSU. A node in Qualnet can be configured as a vehicle or as a RSU using the application file (scenario_name.app). In the current implementation, the first two positions / lines in the application file are reserved for RSUs.

RSU configuration is done as follows: PIRATE <src_id> <dest_id>

The parameters in RSU configuration have the following meaning:

1. Application type:PIRATE
2. <src_id> refers to node identifier of the source RSU
3. <dest_id> refers to node identifier of the destination RSU.

When a node in configured as a RSU, we also specify the node identifier of the neighboring RSU (destination RSU). For the RSU installed at the entrance, the destination

RSU is the RSU installed at the exit. For the RSU installed at the exit, the destination RSU is the RSU installed at the entrance. By doing this the RSUs will be aware of each other. Vehicle configuration in the application file is done as follows:

PIRATE <node_id> <src_id> <dest_id> <time> <min_speed> <max_speed>

Where:

PIRATE	Is the application type
<node_id>	Refers to the node identifier of the vehicle
<src_id>	Refers to the node identifier of the source RSU
<dest_id>	Refers the node identifier of the destination RSU
<time>	Specifies the time when the node enters the simulation
<min_speed>	Indicates the minimum speed of the vehicle
<max_speed>	Indicates the maximum speed of the vehicle

While configuring the vehicle we also mention the source and destination RSU identifiers. The Single-Lane Traffic Simulation with Multi-Agent System [ZhLi05] car-following model has been used for all the vehicles. All the nodes in the experiment should be configured either as a vehicle or as a RSU to actively take part in the simulation. The server instance of the application is called when the node is configured as a RSU and the client instance of the application is called when the node is configured as a vehicle. The initialization function in each instance initializes the parameters and initiates the simulation process.

6.1.2 Guidelines for Designing the System

In this thesis we are addressing the problem of vehicle traffic congestion seen mainly due to increase in traffic volume. We have chosen a road segment of reasonable length and controlled the number of vehicles on this segment to a target threshold by using the APIRATE controller. RSUs should be installed in the entrance and exit of the road segment and these RSUs should be able to communicate with each other over the back bone network. The scenario should be designed in such a way that when the vehicle is communication range of the RSU it should be able send and receive messages from the RSU. Initially a fixed number of vehicles are defined in the experiment. Once the vehicle goes past the exit, it is moved back to the original position near the source RSU and reused for the experiment. This creates a continuous flow of vehicles at the entrance.

The maximum number of vehicles in the network should not exceed 500 as the program is configured to re-use the vehicle once it crosses the exit. The maximum data rate configured at the vehicle should not exceed 54 Mbps as DSRC supports only a maximum of 54 Mbps. We have used a road segment of 2000 meters in length. If the length of the road segment is increased the target queue threshold should also be increased accordingly. Another factor affecting the target queue threshold is the desired speed of the vehicle. If the desired speed of the vehicle is changed, the target queue threshold should be modified accordingly.

6.2 Data Traffic Congestion Management

6.2.1 Controller Design

The APIRATE controller is implemented in network layer to calculate the desirable source sending rate and to advertise the window size to the source. The algorithm updates the window size in the header of each packet traveling towards the destination. The destination node extracts the newly calculated window size and transmits it to the source in the acknowledgement message. Two new fields, window size and average RTT, are added in the TCP (Transmission Control Protocol) header of each packet to provide the algorithm with necessary information. The RTT samples provided by Qualnet are in 500 milliseconds units. In our simulations, the RTT was observed to be much less than 500 millisecond. Therefore we need to add an additional timestamp field in the TCP header in order to calculate the RTT from the source to the destination to facilitate the proper controller operation.

Table 6.1 describes the TCP header structure as used in Qualnet network modeling software. The new fields added are *rtt*, *window_size* and *timestamp*, to carry the roundtrip time, TCP window size and timestamp as discussed earlier. The source (*th_sport*) and destination ports (*th_dport*) are 16 bit unsigned short integers. These parameters identify the sending and receiving ports. The sequence number and acknowledge number are 32 bit unsigned integers. A single unsigned integer is used to store the 4 bit data offset (*th_off*), 8 bit flag (*th_flags*) and 16 bit TCP window (*th_window*). The remaining four bits of the unsigned integer are unused. The checksum is stored in a 16 bit unsigned short integer (*th_sum*). The urgent pointer (*th_urp*) takes another 16 bit unsigned short integer.

Table 6.1: TCP header structure

Data Type	Variable name	Description
unsigned short	th_sport	Source port
unsigned short	th_dport	Destination port
UInt32	th_seq	Sequence number
UInt32	th_ack	Acknowledgement number
unsigned int	th_x2: 4bits	Unused
	th_off: 4 bits	Data offset
	th_flags: 8 bits	Control Bits
	th_win: 16 bits	TCP window
unsigned short	th_sum	Checksum
unsigned short	th_urp	Urgent Pointer
New field added for PI rate control implementation		
float	rtt	Round trip time
float	window_size	Advertised window size
float	timestamp	Timestamp field

The control field is used to carry the following flags:

TH_FIN : No more data from sender

TH_SYN: Synchronize sequence numbers

TH_RST : Reset the connection

TH_PUSH: Push function

TH_ACK: Acknowledgement field significant

TH_URG: Urgent pointer field significant

The regular TCP header is Qualnet had 5 fields so the data offset field was set to 5 during initialization. In order to add three more fields into the TCP header the data offset was changed from 5 to 8 (5 existing field + 3 new fields).

6.2.2 Guidelines for Designing the System

In a data communication network, congestion occurs when there is a demand for more bandwidth than available capacity. The APIRATE controller deals with this problem by controlling the source sending rate. In order to evaluate the performance of PI rate controller it is necessary to identify suitable congestion scenarios. In this thesis we have introduced congestion at the RSU and studied the performance of the algorithm in static and mobile scenarios. We have used GMSF tool to obtain vehicular mobility patterns. Due to the scalability problem in wireless networks the achieved throughput between two wireless nodes is less than its actual bandwidth capacity. So we have defined wireless links with a fixed bandwidth between the RSU and the vehicles. In practice this could be the case as users might have leased a desired bandwidth to use real-time services like streaming media. While using wireless link with a fixed bandwidth, the distance between vehicle and the infrastructure can vary. As a result, the propagation delay varies based on the distance of the vehicle from the RSU, consequently varying the RTT.

There are certain factors that should be considered while designing the system. The size of the buffer is one among them which should be chosen based on bandwidth delay product. The target buffer size should be chosen to be 30% to 40% of maximum buffer size. When minimum and maximum thresholds are needed, these values should be chosen based on the target queue threshold. While using the APIRATE controller, it is necessary to calculate the number of flows. An efficient mechanism should be used to monitor the number of flows that share the link. If the network has sufficient bandwidth to support all the subscribers, then there will not be any congestion. The sufficiency depends on the number of subscribes, the bandwidth demanded by them and the bandwidth capacity of the link in the network delivering the service. If the aggregate bandwidth demanded by the subscribers is greater than the bandwidth capacity of the bottleneck link in the network, then such a condition will result in congestion. For a congestion condition, the bandwidth capacity of the bottleneck link should be less than the aggregate bandwidth demanded by the subscribers. The number of nodes in the network can be varied based on the requirement. The maximum data rate configured at the vehicle should not exceed 54 Mbps as DSRC supports only a maximum of 54 Mbps.

