Spring 5-31-2009

# Road-based routing in vehicular ad hoc networks

Josiane Nzouonta-Domgang
*New Jersey Institute of Technology*

## ABSTRACT

## ROAD-BASED ROUTING IN VEHICULAR AD HOC NETWORKS

**by**
**Josiane Nzouonta-Domgang**

Vehicular ad hoc networks (VANETs) can provide scalable and cost-effective solutions for applications such as traffic safety, dynamic route planning, and context-aware advertisement using short-range wireless communication. To function properly, these applications require efficient routing protocols. However, existing mobile ad hoc network routing and forwarding approaches have limited performance in VANETs. This dissertation shows that routing protocols which account for VANET-specific characteristics in their designs, such as high density and constrained mobility, can provide good performance for a large spectrum of applications.

This work proposes a novel class of routing protocols as well as three forwarding optimizations for VANETs. The Road-Based using Vehicular Traffic (RBVT) routing is a novel class of routing protocols for VANETs. RBVT protocols leverage real-time vehicular traffic information to create stable road-based paths consisting of successions of road intersections that have, with high probability, network connectivity among them. Evaluations of RBVT protocols working in conjunction with geographical forwarding show delivery rate increases as much as 40% and delay decreases as much as 85% when compared with existing protocols.

Three optimizations are proposed to increase forwarding performance. First, one-hop geographical forwarding is improved using a distributed receiver-based election of next hops, which leads to as much as 3 times higher delivery rates in highly congested networks. Second, theoretical analysis and simulation results demonstrate that the delay in

highly congested networks can be reduced by half by switching from traditional FIFO with Taildrop queuing to LIFO with Frontdrop queuing. Third, nodes can determine suitable times to transmit data across RBVT paths or proactively replace routes before they break using analytical models that accurately predict the expected road-based path durations in VANETs.

# ROAD-BASED ROUTING IN VEHICULAR AD HOC NETWORKS

by
Josiane Nzouonta-Domgang

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Computer Science

Department of Computer Science

May 2009

**APPROVAL PAGE**

**ROAD-BASED ROUTING IN VEHICULAR AD HOC NETWORKS**

**Josiane Nzouonta-Domgang**

04/28/2009

Dr. Cristian M. Borcea, Dissertation Advisor — Date
Assistant Professor of Computer Science, NJIT

4-28-09

Dr. Marvin Nakayama, Committee Member — Date
Associate Professor of Computer Science, NJIT

04/28/09

Dr. Guiling Wang, Committee Member — Date
Assistant Professor of Computer Science, NJIT

4/28/09

Dr. Nirwan Ansari, Committee Member — Date
Professor of Electrical and Computer Engineering, NJIT

04/28/2009

Dr. Teunis Ott, Committee Member — Date
Ott Consulting

04/28/2009

Dr. Constantin Serban, Committee Member — Date
Senior Researcher, Telcordia

## BIOGRAPHICAL SKETCH

**Author:** Josiane Nzouonta-Domgang

**Degree:** Doctor of Philosophy

**Date:** May 2009

**Date of Birth:**

**Place of Birth:**

**Undergraduate and Graduate Education:**

- Master of Science in Computer Science,
  Florida Institute of Technology, Melbourne, FL, USA, 2003
- Diploma of Engineering in Electrical Engineering,
  Universite de Lome, Lome, Togo, 2001

**Major:** Computer Science

**Presentations and Publications:**


Josiane Nzouonta, Neeraj Rajgure, Guiling Wang, and Cristian Borcea, "VANET Routing on City Roads using Real-time Vehicular Traffic Information", *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, 2009.

Oriana Riva, Josiane Nzouonta, and Cristian Borcea, "Context-Aware Fault Tolerance in Migratory Services," *in Proceedings of the 5th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous)*, Dublin, Ireland, July 2008.

Josiane Nzouonta, Teunis Ott, and Cristian Borcea, "Impact of Queue Disciplines on Congestion in Wireless Ad Hoc Networks", *Submitted to Elsevier Performance Evaluation*, November 2008.

Josiane Nzouonta and Cristian Borcea, "STEID: A protocol for emergency information dissemination in vehicular networks," *Technical Report, Computer Science Department, NJIT*, December 2006, http://www.cs.njit.edu/ borcea/papers/steid.pdf.

Josiane Nzouonta and Cristian Borcea, "Emergency information dissemination in vehicular networks," *NJIT Graduate Student Association Research Day,* College of Computing Sciences Research Award, Spring 2006.

Josiane Nzouonta, Marius-Calin Silaghi, and Makoto Yokoo, "Secure Computation for Combinatorial Auctions and Market Exchanges," *in Proceedings 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, New York, NY, USA, pp. 1398-1399, August 2004.

*To my family. I am indebted to you.*

## ACKNOWLEDGMENT

I would like to express my sincere gratitude to all who have supported and encouraged me in various ways through the years. This dissertation would not have been possible without you.

I would like to thank my advisor, Dr. Borcea, for giving me the opportunity to work with him and introducing me to this very exciting field of mobile ad hoc networks and vehicular networks. Dr. Borcea puts great attention to the details, and he constantly encourages all his students to try the very best venues (conferences and journals) for their work. I am grateful for his advisership.

Next, I would like to thank Dr. Ansari, Dr. Nakayama, Dr. Ott, Dr. Serban, and Dr. Wang for accepting to serve as members of my dissertation committee. I am grateful for the time they have dedicated to this task and for all their questions and suggestions for improving the work.

I would like to thank Dr. Wang for her generous advice. Dr. Wang is very accessible to all students and I am very grateful for all the time she took out of her schedule to listen to me and offer valuable advice.

I would also like to thank Dr. Ott for all the encouragement he gave me early on in my PhD studies. It was an honor to collaborate with him on various projects.

I thank Dr. Nakayama for taking the time to answer my numerous questions and brainstorm solutions to various issues that arose while working together with him. I am really grateful for all his help.

I thank Dr. Oria and all the faculty and administration of the Computer Science department at NJIT as well as all my colleagues in the Networking Laboratory for their

intellectual and moral support through the years. It has been a privilege to be a part of this institution and to work together with them. I also thank the National Science Foundation which supported this work through grants CNS-0520033 and CNS-0834585.

I am blessed with a wonderful family who has been there for me, and continues to be there for me as I journey to the end. I thank my parents, for their love and unconditional support. They have always pushed me to work harder and try harder and not to give up. I am indebted to my sisters and my whole family for their continual support. None of this would have been possible without them.

I would also like to thank my friends and all the members of First Baptist Peddie Memorial Church for their support. I especially want to thank Pastor Tingson, Pastor Kang, the Praise and Worship team and Moise for all the help, spiritual, moral and intellectual over the years. There have been ups and there have been downs and the cycle is likely to continue.

To all, thank you for putting up with 'a sinner saved by grace'.

May the Lord Bless you and grant you His Peace.

## TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter | Page

TABLE OF CONTENTS
(Continued)

## TABLE OF CONTENTS
## (Continued)

**Chapter** **Page**

## LIST OF TABLES

## LIST OF FIGURES

## LIST OF FIGURES
## (Continued)

**Figure** **Page**

## LIST OF FIGURES
## (Continued)

**Figure** **Page**

## LIST OF FIGURES
## (Continued)

**Figure** **Page**

# CHAPTER 1

# INTRODUCTION

Wireless ad hoc networks (i.e., decentralized networks created on the fly by hosts located in proximity of one another) are no longer just a research concept. Due to their aptitude to require minimal effort to setup, ad hoc networks are suitable for a wide range of applications, including battlefields communications and disaster recovery operations. In August of 2008, researchers at the National Institute of Standards and Technology (NIST) demonstrated an ad hoc network prototype for first responders in building fires and mines collapse [2]. Unmanned vehicles (aerial, terrestrial, and aquatic) with autonomic operation of a few hours, already can be sent to regions where human presence is deemed dangerous [3, 4], and they can form networks on the fly to report observations to command and control centers. When the hosts (or nodes) of an ad network are mobile, the network is called a mobile ad hoc network (MANET). This dissertation focuses on a subset of MANETs, namely vehicular ad hoc networks (VANETs). The rest of this chapter presents several useful applications of vehicular networks and discusses other vehicles-based network solutions in Section 1.1. Section 1.3 discusses the characteristics of vehicular ad hoc networks and the challenges of routing and forwarding in VANETs. The contributions of this dissertation are presented in Section 1.4 and the contributors to this work are recognized in Section 1.5. Finally, Section 1.6 details the structure of this dissertation.

## 1.1 Vehicular Networks

In recent years, most new vehicles come already equipped with GPS receivers and navigation systems. Car manufacturers such as Ford, GM, and BMW have already announced efforts to include significant computing power inside their cars [5, 6] and Chrysler became the first car manufacturer to include Internet access in a few of its 2009 line of vehicles [7]. This trend is expected to continue and in the near future, the number of vehicles equipped with computing technologies and wireless network interfaces will increase dramatically. These vehicles will be able to run network protocols that will exchange messages for safer, entertainment and more fluid traffic on the roads.

Standardization is already underway for communication to and from vehicles. The Federal Communication Commission (FCC) in the United States has allocated a bandwidth of 75MHz around the 5.9GHz band for vehicle to vehicles and vehicles to road side infrastructure communications through the Dedicated Short Range Communications (DSRC) [8] services.

The emergence of vehicular networks would enable several useful applications, both safety and non-safety related, such as automatic road traffic alerts dissemination, dynamic route planning, service queries (e.g., parking availability), audio and video file sharing between moving vehicles, and context-aware advertisement (e.g., [9, 10, 11]). To deploy these services, three types of communications involving moving vehicles are considered, including cellular network, vehicle to roadside infrastructure and ad hoc vehicle communications. Brief descriptions of each of these types of communication are provided below. Note that hybrids means of communication involving combinations of the methods described here can also be used [12, 13].

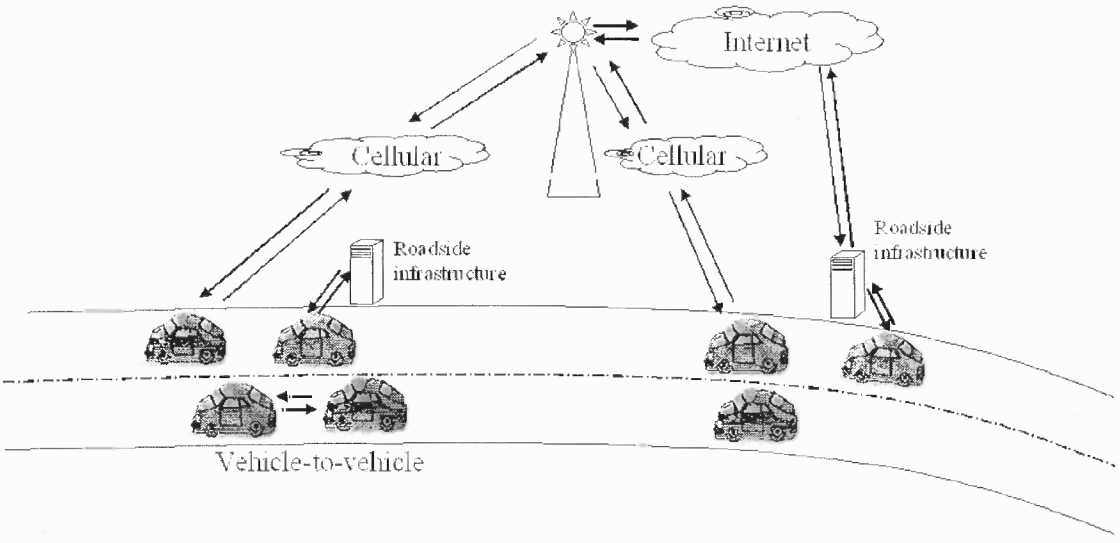



**Figure 1.1** Vehicular networks can be formed in three ways: using cellular network, roadside infrastructure or vehicle-to-vehicle communications.

### 1.1.1 Communications through Cellular Network

The first method connects vehicles to the Internet through cellular data networks using any of the following technologies: EV-DO, 3G, GPRS, etc. [14, 15]. This service is already commercially available from car manufacturers [7] and from other third-parties [16]. In most commercially available solutions, the vehicle is transformed into a IEEE 802.11 (WIFI) hotspot and the Internet connection can be shared by many computers in the car. Usually, a limit is set on the amount of data transfer (e.g., 1GB or 5GB maximum per month). The main advantage of this method of connection is that the vehicle will have Internet access wherever cellular coverage is available. The main drawbacks are the dependence on the cellular operator coverage network and the limited available data rates (rates vary around 500Kbps-800Kpbs) [17].

### 1.1.2 Vehicle to Roadside Infrastructure Communications

The second method uses roadside infrastructure. Here, vehicles connect to other vehicles or to the Internet through roadside access points positioned along the roads. Two main variants can be found in the literature: the access points could be installed specifically for the purpose of providing Internet access to vehicles or the latter could make use of open 802.11 (WiFi) access points encountered opportunistically along city streets [18]. The advantage of this method of connection is that vehicles will be able to connect to the Internet using much higher data rates (e.g., 11Mbps) than through the cellular network. The drawbacks include the cost related to installing access points along the roads to obtain reasonable coverage. Additionally, in the case where open access points are used, the access points owners' consent would legally be required before such a service is deployed [18].

### 1.1.3 Vehicle-to-vehicle (ad hoc) Communications

Using Internet-based communications to and from vehicles will probably remain the method of choice for communications as long as the ratio of WiFi-enabled vehicles remains low. However, the prevalence of WiFi-ready vehicles will open the way for ad hoc networks of moving vehicles [11, 19]. The advantage here is the addition of a distinct, high bandwidth network to the existing infrastructure network. The main drawback is that these networks could require new set of protocols as the viability of vehicular networks applications described above is conditioned by whether or not VANET routing protocols are able to satisfy the throughput and delay requirements of these applications.

## 1.2 Problem Statement

This dissertation addresses the problem of efficient routing and forwarding in VANETs. VANETs were selected for this study because, among the vehicular networks, the ad hoc configuration has the greater potential of widespread use: it is scalable (compared to cellular communication), low-cost, and provides higher bandwidth. Even though VANETs show great promise, their success is dependent on whether VANET routing protocols are able to satisfy the throughput and delay requirements of applications deployed on these networks. Thus, this dissertation aims to answer questions such as: Do existing MANET routing protocols work well in VANET? If not, what are the main characteristics of VANETs that influence routing and how can they be incorporated in better protocols? Are current forwarding protocols enough or can they be optimized for VANET characteristics?

## 1.3 Routing and Forwarding Challenges in VANETs

To better understand the challenges brought by VANETs, it is important first to understand the characteristics of these networks.

### 1.3.1 Characteristics of Vehicular Ad Hoc Networks

VANETs are characterized by (a) high node mobility, (b) constrained nodes movements (c) obstacles-heavy deployment fields, and (d) large number of nodes, which all add to the communication challenges. First, vehicles are continually moving along the roads at higher speeds than in a MANET. Thus a VANET will present a continually changing structure, and communication links are expected to be valid for few minutes or seconds. Next, the movements of vehicles are constrained on roads, hence the existing roadmaps put a limit to the topologies available in VANETs, when compared to MANETs. Then, the presence of

high-rise buildings and houses between streets impacts the propagation of wireless waves through reflections and refractions [20, 21]. Finally, VANETs have the potential to contain a very large number of nodes as any vehicle can be part of the network. It is assumed that each vehicle is equipped with a Geographical Positioning System (GPS), digital maps or navigation system and an ad hoc wireless communication device.

### 1.3.2 VANET Routing Challenges

Analyzes of traditional routing protocols for MANETs demonstrated that their performance is poor in VANETs [22, 23]. The main problem with these protocols (e.g., [24, 25]) in VANETs environments is their route instability, which leads to packets drops, increased overhead from route repairs, low delivery ratios and high transmission delays.

An alternative routing approach is offered by geographical routing protocols (e.g., GPSR [26]), which decouple forwarding from the nodes identity; they do not establish routes, but use the position of the destination and the position of the neighbor nodes to forward data. Any node ensuring progress toward the destination can be used for forwarding. Yet, it runs the risk of packets being dropped at dead end streets because no consideration is given to the roads layouts. The question then, is whether integrating VANETs features (road topology, real-time road traffic flow, presence of building, etc.) in the design of routing protocols would lead to better performance. And if so, what is the best way to integrate them?

### 1.3.3 VANET Forwarding Challenges

The characteristics of VANETs also impact the forwarding of packets. Three main forwarding challenges were identified: next hop selection, queuing disciplines, and paths durations.

Protocols such as DSR or GPSR maintain lists of neighbors, which are used to determine the next hop. If the lists are not accurate, the best next hop could be missed, or even worse, a vehicle node which is already out of the transmission range could be chosen. Maintaining up-to-date lists requires frequent "hello" packet broadcasting. Yet, too much broadcasting will result in a large communication overhead. Thus, the question is how to use accurate node positions in the selection of the next hop without incurring too much overhead.

Vehicular ad hoc networks often experience congestion faster than well-designed wired networks, leading to high end-to-end delays and jitter even for moderate traffic. This particularly impacts delay sensitive but loss tolerant applications such as traffic or accident monitoring. The choice of queuing discipline had been shown to impact the performance of data transfers in wired IP networks [27], where TCP was proved to perform better under congestion when routers use FIFO with Frontdrop instead of FIFO with Taildrop or RED [28]. The question then is whether ad hoc networks can achieve better end-to-end delay and jitter with a different queuing discipline.

The final forwarding challenge considered deals with exploiting the knowledge of routing paths duration to improve the performance of RBVT. Often, a node in a vehicular ad hoc network will try to establish a communication path when the destination is unreachable. Other times, the path will be established only to have it break a few seconds later due to the movements of nodes. The questions then are: Can vehicular traffic information be used to accurately estimate the length of connection/disconnection periods between nodes in VANETs? Can these estimates be used to optimize the route selection and data transfers?

## 1.4 Contributions of Dissertation

The focus of this dissertation is how to improve data routing and forwarding in VANETs by incorporating VANETs characteristics into protocol designs. The main contributions of this dissertation are:

- The **Road-Based with Vehicular Traffic (RBVT)** class of routing protocols for VANETs. RBVT protocols leverage real-time vehicular traffic information to create road-based paths consisting of successions of road intersections that have, with high probability, network connectivity among them. Geographical forwarding is used to move packets between intersections on the path. Two variants are designed and implemented: a reactive protocol, RBVT-R, and a proactive protocol, RBVT-P. The main difference from existing approaches is the reduced impact of individual node movements on the routing path sensitivity. This is enabled by the combination of: (1) greater adaptability to network conditions through incorporation of real-time vehicular traffic information, and (2) route stability through road-based routes and geographical forwarding. Evaluations of RBVT protocols working in conjunction with geographical forwarding show delivery rate increases as much as 40% and delay decreases as much as 85% when compared with existing protocols.

- **Distributed multi-criteria receiver-based forwarding technique** for VANETs is a distributed, beaconless forwarding method based on a light modification of the RTS/CTS mechanism in IEEE 802.11 standard. This method enables geographical forwarding without the overhead associated with periodic "hello" messages to maintain accurate neighbor lists With only slight modification, the RTS/CTS messages will enable nodes to self-determine whether they should act as en-route nodes for forwarding

packets. This method improves upon previous approaches by identifying three key parameters that affect the quality of the selected next-hop in VANETs. The parameters are: the distance between the next hop and the destination, the received power level (which could be affected by noise and channel fading) and the distance of the relay to the transmitter. Evaluations show as much as 3 times higher delivery rates in highly congested networks.