Chapter 7
Conclusions

In this thesis, we have addressed the problem of data traffic congestion and vehicle traffic congestion seen in VANETs using the APIRATE (Adaptive Proportional Integral rate) controller. This controller was originally proposed to regulate data traffic in heterogeneous networks. We have extended the APIRATE controller to regulate vehicle traffic in the vehicular networks in addition.

In order to deal with the data traffic problem in vehicular networks, we have implemented the algorithm in Qualnet simulation tool. As demonstrated in our evaluation, the algorithm addresses the problem of congestion by controlling the queue length to a target threshold. The controller calculates the desirable source sending rate based on the instantaneous queue length of the congested node and advertises it to the source in the form of window size. The source node tunes its sending rate based on the newly received window size. Based on our simulation evaluation, the PI rate controller is capable of providing a steady throughput at the source avoiding congestion at intermediate nodes as desired by most of the streaming multimedia applications. The performance of the APIRATE controller has been evaluated in both static scenario and mobile scenarios. The controller behavior was investigated in terms of throughput, window size, queue size and RTT. The performance of the PI rate controller was compared with the combined performance of TCP NEWRENO along with RED queue. We observed that the PI rate controller outperformed the RED with TCP NEWRENO. Even with two additional overheads (two new fields added to the header of each packet), the APIRATE controller is ideal to address the problem of data traffic congestion in vehicular networks. The implementation is simple with out requiring complex modifications. Moreover the APIRATE controller has a positive influence on the source throughput.

Another nature of congestion addressed in this thesis is the vehicular traffic congestion. There are various causes for vehicular traffic congestion. One of the main reasons is the increase in the traffic volume above the road capacity. We have used the APIRATE controller to deal with this problem. The algorithm controls the queue length, i.e. number of vehicles on the road, to a target threshold by controlling the rate at which vehicles enter the road segment. In order to implement this algorithm in Qualnet, we have added a new application layer protocol. Both the vehicle and the RSU (Roadside Unit) behavior are defined in this application. We have evaluated the performance of the APIRATE controller in a road segment of a complex network. We have assumed that RSUs are installed at the entrance and the exit of the road segment and are connected to each other over the backbone network. The algorithm running at the destination RSU calculates the desirable source sending rate based on the instantaneous queue length and transmits it to the source RSU over the backbone network. The source RSU utilizes this information to control the rate of vehicles entering the road segment of interest. The performance of the PI rate controller was investigated in terms of queue size, throughput and average trip time. It was observed that the APIRATE controller was able to control the queue length to a target threshold at different vehicular speeds, consequently controlling the traffic volume in the road segment. Our simulation results show the feasibility of using the APIRATE algorithm to deal with vehicle traffic congestion. By avoiding congestion in regions that are subject to vehicle traffic congestion, end-to-end delay can be reduced.

The design and implementation of the APIRATE controller to address two different problems in VANET has been a challenging task. Because of the limitations in Qualnet, we had to calculate the algorithm parameters as desired and to use it in the operation of the algorithm to manage data traffic. It was challenging to use a data traffic controller to perform vehicle traffic management because the algorithm was originally designed for computer communication networks. The design of a car-following model and dynamic vehicular mobility was also tricky. Additional factors like road map, intelligent car-following model etc. should be considered in order to extend the application of this algorithm to perform vehicle traffic management in complex road networks.

7.1 Future Work

To promote TCP for real-time multimedia transmission, the performance of APIRATE controller should be compared with UDP using suitable applications. One should also consider evaluating the performance of the algorithm using DSRC at the physical layer. In the case of vehicle traffic congestion management, one could study the performance of the algorithm in more complex road networks with multiple lanes. Interference from other communicating vehicles should be taken into account. Alternatives for vehicles that are not equipped with wireless device should be considered. Significant research still needs to be done to achieve complex dynamic vehicular mobility in Qualnet. In future, one could also work towards combining the vehicle traffic and data traffic controller designed in this thesis.

References

[AlPa99] M. Allman, V. Paxson and W. Stevens, "TCP congestion control", IETF RFC 2581, April 1999.

[AsGo95] K. J. Astrom, G. C. Goodwin and P. R. Kumar, "Adaptive Control, Filtering and Signal Processing", Vol. 74, The IMA Volumes in mathematics and its Applications, Springer-Verlag, April 1995.

[AtLi01] S. Athuraliya, V. H. Li, S. H. low and Q. Yin, "REM: Active Queue Management", *IEEE Network*, Vol. 15, Issue 3, May/June 2001, Pages: 48-53.

[AtPo06] W. Pattara-atikom, P. Pongpaibool and S. Thajchayapong, "Estimating Road Traffic Congestion using Vehicle Velocity", *6th International Conference on ITS Telecommunications*, June 2006, Pages: 1001-1004.

[AwOu02] J. Aweya, M. Ouellette and D. Y. Montuno, "Using TCP rate control for queue input-output rate matching", *International Journal of communication System*, Vol. 15, Issue 10, December 2002, Pages: 907 – 920.

[BaKr97] B. S. Bakshi, P. Krishna, N. H. Vaidya and D. K. Pradhan, "Improving Performance of TCP over Wireless Networks", *17th International Conference on Distributed Computing Systems*, May 1997 Pages: 365 – 373.

[BaLe08] R. Baumann, F. Legendre and P. Sommer, "Generic mobility simulation framework (GMSF)", MobilityModels '08, *1st ACM SIGMOBILE workshop on Mobility models*, May 08, Pages 49-56.

[BaPa97] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links", *IEEE/ACM Transactions on Networking*, Vol. 5, Issue 6, December 1997, Pages: 756 - 769.

[BaSa07] P. G. Balaji, G. Sachdeva, D. Srinivasan and C. Tham, "Multi-agent System based Urban Traffic Management", *IEEE Congress on Evolutionary Computation*, September 2007, Pages:1740 – 1747.

[BrVl91] M. G. M. Brocken and M. J. M. v. Vlist, "Traffic control with variable message signs", *Vehicle Navigation and Information Systems Conference,* 20-23 October 1991, Pages: 27 – 45.

[CaIf94] R. Caceres and L. Iftode, "The Effects of Mobility on Reliable Transport Protocols", *14th International Conference on Distributed Computing Systems*, June 1994, Pages: 12 – 20.

[ChNa05] K. Chen, K. Nahrstedt and N. Vaidya, "The Utility of Explicit Rate-Based flow control in Mobile Ad-hoc Networks", *IEEE Wireless Communication and Networking Conference*, WCBC 2004, Atlanta, Georgia, Vol. 3, March 2004, Pages: 1921-1926.

[CoAg95] J. A. Cobb and P. Agrawal, "Congestion or corruption? A strategy for efficient TCP wireless sessions", *IEEE Symposium on Computers and Communications*, July 1995, Pages: 262-268.

[DoJo07] S. Dornbush and A. Joshi, "StreetSmart Traffic: Discovering and Disseminating Automobile Congestion Using VANET's", *IEEE 65th Vehicular Technology Conference*, April 2007, Pages: 11-15.

[FeSh02] W. Feng, K. Shin, D. kandlur and D. Saha, "The Blue: Active Queue Management Algorithms", *IEEE/ACM Transactions on Networks*, Vol. 10, Issue 4, August 2002, Pages: 513-528.

[FlGu01] S. Floyd, R. Gummadi and S. Shenker, "Adaptive RED: An Algorithm for Increasing the Robustness of RED", *UC Berkeley ICIR Technical Report*, http://www.icir.org/floyd/papers/adaptiveRed.pdf, August 2001.

[FlHa00] S. Floyd, M. Hanley, J. Padhye and J. Widmer, "Equation based congestion control for Unicast applications", *ACM SIGCOMM*, 2000, Pages: 43-56.

[FlHe99] S. Floyd and T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm", IETF RFC 2582, April 1999.

[FlJa93] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", *IEEE/ACM Transactions on Networking*, Vol. 1, Issue 4, August 1993, Pages: 397 – 413.

[FuLu05] Z. Fu, H. Luo, P. Zerfos, S. Lu, L. Zhang, and M. Gerla, "The Impact of multi-hop wireless channel on TCP performance", *IEEE transactions on Mobile Computing*, Vol. 4, Issue 2, March-April 2005, Pages: 209-221.

[Floy03] S. Floyd, "High Speed TCP for Large Congestion Windows", IETF RFC 3649, December 2003.

[GeLo02] M. Gerla, R. Locigno, S. mascolo and W. Weng, "Generalized Window Advertising for TCP Congestion Control", *European Transactions on Telecommunication*, Vol. 13, Issue 6, December 2002, Pages: 549 – 562.

[GeWe00] M. Gerla, W. Weng and R. L. Cigno, "Bandwidth feedback control of TCP and real time sources in the Internet", *IEEE Globecom 2000*, November 2000, Pages: 561 – 565.

[GiEl94] J. F. Gilmore, K. J. Elibiary, and H. Forbes, "Intelligent Control in Traffic Management", *Vehicle Navigation & Information System*, 1994, Pages: 491 - 496.

[HaBe07] M. M. Hassani, R. Berangi and H. Tavakolaie, "An analytical model for evaluating utilization of TCP TAHOE using markovian model", *International Conference on Networking, Architecture, and Storage*, NAS 2007, Pages: 257-258, July 2007.

[HaFl03] M. Hanley, S. Floyd, J. Padhye and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", IETF RFC 3448, January 2003.

[HaMu99] G. Hasegawa, M. Murata and H. Miyahara, "Fairness and Stability of Congestion Control Mechanisms of TCP", *Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, INFOCOM99, March 1999, Pages: 1329 – 1336.

[HeGu06] S. He and W. Guant, "Evaluation of the Effects of Variable Message Signs at Urban Traffic Network", *ICIT 2006, IEEE International Conference on Industrial Technology*, December 2006, Pages: 2955 – 2959.

[HoYa05] Y. Hong and O. W. W. Yang, "Self tuning PI Rate Controller for AQM Router Supporting Best Effort Traffic", *IEEE International conference on communication*, ICC 2005, Seoul Korea, May 2005, Pages: 1539 – 1543.

[HoYa07] Y. Hong and O. W.W. Yang, "Design of Adaptive PI Rate Controller for Best-Effort Traffic in the Internet Based on Phase Margin", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 18, Issue 4, April 2007, Pages: 550-561.

[HuXu03] C. Huang and L. Xu, "SRC: Stable rate Control for Streaming Media", *IEEE Globecom*, San Francisco, U.S.A, December 2003, Pages: 4016 – 4021.

[InSh07] S. Inoue, K. Shozaki and Y. Kakuda, "An Automobile Control Method for Alleviation of Traffic Congestions Using Inter-Vehicle Ad Hoc Communication in Lattice-Like Roads", *2007 IEEE Globecom Workshops*, November 2007, Pages: 1 – 6.

[JaBe05] S. Jaap, M. Bechler and L. Wolf, "Evaluation of Routing Protocols for Vehicular Ad Hoc Networks in City Traffic Scenarios", *5th Int. Conference on Intelligent Transportation Systems Telecommunications* (ITST), 2005.

[JaCh84] R. Jain, D. Chiu and W. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in shared systems", *DEC Technical Report DEC-TR-301*, 1984.

[JaKa88] V. Jacobson and M. J. Karels, "Congestion avoidance and control", *SIGCOMM '88 Workshop*, 1988, Pages: 314-329.

[KaHa94] S. Katsuki and E. Hato, "A Study of Drivers Behavior and Traffic Management", *Vehicle Navigation and Information Systems Conference*, September 1994, Pages: 255 – 258.

[KaHa02] D. Katabi, M. Handley and C. Rohrs, "Congestion Control for High bandwidth Delay Product Networks", *ACM SIGCOMM*, October 2002, Pages: 89 – 102.

[KaKa00] S. Karandikar, S. Kalyanaramanm, P. Bagal and B. Packer, "TCP rate control", *ACM Computer Communications Review*, Vol. 30, January 2000, Pages: 45-58.

[KaKo05] S. Kamijo, H. Koo, X. Liu, K. Fujihira and M. Sakauchi, "Development and Evaluation of Real-time Video Surveillance System on Highway based on Semantic Hierarchy and Decision Surface", *IEEE International Conference on Systems, Man and Cybernetics*, 10-12 October 2005, Pages: 840 – 846.

[KiCh00] S. Kim, S. Choi and C. Kim, "Instantaneous variant of TCP NewReno", *Electronic Letters*, October 1999, Pages: 1818 – 1820.

[KiLe97] K. Kim, J. Lee and B. G. Lee, "Congestion data acquisition using high resolution satellite imagery and frequency analysis techniques", *IEEE International Geoscience and Remote Sensing*, IGARSS '97, Remote Sensing - A Scientific Vision for Sustainable Development, 3-8 August 1997, Pages: 331 – 334.

[KuKn06] A. Kuzmanovic and E. W. Knightly, "TCP-LP: Low-Priority Service via End-Point Congestion Control", *IEEE/ACM Transactions on Networking*, August 2006, Pages: 739 – 752.

[KuSr04] S. Kunniyur and R. Srikant, "An adaptive virtual queue (AVQ) algorithms for active queue management", *IEEE/ACM Transactions on Networking*, Vol. 12, Issue 2, April 2004, Pages: 286-299.