- **Theoretical and simulation analysis on the impact of queuing disciplines on end-to-end delay in ad hoc networks.** This analysis focuses on the effects of FIFO vs LIFO and Taildrop vs Frontdrop queuing disciplines on performance. It shows that LIFO with Frontdrop works better in high congestion scenarios while FIFO (especially FIFO with Frontdrop) works better for low data traffic. Simulation evaluations confirmed the analytical results. This analysis is different from previous work in networking, which seldom consider LIFO because of its perceived unfairness. In highly congested networks, delay can be reduced by half by switching from traditional FIFO with Taildrop queuing to LIFO with Frontdrop queuing.

- **Theoretical characterization of RBVT path durations:** The knowledge of expected path durations can be used to decide when to start a data transfer or to generate a new route discovery before the current route is about to break. In this way, the average throughput can be improved and the overhead further reduced. This contribution focuses on theoretical characterization of RBVT path durations in VANETs. Two analytical models are proposed to estimate the lifetime of RBVT paths. The first model, DTMC-CA, is a discrete-time and discrete-space Markov chain model based on the microscopic Cellular Automaton freeway traffic model. Two methods are

employed to significantly reduce the state space of the Markov chain. Evaluations show that DTMC-CA provides a high level of accuracy when compared with simulation results (within few seconds of simulation measurements in most cases). DTMC-CA accuracy is high for short paths while good approximations are provided for longer paths. The predictions obtained can be incorporated in RBVT protocols for improved performance. The second model, Connectivity Window, generalizes DTMC-CA to any microscopic vehicular traffic model by abstracting the details of microscopic traffic movement from the model.

## 1.5 Contributors to this Dissertation

This dissertation has been made with contributions from Neeraj Rajgure, Teunis Ott, Guiling Wang, and Marvin Nakayama. Neeraj Rajgure, who designed RBVT-P, co-authored [1] along with Guiling Wang. Teunis Ott co-authored the study on the impact of queuing disciplines on latency in ad hoc networks [29]. Marvin Nakayama co-authored the study on deriving estimate of path durations with RBVT.

## 1.6 Structure of Dissertation

The remaining of this thesis proposal is structured as follows: chapter 2 reviews related work. Chapter 3 presents the RBVT class of protocols and shows two examples RBVT protocols implemented, a reactive protocol and a proactive protocol. Then chapter 4 describes the MAC layer optimizations which enable a beaconless geographic forwarding. Chapter 5 presents the analytical and simulation study of the impact of queuing disciplines on end-to-end delay in ad hoc networks while the RBVT path durations analysis is presented in chapter 6. Finally, chapter 7 summarizes this thesis and presents future work.

# CHAPTER 2

# RELATED WORK

This chapter presents background and related work literature in the domain of routing in MANETs (section 2.1), routing in VANETs (section 2.2), distributed next hop selection (section 2.3), methods specifically aimed at reducing delay in ad hoc networks (section 2.4) and distribution of path durations in ad hoc networks (section 2.5). The chapter concludes in section 2.6.

## 2.1 Routing in MANET

Routing has been a major research topic in MANETs. DSDV [30] and DSR [25] are protocols that focus on the topology of the network to select the end-to-end communication path. DSDV is a proactive protocol that may work well in small static environments, but does not scale well in larger, dynamic environments in which link information are frequently updated. RBVT-P is a proactive algorithm as well, but unlike DSDV it is not tied to individual nodes. RBVT-P constructs a real-time view of the vehicular traffic on the roads. DSR is a source-based routing that creates routes on-demand. RBVT-R creates routes on-demand, but the two protocols differ in the representation of the routes. In DSR, routes are sequences of nodes, thus leading to frequent route break in VANETs, while in RBVT-R, the routes are sequences of road intersections defining connected road segments.

To improve on traditional node-centric protocols that do not consider the road topology, a few protocols for VANETs [31, 32] exploit the fact that movements of vehicles are constrained on roads to either predict the lifetime of routes in node-centric protocols (and

repair routes before they break) or reduce the number of route breaks by selecting, during the route creation, neighbors moving in the same direction and with a small relative speed. RBVT-R routing differs from these protocols in that the routes are road-based and their main components are the road intersections traversed on the path from source to destination.

Geographical routing protocols, such as GPSR [26], GFG [33], and GOAFR [34], use node positions to route data between end-points. In static ad hoc networks, for which they have been originally designed, they scale well because the only overhead is generated by "hello" messages to update the neighbor lists. However, under high mobility VANETs, the recovery strategies proposed in the literature (when a forwarding node cannot be found) are often based on planar graph traversals, which were shown not to be as effective in VANETs due to radio obstacles, high node mobility, and the fact that vehicle movements are constrained on roads, rather than being uniformly distributed across a region [22].

## 2.2 Routing in VANET

The main concepts of anchor-based routing in sensor networks [35, 36] have been adapted to vehicular networks environments. GSR [22] and SAR [37] integrate the road topologies in routing using those concepts. In these protocols, a source computes the shortest road-based path from its current position to the destination. Similar to RBVT, they include the list of intersections that defines the path from source to destination in the header of each data packet sent by the source. However, [22], [37] do not consider the real-time vehicular traffic, and consequently, they could include empty roads or roads with network partitions. To alleviate this issue, A-STAR [38] modifies GSR by giving preference to streets served by transit buses each time a new intersection is to be added to the source route. The recently introduced CAR [23] protocol finds connected paths between source and destination pairs

considering real-time traffic. CAR uses "guards" added to "hello" messages to reflect the movements of the source and destination nodes on the paths. Gytar [39] does not store the full intersection-based route in the packets. Instead, the selection of the next intersection is made dynamically, each time choosing the next road segment with the best balance of road density and road length.

MDDV [40] and VADD [41] use opportunistic forwarding to transport data from source to destination. VADD uses historic data traffic flow to determine the best route to the destination. MDDV considers the road traffic conditions as well as the number of lanes on each road segment to select the best road-based trajectory to forward data. In both protocols, when no vehicle node can be found by along the forwarding trajectory, a carry-and-forward approach is used. The vehicle node which is unable to transmit the data packet will store it until it finds a more suitable relay. These protocols are well suited for delay-tolerant applications i.e. applications for which the users can tolerate a certain level of delay (up to a minute or more), as long as the data eventually arrives. The RBVT protocols on the other hand provide support for applications that are not delay tolerant. RBVT protocols require that an end to end path exists for data to reach the destination. Under very sparse vehicular traffic, as well as at the early stages of the deployment of wireless technology in vehicles (while many vehicles do not have wireless interfaces), opportunistic forwarding solutions, such as these, will be needed for car-to-car ad hoc communications.

Note that real-life measurements with commercial GPS receivers [42] showed errors in reporting of GPS positions in urban environment. Because RBVT protocols follow paths made of road segments, they are more resilient to vehicle node positions errors of a few meters. The integration of inertial navigation system to GPS receivers is expected to improve the detection and handling of GPS position errors.

## 2.3 Distributed Next-Hop Selection

Previous work on distributed next-hop selection is available in the literature. Receiver-based selection is proposed at the routing layer in [43, 44] and at the MAC layer [45, 46, 47, 48]. In [43], all neighbors of a transmitting node receive the entire packet, but only one neighbor will re-broadcast it. This neighbor is the one which wins a time-based contention phase in which the node closest to the destination is favored. Minimizing the remaining distance to the destination is also the objective of the schemes in [45, 46, 48], which operate at the MAC layer. In [45], the area covered by the sender transmission range is divided into circular regions of different priorities with the nodes in the outermost ring given the highest priority. However, all these methods consider the unit disk assumption, which does not hold in real-life VANETs. RBVT's next hop self-election is capable to work in realistic conditions, where obstacles and noise affect the wireless communication frequently, because it incorporates multiple criteria in the selection of the best next hop (forwarding progress, optimal transmission area and received power).

Multiple criteria receiver-based next hop selection has been described in a general form in [47]. The authors demonstrated that using carefully selected criteria can improve the election of the optimal next hop. However, actual criteria to be used in practice were not defined. In this dissertation, the idea is applied to vehicular networks and criteria to optimize the election of the next hop are defined.

## 2.4 Effect of Queuing Discipline on Delay

Long queuing delays and packet losses are typical signs of congestion in wired networks. These problems are exacerbated in ad hoc networks by the shared medium contention, higher bit-error rates, changing channel quality during data transfers, and potential mobility.

To improve the quality of delay sensitive applications, several solutions that adapt techniques dealing with congestion in wired networks to the conditions encountered in wireless networks have been proposed.

Rate-based approaches [49, 50, 51] reduce the sending rate when congestion is detected. In [49], the authors adapt TFRC (TCP Friendly Rate Control [52]) to wireless networks. With TFRC, the destination host continuously sends estimates of the loss rate to the source host, which adjusts its sending rate to these estimates. The solutions proposed in [51] point out that hop-by-hop techniques are able to react to congestion faster, providing improved performance compared to end-to-end schemes such as [49] and [50]. Indeed, in ad hoc networks, multi-hop communications often use routing protocols such as [24, 25] which perform symmetric routing (the same path is used for communications to and from the source). Thus, the notification packets sent by the destination are likely to experience some delay as well. Furthermore, notification packets increase the amount of data in the network, thus adding to the channel contention [53].

A cross-layer approach presented in [54] proposes adaptations to all the layers of the TCP/IP protocol stack. At the link layer for example, the authors propose adapting the packets' lengths to the current SINR (signal to interference plus noise ratio) to decrease the packet error rate. The optimizations are carried out by each node individually, adapting to wireless link conditions and traffic flows. Cross-layer solutions are attractive in that they provide a holistic approach to the problem and can be applied locally at each node. The disadvantage is the corresponding complexity: a complete overhaul of the protocols on all the layers is needed.

A priority-content approach [55], focused on video transmissions, proposes to use a congestion-distortion scheduler to prioritize certain packets based on their content. For

example, the I and SI frames of an MPEG-4 stream are given priority over B frames because the loss of I or SI frames impacts the decoder significantly compared to the loss of a B frame. Prioritizing certain packets can be accomplished at the original source of the transmission. However, congestion in a multi-hop ad hoc network may appear at any intermediate node which acts as a router for the packet and the quality of the transmission would still suffer. The solution proposed in [56] describes a hybrid Automatic Repeat Request (ARQ) algorithm which combines ARQ and Forward Error Correction (FEC) to improve unicast communications, while the one in [57] presents a connection-oriented unreliable transport protocol which operates like UDP with a congestion control scheme.

Different from all these solutions, our work demonstrates that end-to-end delay and jitter can be improved significantly by simply allowing the nodes to dynamically switch between different queuing disciplines. The decision is made locally as function of the observed traffic load. This solution does not generate network overhead, does not require changes to higher layer protocols, and can work in conjunction with optimizations at other layers.

## 2.5 Paths Duration in Vehicular Ad Hoc Networks

A few continuous and discrete time/space models have been proposed for estimating path duration in mobile ad hoc networks. A discrete-time and discrete-space model for MANET path durations is proposed in [58]. Mobile nodes move on an open plane according to the Random Way-point (RWP) model. The area is divided into hexagonal cells of radius $r$. A link between two nodes, with respective coordinates $(x, y)$ and $(x', y')$, is represented as a vector of the differences of coordinates $(x' - x, y' - y)$. The link vectors are elements of the state space of a Markov chain. Then, assuming independence between consecutive links on

a path, the authors derive an estimate of path duration in the network. A continuous model for the distribution of residual lifetime of link and path duration in MANETs using the RWP model is presented in [59]. By application of Palm's theorem, the authors show that path durations converge to an exponential distribution. The authors also assessed the impact of the assumption of independence between consecutive links and find a weak correlation coefficient between them.

In vehicular networks, link durations and spatial node distributions have been studied through analytical derivations [60, 61]. The authors derived the probability of two vehicles being connected at a time $t$ as well as the distribution of the number of vehicles in communication range. The distribution of duration of one-hop links is also computed. Similar metrics are measured through simulations in [62]. The work in this dissertation focuses on the duration of uninterrupted wireless connectivity for multihop communications in a vehicular networks. The duration of connectivity on a closed loop road obtained through simulations is presented in [63]. The difference with the present work is the analytical method proposed here for deriving those measures.

## 2.6 Chapter Summary

This chapter presented background and related work literature on routing in MANETs and VANETs. Also discussed were techniques proposed to reduce overhead in ad hoc networks through self-election mechanisms. Next, previous work addressing the problem of delay in ad hoc networks was also reviewed. Finally, existing methods to compute connectivity related probabilities were presented.

# CHAPTER 3

# ROAD-BASED ROUTING PROTOCOLS

This chapter presents a class of city-based VANET routing protocols, called RBVT, which leverage real-time vehicular traffic information to create road-based paths consisting of successions of road intersections that have, with high probability, network connectivity among them. Geographical forwarding allows the use of any node present on a road segment to forward packets between two consecutive intersections on the path from source to destination, reducing the path's sensibility to individual node movements. Simulation results show that the RBVT protocols outperform existing protocols in the studied scenarios. In terms of successful data delivery, RBVT-R performed best, with an increase as much as 40% compared to AODV and 30% compared to GSR using the IEEE 802.11 standard. In terms of average delay, RBVT-P performed best, with delays as much as 85% lower than some existing solutions.

The rest of the chapter is organized as follows. Section 3.1 presents the motivation behind RBVT design. Section 3.2 describes RBVT-R, the reactive protocol, and RBVT-P, the proactive protocol. Section 3.3 presents the simulation results. The chapter concludes in Section 3.4.

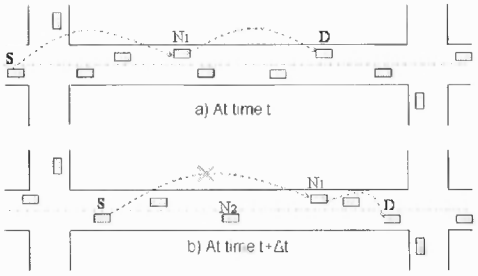
## 3.1 Motivation

Vehicular ad hoc networks (VANETs) are expected to support a large spectrum of mobile distributed applications ranging from traffic alert dissemination and dynamic route planning to context-aware advertisement and file sharing [9, 10, 11, 64, 18]. Considering the large

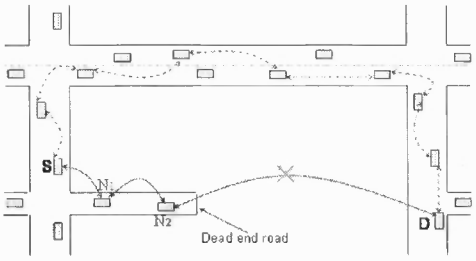
number of nodes participating in these networks and their high mobility, debates still exist about the feasibility of applications using end-to-end multi-hop communication. The main concern is whether the performance of VANET routing protocols can satisfy the throughput and delay requirements of such applications. Since routing performance depends on the type of VANETs (i.e., network characteristics such as node density and average speed differ in cities, highways, and suburban areas), this chapter focuses on VANET routing in city-based scenarios.

Analyzes of traditional routing protocols for mobile ad hoc networks (MANETs) demonstrated that their performance is poor in VANETs [22, 23]. The main problem with these protocols (AODV [24], DSR [25], etc) in VANETs environments is their route instability. The traditional node-centric view of the routes (i.e., an established route is a fixed succession of nodes between the source and destination) leads to frequent broken routes in the presence of VANETs' high mobility, as illustrated in Figure 3.1(a). Consequently, many packets are dropped and the overhead due to route repairs or failure notifications increases significantly, leading to low delivery ratios and high transmission delays.

An alternative approach is offered by geographical routing protocols (e.g., GFG [33], GOAFR [34], GPSR [26]), which decouple forwarding from the nodes identity; they do not establish routes, but use the position of the destination and the position of the neighbor nodes to forward data. Unlike node-centric routing, geographical routing has the advantage that any node ensuring progress toward the destination can be used for forwarding. For instance, in Figure 3.1(a), using geographical forwarding, node N2 could be used instead of N1 to forward data to D. Despite better path stability, geographical forwarding does not perform well in city-based VANETs either [65, 22] Its problem is that many times it cannot




(a) Routes established as fixed successions of nodes break frequently in highly mobile VANETs. Route (S, N1, D) established at time $t$ breaks at time $t + \Delta t$ when N1 moves out of the transmission range of S.




(b) Geographical forwarding can route packets toward dead ends causing unnecessary traffic overhead in the network and longer delays for packets. Instead of forwarding data on the dotted path, geographical forwarding sends data to N1 and N2, following the shortest geographical path from S to D on a dead end road.

**Figure 3.1** Problems with traditional routing approaches in VANETs

find a next hop (i.e., a node closer to the destination than the current node). For example, as shown in Figure 3.1(b), it could take wrong road paths that do not lead to destination. The recovery strategies proposed in the literature are often based on planar graph traversals, which were shown not to be as effective in VANETs due to radio obstacles, high node mobility, and the fact that vehicle movements are constrained on roads rather than being uniformly distributed across a region [22].

A number of road-based routing protocols [22, 37, 65, 23] have been designed to address this issue. However, many of them [22, 37] do not factor in the vehicular traffic flow by using the shortest road path between source and destination. As depicted in Figure 3.2, it is possible that the roads segments on the shortest path are empty (or have network partitions). Other projects [39, 41, 40, 65] try to alleviate this issue by using historical data about average daily/hourly vehicular traffic flows. Unfortunately, historical data are not an accurate indicator of the current road traffic conditions, as events such as road constructions or traffic redirection are not rare.

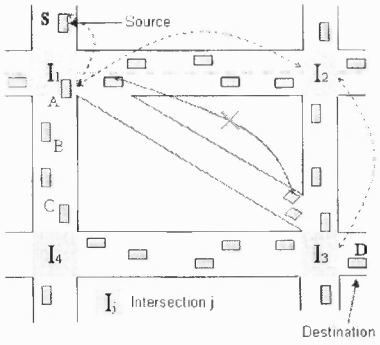



**Figure 3.2** The proposed solution creates a route (S, I1, I2, I3, D) using the road intersections. Since it considers the real-time vehicular traffic, the proposed solution is able to avoid the shorter path (S, I1, I3, D) that would lead to a broken route. Once the road-based route is established, geographical forwarding is used to route data between any two intersections.

## 3.2 RBVT Protocols

The RBVT class of routing protocols leverage real-time vehicular traffic information to create road-based paths. Figure 3.2 shows an example that illustrates the main idea of this class of routing. RBVT presents two main advantages: (1) adaptability to network conditions by incorporating real- time vehicular traffic information, and (2) route stability through road-based routes and geographical forwarding.

RBVT paths can be created on-demand or proactively. Two RBVT protocols were designed and implemented, each illustrating a method of path creation: a reactive protocol, RBVT-R, and a proactive protocol RBVT-P. RBVT-R discovers routes on-demand and includes them in the packets headers (i.e., source routing). RBVT-P on the other hand periodically generates connectivity packets that visit all the connected road segments and stores the graph of the real-time vehicular traffic.

RBVT protocols assume that each vehicle is equipped with a GPS receiver, digital maps (e.g., Tiger Line database [66]), and a navigation system that maps GPS positions on roads. Vehicles exchange packets using short-range wireless interfaces such as IEEE 802.11 [67] and DSRC (Dedicated Short Range Communication)[8].

### 3.2.1 RBVT-R: Reactive Routing Protocol

RBVT-R is a reactive source routing protocol for VANETs that creates road-based paths (or routes) on-demand, using "connected" road segments. A connected road segment is a segment between two adjacent intersections with enough vehicular traffic to ensure network connectivity. These routes, represented as sequences of intersections, are stored in the data packet headers and used by intermediate nodes to geographically forward packets between intersections.