[LiLe07] Y. Li, D. Leith and R. N. Shorten, "Experimental Evaluation of TCP Protocols for High-Speed Networks", *IEEE/ACM Transactions on Networking*, October 2007, Pages: 1109 – 1122.

[LiMo97] D. Lin and R. Morris, "Dynamics of Random Early Detection", *ACM SIGCOMM '97 conference on Applications, technologies, architectures, and protocols for computer communication*, 1997, Pages: 127 – 137.

[LoPa02] S. Low, F. Paganini, J. Wang, S. Adkakha and J. C. Doyle, "Dynamics of TCP/RED and Scalable Control", *IEEE INFOCOM*, June 2002, Pages: 239 – 248.

[MaBo99] M. May, J. Bolot, C. Diot and B. Lyles, "Reasons Not to Deploy RED", *7th International workshop on Quality of Service*, IWQoS, June 1999, Pages: 260 – 262.

[MaSe97] M. Mathis, J. Semke, J. Mahdavi and T. Ott, "The Macroscopic Behavior of TCP Congestion Avoid Algorithm", *ACM Computer Communication Review*, Vol. 27, Issue 3, July 1997.

[MoWa00] J. Mo and J. Walrand, "Fair End-to-End Window-Based Congestion Control", *IEEE/ACM Transactions on Networking*, Vol. 8, Issue 5, October 2000, Pages: 556 – 567.

[OtLa99] T. J. Ott, T. V. Lakshman and L. H. Wong, "SRED: Stabilized RED", *In IEEE / INFOCOM*, 1999, Pages: 1346 – 1355.

[PaKo02] M. Papageorgiou and A. Kotsialos, "Freeway Ramp Metering: An Overview", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 3, Issue 4, December 2002, Pages: 271 – 281.

[PaMa06] N. Parvez, A. Mahanti and C. Williamson, "TCP NewReno: Slow-but-Steady or Impatient?", *IEEE International Conference on Communications*, June 2006, Pages: 716 – 722.

[PaPe07] W. Pattara-atikom and R. Peachavanish, "Estimating Road Traffic Congestion from Cell Dwell Time using Neural Network", *7th International Conference on ITS Telecommunications*, 2007, ITST '07, 6-8 June 2007, Pages: 1 – 6.

[PaPr99] R. Pan, B. Prabhakar and K. Psoynis, "CHOKe: A stateless active queue management scheme for approximating fair bandwidth allocation", *IEEE INFOCOM*, Tel Aviv, March 2000, Pages: 942-951.

[Post81] J. Postel, "Transmission control protocol", IETF RFC 793, September 1981.

[Qual08] Qualnet Software, http://www.scalable-networks.com/, 2008.

[Rowe91] E. Rowe, "The Los Angeles Automated Traffic Surveillance and Control (ATSAC) System", *IEEE Transactions on Vehicular Technology*, February 1991, Pages: 16 – 20.

[SaMa05] K. Sato, Y. Matsui and T. Koita, "Adaptation of Mobile Ad-Hoc Network Protocols for Sensor Networks to Vehicle Control Applications", *SPIE*, Vol. 6041, 2005.

[SmCh95] R. H. Smith and D. C. Chin, "Evaluation of an adaptive traffic control technique with underlying system changes", *Winter Simulation Conference*, 3-6 December 1995, Pages: 1124 – 1130.

[Stal03] W. Stallings, "Data and Computer Communications", 7th Edition, Prentice Hall, 2003.

[Stev94] W. R. Stevens, "TCP/IP Illustrated", Vol. 1, Addison-Wesley, Boston, 1994.

[Stev97] W. R. Stevens, "TCP Slow start, congestion avoidance, fast retransmit and fast recovery algorithms", IETF RFC 2001, September 1997.

[StWr95] W. R. Stevens and G. R. Wright, "TCP/IP Illustrated, Vol. 2: The Implementation", Addison-Wesley, 1995.

[Sumo08] Simulation of Urban MObility, http://sumo.sourceforge.net/, 2008.

[Tanen02] A. S. Tanenbaum, "Computer Networks, Fourth Edition", Prentice Hall, 2002.

[ThPa06] S. Thajchayapong, W. Pattara-atikom, N. Chadil, and C. Mitrpant, "Enhanced Detection of Road Traffic Congestion Areas using Cell Dwell Times", *2006 IEEE Intelligent Transportation Systems Conference*, September 17-20, 2006, Pages: 1084-1089.

[TuQi08] M. Tubaishat, Q. Qi, Y. Shang and H. Shi, "Wireless Sensor-Based Traffic Light Control", *5th IEEE Consumer Communications and Networking Conference*, CCNC 2008, 10-12 January 2008, Pages: 702 – 706.

[WaKu07] Z. Wang, L. Kulik, and K. Ramamohanarao, "Proactive Traffic Merging Strategies for Sensor-Enabled Cars", *Fourth ACM international workshop on Vehicular ad-hoc networks*, VANET'07, September 10, 2007, Pages: 39 – 48.

[WaZh02] F. Wang and Y. Zhang, "Improving TCP performance over Mobile Ad-hoc Networks with out-of-order detection and response", *3rd ACM International Symposium on Mobile Ad-hoc Networking and computing,* MobiHoc 2002, Switzerland, 2002, Pages: 217-225.

[WeAl04] A. Wennström, S. Alfredsson, and A. Brunstrom, "TCP over Wireless Networks", *Technical report, Karlstad University Studies*, ISSN 1403-8099, 2004.

[WeJi06] D. X. Wei, C. Jin, S. H. Low and S. Hegde, "FAST TCP: Motivation, Architecture, Algorithms, Performance", *IEEE/ACM Transactions on Networking*, Vol. 14, December 2006, Pages: 1246 – 1259.

[WeLe07] H. F. Wedde, S. Lehnhoff and B. v. Bonn, "Highly Dynamic and Scalable VANET Routing for Avoiding Traffic Congestions", *Fourth ACM international workshop on Vehicular ad hoc networks*, VANET'07, September 10, 2007, Pages: 81 – 82.

[WiRo05] L. Wischhof and H. Rohling, "Congestion Control in Vehicular Ad Hoc Networks", *IEEE International Conference on Vehicular Electronics and Safety*, October 2005, Pages: 58- 63.

[XuGe03] K. Xu, M. Gerla, L. Qi and Y. Shu, " Enhancing TCP Fairness in Ad-hoc wireless networks using neighborhood RED", *9th Annual International conference on mobile computing and networking ACM*, Mobicom 2003, San Diego, CA, USA, September 2003, Pages: 16-28.

[YiSh04] Y. Yi and S. Shakkottai, "Hop-by-Hop Congestion Control over Wireless Multi-hop Network", *23rd Annual Joint Conference of the IEEE Computer and communications Societies, Infocom 2004*, Vol. 4, March 2004, Pages: 2548-2558.

[ZaSt07] Y. Zang, L. Stibor, X. Cheng, H. Reumerman, A. Paruzel and A. Barros, "Congestion Control in Wireless Networks for Vehicular Safety Applications", *13th European Wireless Conference*, April 2007.

[ZhJi03] L. Zhao, Y. Jing and G. M. Dimirovski, "Improvement of AIMD algorithm considering multimedia transfer over Internet", *6th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Service*, TELSIKS 2003, October 2003, Pages: 392 – 394.

[ZhLi05] F. Zhang, J. Li, and Q. Zhao, "Single-Lane Traffic Simulation with Multi-Agent System", *8th International IEEE Conference on Intelligent Transportation Systems*, Austria, September 2005, Pages: 56 – 60.

Appendix A
Adaptive PI Rate Controller [HoYa07]

The APIRATE (Adaptive Proportional Integral rate) controller is a rate-based controller derived from classical control theory [HoYa07]. The adaptive nature of the algorithm re-calculates the control parameters to self-tune the controller whenever there is a dramatic change in network traffic. The controller calculates the desirable window size based on instantaneous queue length of the intermediate node and advertises it to the sources.

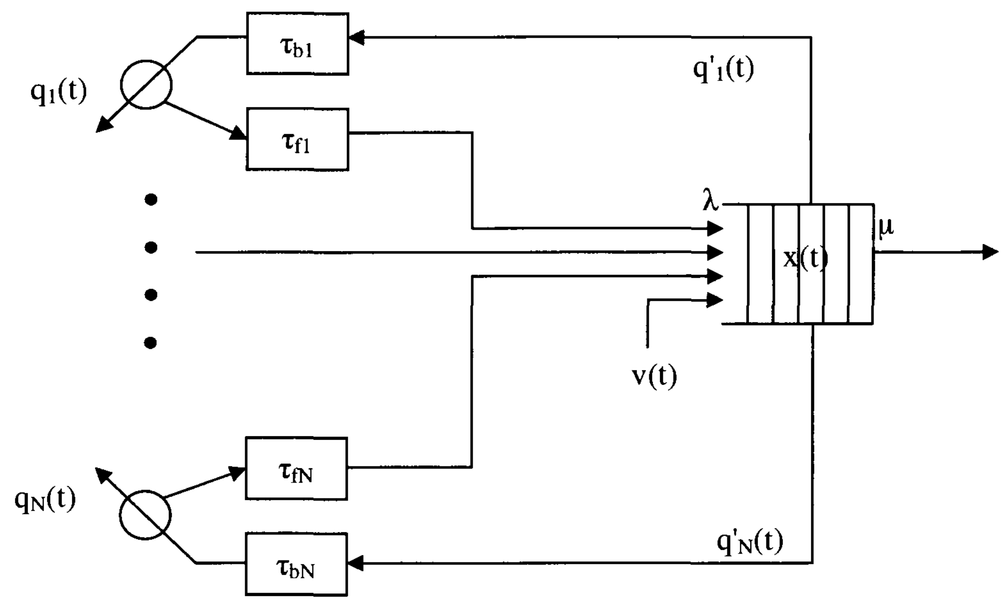


Figure A.1: System model of the intermediate node

As shown in Figure A.1, we consider N TCP flows to an RSU capable of servicing information at μ data packets per second. We also consider an uncontrolled traffic (UDP traffic) using the bottleneck link at a rate $v(t)$. The APIRATE controller is installed in the RSU. Each RSU has a finite buffer K to store the incoming packets serviced at rate μ. The aggregate arrival rate at the congested node in λ. Forward time delay (τ_{fi}), is the delay experienced by the packet while traveling from source node to the RSU. The feedback time delay (τ_{bi}), is the delay experienced from the RSU to the source node via the destination RSU. The roundtrip time $\tau_i = \tau_{fi} + \tau_{bi}$. The APIRATE controller calculates the source transmission rate $q'_i(t)$ based on instantaneous queue length of the congested node and advertises it to the sources using the ACK message so that the source can regulate its current sending rate $q_i(t)$ to provide best effort service traffic.

From the system model, we can see that the queue length $x(t)$ at the congested node is the sum of the N source transmission rates and the uncontrolled traffic minus the service rate.

$$x(t) = \sum_{i=1}^{N} q_i(t - \tau_{fi}) + v(t) - \mu \quad \text{(A1)}$$

Let K_P and K_I be the proportional and integral gain of the PI rate controller. Let x_0 be the target buffer occupancy of the congested node and $e(t)$ be the queue deviation. The source transmission rate $q'_i(t)$ is calculated using the following equation:

$$\begin{aligned} q'_i(t) &= K_P\, e(t) + K_I \int_0^t e(\tau)d\tau \\ &= K_P\,(x_0 - x(t)) + K_I \int_0^t (x_0 - x(\tau))d\tau \quad \text{(A2)} \end{aligned}$$

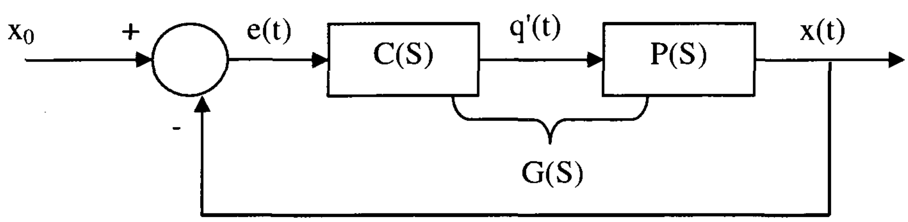


Figure A.2: Control System

We consider the active queue management system shown in Figure A.2 for our controller design. In this system P(s) is the plant and C(s) is the PI rate controller. The open loop transfer function of the active queue management system is defined as follows:

$$G(s) = P(s)\,C(s) = \frac{Ne^{-s\tau}}{s}\left(K_P + \frac{K_I}{s}\right) \quad \text{(A3)}$$

where τ is the average roundtrip time. The control parameters K_P and K_I are calculated using the following equations:

$$K_P = \frac{\sqrt{2+\sqrt{3}}\,(5\pi/12 - \phi_m)}{2N\tau} \quad \text{(A4)}$$

$$K_I = \frac{(5\pi/12 - \phi_m)^2}{2\sqrt{2+\sqrt{3}}\,N\tau^2} \quad \text{(A5)}$$

In equation (4) and (5), ϕ_m is the phase margin. It has been determined in [HoYa07] that the optimal value of phase margin is 45°.

The APIRATE controller also defines a monitoring parameter, $\alpha = N * \tau$, based on number of flows (N) and average round trip time (τ). The control parameter are recalculated if $|\alpha - \alpha_{new}|$ is greater than $\min\{\alpha, \alpha_{new}\}/2$ [HoYa07]. Knowing the roundtrip time, the TCP window size can be derived from source transmission rate. The size of the TCP window is restricted between 1 and maximum size negotiated during the connection setup.