**Route Discovery** When a source node needs to send information to a destination node, RBVT-R initiates a route discovery process, as illustrated in Figure 3.3(a). The source creates a route discovery (RD) packet, whose header includes the address and location of the source, the address of the destination, and a sequence number. It is assumed that nodes have unique IP addresses. RD is flooded in the region around the source to discover a route toward the destination. The flooding is necessary because RBVT-R does not assume a location service that can be queried to find out the location of the destination. For scalability reasons, the flooding region is limited by a TTL value set in the header.

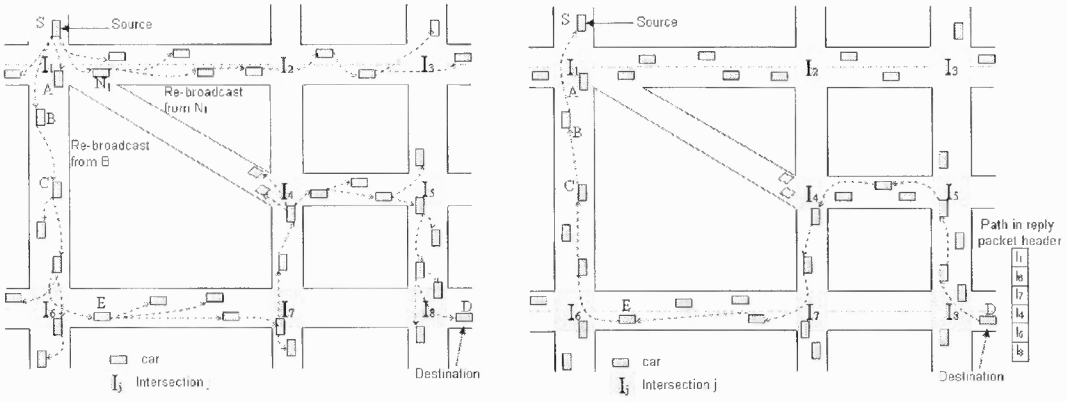
To reduce the effects of the broadcast storm problem [68], RBVT-R uses an improved flooding mechanism [69]. If a node receives an RD packet with the same source address and sequence number with a previously received packet, it discards it. When a node receives a new RD, it does not directly rebroadcast this packet; the node holds the packet for a period of time inversely proportional to the distance between itself and the sending node. Once the waiting period is over, a node re-broadcasts the RD packet only if it did not notice that this packet was re-broadcast by farther-away nodes located on the same road segment. In

this way, farther-away nodes can rebroadcast the request first, thus ensuring faster progress and less traffic in the network.

In RBVT-R, the route is built gradually. Initially, the route structure is an empty list. When a vehicle node receives the RD packet for the first time, it checks if it is located on a different road segment from the transmitter of the packet. If so, the receiving node appends to the route list the road intersections that were "traversed" by the RD packet from the transmitter position.

The route creation process is illustrated on Figure 3.3(a). The source vehicle S creates a RD packet to discover a route to destination D. S adds its own position in the packet and broadcasts it. Both nodes A and B receive the packet on segment $I_1 - I_6$, but only node B will re-broadcast it in the improved flooding mechanism. Before this re-broadcast, node B appends intersection $I_1$ to the route in header of the packet. However, node C will not update the route because it is located on the same road segment with B. A new route update is done when RD is received at node E (intersection $I_6$ is added to the route). This process continues until the packet reaches the destination or the TTL expires.

The RD packet may sometimes be received by nodes on parallel streets. In this case, the RD packet is updated only if the sequence of junctions implicitly traversed can be determined. If this is not possible, the current implementation prevents those vehicles nodes from updating the RD packets. Since the route structure is stored in the header of the RD packet, the number of intersections which can be appended to a route is limited by the size of the IP packet header options and the number of bytes used to identify each road intersection. Techniques such as hierarchical naming of intersections (identifying city, then intersection within city) can increase the maximum number of intersections stored in RD.




(a) A source node uses the improved flooding mechanism to send a route discovery packet in the network to find the destination. The route discovery packet is broadcast along the roads and stores the traversed intersections in its header.

(b) The destination unicasts a route reply packet back to the source. The reply follows the route stored in the route discovery packet, and geographical forwarding is used between intersections.

**Figure 3.3** Route establishment in RBVT-R

**Route Reply** Upon receiving the RD packet, the destination node creates a route reply (RR) packet for the source. The route recorded in the RD header is copied in RR header. As shown in Fig. 3.3(b), this route defines a connected path, composed of road intersections, from source to destination. The destination also adds its current position in the RR header. The RR packet is forwarded along the road segments defined by the intersections stored in its header. Geographical forwarding is used between intersections to take advantage of every available node on the path. The destination may receive duplicates of an RD packet. A new reply is generated only if the newly received packet contains a better quality route. The quality of a route can be expressed using a combination of metrics such as node density on the road segments, the number of lanes, the traffic flow rates, etc. In the current implementation, routes with smaller number of intersections are privileged. Upon receiving the RR packet, the source starts sending data. Each data packet stores the route

in its header and it is geographically forwarded along this route. Protocol 1 presents the pseudo-code for the route discovery and route reply phases.

**Protocol 1** Route Discovery and Route Reply in RBVT-R at node $n_i$

Notation:

$n_S$, $n_D$: ID of the source and destination
$Path$,$TempPath$: The best and temporary paths from $n_S$ to $n_D$
$|Path|$: Path length
$RS(n_i)$: Road segment where node $n_i$ is located
$\alpha$: Waiting time parameter
$RD$: Route discovery packet
$RR$: Route reply packet

Upon receiving $RD(n_S, n_D, TempPath)$ from $n_j$:

**if** $(n_i == n_D)\&(|TempPath| \leq |Path|)$ **then**
    $Path = TempPath$
    Send $RR(n_D, n_S, Path)$
    Return
**end if**
**if** $RD$ not seen before **then**
    **if** $RS(n_i) \neq RS(n_j)$ **then**
        Add $RS(n_i)$ to $TempPath$
    **end if**
    Set timer = $\alpha * distance(n_j, n_i)$
**else**
    **if** $RS(n_i) == RS(n_j)$ **then**
        Cancel timer $/* n_j$ is a better broadcast node$*/$
    **end if**
**end if**

Upon timeout

Broadcast $RD(n_S, n_S, TempPath)$

Upon receiving $RR(n_D, n_S, Path)$ from $n_j$:

**if** $n_i == n_S$ **then**
    Store $Path$
    Forward $Data(Path)$
**else**
    Forward $RR(n_D, n_S, Path)$
**end if**

**Route Maintenance** Existing routes are updated to adapt to the movements of the source and destination over time as well as to repair broken paths. Since sources and destinations

are moving vehicles, the route created during the route discovery phase is not expected to remain constant. A dynamic route updating technique (at the source) keeps the route consistent with the current road segment positions of the source and destination nodes. For example, if node S in Fig. 3.3(b) moves to segment $I_1$-$I_6$, $I_1$ is no longer a valid intersection along the route and should be removed. This change takes place at the source, which also informs the destination of the new path using route update control packets. Similarly, node D may move to the road segment $I_5$-$I_8$. When this happens, $I_8$ should be removed from the list of intersections in the route. Consequently, the destination sends a route update packet to the source. If this update is received at the source, it means the route is valid and it can therefore be used for future data transmissions.

In some situations, the vehicle node may transmit the route update packet before changing road segment. For example, if a vehicle node is about to make a turn which will result in the addition of an intersection to the path, the presence of obstacles may temporarily cause a loss in connectivity [20], which may prevent the successful transmission of the update packet. To avert this problem, vehicle nodes with RBVT-R can transmit the route update packet before the turn to the new segment is complete.

### 3.2.2 RBVT-P: Proactive Road-Based Routing

For a complete description of RBVT protocols, a brief overview of RBVT-P, a proactive routing protocol, is presented. RBVT-P was co-designed and implemented by Neeraj Rajgure; more details can be found in [1, 70]. This chapter will present simulation results for both RBVT-P and RBVT-R.

RBVT-P periodically discovers and disseminates the road-based network topology in order to maintain a relatively consistent view of the network connectivity at each node.

The road-based network topology is constructed using connectivity packets (CPs). CPs are unicast in the network. As they traverse road segments, CPs store their end points (i.e., intersections) in the packet. This traversal method allows CPs to limit most of the overhead associated with common MANET proactive protocols.

Once the network traversal is complete, the topology information in the CP is extracted and stored in a route update (RU) packet that is disseminated to all the nodes in the network (i.e., in the region covered by the CP). The dissemination is made using a controlled flooding process which reduces redundant broadcasts from close nodes. Upon receiving the RU packet, each node locally updates the shortest paths to other road segments in the connected graph. A source node computes the shortest path to the destination using only those road segments that are marked as reachable in its routing table. The sequence of intersections denoting the path is added to the header of each data packet, which is then forwarded geographically along the defined path. Additionally, the header includes the timestamp associated with the route to allow for loose source routing. In case of route break, the intermediate node continues with geographical forwarding until it reaches a node that has fresher information, in which case a new route is stored in the packet header.

## 3.3 Performance Evaluation

This section presents the evaluation of the RBVT protocols using the Network Simulator NS-2.30 [71]. A scenario with obstacles, to model buildings, is used to evaluate the performance. Geographical forwarding is enabled by periodic "hello" messages broadcast by each node to advertise its position. RBVT-R and RBVT-P are compared against four existing VANET/MANET routing protocols. The following presents the evaluation methodology, the metrics used to compare the protocols, and the analysis of the simulation results.

### 3.3.1 Evaluation Methodology

The performance of RBVT protocols is compared against representatives from the main classes of routing protocols: AODV [24], a MANET reactive, distance-vector routing protocol, OLSR [72], a MANET proactive, link-state routing protocol, GPSR [26], a MANET position based routing protocol and GSR [22], a VANET position based routing protocol which takes into account the road layouts in the forwarding decisions. A brief review of how each of these protocols operate follows.

In AODV, a route is created on-demand, when a source vehicle node wants to communicate with a destination node. The route creation involves flooding a route request message and establishing, at each hop, a backward pointer (last transmitter of the request) to the source. A reply is unicasted along this path, using the backward pointers while establishing forward pointers to the destination. In OLSR, each node maintains sets of 1-hop and 2-hops neighbors and selects some neighbors as multipoint relays. OLSR proactively discovers and disseminates link state information over the multipoint relays backbone. Using this topology information, each node computes the next hop to every other node in the network using shortest path hop count forwarding. GPSR is a position based routing protocol which forwards data packets using greedy geographical forwarding from the source node to the destination node. When a node cannot find a neighbor node closer to the destination position than itself, a recovery strategy based on planar graph traversal is applied. In GSR, every vehicle node is equipped with a GPS receiver and holds a digital map of the region. A source vehicle that wishes to communicate with a destination vehicle creates the shortest path based on the roads layout from its position to the destination position. This route is made of a sequence of road intersections. Data packets are forwarded using greedy geographical forwarding along this path. No consideration is given to the vehicular traffic.

### 3.3.2 Metrics

The performance of the routing protocols was evaluated by varying CBR (Constant Bit Rate) data rates, network densities, and numbers of concurrent flows. The metrics used to assess the performance are the following:

- **Average delivery ratio.** This metric is defined as the number of data packets successfully delivered at destinations per number of data packets sent by sources (duplicate packets generated by loss of acknowledgments at the MAC layer are excluded). The average delivery ratio shows the ability of the routing protocol to transfer data successfully end-to-end.

- **Average delay.** This metric is defined as the average delay incurred in the transmission of all data packets delivered successfully. The average delay characterizes the latency generated by the routing approach.

- **Average path length.** This metric is defined as the average number of nodes which participated in the successful forwarding of packets from source to destination. Historically, the average path length was a measure of path quality. This metric to verify if there is a correlation between the path length and average delivery ratio and average delay, respectively.

- **Overhead.** This metric is defined as the number of extra routing packets per number of unique data packets received at destinations. The overhead measures the additional traffic generated by the routing protocol for packets successfully delivered.

### 3.3.3 Simulation Setup

The simulation scenario is a 1500m x 1500m area extracted from the TIGER/Line database of the US Census Bureau [66]. Figure 3.4 shows the map used, which is an area of Los Angeles, California. SUMO, an open-source microscopic, space-continuous and time discrete vehicular traffic generator package [73] is used to generate the movements of the vehicle nodes. SUMO uses a collision-free car-following model [74] to determine the speeds and positions of the vehicles. The map extracted from the Tiger/Line database is input to SUMO along with specifications about the speeds limits and number of lanes

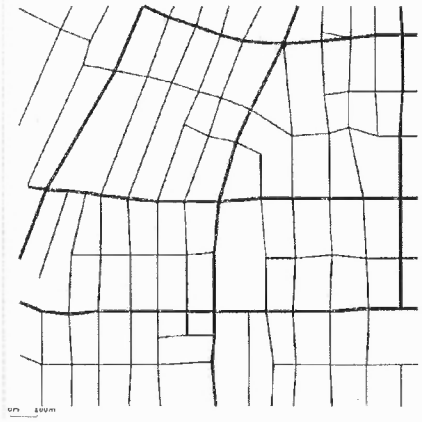


**Figure 3.4** City map used in the first simulation scenario

of each road segment on the map. Traffic lights operated junctions as well as priority junctions are also specified. Less that one-fifth of the junctions are regulated using traffic lights. The first 2000 seconds of SUMO output are discarded to obtain more accurate node movements. This configuration is used with different number of vehicle nodes and the output from SUMO is converted into input files for the movement of nodes in the NS-2 simulator.

For the wireless configuration, the IEEE 802.11 with Distributed Coordination Function (DCF) standard [67] is used at the MAC layer. At the physical layer, the Shadowing propagation model characterizes physical propagation. The communication range is set at 400m with 80% probability of success for transmissions. Some studies [75] have reported real-life measurements between moving vehicles in the range of 450m and 550m. Additionally, while the DSRC standard specifies a range up to 1000m for safety applications, many non-safety applications are expected to reach 400m [8]. The values of path loss exponent $\beta = 3.25$ and deviation $\sigma = 4.0$ are used for the Shadowing propagation [76].

Simulation of buildings in a city environment is done using the following obstacle model. The contour of each street can either be a building wall (of various material) or

| Parameter | Value |
|---|---|
| Simulation area | 1500m x 1500m |
| Number of vehicles | 150-250-350 |
| Number of CBR sources | 1-20 |
| Transmission range | 400m |
| Simulation time | 300s |
| Vehicle velocity | 25 - 55 miles per hour |
| CBR rate | 0.5 - 5 packet per second |
| MAC protocol | IEEE 802.11 DCF |
| Data packet size | 512 bytes |

**Table 3.1** Simulation Setup

an empty area. Thus for each street border, a random signal attenuation value is selected between $0dB$ and $16dB$. For a given pair of transmitter-receiver nodes, the attenuation of the signal at the receiver is computed as follows: first the attenuation from each wall in the direct line of sight between the nodes are summed; then the sum value is added to the attenuation determined through the shadowing propagation model. The signal attenuation values obtained were comparable to values reported from field experiments at 5.3GHz [21]. The simulation parameters are summarized in Table 3.1.

The experiments were conducted in networks of different node densities: 350 nodes scenario is used as a representative of relatively dense networks, 250 nodes scenario for medium density networks and 150 nodes for sparse networks. The implementation of AODV is the one provided by NS-2.30 (with link layer feedback enabled), while the implementation of GSR is based on [22]. GPSR implementation code is taken from [77]

and OLSR implementation code is taken from [78]. To allow the vehicle nodes to have more accurate neighbor information, the hello interval is 0.8 seconds and neighbors are purged from the cache after 1.6 seconds of inactivity. The topology control interval in OLSR was set to 2 sec.

### 3.3.4 Simulation Results

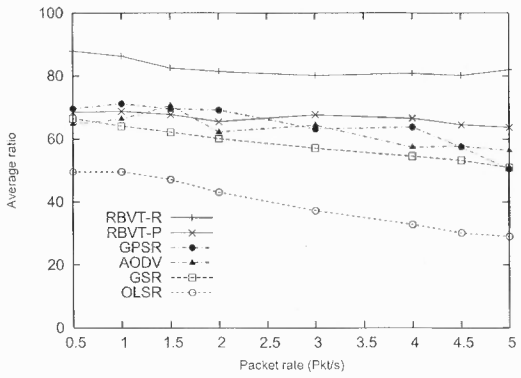
**Average delivery ratio:** Figure 3.5 shows that RBVT-R outperforms the other protocols, with as much as 40% increase compared to AODV and as much as 30% increase compared to GSR. For most cases, a decrease in the average delivery ratio is observed as the data traffic rate increases, for all the protocols. The descending slope is not acute, which means that the protocols are able to contain the increase in number of packets in the network. This behavior can be attributed, in part, to the presence of obstacles on the map area, which limit the level of contention in the wireless network. RBVT-P performs better in medium and dense networks than in sparse networks. The reason is that when the density is small (Figure 3.5(a)), network partitions prevent the CPs from covering large sections of the map, thus limiting the information gathered by the CPs.

Across network densities, the delivery ratio of protocols which integrate road layouts (RBVT protocols and GSR) increases as the network becomes denser. Both RBVT protocols perform better than GSR for all the densities because they integrate real-time knowledge of the vehicular traffic on the roads. When the network is sparse, GSR does not perform as well as some node-centric protocols (Fig. 3.5(a)). However, as the node density increases, the shortest path along the roads map becomes more likely to have enough nodes, thus the increase in average delivery ratio.

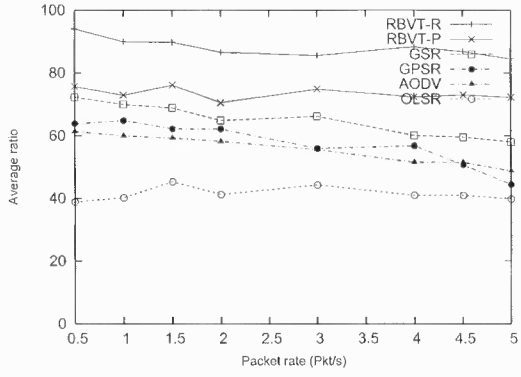
Higher node density does not necessarily mean improved performances for protocols which do not consider the road layouts. For example, in OLSR, the increase in the number of nodes translates into an increase of the link state updates. Two observations can be made on GPSR. First, given that city roads include irregularities such as dead-end streets, following the shortest euclidean distance is not always equivalent to following the shortest path through the roads. Second, the GPSR protocol is stateless and this generally provides many advantages for routing of data packets. However, if a local maxima forms in the network, the stateless nature of the protocol means that packets after packets will follow the same path to the position of the local maxima, and once there, the forwarding mode of each packet will be set to perimeter forwarding. This is unlike protocols that implement feedback mechanisms, such as AODV, which are able to perform a local repair or send a route error notification to the data source node.

**Average delay:** Figure 3.6 shows that RBVT-P has the smallest average delay among the protocols studied. RBVT-P performs better than RBVT-R due to the proactive versus reactive nature of the two protocols. In RBVT-P, the routes already exist at the time of data transmission, while in RBVT-R, route discovery processes are started. Furthermore, the cost of gathering and disseminating routes in RBVT-P is shared among all the data flows, whereas in RBVT-R, each new flow adds its own routing cost. Thus, unlike in MANETs, proactive road-based protocols with real-time traffic awareness can be a viable approach in vehicular networks, especially for delay sensitive applications, such as video streaming.

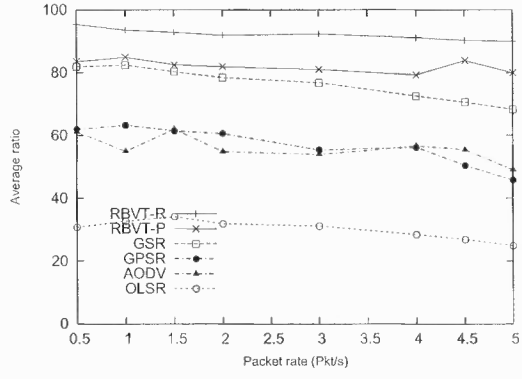
Observe through the graphs of delay that the average delay for RBVT-P consistently remains less than 1 *sec*, from 150 nodes to 350 nodes, while the average delay of RBVT-R decreases consistently with the increase in density. The reason is that RBVT-R routes remain active for longer periods of time (as the number of nodes increases). Thus, less




(a) 150 nodes

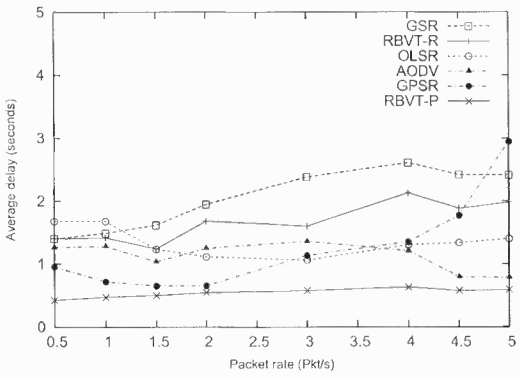



(b) 250 nodes

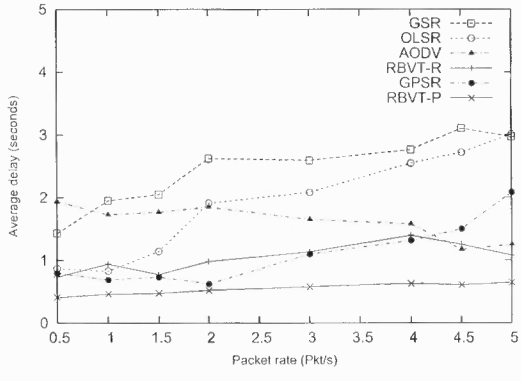



(c) 350 nodes

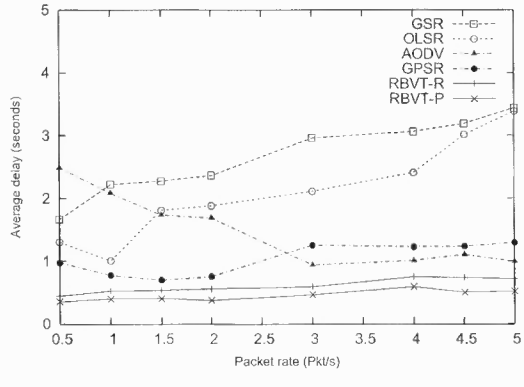
**Figure 3.5** Average Delivery Ratio for RBVT-R, RBVT-P, AODV, OLSR, GPSR and GSR in Networks with Different Node Densities




(a) 150 nodes




(b) 250 nodes



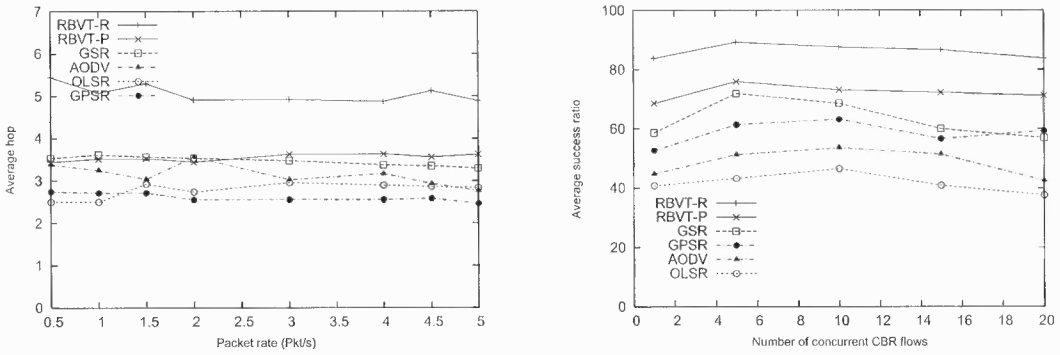

(c) 350 nodes

**Figure 3.6** Average Delay for RBVT-R, RBVT-P, AODV, OLSR, GPSR and GSR in Networks with 15 flows and Different Node Densities

packets need to be buffered because the source is repairing the route. The average delay of GSR on the other hand continually increases. This is because GSR forwards data on road segments selected solely based on the positions of the communication endpoints. A side effect is that some road segments may become congested, but because there is no communication quality feedback sent back to the source vehicle, the overall communication performance suffers. This suggests that altering the paths used in GSR, using feedback from the network, may improve the protocol performances.

**Average path length:** Figure 3.7(a) plots the average path length of packets received at destination for the protocols. This plot is similar for the different network densities and only results with 250 nodes are shown. In general, RBVT-R protocol has longer average paths than the other protocols. There are two reasons for this result: (1) RBVT-R protocol gives preference to link quality over forward progress when selecting the next neighbor node and (2) unlike RBVT-P which consistently select the shortest connected path, a route established with RBVT-R is used until the source considers it broken, even if shorter routes form at a later time. This suggests that RBVT-R could benefit from a method of assessing the quality of the routes used for communications, even when they are not broken. Additionally, note that longer path lengths do not necessarily translate, as expected, into worse performance. On the contrary, selecting better forwarding nodes leads to better performance (RBVT-R has the highest delivery ratio despite having longer paths).

**Impact of number of flows:** The impact of the number of concurrent flows on the protocols performance is shown in Figure 3.7(b). The packet sending rate is 4 packets/second. RBVT protocols perform best in terms of delivery ratio. It is observed that all the protocols scale well to the increase in the number of CBR flows, in this scenario. The drop in performance is small from 1 to 20 CBR pairs. All protocols considered are able to sustain




(a) Average path length

(b) Average Delivery Ratio

**Figure 3.7** Average Path Length for variable data sending rate (a). Average Delivery Ratio and Average Delay with variable number of concurrent flows. The data rate is fixed at 4 packets/second and the network size is 250 nodes (b)

well multiple concurrent flows. Additional experiments compared running M flows with data rate N packet/sec versus running N flows with data rate M packet/sec. The results showed that the performance is slightly better when there are more flows (and lower data rates per flow) as the traffic is distributed more evenly across the network.

## 3.4 Chapter Summary

This chapter presented RBVT, a class of VANET routing protocols for city-based environments that take advantage of the roads topologies to improve the performance of routing in VANETs. RBVT protocols use real-time vehicular traffic information to create road-based paths between end-points. Geographical forwarding is used to find forwarding nodes along the road segments that form these paths. Simulation results showed that the two RBVT protocols, RBVT-R and RBVT-P, outperform existing approaches in terms of average delivery ratio and average delay. Because the RBVT protocols forward data along the streets, not across the streets, and take into account the real traffic on the roads, they perform well

in realistic vehicular environments in which buildings and other road characteristics such as dead end streets are present. These results demonstrate that distributed applications that generate a moderate amount of traffic can be successfully implemented in VANETs. Furthermore, these applications should use RBVT-R when throughput is their main requirement and RBVT-P if they are delay sensitive.

# CHAPTER 4

# VANET FORWARDING USING DISTRIBUTED NEXT-HOP SELF-ELECTION

Initial evaluation of RBVT with an IEEE 802.11 VANET showed that when the wireless medium became congested, the overhead introduced by the periodic "hello" packets used to maintain the list of neighbors severely degraded the end-to-end data transfer performance. To eliminate this overhead, a beaconless, distributed self-election method based on a light modification of the RTS/CTS mechanism in IEEE 802.11 standard is introduced. The three criteria-based technique accounts for non-uniform radio propagations. The next hop election process piggybacks its data on the IEEE 802.11 RTS/CTS frames [67], thus introducing no overhead.

The remainder of this chapter is organized as follows: Section 4.1 motivates the study. A review of the IEEE 802.11 RTS/CTS exchange sequence is presented in Section 4.2. Section 4.3 presents the distributed self-election mechanism and Section 4.4 describes the prioritization function used to self-elect the best next hop. Evaluation results are shown in Section 4.5 and the chapter concludes in Section 4.6.

## 4.1 Motivation

Initial simulations results with RBVT protocols showed that as the network became congested, the overhead traffic from periodic hello messages negatively impacted the end-to-end data transfers. In this Section a solution is proposed to this problem: a distributed next hop election method, which increases significantly the average data delivery ratio by reducing the overhead associated with the selection of the next hop node in congested networks.

In RBVT, geographical forwarding is used to transfer data packets between intersections. In previous works on geographical or position-based forwarding [26, 33, 34], each forwarding node picks the next hop using its list of neighbors and their geographical positions. The next hop is chosen in such a way as to maximize the forwarding progress (e.g., typically, this is the neighbor closest to the destination). This process continues until the packet reaches the destination. Therefore, to successfully choose next hops, it is vital for each en-route node to keep a precise neighbor list. If the lists are not accurate, the best next hop could be missed, or even worse, a node which is already out of the transmission range could be chosen. Maintaining up-to-date lists requires frequent "hello" packet broadcasting. However, this broadcasting results in a large communication overhead.

The solution proposed, to eliminate "hello" packets, is inspired by the receiver-side relay election approaches (e.g., [43, 46, 48]) in ad hoc and sensor networks. In the receiver-based relay selection approaches, the sender broadcasts a control packet informing its neighbors about a pending data packet transmission. Each receiver uses certain criteria to determine if it should elect itself as a next hop candidate, and if so, it computes a waiting time. This waiting time is used to allow better receivers to answer first. If a receiver does not overhear a better candidate before its waiting time expires, it informs the sender that it is the best next hop.

The current implementations of these approaches use one criterion to compute the waiting time, namely the distance between potential next hops and the destination. This method works well under the unit disk assumption [79] (i.e., the transmission range is a circle of a fixed radius). However, previous studies (e.g., [80, 81]) have shown that real wireless radios do not follow the unit disk assumption. This is especially true in vehicular networks where buildings and other obstacles impact radio propagation through

signal fluctuations and fading. In this context, selecting the neighbor that optimizes the forward progress alone does not guarantee an optimal selection of the next hop [47].

The method proposed here accounts for non-uniform radio propagation using two additional criteria, optimal transmission area and received power. Furthermore, the next hop election protocol piggybacks its data on the IEEE 802.11 RTS/CTS frames [67], thus introducing no overhead. To help with the understanding of this protocol, a brief overview of the IEEE RTS/CTS mechanism is presented.

## 4.2 802.11 RTS/CTS Background

In IEEE 802.11 with Distributed Coordination Function (DCF) standard, the RTS (Request-To-Send) and CTS (Clear-to-Send) frames are used to address the hidden terminal problem [82] inherent to wireless communications. This problem and the functionality of RTS/CTS frames are illustrated in Fig. 4.1. In this example, both node S and node C are node B's neighbors. When S is sending a frame to B, C should not send any frame to B; otherwise, there would be a collision at B. However, node C is out of the communication range of node S, and it does not detect a busy channel while S is transmitting. Once node C starts its transmission, a collision happens at B, which cannot be detected by S until after it times out without receiving an acknowledgment from B.

IEEE 802.11 with DCF addresses the hidden terminal problem by deploying the RTS/CTS exchange. Before a node transmits a frame, it sends a very short RTS frame to the intended receiver, including the transmission time of the follow-up data and acknowledgment frames. The receiver broadcasts a CTS message, received by all its neighbors, once it receives the RTS with the needed channel clear time. The neighbors will consequently defer their transmissions until this transmission is completed. In the example, node C will

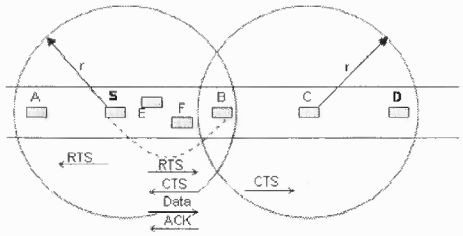



**Figure 4.1** Example of RTS/CTS exchange in IEEE 802.11

never send to node B while node S is sending to node B since node C has heard the CTS from node B. Node C will wait for the time specified in the CTS to guarantee that the transmission from S to B is successful.

### 4.3 Election using RTS/CTS

**Protocol 2** Self-Election Algorithm at Node $n_i$

Notation:

$t_{DATA}, t_{CTS}, t_{RTS}, t_{ACK}$: time to transmit the data frame, the CTS, the RTS, and the ACK
$t_i$: waiting time of node $n_i$
$loc_i$: location of node $n_i$
$loc_D$: location of the destination
$n_S$: ID of the sender looking for the next hop

Upon receiving $RTS(loc_s, loc_D, t_{DATA})$ from node $n_s$:
1: Call waiting function and calculate $t_i$
2: Set timer to $t_i$
3: Defer transmissions, if any, to node $n_s$ for $t_{DATA} + t_{RTS}$

Upon receiving an $CTS(n_j, n_s, t_{DATA})$ from $n_j$ before the timeout
4: Cancel the timer /* $n_j$ is the best next hop candidate */
5: Defer transmissions, if any, to $n_j$ for $t_{DATA}$

Upon overhearing $DATA$ from node $n_s$
6: Defer transmissions, if any, to $n_s$ for $t_{ACK}$

Upon timeout:
7: Broadcast $CTS(n_i, n_S, t_{DATA})$ /* $n_i$ is the best candidate */

RBVT leverages the RTS/CTS exchange to replace the sender selection of the next hop with a receiver self-election, and implicitly eliminates the overhead associated with

frequent "hello" messages in geographical forwarding in congested networks. Essentially, broadcast of RTS frames become requests for next hop self-election. RTS frames are modified to carry the position of the sender and the position of the target destination, which are used during the self-election. RTS frames also carry a flag to indicate to all receiving nodes that they should process and possibly answer the frame (in the original mechanism, only the intended receiver processes and answers an RTS frame).

Specifically, each node which receives the modified RTS frame calculates a waiting time, after which it will send a CTS frame back to the sender. The waiting time is an indicator of how good a forwarding candidate the node is: the shorter the waiting time, the better candidate a node is. Section 4.4 explains how this waiting time is calculated. A CTS from one of the receivers tells that a better candidate exists and no candidate receivers that overhears it will reply. The sender receives the CTS from the best next hop candidate and forwards the data frame to this node, which then acknowledges the data frame. The detailed protocol is presented in Protocol 2.

Fig. 4.2 shows an example that illustrates this protocol. Sender $n_S$ needs to forward a data frame to the best next hop en-route to the destination D. It broadcasts an RTS frame specifying its own location, the location of the destination D, and the transmission time of the data frame. Nodes $n_1$, $n_2$, and $n_3$ hear the RTS, calculate their waiting time and set their timers waiting to reply to $n_S$ with CTS. Note that $n_4$ does not perform the computation as it is farther from the destination than the sender. Node $n_2$ has the shortest waiting time and replies with the CTS first. Once $n_1$ and $n_3$ overhear the CTS from $n_2$, they will cancel their timers. In addition, all the neighbors of $n_2$, including $n_5$, $n_6$, $n_1$ and $n_3$, will know that they should not send any frame to $n_2$ until it completes the transmission. Once $n_S$ receives the CTS from $n_2$, it will send the data frame to $n_2$. At the same time, all of $n_S$'s neighbors

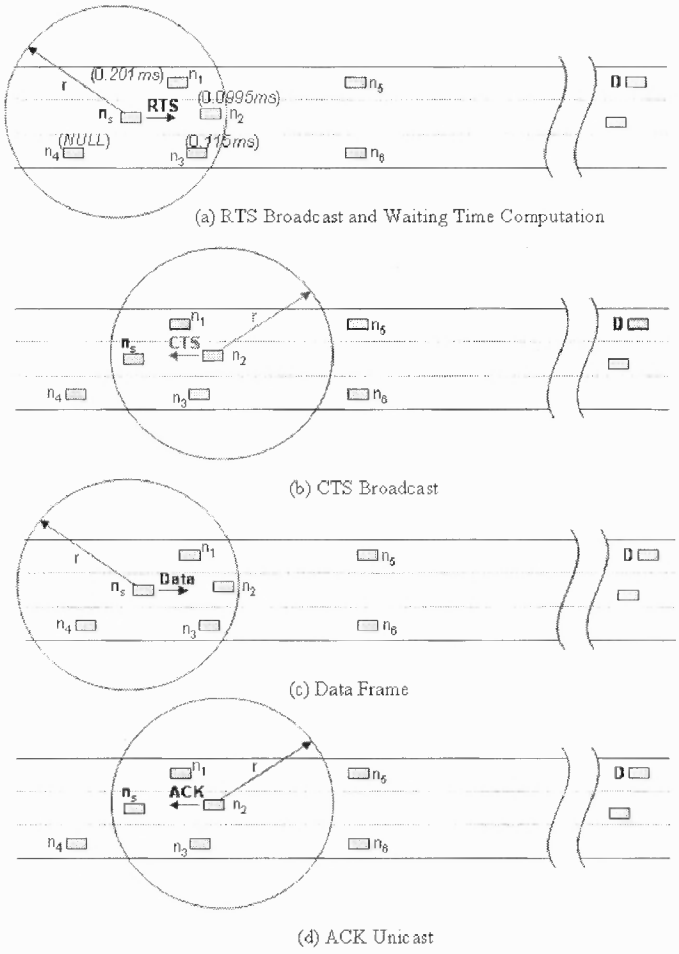



**Figure 4.2** Next Hop Self-Election Example

overhearing the data frame learn that they should not send any frame to $n_S$ until $n_2$ finishes sending the acknowledgment to $n_S$. This example shows how this forwarding method can effectively choose the next hop without "hello" messages overhead.

## 4.4 Waiting Function

Determining the best next hop depends on the waiting time. An effective calculation of this waiting time should meet three objectives: (1) the waiting time of the best next hop candidate is the shortest such that this node replies first, (2) the waiting time difference

between the best next hop candidate and the second best candidate is large enough such that collisions are minimized between nearby nodes, and (3) the waiting time is as short as possible to avoid unnecessary delays. To achieve these goals, first, three key parameters are identified, forward progress, optimal transmission area, and received power, which characterize the best next hop node. Next, the parameters are incorporated with different weights, into a low-complexity function that computes the waiting time.

**Function Parameters** *Forward progress* $d_i$ of a node $N_i$ from a sender $S$ is defined as $d_i = d_{SD} - d_{N_iD}$, where $d_{SD}$ is the distance between the sender $S$ and the destination $D$, and $d_{N_iD}$ is the distance between $N_i$ and $D$. This parameter is commonly used in geographical forwarding of single criterion receiver-oriented schemes [45, 43, 46]. It denotes the actual progression made by the packet toward the destination if $N_i$ would be the next hop. A node with $d_i$ closest to $d_{SD}$ is the node closest to the destination.

*Optimal Transmission Area* $f_i$ of a node $N_i$ describes the probability that the node can receive the sender's data packet successfully. Since wireless channels are error-prone, a node located much farther than the nominal transmission range may not be able to receive long data frames successfully even though it can receive short RTS frames without errors. This situation could happen because real wireless radios do not follow the unit disk assumption [80].