Appendix B
Single-Lane Traffic Simulation with Multi Agent System (Car following model) [ZhLi05]

Realistic mobility pattern is an important aspect of vehicle traffic simulation. While traveling from one point to another, the vehicle should move along the road and should be able to cruise, accelerate or decelerate without colliding with other vehicles. In order to achieve this realistic behavior, we have used the Single-Lane Traffic Simulation with Multi-Agent System car following model presented in [ZhLi05]. In this model, the system (road network system with vehicle, traffic light, road segment etc) is divided into agents like road segment agent, driver vehicle agent, traffic light agent, vehicle dispatching agent etc. From the multi-agent point of view, the framework for single-lane traffic simulation system is derived. In this framework, driver-vehicle agent and road segment agent are most important. The Driver-Vehicle agent presented in [ZhLi05] has been used in this thesis to achieve realistic vehicular mobility.

In single lane traffic simulation, longitudinal movement is what matters. The driver-vehicle agent is the community of a driver and a vehicle. This agent decides in real-time which maneuver (Accelerate, cruise or decelerate) has to be performed in every situation. A longitudinal movement decision tree has been presented in order to achieve this. The driving logic consist of two main steps, the first step is to select a basic maneuver (accelerate, decelerate or cruise) based on the traffic condition. The next step it to calculate the acceleration or deceleration rate. The decision tree presented in Figure B.1 depicts the maneuver selection procedure. v_L is the speed of the leader, v_F is speed of the follower, v_{desire} is the desired speed of the follower, T_P is the time headway the follower prefers to maintain in steady state of the car-following model, and $gap_{desire} = v_F * T_P$. A driver selects a basic maneuver based on the gap and relative speed of leader and follower. Once the basic maneuver is selected, the acceleration or deceleration rate is calculated. Based on the newly calculated speed and speed of the leader, a safe speed (v_{safe}) is calculated in order to avoid collision.

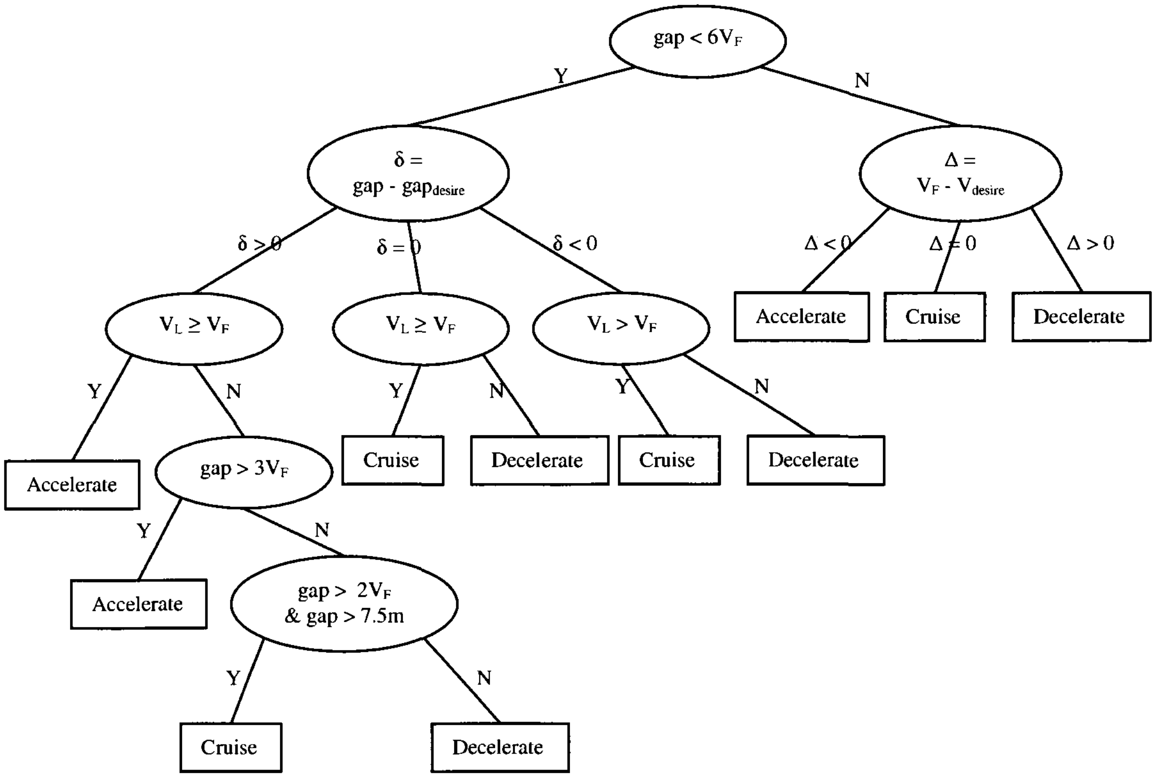


Figure B.1: Decision tree for longitudinal movement

Acceleration: Drivers have a higher acceleration rate at low speeds and lower acceleration rate at high speeds. Acceleration is calculated using following equations:

$$a^{+}(t) = \begin{cases} a_1 = 1.1 m/s^2, & if \quad v(t) \le 12.19 m/s \\ a_2 = 0.37 m/s^2, & if \quad v(t) > 12.19 m/s \end{cases} \tag{B1}$$

Deceleration: A vehicle decelerates in order to achieve the desired gap. The deceleration factor is calculated using the following equation:

$$a_F^{-} = \frac{v_F^2 - v_L^2}{2 * gap} \tag{B2}$$

Collision avoidance condition: In a real-world scenario, reaction delay for vehicles cannot be neglected. If τ is the reaction delay, the safe velocity in order to avoid collisions is calculated using the following equation:

$$v_{safe} = v_L(t) + \frac{gap - v_L(t)\tau}{\tau + \frac{v_F(t) + v_L(t)}{2a_{\max}^{-}}} \quad \text{(B3)}$$

Based on v_{max}, $v(t)+a\Delta t$ and v_{safe}, the ideal velocity for the vehicle is calculated.

Ideal velocity v_{ideal}= min{ v_{max}, $v(t) + a\Delta t$, v_{safe} } (B4)

Where v_{max} is the maximum velocity of the vehicle. The velocity of the vehicle is chosen based on ideal velocity and randomization factor.

$v(t+\Delta t) = \max[0, v_{ideal} - b\eta]$ (B5)

Where b is strength of noise and η is uniform random variable.

Appendix C
Designing Scenarios in Qualnet [Qual08]

The performance analysis of the APIRATE controller to control data traffic and vehicle traffic congestion has been done using the Qualnet simulation tool. Qualnet is a network modeling software that determines the performance of different protocols using simulation. Qualnet provides a comprehensive environment for designing protocols, creating and animating experiments and analyzing the results. Qualnet is composed of six main tools: Qualnet Scenario Designer, Qualnet Animator, Qualnet Analyzer, Qualnet Simulator, Qualnet Tracer, and Qualnet 3D Visualizer. The Qualnet scenario designer is used to design experiments. It includes a comprehensive set of network components, links and several applications which allow users to create desired scenarios as well as set their configuration using click and drag options.