A translation function is deployed to express the optimal transmission area. The function takes the distance to the sender as input and outputs the distance to the optimal transmission area. A sample translation function is shown below and sample graphs for two translation functions are depicted in Fig. 4.3.

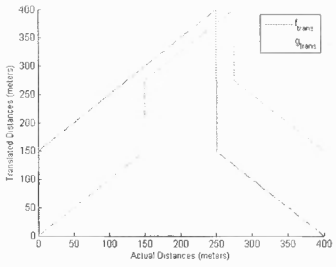



**Figure 4.3** Sample translation functions for optimal transmission area

$$f_{trans}(x) = \begin{cases} \text{x} + d_{trans} & \text{if } x \leq d_{opt} \\ \text{-x} + d_{trans} & \text{if } x > d_{opt} \end{cases}$$

where $d_{opt}$ represents the optimal transmission range, $d_{max}$ represents the estimated maximum transmission range for acceptable error rate, and $d_{trans}$ represents the translation distance ($d_{trans} = 150m$ for $f_{trans}$ in Fig. 4.3). These parameters may be adjusted based on the network conditions in the area.

*Received Power* $p_i$ of a node $N_i$ is the received power level of the RTS frame. Priority is given to nodes with stronger $p_i$. This parameter indicates the true channel quality from a sender to a receiver. Empirical studies and theoretic analysis can provide an optimal transmission area, but in reality, there might be obstacles or noise around nodes. The received power can also help differentiate nodes at comparable distances. The fact that the reporting of the received signal power is made while the vehicles are moving does not affect the quality of the reported data because the distance traveled by a vehicle while receiving an RTS is negligible.

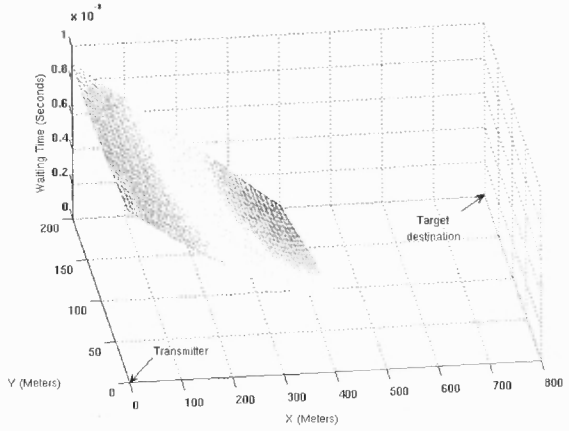
**Function Definition** The multi-variable function proposed in [47] is adapted to a three-variable polynomial of the selected parameters. The waiting time, $t_i$, returned by this function is in the interval $[0, T_{max}]$, where $T_{max}$ is the maximum waiting time.

$$f(d_i, d_{SN_i}, p_i) = A d_i^{\alpha_1} f_i^{\alpha_2} p_i^{\alpha_3} + T_{max}$$

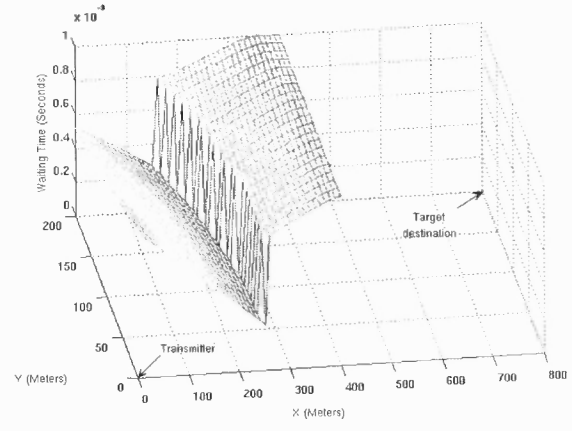
where $A = \frac{-T_{max}}{d_{max}^{\alpha_1} f_{max}^{\alpha_2} p_{max}^{\alpha_3}}$ and $\alpha_i (i = 1, 2, 3)$ is the weight of each parameter.

The greater the weight value, the more impact the parameter has in the election process. All next hop candidates use the same values of factors $\alpha_i$. Currently, static values are used for these factors, but they could be determined and adjusted dynamically based on the network and traffic conditions in the area.

**Function Evaluation** Figure 4.4 shows a comparison between the next hop selection using the proposed function and the selection using only forward progress. The transmitter is at location (0,0) and a destination at location (800,200). The figures show the waiting time for nodes located at various locations around the transmitter after they receive an RTS frame. In this comparison, the function coefficients used are: $\alpha_1 = 0.2$, $\alpha_2 = 1.2$ and $\alpha_3 = 0.03$. The optimal wireless transmission range is set to be 250 meters, and the translation function $f_{trans}$ shown in Figure 4.3 is deployed. Note that perfect reception within a specific range around the transmitter is not assumed. Rather, the received power is calculated using the Shadowing propagation model [76]. In this model, the power level at a receiving node is not solely a function of the distance to the transmitting node, but randomness is added to account for fluctuations in signal propagation. The formula used to compute the received power is




(a) Waiting time determined using forward progress only




(b) Waiting time determined using multi-criteria function

**Figure 4.4** Waiting times experienced by receivers located at various positions around a transmitter

| | Packet Loss Rate | Frames per Received Packet |
|---|---|---|
| Forward progress only | 37.54% | 36.50 |
| Multi-criteria function | 7.52% | 2.62 |

**Table 4.1** Using the multi-criteria function to select next hops leads to significantly lower packet loss and overhead compared to using forward progress only.

$$\left|\frac{P_r(d)}{P_r(d_0)}\right|_{dB} = -10\beta log(\frac{d}{d_0}) + X_{dB} \tag{4.1}$$

where $X_{dB}$ is a normal random variable with mean zero and standard deviation $\sigma_{dB}$. $\sigma_{dB}$ is the shadowing deviation and $\beta$ represents the path loss exponent.

The comparison in Fig. 4.4 shows that the proposed scheme favors nodes around the optimal transmission range and assigns shorter waiting times for the nodes within this range. The forward progress only approach favors nodes beyond the optimal transmission range (in case they receive the RTS), which could lead to many data packet losses. Table 4.1 validates this observation, as it presents a comparison between the two methods in terms of packet loss and the number of MAC layer frames transmitted in the network per data packet received successfully at destinations (which is a measure of traffic overhead). These simulation results were obtained using RBVT-R in a network with 250 nodes. 15 source-destination pairs exchanged 10000 packets at the rate of 2 packets/second. Using the multi-criteria function leads to a packet loss 5 times lower than using forward progress only. Also, the traffic overhead is more than one order of magnitude lower when using the proposed scheme. This result is due, in part, to the high number of re-transmissions experienced by nodes located farther from the optimal transmission range.

## 4.5 Performance Evaluation

This section presents the evaluation of the distributed self-election mechanism on RBVT performance using the Network Simulator NS-2.30 [71]. To evaluate the performance, two urban scenarios are considered: (1) a scenario with obstacles, to model buildings, in which periodic "hello" messages and the IEEE 802.11 MAC standard are used, and (2) a scenario without obstacles to simulate high contention networks. The metrics as well as the settings of the scenario with obstacles are the same as in Section 3.3. In the following, the scenario without obstacles is described and the simulation results showing the impact of the forwarding optimizations on RBVT performance are presented.

### 4.5.1 Simulation Setup for Scenario Without Obstacles

The simulation scenario without obstacles uses a 1500m x 1500m area extracted from the TIGER/Line database of the US Census Bureau [66], which forms a grid layout with a total of 22 road segments. It is an area of Fellsmere, FL with center point coordinates latitude 27.784728° North and longitude -80.604385° West. Each road has bi-directional traffic with two lanes in each direction. In order to evaluate the protocols under increased network congestion, obstacles are not added to this scenario. This way, a small increase in data sending rate will provide a noticeable increase in the level of contention in the network.

The microscopic mobility generator employed is one developed during this dissertation work, and based on the car-following and lane-changing models proposed by Gipps [83, 84]. The Gipps model belongs to the class of collision avoidance vehicular mobility models. The main goal of these models is to enable a vehicle to move at the maximum safest speed that avoids collision with the preceding vehicle. The traffic generator supports

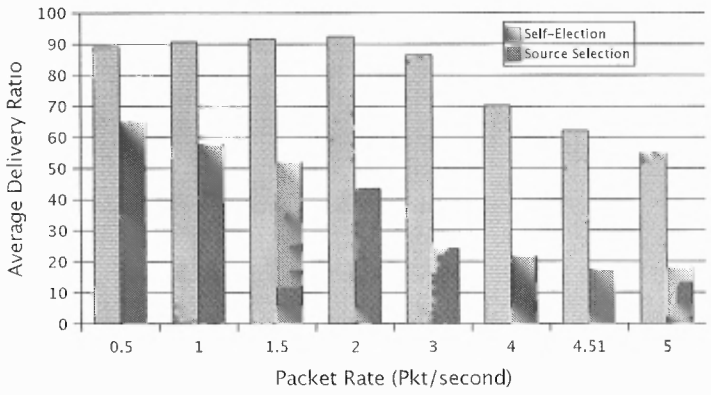
traffic lights at road intersections as well as bidirectional and multi-lane traffic. The input to the generator is a map of the roads with specifications of the average speed and the average traffic flow on each road. When a vehicle enters a road segment, its action at the end of the segment (i.e., left turn, right turn, u-turn, or straight ahead) is based on the average traffic flows of the roads crossing the end intersection. The first 2000 sec of output are discarded to obtain more accurate movements of nodes.

The IEEE 802.11 with DCF standard is used for AODV, OLSR, GPSR and GSR, and the forwarding optimizations for the RBVT protocols. The "hello" interval is set to 2 sec because it provided better results in this scenario. At the physical layer, the Shadowing propagation model characterizes physical propagation. For these simulations, the wireless range is set to 250 meters, to prevent communication between vehicles on parallel streets (minimum distance between streets is 400m). The values of exponents of the waiting function (in the next hop self-election mechanism) are $\alpha_1 = 0.07$, $\alpha_2 = 0.5$ and $\alpha_3 = 0.03$. LIFO was used instead of FIFO as the queuing discipline for RBVT in this scenario because it provided better latency when experiencing high contention [29].

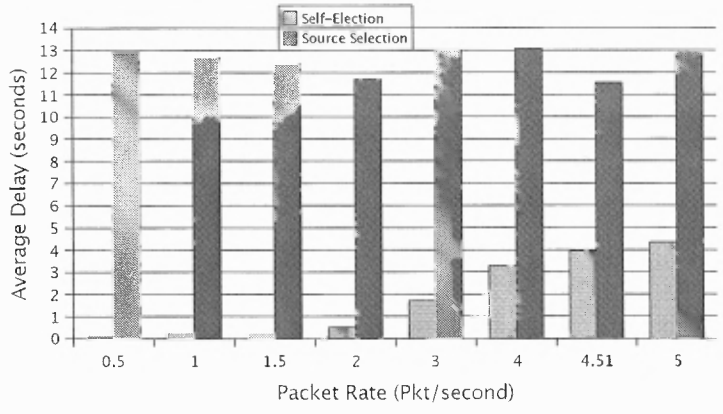
### 4.5.2 Simulation Results

**Forwarding Optimization:** Figure 4.5 and Figure 4.6 assess the impact on performance of the proposed geographical forwarding mechanism, which takes advantage of the 802.11 RTS/CTS to choose the next hop using receiver self-election, compared with a traditional approach using "hello" packets to create the list of neighbors at nodes. Both scenarios with and without obstacles are considered.

In the scenario without obstacles, the "hello" packets were generated every 2 seconds (which produces the best results for this approach). Figure 4.5(a) shows that the forwarding

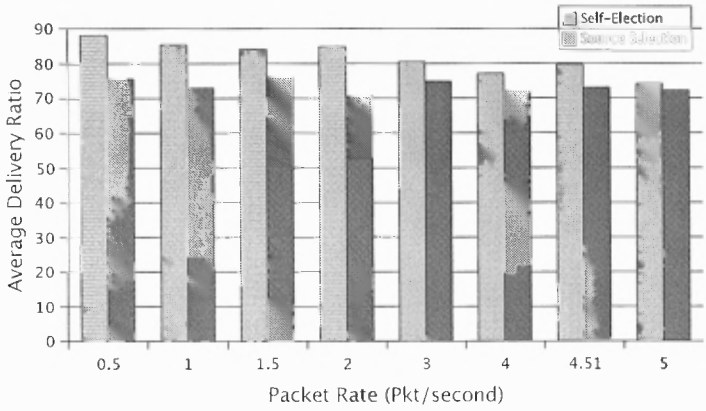



(a) Average Delivery Ratio

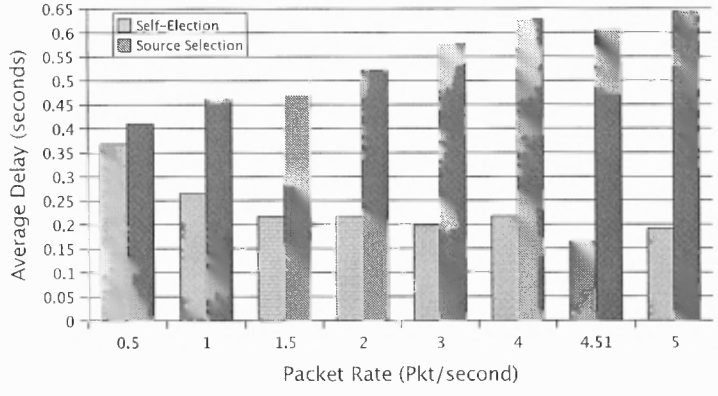



(b) Average Delay

**Figure 4.5** Average Delivery Ratio and Average Delay comparison between two types of geographical forwarding: source selection using "hello" packets and receiver self-election using the RTS/CTS-based mechanism under the scenario without obstacles. The routing protocol is RBVT-R, and the network size is 250 nodes.

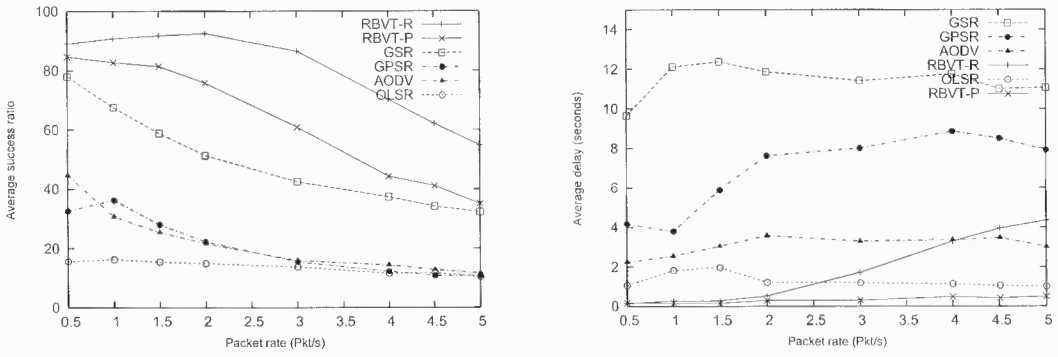



(a) Average Delivery Ratio




(b) Average Delay

**Figure 4.6** Average Delivery Ratio and Average Delay comparison between two types of geographical forwarding: source selection using "hello" packets and receiver self-election using the RTS/CTS-based mechanism under the scenario with obstacles. The routing protocol is RBVT-P, and the network size is 250 nodes.




(a) 250 nodes

(b) 250 nodes

**Figure 4.7** Average Delivery Ratio and Average Delay for RBVT-R, RBVT-P, AODV, OLSR, GPSR and GSR in Networks under high contention

optimizations lead to a delivery ratio as much as 3 times higher in congested environments. Similarly, Figure 4.5(b) shows that the delay is three times lower on average with the improvements. There are two reasons for these high improvements. First, the absence of periodic hello messages means less overhead in the network. This overhead reduction leads to much higher link utilization for data transfers. It also leads to improved delays as fewer re-transmissions and exponential backoffs are necessary. Second, the multi-criteria waiting function used in the election of the next hop favors link quality over greediness as explained in Section 4.4.

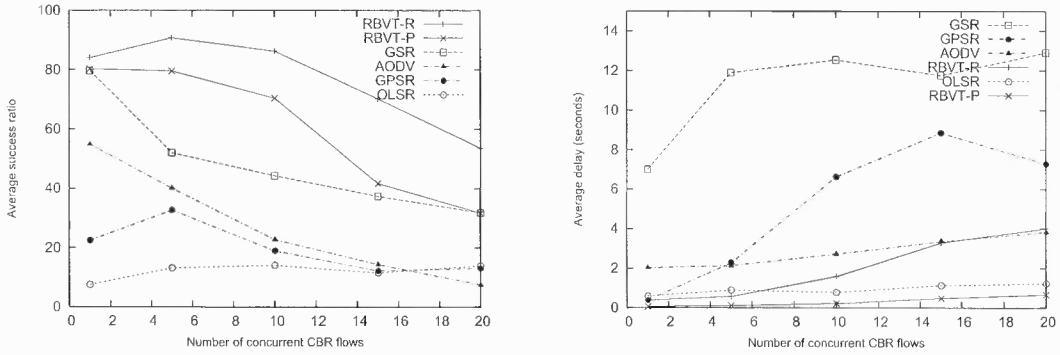
In the scenario with obstacles (Figure 4.6(a)), the forwarding optimizations lead to an increase in the packet delivery ratio of up 14%. The difference between the self-election and the source selection results in this scenario is smaller, when compared to the scenario without obstacles (Figure 4.5(a)). This is due to the reduced level on contention because of the obstacles. The average delay is reduced as well.

Using the scenario without obstacles, RBVT-R and RBVT-P are compared against the four existing VANET/MANET routing protocols listed in Section 3.3.

**Average delivery ratio:** Figure 4.7 shows that in this scenario, the RBVT protocols outperforms the other protocols in terms of successful data delivery ratio. Note that all the protocols are more sensible to the increase in the data packets sending rate than previously (Figure 3.5). Both RBVT protocols perform better under added congestion because of the forwarding optimization. At 3 packet/second for e.g., only RBVT-R and RBVT-P have a data delivery ratio above 50%. Comparing RBVT-P with OLSR, one observes that OLSR performance is more affected by contentions in the network. RBVT-P maintains only the overall connectivity between the road intersections in the network, while OLSR proactively maintains the link state between the multipoint relays.

**Average delay:** Figure 4.7(b) shows that for most packet rates, RBVT-P has the best performance in terms of delay. The contention on the wireless channel can be clearly observed here, with the values of the average delay for GPSR and GSR increasing well above 5 seconds.

**Impact of number of flows:** The impact of the number of concurrent flows on the protocol performances in this scenario is shown in Figure 4.8. Generally, the fewer the number of flows, the better the protocol performance in terms of delivery ratio. Among the protocols, the RBVT protocols scale better than the other protocols. AODV shows the most accentuated drop in delivery ratio, with a 50% decrease from the 1 flow simulation to the 20 flows simulation. The AODV protocol is able to keep the average delay of the transmitted data in check by dropping packets for which it does not have a route. GSR does not scale very well with the variation in the number of flows either, especially for the delay that practically doubles for 20 flows compared to 1 flow. RBVT-R has the minimum decrease in delivery ratio among the simulated protocols (Figure 4.8(a). However, RBVT-R average




(a) Average Delivery Ratio

(b) Average Delay

**Figure 4.8** Average Delivery Ratio and Average Delay with variable number of concurrent flows. The data rate is fixed at 4 packets/second and the network size is 250 nodes

delay is more sensitive to the added flows than RBVT-P which consistently maintains a small average delay.