We have used the scenario designer for creating the desired scenarios for both data traffic and vehicle congestion management in VANET (Vehicular Ad-Hoc Network). Once the scenario is created the Qualnet animation tool is used to run the experiments. The animator also provides options to animate various network events. The animator uses the Qualnet Simulator as a processing engine to run a scenario. Initially we used the Qualnet animator to run the experiments, later in order to speed up the execution process we executed each scenario in command line using the command line option of Qualnet simulator. Qualnet Simulator is an advanced discrete-event simulator that enables fast, scalable and hi-fidelity modeling of networks. The command line mode runs faster than the Qualnet animator since there are no graphical events associated and produces the same results, as both utilize the same simulation engine.

The results obtained from simulation are analyzed using the Qualnet Analyzer. The Qualnet Analyzer facilitates graphical analysis of the statistics collected during the simulation. The average statistics provided by Qualnet provides a better understanding of using the communication layer protocols, the parameter values or the network architecture. In order to obtain instantaneous values we have made changes in the Application, Transport and Network Layers. We have used the Microsoft Excel Application to display the instantaneous characteristics of different metrics. The Qualnet Packet Tracer provides per-packet analyses of the simulation. If trace option is enabled in configuration settings, Qualnet generates a trace file which contains in depth information about the life cycle of a packet. The packet tracer displays information about the packet header and the fields in an organized way. The packet tracing tool has been mainly used for debugging purpose. The Qualnet 3D Visualizer is a 3D graphical imaging tool.

Appendix D
Large Mobile Network Results for Rural Traffic Scenario

In this appendix we shall discuss the results obtained for large rural traffic scenario. The behavior for performance metrics is studied at different traffic loads (Section 5.3.2.2).

a) Throughput:

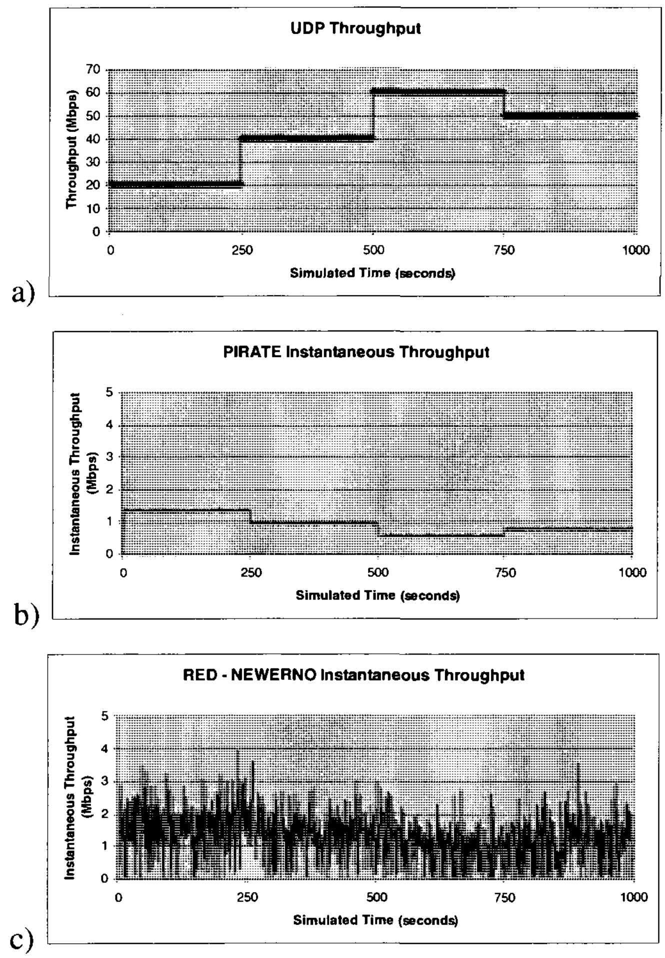


Figure D.1: Application throughput for a large rural network

Figure D.1 shows the instantaneous application throughput for one of the TCP flows. The PI rate controller controls the source transmission rate based on instantaneous queue length of the congested node. As a result, when the uncontrolled traffic is varied the source transmission rate of the TCP flows is controlled accordingly. While using the PI rate controller, the throughput fluctuates with a very small amplitude. In the case of TCP

NEWRENO with RED queue, the application throughput fluctuates with a very high amplitude.

b) Queue Length:

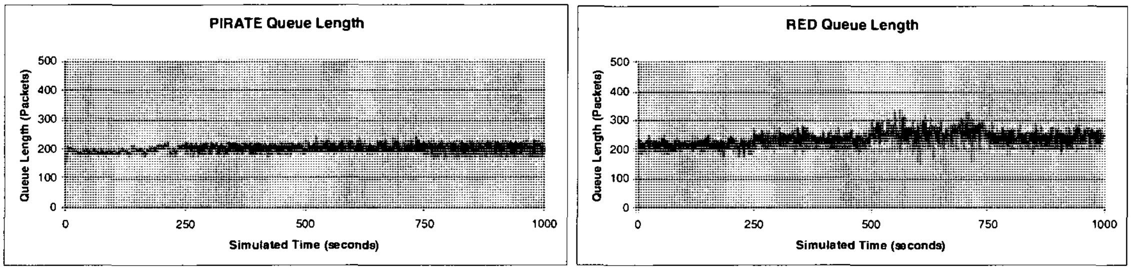


Figure D.2: Instantaneous queue length behavior for a large rural network

Figure D.2 shows the queue length behavior of congested node (RSU) while using the PI rate controller and the RED queue. The PI rate controller is able to control the queue length under the configured target threshold of 102 (± 0.638) Kbytes even in the presence of uncontrolled UDP traffic. In the case of RED queue, the queue length fluctuates with much higher amplitude between the minimum and maximum threshold with respect to an average of 119 (± 0.587) Kbytes. In the case of RED queue, the average queue length increases as the uncontrolled traffic is increased.