**Overhead:** As expected based on the results of Table 4.1, the next hop self-election mostly eliminates the overhead of the RBVT protocols when compared to other protocols such as AODV and GSR. Although RBVT-R floods route discovery requests and RBVT-P floods the routing update packets, these overheads are very small compared with the overhead introduced by frequent route errors in AODV and the "hello" packets overhead in GSR. Using the roads layout and the real-time vehicular traffic information lead to more stable paths, and hence lower overhead for RBVT protocols.

## 4.6 Chapter Summary

This chapter described a beaconless, distributed self-election mechanism for vehicular ad hoc networks based on a slight modification of the RTS/CTS mechanism. The three criteria prioritization function, used in the self-election, takes into account realistic channel propagation and does not introduce additional overhead. Simulation results showed that the

forwarding optimizations lead to a delivery ratio as much as 3 times higher in congested environments.

# CHAPTER 5

# EFFECT OF QUEUING DISCIPLINES ON FORWARDING PERFORMANCE

This chapter analyzes the relative performance of FIFO versus LIFO and of Taildrop (TD) versus Frontdrop (FD) in all four possible combinations, namely FIFO-TD, FIFO-FD, LIFO-TD and LIFO-FD. Analytical results at a single node queue show that FIFO disciplines perform better in terms of expected waiting time and variance of the expected waiting time for non-congested situations (i.e., packet arrival rate is less than packet service rate), while LIFO-FD performs better for congested situations (i.e., packet arrival rate is greater than packet service rate). Simulations in larger scale ad hoc networks show that LIFO-FD leads to less than half the UDP delay of FIFO-TD under high data traffic load, while maintaining similar jitter. When the traffic load is low, the four disciplines present comparable end-to-end delays, but LIFO shows greater jitter than the other disciplines (FIFO-FD has the lowest jitter).

The rest of this chapter is organized as follows. Section 5.1 motivates this study. A brief overview of the service disciplines and drop policies studied is presented in Section 5.2. Section 5.3 contains the theoretical analysis and results for single node queuing. The simulation results are presented in Section 5.4. The chapter concludes in Section 5.5.

## 5.1 Motivation

Many VANET applications are delay sensitive, but loss tolerant. For example such applications can send video stream from the proximity of an accident or monitor traffic conditions. In the more general case of MANET, multimedia streaming can be used for video surveillance,

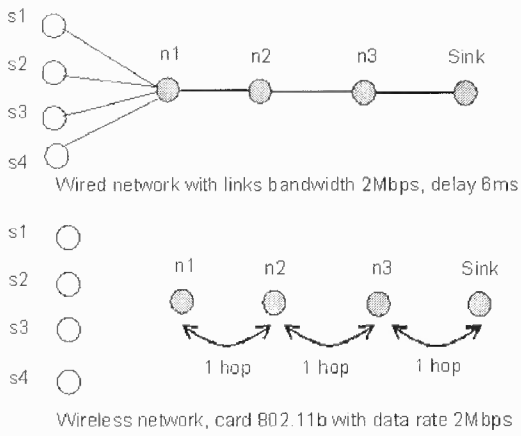



**Figure 5.1** The shared nature of a wireless network creates contention and leads to congestion faster than wired networks. Hosts ($S1$, $S2$, $S3$, $S4$) transmit UDP traffic at rate of 20 packet/sec to sink. Duration 100sec. Average end-to-end delays greater than 2sec in the wireless network while the wired network has average delay of 87ms.

entity tracking, or search and rescue missions. Unfortunately, the performance of these applications (i.e., delay and jitter) degrades quickly in ad hoc networks, even for moderate traffic. For many reasons, including varying connectivity and interference between links, ad hoc networks experience problematic effects of congestion, such as high end-to-end delay and jitter, faster than well-designed wired networks. Figure 5.1 illustrates this problem with two networks, one wired and one wireless (IEEE 802.11), having the same topologies and queuing discipline (FIFO with Taildrop). Under identical traffic load, the wired network results in an average end-to-end delay of 87ms while the wireless network experiences more than 2sec average delay. The difference between these results can be partly explained by the contention algorithm used in the wireless network, which prevents any two nodes from the set ($S1$,$S2$, $S3$, $S4$, $n1$, $n2$) from transmitting data simultaneously. Consequently, packets spend longer time in transit.

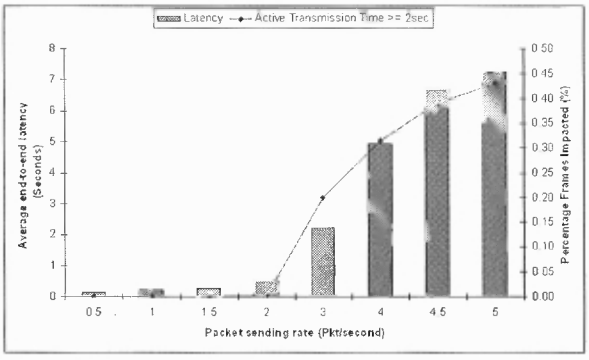



**Figure 5.2** Average end-to-end delay as function of percentages of frames experiencing transmission delays greater than 2 seconds. The results were obtained using RBVT-R routing [1] and 15 source-destination pairs exchanging packets at rates varying from 0.2 to 5 packets/sec.

To better understand the combined effect of contention and queuing discipline on end-to-end delay, simulations are conducted in a vehicular ad hoc network with 250 wireless vehicles moving at normal speeds on city roads. The simulations showed that during high contention periods the IEEE 802.11 exponential back-off mechanism can make a frame, ready to be transmitted, wait more than 2 seconds at a node queue before completing the transmission. Although the number of frames that experience these unusually high transmission delays is relatively low, their effect propagates to all the frames in the node queue. Figure 5.2 shows that a very low percentage of "problem" frames could lead to more than one order of magnitude increase in the average end-to-end delay. This observation follows from the characteristics of the FIFO Taildrop queuing discipline. When the queue becomes full, newly arrived packets are dropped. On the other hand, the old packets that experienced long queuing delays are delivered, contributing to the substantial increase in the end-to-end delay.

The main questions drawn from these preliminary experiments are: Can ad hoc networks in general, and VANETs in particular, achieve better delay and jitter with a different queuing discipline? Is there a queuing discipline that performs better in all cases or different disciplines work better for different network conditions? To answer these questions, this chapter analyzes the relative performance of FIFO versus LIFO and of Taildrop (TD) versus Frontdrop (FD) in all four possible combinations, namely FIFO-TD, FIFO-FD, LIFO-TD and LIFO-FD. Note that the choice of queuing discipline has been already shown to impact the performance of data transfers in wired IP networks [27], where TCP was proved to perform better under congestion when routers use FIFO with Frontdrop instead of FIFO with Taildrop or RED [28].

## 5.2 Background

In the following, the perceived advantages and disadvantages of the queuing disciplines considered in this chapter, specifically the service disciplines and the drop disciplines, are discussed as they pertain to TCP and UDP traffic. This preliminary discussion will help with the understanding of the experiments, results, and conclusions. Figure 5.3 illustrates how each discipline works.

### 5.2.1 Service Disciplines

The FIFO service discipline selects the packet at the head of the queue as the next packet to service. FIFO allows packets forwarded along the same path to be delivered at the destination in the same order as they were produced at the source. Protocols, such as TCP, benefit from the in-order delivery of packets because the destination can remove data from

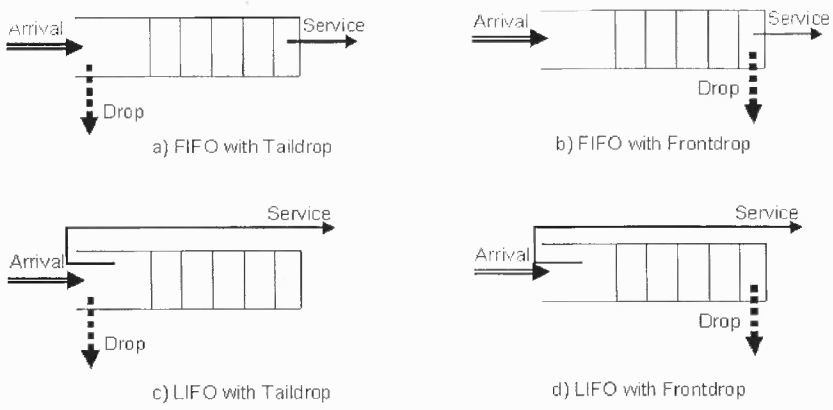



**Figure 5.3** FIFO and LIFO with Taildrop and Frontdrop in all four possible combinations.

buffers as soon as in-order data is received. FIFO also provides a sense of fairness because the packet which is serviced is the one which has been waiting the longer.

However, FIFO leads to long waiting times under congestion. Each newly received packet is inserted at the end of queue and will be serviced after all the packets already waiting in the queue at its arrival. When this happens over multiple queues and the wait time in a specific queue is not negligible, the total time spent by a packet in transit may increase considerably. In applications such as live streaming, this may result in packets arriving too late to be played, thus unnecessarily using network resources.

The LIFO service discipline, on the other hand, selects the packet at the tail of the buffer as the next packet to service. The advantage is that the newest packets will be served first. UDP performance for live streaming could be improved if the packets that stayed for too long in the queues, which will probably be dropped at destination anyway, do not affect the delivery of newer packets. More details about this idea will be presented in Section 5.4.

A number of issues might arise with LIFO. First, LIFO provides a sense of unfairness because newer packets are prioritized over packets which have been waiting longer. Additionally,

while packet-switched networks do not expect in-order delivery of packets, it is expected that LIFO will result in more out-of-order packets than FIFO.

### 5.2.2 Drop Disciplines

When the queues become full, the node/router applies the drop discipline (Taildrop or Frontdrop) to each newly received packet. Taildrop drops the newly received packet. The advantage is not only that the waiting time to drop a packet is null, but also this policy provides a sense of fairness by keeping the packets already admitted in the system. On the other hand, one might argue that dropping newly received packets, as Taildrop does, is a disadvantage because some packets may never get a chance to enter the queue and be serviced.

With Frontdrop, the oldest packet in the queue (the one at the head) is dropped, and the newly received packet is added to the tail of the queue. The advantage of this policy is that aged packets are dropped. However, it provides a sense of unfairness because packets which are close to being serviced are dropped from the queue.

## 5.3 Analytical Analysis

In this section, the analytical expressions for the expected waiting time and the variance of waiting time for packets in a single queue are derived under FIFO-TD, FIFO-FD, LIFO-TD, and LIFO-FD. The goal of this analysis is to provide insights into the differences between these queuing disciplines to understand which ones are the most promising and under what conditions.

### 5.3.1 System Model

The model used is an exponential queue [1]. If there is space for $N$ waiting packets in the queue, the system has the states $\{-1, 0, \cdots, N, N+1\}$. In state $(-1)$, the system is empty: no packet waiting or being served. In state $(0)$, there is one packet being served and there are no waiting packets. In state $(k)$ with $1 \leq k \leq N$, there is one packet being served and there are $k$ waiting packets. State $(N+1)$ is not a real state but is used to facilitate the description: when a packet arrival occurs while the system is in state $(N)$, the system first goes to the hypothetical state $(N+1)$ and then instantaneously drops a packet and goes back to state $(N)$.

This model leads to a Markov Chain $X(t)$ on the state space $\{-1, 0, \cdots, N\}$, where $X(t)$ is the state at time $t$. The stationary distribution of this Markov chain does not depend on FIFO–LIFO or Taildrop–Frontdrop. The traffic intensity is defined as $\rho = \frac{\lambda}{\mu}$, where $\lambda$ is the arrival rate and $\mu$ is the service rate. For all queuing disciplines, the stationary probabilities $\pi_k = P\{X(t) = k\}$ are: $\pi_{N+1} = 0$ and

$$\pi_k = \rho^{k+1}\pi_{-1}, \quad \pi_k = \rho^{k+1}.\frac{1-\rho}{1-\rho^{N+2}} \quad \text{for} \quad -1 \leq k \leq N. \tag{5.1}$$

It must be noted that it is *not* required that $\rho < 1$. The finite state space makes the process ergodic anyhow, and in fact, the problem is of most interest when $\rho$ is close to 1 or even larger. If $\rho = 1$, then according to (5.1): $\pi_k = \frac{1}{N+2}$ for $-1 \leq k \leq N$. In addition to (5.1), a fraction $1 - \pi_N$ of all arriving packets will eventually be served and a fraction $\pi_N$ will not be served (i.e., they will be dropped). For all disciplines considered, if a packet arrives in state $(k)$, $k < N$, then the state changes to $(k+1)$: all packets already in the

[1] A discussion of the question of whether this model is appropriate for packet queues is presented in the next sub-section.

system keep their position, and the new packet goes into position $k+1$. If there is a service completion in state $(k)$ (this implies $k \geq 0$), then the state changes to $(k-1)$. Under FIFO, the packet in position $j$ $(0 \leq j \leq k)$ moves to position $j-1$. Under LIFO, the packet in position $k$ moves to position 0, and all the other packets stay in the position they were in.

If a packet arrives while the system is in state $(N)$, the system remains in this state; as explained, the system moves to state $(N+1)$ for an infinitely small amount of time, then moves back to state $(N)$. Under Taildrop, the arriving packet is dropped and all the other packets remain in their old positions. Under Frontdrop, the packet in position 1 is dropped, all the other packets already in the system move up one position $(j \rightarrow j-1)$, and the arriving packet goes to position $N$.

Let $T$ denote the elapsed time from arrival of a packet to either being dropped or serviced. Define "eventually being dropped" as "failure" and "eventually being served" as "success". The probabilities of success and failure are defined as follows:

$$\eta_S(s) = E[e^{-sT}\chi(Success)], \quad \eta_F(s) = E[e^{-sT}\chi(Failure)], \tag{5.2}$$

where $\chi(.)$ is the indicator function.

The probabilities of success and failure given that the packet arrives when the system is in state $k$ are as follows:

$$\eta_{k,S}(s) = E[e^{-sT}\chi(Success)|\text{packet arrives while system in state } (k)],$$

$$\eta_{k,F}(s) = E[e^{-sT}\chi(Failure)|\text{packet arrives while system in state } (k)], \tag{5.3}$$

such that

$$\eta_S(s) = \sum_{k=-1}^{N} \pi_k \eta_{k,S}(s), \quad \eta_F(s) = \sum_{k=-1}^{N} \pi_k \eta_{k,F}(s), \tag{5.4}$$

where for all disciplines considered

$$\eta_{-1,S}(s) \equiv 1, \quad \eta_{-1,F}(s) \equiv 0. \tag{5.5}$$

Section 5.3.3 shows how the expressions for $\eta_{k,S}(.)$ and $\eta_{k,F}(.)$ are derived for all the four disciplines. While standing for the same concepts, the expressions are different for the four disciplines considered.

### 5.3.2 On the Exponential Model Used

An important question is whether the exponential model is appropriate for packet queues. Before discussing this question further, note that the modeling stage does not lead to a choice of parameters for the actual queuing disciplines to be used, but to broad categories of queuing disciplines that are the most promising. These broad categories will be further investigated through simulations.

A model better than the M/M/1/K (exponential arrival rate and exponential service rate), which is employed here, might be M/D/1/K (exponential arrival rate and deterministic service rate). This model is attractive when most packets are of almost the same size such as the MSS (maximum segment size) in TCP. However, this is certainly not the case: with TCP, at least one third of the packets are acknowledgments, SYNs, and FINs. A counterargument, in favor of using M/D/1/K, is that the fraction of MSS-sized packets may temporarily be very high in one direction. Thus, it is hard to find a model that is clearly better than the M/M/1/K model.

Intuitively, it seems likely that this model captures the interleaving of arrivals and departures reasonably well. Therefore, it should do a good job in predicting the probability of being served ("Success") and being dropped ("Failure") given the number of prior

packets found upon arrival. Additionally, it should also do a good job in predicting average or expected waiting times until such events.

The model quite likely does not capture very well the variability of waiting times. For example, in M/M/1/K with FIFO-TD, if a packet has exactly $k$ packets in front of it (waiting or being served), the expected residual waiting time is exactly $\frac{k}{\mu}$ and the variance of the residual waiting time is exactly $\frac{k}{\mu^2}$. In the M/D/1/K model, the expected residual waiting time is very close to $\frac{k}{\mu}$ for large $k$, but the variance of the residual waiting time is actually $O(1)$ for $k \to \infty$.

Therefore, estimates of variances of waiting times based on the M/M/1/K model are probably not of exact accuracy, and in particular, are likely to be too high. Assuming that the model does a good job in predicting numbers of service completions of other packets between the arrival of a packet and its departure from the queue (i.e., serviced or dropped), it is possible to derive reasonable correction on the estimated variances.

### 5.3.3 Analytical Results

**FIFO with Taildrop** In the stationary situation, an arriving packet finds, with probability $\pi_k$ as in (5.1), that the system is in state $(k)$. If $k = -1$ the arriving packet does not need to wait: waiting time is zero. Under FIFO with Taildrop, if the arriving packet finds the system in state $(N)$, the arriving packet is dropped (and encounters no waiting). If the arriving packet finds the system in state $(k)$, with $0 \leq k \leq N-1$, the wait is the sum of $(k+1)$ independent random variables, each exponentially with parameter $\mu$. Therefore

$$\eta_S(s) = \sum_{k=-1}^{N-1} \pi_k \left(\frac{\mu}{\mu + s}\right)^{k+1}. \tag{5.6}$$

$$\eta_F(s) = \pi_N. \tag{5.7}$$

**FIFO with Frontdrop** Consider pairs $(k, L)$ with $0 \leq k \leq L \leq N$, where $(L) = X(0^+)$ is the state of the system (number of waiting packets) and $k$ is the location of a "marked" packet, both at time 0. "Success" means that the marked packet is eventually served, and "Failure" means that the marked packet is eventually dropped. Dropped means the packet makes it to location 1 and then (so close to success) is dropped. Let $T_{k,L}$ be the time until the marked packet either starts being serviced or is dropped. Define

$$\zeta_{k,L,S}(s) = E[e^{-sT_{k,L}}\chi(Success)], \zeta_{k,L,F}(s) = E[e^{-sT_{k,L}}\chi(Failure)]. \tag{5.8}$$

Thus, $\zeta_{k,L,S}(0)$ is the probability the marked packet (eventually) gets served, and $\zeta_{k,L,F}(0)$ is the probability that the marked packet eventually gets dropped. Also, for $0 \leq k \leq N-1$

$$\eta_{k,S}(s) = \zeta_{k+1,k+1,S}(s), \quad \eta_{k,F}(s) = \zeta_{k+1,k+1,F}(s), \tag{5.9}$$

and

$$\eta_{-1,S}(s) = 1, \quad \eta_{-1,F}(s) = 0, \eta_{N,S} = \zeta_{N,N,S}(s), \quad \eta_{N,F} = \zeta_{N,N,F}(s). \tag{5.10}$$

The expression of $\zeta_{k,k,S}(s)$ is computed in Appendix B.

**LIFO with TailDrop** As before, if the arriving packet finds the system in state $(-1)$, the packet is served and there is no waiting. If the packet finds the system in state $(N)$, the packet is dropped and there is no waiting.

If the arriving packet finds the system in state $(k)$, with $0 \leq k \leq N-1$, the packet takes position $k+1$ and will certainly be served; in fact, it will be served before all already

waiting packets. Therefore, it is as a system with a buffer for $N-k$ waiting packets, the just arriving packet has position 1 in that new system which starts in state $(1)$, and the waiting time of that packet is the first passage time to state $(0)$ in that new system.

Therefore

$$\eta_S(s) = \pi_{-1} + \sum_{k=0}^{N-1} \pi_k \left( \sigma_{1,N-k}(s) + \frac{\phi_{1,N-k}(s)\sigma_{N-k,N-k}(s)}{1-\phi_{N-k,N-k}(s)} \right), \tag{5.11}$$

As always with Taildrop:

$$\eta_F(s) = \pi_N. \tag{5.12}$$

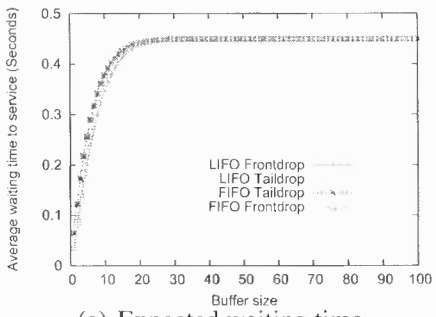
The entities $\sigma_{k,N}(.)$, $\phi_{k,N}(.)$, $\sigma_{k,N} = \sigma_{k,N}(0)$, $\phi_{k,N} = \phi_{k,N}(0)$ are derived in Appendix A.

**LIFO with Frontdrop** This case is somewhat similar to the previous case. The difference is that every time a drop occurs, the packet in location 1 is dropped and all the other packets go to one lower location. Also, while a packet which arrives in state $(k)$, with $0 \leq k \leq N-1$, goes into location $k+1$, a packet which arrives while the system is in state $(N)$ goes into position $N$. A packet in position $k+1$, while the state is $(k+1)$, either is served before the next drop or moves to position $k$, state $N$, at the next drop. Hence, for $0 \leq k < N$:

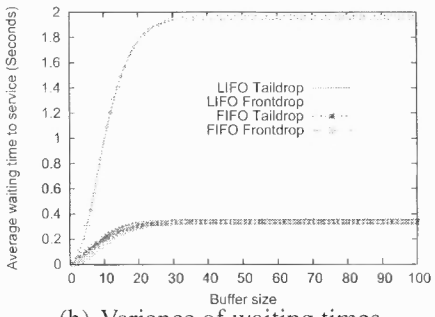
$$\eta_{k,S}(s) = \sigma_{1,N-k}(s) + \phi_{1,N-k}(s) \left( \sum_{j=1}^{k} \left( \prod_{\nu=1}^{j-1} \phi_{N-k+\nu,N-k+\nu}(s) \right) \sigma_{N-k+j,N-k+j}(s) \right) \tag{5.13}$$

And

$$\eta_{N,S}(s) = \eta_{N-1,S}(s). \tag{5.14}$$




(a) Expected waiting time




(b) Variance of waiting times

**Figure 5.4** Expected waiting times and variance of waiting times for packets eventually served with a traffic intensity $\rho = 0.75$

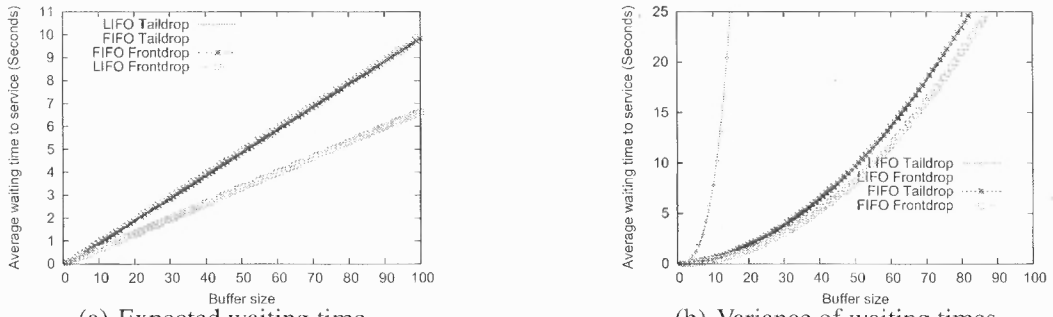
In (5.13), the sum is for $j$ from 1 to $k$, because the $(k+1)$-th packet to drop before the packet starting in position $k+1$ is served would be that packet itself.

### 5.3.4 Numerical Results

Using the expressions derived in Section 5.3.3, the expected waiting time as well as the variance of waiting times of packets eventually served are assessed for different traffic intensities using all four queuing disciplines. The results are presented in Figures 5.4 to 5.6.

**Queue with low traffic:** $\rho < 1$ Two useful observations can be drawn from Figure 5.4. First, the values of the expected waiting time and variance of waiting times stabilize to constant values as the queue size increases, independently of the discipline. Because the service rate is greater than the arrival rate, there is (practically) no drop of packets, thus no difference between Taildrop and Frontdrop performance.

Second, the variance of waiting times of packets eventually served are higher under LIFO than FIFO. With LIFO, certain packets will stay longer in the queue because of new




(a) Expected waiting time

(b) Variance of waiting times

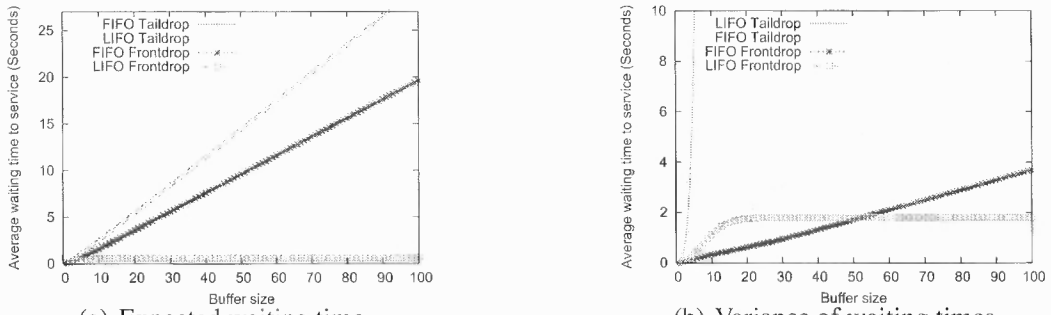
**Figure 5.5** Expected waiting times and variance of waiting times for packets eventually served with a traffic intensity $\rho = 1.0$

packet arrivals, yet they are eventually served because the traffic intensity makes it unlikely that the queue becomes full ($\rho < 1$).

**Queue with even traffic:** $\rho = 1$ Figure 5.5 shows that when $\rho = 1$, unlike the case of $\rho < 1$, the expected waiting time of successful packets increase linearly with the buffer size for all four disciplines, albeit with different slopes. In the context of FIFO, the linear increase results from packets gradually moving to the head of the queue to be serviced: the longer the queue, the longer the time needed to reach the head position.

Another observation in the case of Taildrop, the average waiting time does not depend on FIFO–LIFO. To understand why, let's consider $\overline{W}$ as the average waiting time of all packets, and $\overline{W_s}$, $\overline{W_d}$ as the average waiting time until service and drop, respectively. Then, regardless of the discipline, $\overline{W} = (1 - \pi_N)\overline{W_s} + \pi_N \overline{W_d}$. In case of Taildrop, packets are dropped upon arrival and encounter no waiting at all. Thus $\overline{W_d} = 0$ and $\overline{W_s} = \frac{\overline{W}}{1-\pi_N}$ regardless of whether it is FIFO or LIFO and independently of the traffic intensity.

The issue is different for the variances, with LIFO leading to very high variances of waiting times compared to FIFO (Figure 5.5(b)). This high variance can be explained by the




(a) Expected waiting time

(b) Variance of waiting times

**Figure 5.6** Expected waiting times and variance of waiting times for packets eventually served with a traffic intensity $\rho = 1.5$

fact that LIFO has a higher probability of almost no waiting, but also a higher probability of a long wait. Because the packets arrival at the queue equals the rate of departure from the queue, little difference is observed between FIFO-TD and FIFO-FD. The asymptotes of the variance with FIFO are similar for both Taildrop and Frontdrop.

**Queue with high traffic:** $\rho > 1$ Figure 5.6 shows the differences in performance of the four disciplines when $\rho > 1$. The expected waiting time of packets eventually served is proportional to the size of the queue under FIFO-TD, FIFO-FD, and LIFO-TD. Figure 5.6(a) confirms the previous analysis showing identical expected waiting times results for FIFO-TD and LIFO-TD. Comparatively, FIFO-FD has lower values of expected waiting times than FIFO-TD. More specifically, for $\rho > 1$ and large buffer sizes, the expected waiting time with FIFO-TD is roughly $\rho$ times the expected waiting time of FIFO-FD. Indeed, with $\rho > 1$ and a buffer size $N$, the waiting time under FIFO-TD is about $\frac{N}{\mu}$ time units, while the waiting time under FIFO-FD is about $\frac{N}{\lambda}$ time units. In both cases this is the waiting time from entering into a full system until reaching the head of the queue.

In contrast to the other disciplines, under LIFO-FD, the expected waiting time values stabilize as the queue size increases. Because the arrival rate in the queue is greater than the departure rate, packets served are more likely not to have reached the front of the queue.

Figure 5.6(b) shows the variances of waiting times, focusing on FIFO and LIFO-FD. The variance of expected waiting times have linearly increasing asymptotes under FIFO-FD and FIFO-TD while the asymptote is constant for LIFO-FD. With Frontdrop and $\rho > 1$ and a large buffer, a packet will be discarded without being served with probability $\frac{\mu}{\lambda+\mu}$. The discarded packets are also the oldest in the queue, which explains the smaller increase of variance of waiting times for FIFO-FD and LIFO-FD.

**Summary of Results** These results suggest that the Frontdrop policy enables FIFO and LIFO to have smaller expected waiting times as well as smaller variance of waiting times under congestion when compared to Taildrop. In particular, the ability of LIFO-FD to stabilize both the expected waiting times and the variance of waiting times of packets served indicates that it may be a suitable queuing discipline under congestion.

The results also suggest that FIFO requires the buffer size to be carefully chosen to make it work under light load ($\rho < 1$) as well as heavy load ($\rho > 1$). The buffer size should not be too small (to prevent loss in light traffic) or too large (to prevent huge delays in heavy traffic). LIFO is much less sensitive. In particular, LIFO-FD works as long as the buffer is large enough.

LIFO-TD would need a separate mechanism to discard old packets (packets no longer of interest to anybody) to still work when $\rho > 1$ even with large buffers. LIFO-FD has this mechanism built-in implicitly.

## 5.4 Network Simulations

This section presents the evaluation through simulations of data transfers in ad hoc networks under the four queuing disciplines. The goals of these simulations are to (1) validate the results for single queue obtained in the previous section, (2) understand whether the theoretical conclusions for single queue (i.e., LIFO-FD works best for high traffic situations, and FIFO works best for low traffic situations) can extend to multi-hop static ad hoc networks, (3) quantify the benefits of different queuing disciplines in terms of end-to-end UDP delay and jitter (and implicitly quantify the benefits for multimedia streaming in ad hoc networks), (4) identify any potential negative effects of queuing disciplines on fairness and/or TCP, and (5) understand the impact of node mobility on the end-to-end performances (i.e. whether the benefits observed in static ad hoc networks apply to mobile ad hoc networks as well).

The simulations are conducted using the Network Simulator NS-2.30 [71]. Table 5.1 summarizes the simulations settings. Simulations are conducted for both UDP and TCP. In the UDP simulations, the data sources inject UDP traffic at a Constant Bit Rate (CBR) with variable sending rates. In the TCP simulations, the sources use FTP and TCP-Reno to transfer files of 2MB-5MB size.

### 5.4.1 Metrics

The network performance under all four queuing disciplines is evaluated by varying the queue size, the CBR data rates, and the numbers of concurrent flows. The metrics used to assess the performance are the following:

- **Average end-to-end delay.** This metric is defined as the average delay incurred in the transmission of all data packets delivered successfully. This does not include

| Parameter | Value |
|---|---|
| Simulation area | 1200m x 600m |
| Number of nodes | 16, 250 |
| Number of TCP/UDP sources | 1-10 |
| Transmission range | 250m |
| Simulation time | 100s-1000s |
| CBR rate | 1 - 20 packet per second |
| MAC protocol | IEEE 802.11 DCF |
| Data packet size | 1000 bytes |

**Table 5.1** Simulation Setup

the delay due to re-sequencing at the destination (i.e. waiting for packet with lower sequence number or lower timestamp). The average end-to-end delay is directly influenced by the queuing approach.

- **End-to-end jitter.** This metric is defined as the variation in the end-to-end delay values of successive data packets successfully received at destination. Lost packets are ignored in this evaluation. The lower the jitter, the better the performance of multimedia streaming applications. The jitter also quantifies the effect of the out-of-order delivery of LIFO discipline compared to FIFO to answer questions such as: will LIFO impact negatively the TCP performance?

- **Throughput.** In this chapter, this metric is defined as the number of data packets successfully delivered at destinations per time unit. The throughput shows the effect of the queuing discipline on the useful data transferred end-to-end.

### 5.4.2 Static Network Simulations

**Topology** The static ad hoc network topology is shown in Figure 5.7. The topology contains 16 wireless nodes and data is transferred between pairs of nodes ($s_i$,$r_i$). This topology was selected to ensure, in a simple and controlled way, that the set of nodes $n_i$

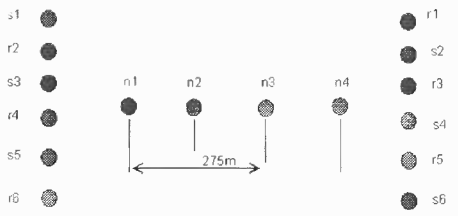
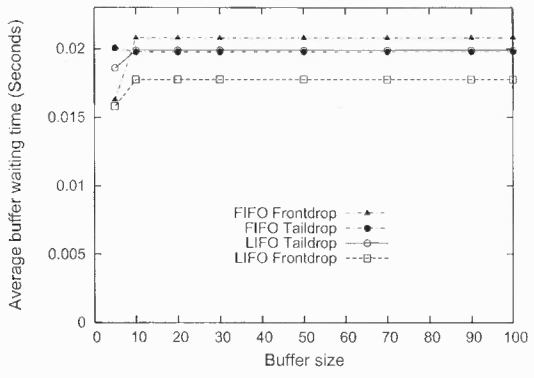



**Figure 5.7** Topology used in the static ad hoc network simulation study

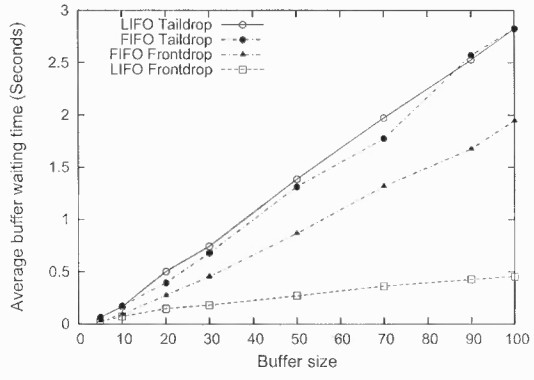
become a bottleneck during the transmissions due to both shared medium contention and congestion. Besides increasing the data sending rates, the addition of transmitting pairs ($s_i$,$r_i$) increases the data traffic forwarded through nodes $n_i$, making them bottlenecks. The distance between the $n_i$ ensures that each packet forwarded from source to destination will hop through all four $n_i$ nodes (i.e., the order $n_1$, $n_2$, $n_3$ and $n_4$, or vice-versa, is always maintained). This topology allows us to examine the queues of nodes $n_i$ to understand single queue behavior, while the end-to-end measurements will give the performance of the network.

In all the simulations, the value of the queue (buffer) size is set to the same value for all nodes. At the network level, greedy geographical routing is used to transfer data between communicating pairs. Each node periodically broadcasts "hello" packets to maintain the neighborhood tables required by geographical routing.

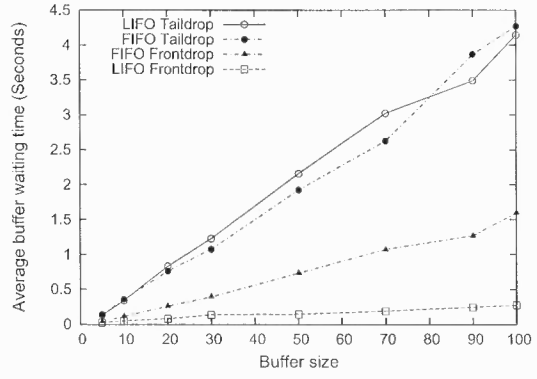
**Single buffer** First, one would like to confirm the results of the analytical model described in Section 5.3. For this purpose, one considers the individual $n_i$ queues and observe the time spent by packets at each queue. As expected, the nodes $n_1$ and $n_4$ contribute for most to the end-to-end delay. This is because the source flows use these nodes as the entry points toward the destination nodes, leading to high traffic contention around $n_1$ and $n_4$.




(a) UDP CBR sending rate = 5pkt/s




(b) UDP CBR sending rate = 10pkt/s




(c) UDP CBR sending rate = 20pkt/s

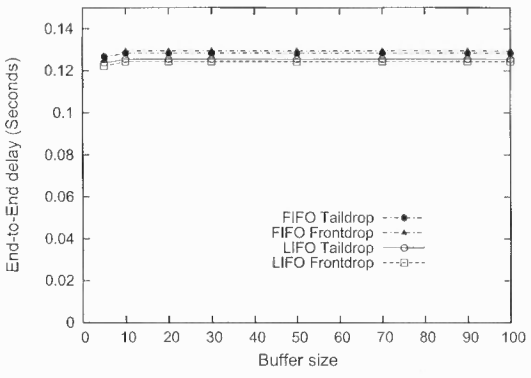
**Figure 5.8** Average buffer waiting time vs. buffer size at node $n_1$, UDP CBR data traffic with different sending rates

Figure 5.8 presents the average buffer waiting times at host $n_1$ for the four disciplines, under different traffic intensities. The data sending rate is used as an indicator of the traffic intensity, with 5pkt/s corresponding to light traffic (i.e. $\rho < 1$) and 20pkt/s corresponding to heavy traffic (i.e. $\rho > 1$). Identifying a data rate corresponding to $\rho = 1$ is less trivial in wireless network and the data sending rate of 10pkt/s is used as a middle traffic load between light and heavy traffic.

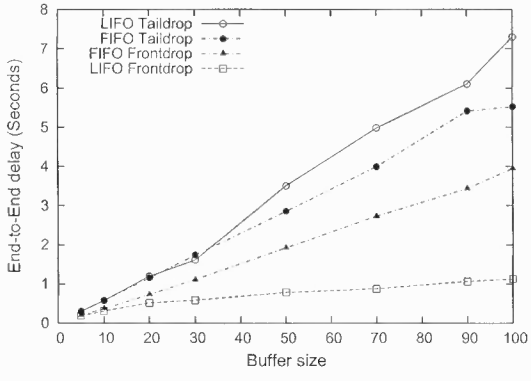
For light data traffic (figure 5.8(a)), the performance of the four disciplines are comparable. Under heavy load (figure 5.8(c)), however, LIFO-TD performs best as it enables packets to spend less than 300ms waiting time at the queue, independent of the queue size. These results are comparable to those of the analytical model (Figure 5.6), and the explanation presented in Section 5.3 remains the same.