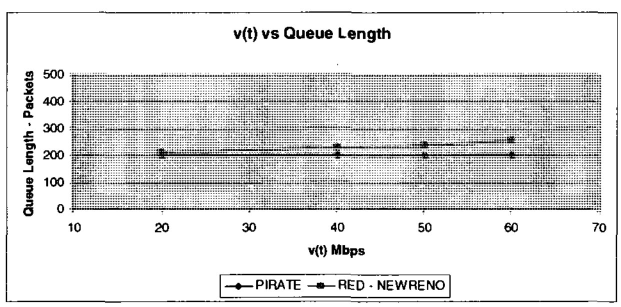


Figure D.3: Average queue length vs. uncontrolled traffic rate for a large rural network

Figure D.3, shows the behavior of queue length when the UDP traffic (*v(t)* in network model) is varied. In the case of PI rate controller, the average queue length is observed to be close to configured target threshold. In the case of TCP NEWRENO with RED queue, the average queue length increases with increase in uncontrolled UDP traffic. The increase in average queue length also results in increase in RTT.

c) TCP Window:

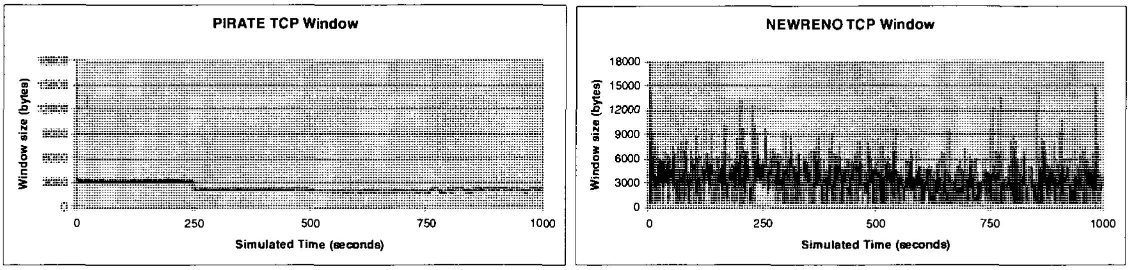


Figure D.4: Source TCP window behavior for a large rural network

Figure D.4 shows the window size behavior at one of the TCP sources for both PI rate controller and TCP NEWRENO with RED queue. In the case of PI rate controller the TCP window fluctuates with a very small amplitude with respect to an average of 2323 (±9.86) bytes. In the case of TCP NewReno the TCP window fluctuates with a very large amplitude with respect to an average of 3298 (±44.32) bytes, as seen in the figure.

d) Round Trip Time (RTT):

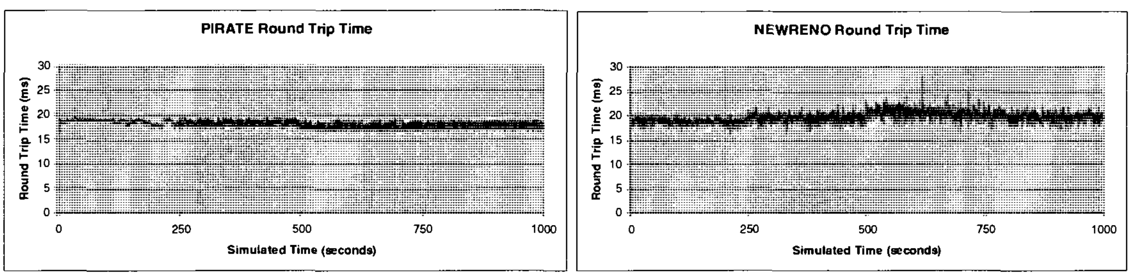


Figure D.5: RTT behavior for a large rural network

Figure D.5 shows the RTT behavior for PI rate and TCP NewReno. While using the APIRATE controller the average RTT was observed to be around 17.987 (± 0.0092) milliseconds. In the case of TCP NewReno, the average RTT was around 19.601 (±0.0648) milliseconds. The average RTT in the case of TCP NEWRENO was observed to be higher than PI Rate controller.

Figure D.6 shows the impact of uncontrolled traffic on RTT. In the case of TCP NEWRENO with RED queue, as the uncontrolled traffic is increased the average RTT increases. The reason for this is the increase in average queue length at the congested node. In the case of PI rate controller, the average RTT reduces with the increase in uncontrolled traffic. As the uncontrolled traffic is increased, the APIRATE controller reduces the source transmission rate by reducing the TCP window at the source. As a result, the average RTT reduces.

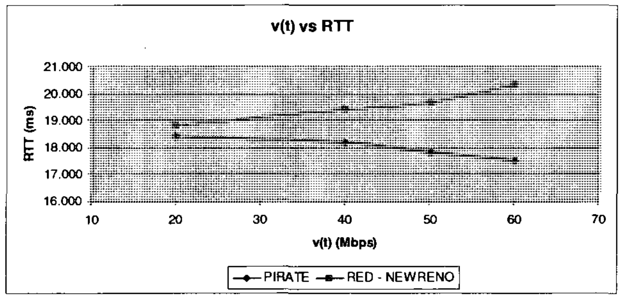


Figure D.6: Average RTT vs. uncontrolled traffic for a large rural network

Appendix E
Small Mobile Network Results for City Traffic and Rural Traffic Scenarios

In this appendix we have presented the results obtained for city traffic small network scenario and rural traffic small network scenario. These scenarios are inherited from the scenario described in Section 3.4.2.1 by adding the city traffic and rural traffic mobility characteristics.

Table E.1: Performance summary for a city network with 20 vehicles

Small City Network			
Parameter	**Node ID**	**PIRATE**	**TCP NEWRENO**
Average Throughput (bps)	1	912882	967184
	2	912907	948815
	3	912971	935061
	4	913146	962293
	5	913192	998077
	6	912970	737999
	7	913185	938214
	8	913000	908375
	9	913197	987845
	10	912930	895457
Fairness Index		0.9999	0.9943
Average RTT (milliseconds)		18.310	19.710
Average Window (bytes)		2321.553	3479.805
Average Queue Size (packets)		39.905	45.542
Packets drops at network layer		0	71547

In Table E.1, we have presented the results obtained for city traffic mobility scenario with 20 vehicles. The road network used for this scenario is shown in Figure 3.13. The APIRATE controller has successfully managed to control the queue length to a target threshold as in other scenarios. Furthermore, the APIRATE controller has dominated TCP NewReno with RED queue in terms of throughput, fairness, RTT, window size and queue size. For TCP NEWRENO with RED queue, 1.64% of the transmitted packets were dropped by the congested node.

Table E.2: Performance summary for a rural network with 20 vehicles

Small Rural Network			
Parameter	**Node ID**	**PIRATE**	**TCP NEWRENO**
Average Throughput (bps)	1	912905	960444
	2	912762	918628
	3	913303	937535
	4	912896	953218
	5	913418	923584
	6	913143	829804
	7	913178	956956
	8	912847	962473
	9	912538	902614
	10	913346	935024
Fairness Index		0.9999	0.9983
Average RTT (milliseconds)		18.308	19.829
Average Window (bytes)		2317.410	3449.900
Average Queue Size (packets)		40.050	45.950
Packets drops at network layer		0	71743

In Table E.2, we have presented the results obtained for rural traffic mobility scenario with 20 vehicles. The road network used for this scenario is shown in Figure 3.15. The behavior of the performance metrics for rural traffic scenario and city traffic scenario in small network was close to the behavior shown in small static network scenario. In all the scenarios, the APIRATE controller has successfully addressed congestion by controlling the source transmission rate and without sacrificing the quality of service. Furthermore, the controller has dominated TCP NewReno with RED queue in terms of all the performance metrics.