**End-to-end delay** Figure 5.9 presents the measured average end-to-end delay, obtained with 6 simultaneous UDP CBR flows, as function of the queue (buffer) size. Observe that the drop policy impacts the end-to-end delay sooner than the service discipline. For queue sizes of 20 or less, the end-to-end delay values are about identical for LIFO and FIFO. The reason is that the time required by a packet to reach the head of the queue is low for relatively small size queues. Thus, the impact of servicing the next packet from the head or the tail of the queue is reduced in small size queues.

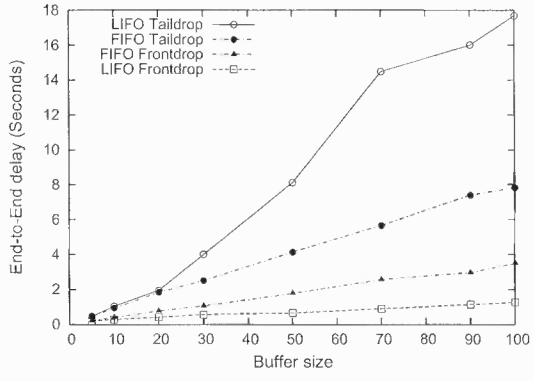
Figure 5.9 also shows that the end-to-end delay values are comparable under light data traffic (5pkt/s). Under high data traffic (20pkt/s), LIFO-FD provides the lowest measured delays. This result confirms that the measured end-to-end performance results match with the results of a single queue.




(a) UDP CBR sending rate = 5pkt/s




(b) UDP CBR sending rate = 10pkt/s




(c) UDP CBR sending rate = 20pkt/s

**Figure 5.9** Average end-to-end delay vs. buffer size, UDP CBR data traffic with different sending rates

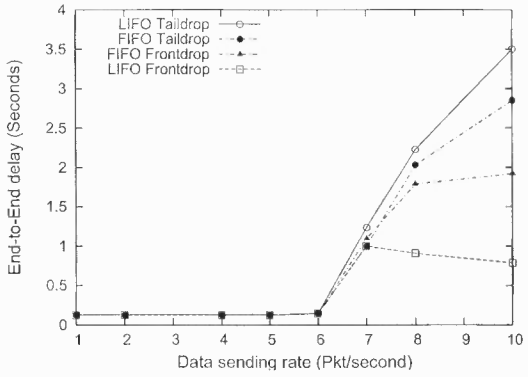



**Figure 5.10** Average end-to-end delay vs. UDP CBR sending rate, buffer size = 50 slots

Next, the impact of UDP CBR sending rate on a single queue size is evaluated (figure 5.10). Before analyzing this figure, it is important to address the selection of the queue size (50 slots) used for this set of simulations. From the analytical results (Section 5.3.3), it is known that the FIFO disciplines are particularly impacted by the size of the queue. How to determine the correct queue size is a current research topic [85, 86] both for simulation and industrial purposes. The number of slots was set to 50 slots because it is the most common configuration in simulations (default value in NS2 [71]) and it is the value recommended for routers used in some real-life networks [87].

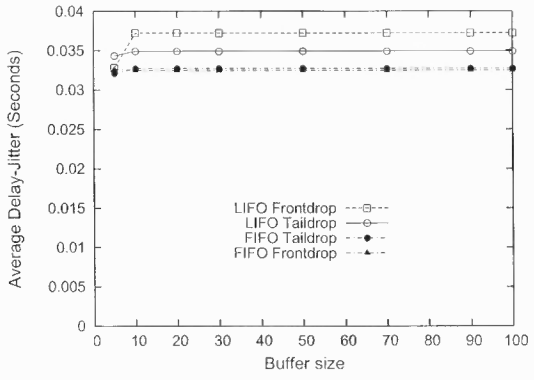
Figure 5.10 shows that the Frontdrop policy leads to shorter average end-to-end delay for both FIFO and LIFO. For small values of the sending rates (up to 6pkt/s for this topology), all the queuing disciplines have similar results. However, for rates of 7pkt/s and up, the queues become full and the results differ considerably. LIFO-FD has the shortest average end-to-end delay followed by the FIFO-FD. These results are similar to Figure 5.4(a) (for up to 6pkt/s) and Figure 5.6(a) for 7pkt/s and up. The reasons explained for those figures apply here as well.

**End-to-end jitter** So far, the results showed that LIFO-FD performs best in terms of end-to-end delay under high traffic load, while the four disciplines showed comparable results under light traffic load. Now one examines the end-to-end jitter for the same transmissions. Considering that LIFO is expected to have more out-of-order packets at destinations, the question is whether or not its end-to-end jitter has acceptable values that do not counterbalance the gains in terms of end-to-end average delay.

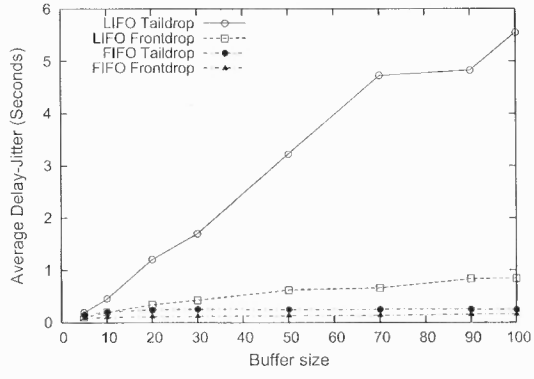
To answer this question, one measures the end-to-end jitter between successive data packets at the destination nodes. In the event of a missing packet (e.g., dropped), that packets is ignores and the next successive packets to reach the destination are considered. The end-to-end jitter of packet $i+1$ is obtained as $jitter(i+1) = |(recv(i+1) - send(i+1)) - (recv(i) - send(i))|$, where $send(i)$ is the time packet $i$ is sent at the source of the flow and $recv(i)$ is the time packet $i$ is received at its intended destination.

Figure 5.11 shows that FIFO-FD performs best in terms of end-to-end jitter, across all the CBR traffic rates. FIFO-FD takes advantage of the Frontdrop policy and additionally serves the packets in order, causing fewer end-to-end out of order deliveries. LIFO-FD and FIFO-TD present similar average end-to-end jitter for higher data traffic rates. This result in conjunction with the one that showed significantly lower end-to-end delay for LIFO-FD demonstrate that LIFO-FD should be the selected discipline when networks experience high traffic periods.

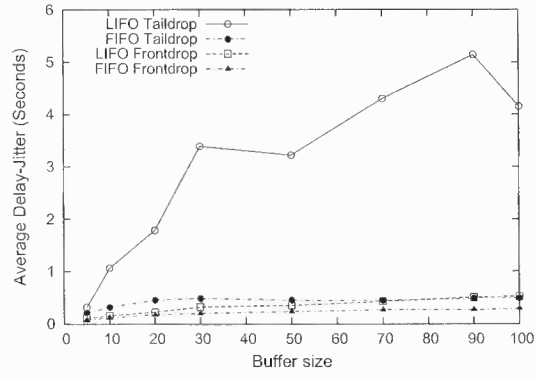
**UDP throughput** The preceding experiments showed that LIFO-FD provided the smallest end-to-end delay and similar jitter compared to FIFO-TD. These improvements do not adversely affect UDP data throughput as shown in Figure 5.12. The amounts of data transferred are similar when using FIFO-FD or LIFO-FD. This is expected because in both




(a) UDP CBR sending rate = 5pkt/s




(b) UDP CBR sending rate = 10pkt/s




(c) UDP CBR sending rate = 20pkt/s

**Figure 5.11** End-to-end jitter with UDP CBR traffic, UDP CBR data traffic with different sending rates

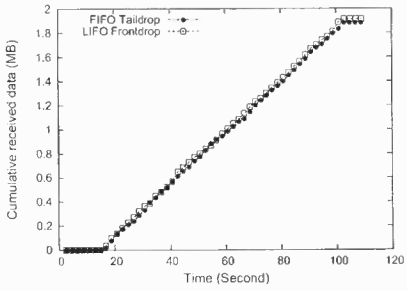
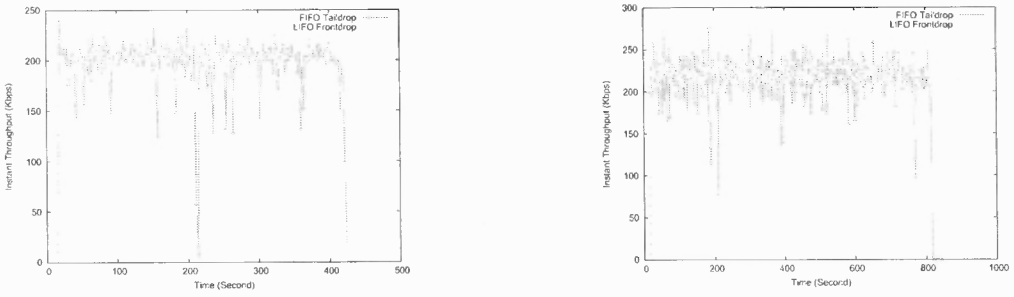



**Figure 5.12** Total instantaneous UDP data received, UDP CBR sending rate = 10pkt/s, buffer size = 50 slots



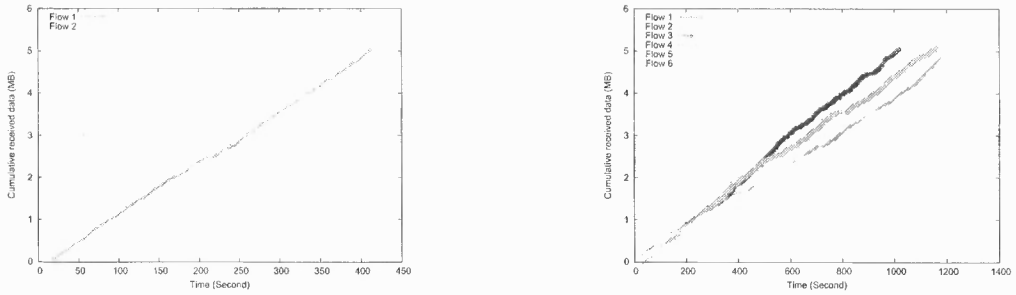

(a) 2 flows, 5MB each

(b) 10 flows, 2MB each

**Figure 5.13** Total instantaneous TCP throughput, Buffer size = 50 slots

disciplines a direct consequence of a host overload is the continuous presence of a packet to serve at that queue. Consequently, while LIFO-FD does provide significant reduction in the end-to-end delay, it does not adversely affect the amount of transferred data with UDP.

**TCP throughput and fairness** While LIFO-FD proved to work best for UDP transfers overall, the following experiments will compare the TCP throughput under LIFO-TD and FIFO-TD (i.e., the commonly used queuing discipline). Specifically, one would like to see if LIFO-FD leads to too many duplicate acknowledgments and consequently to window




(a) 2 flows, 5MB each

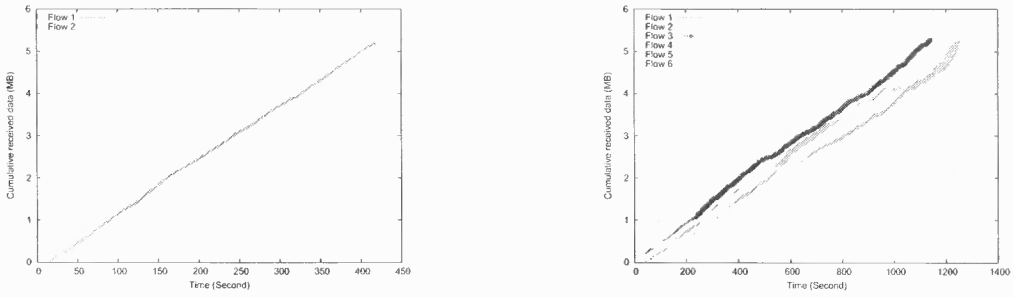
(b) 6 flows, 5MB each

**Figure 5.14** TCP fairness: per flow cumulative received data, FIFO-TD, Buffer size = 50 slots

collapse due to out-of-order deliveries. The maximum TCP window size in NS2 corresponds to 20000 bytes in these simulations, and the buffer size at a host is set to 50 slots.

Figure 5.13 shows how the total instantaneous TCP throughput (over all flows) varies over time for the two queuing disciplines. The instantaneous throughput is the measured throughput for every 2 seconds interval. Although the instantaneous throughput differs between the two disciplines on specific intervals, the results show that the transfers complete almost at the same time. Obviously, these results show that the potential problem of TCP window collapse does not happen in the simulations of LIFO-FD performed. In none of the experiments did network congestion happen (i.e., dropped packets due to full queues) because the TCP congestion control algorithm kicked in earlier due to long delays caused by contention or lost packets caused by channel errors. This also explains the relatively low throughput of about 220Kbps for both disciplines. Such throughput results are not surprising in multi-hop ad hoc networks, as documented in [88].

The question addressed next is whether or not LIFO-FD provides the same level of fairness for TCP flows as FIFO-TD. The level of fairness is quantified by checking if some




(a) 2 flows, 5MB each

(b) 6 flows, 5MB each

**Figure 5.15** TCP fairness: per flow cumulative received data, LIFO-FD, Buffer size = 50 slots

flows are starved during transmissions. The literature [89] has proposed ways to improve the fairness of 802.11 DCF networks, mainly assuming the FIFO-TD policy. The goal here is not to propose ways to improve the fairness of 802.11 ad hoc networks, but to assess whether under similar circumstances, a queuing discipline results in flow starvation.

Figure 5.15 shows that no particular flow is starved under LIFO-FD, and the completion times are similar to those of FIFO-TD (figure 5.14) when multiple flows are transferred (figure 5.15(b) and figure 5.14(b)). Thus, LIFO-FD does not disproportionally alter how TCP flows are transmitted, when compared to FIFO-TD.

### 5.4.3 VANET Simulations using RBVT-R

**Simulation Setup** Since the expected live streaming applications may often run over mobile ad hoc networks with a larger number of nodes than in the previous experiments, the second simulation scenario uses a 250-node vehicular ad hoc network on a 1500m x 1500m area extracted from the TIGER/Line database of the US Census Bureau [66]. The region simulated is an area of Fellsmere, FL with center point coordinates latitude

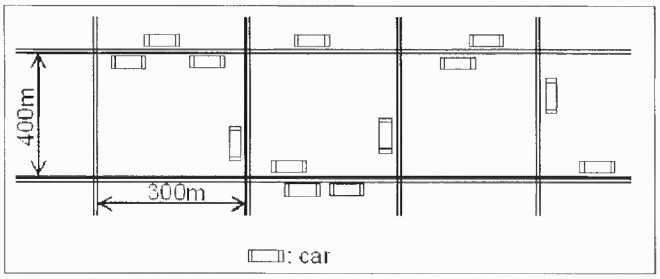


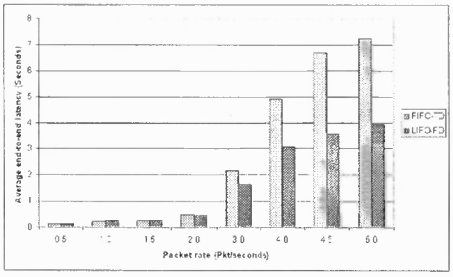

**Figure 5.16** Topology used in the mobile ad hoc network simulation study

27.784728° North and longitude 80.604385° West. The roads form a grid layout as shown in Figure 5.16 with bi-directional traffic and two lanes in each direction.

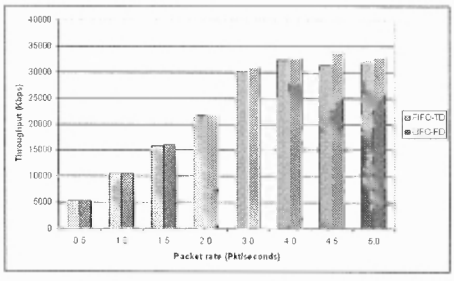
The vehicles movements are generated using a microscopic mobility generator based on the car-following and lane-changing models proposed by Gipps [83, 84]. When a vehicle enters a road segment, its action at the end of the segment (i.e., left turn, right turn, u-turn, or straight ahead) is based on the average traffic flows of the roads crossing the end intersection. The first 2000 seconds of output are discarded to obtain more accurate movements of nodes. The simulations run for 400 sec.

The routing protocol used is RBVT-R [1]. RBVT-R is a reactive source routing protocol which creates stable paths consisting of road segments that have enough vehicles on them to avoid broken routes when geographical forwarding is employed.

**Simulation Results** Figure 5.17 demonstrates not only that LIFO-FD results in lower delay (as much as 45%) than FIFO-TD but also that it does not adversely affect the delivery ratio in mobile ad hoc networks. When one compares the static network results, Section 5.4.2, to those obtained in the mobile network, one sees that the static network provides slightly better results than the mobile network. The reasons for this can be attributed, in part, to the average number of hops which was higher in the mobile network simulations, 7




(a) Average end-to-end delay




(b) Average throughput

**Figure 5.17** Average end-to-end delay and average throughput comparison: 802.11 LIFO-FD queuing vs. 802.11 FIFO-TD queuing. The routing protocol is RBVT-R, and the network size is 250 nodes. 30 nodes exchange UDP CBR data traffic.

hops compared to 5 hops used for the static network. Additionally, the node movements sometimes lead to periods of broken connectivity between certain source and destination pairs. In those instances, the routing algorithm must repair or recreate a new communication path, which increases the overall end-to-end transmission delay.

## 5.5 Chapter Summary

This chapter studied the relative performance of four simple queuing disciplines, FIFO-TD, FIFO-FD, LIFO-TD, and LIFO-FD, to understand if another discipline can perform better than the commonly-used FIFO-TD in vehicular ad hoc networks. The focus was on improving the performance of real-time multimedia applications, which are already used in real-life scenarios and have the potential to become the main driving force behind vehicular ad hoc networks. More exactly, the goal was to find which queuing discipline works best for UDP traffic in terms of latency and jitter, without compromising other parameters such as throughput, or protocols such as TCP or RBVT.

The main conclusion of this chapter is that LIFO-FD performs best for high traffic periods, and FIFO-TD performs best for low traffic periods. While LIFO is generally perceived as unfair, the theoretical analysis and simulation results demonstrated that this unfairness is tied only to LIFO-TD. LIFO-FD achieves similar TCP fairness and throughput with FIFO-TD.

# CHAPTER 6

## VANET FORWARDING USING ESTIMATED PATH DURATION

The expected lifetime of a communication route can be incorporated in the forwarding process to improve the performance of data transfers. In node-centric routing, this expected lifetime depends on the mobility of the intermediate nodes belonging to that route. In RBVT, on the other hand, it does not depend on intermediate nodes (i.e., geographical forwarding); it depends only on the existence of network connectivity between the endpoints. Therefore, estimating the expected lifetime of an RBVT route can be determined by estimating the expected connectivity duration between the intersections defining that route.

This chapter presents two discrete-time and discrete-space Markov chain (DTMC) models for estimating the expected lifetime of RBVT routes. The first model is *DTMC-CA*, an analytical model based on the Cellular Automaton (CA) freeway traffic model [90]. DTMC-CA computes the expected duration of connectivity and disconnectivity between 2 nodes. DTMC-CA can also be used to determine MANET-type average path durations. Evaluation results show the validity of the proposed model. Second, a generalization of the DTMC-CA model called *Connectivity Window* (CW) is presented. CW can be used in conjunction with any vehicular traffic generator because it abstracts the microscopic information of traffic. Unlike analytical models for MANETs, the proposed models do not assume independence between successive links on the path. This, however, comes at the cost of a potentially large state space. Two methods are proposed to decrease the state space by more than 90% without any loss of information for the measures of interest.

The remainder of this chapter is organized as follows: Section 6.1 motivates the study. The cellular automaton traffic model used in the analysis is described in Section 6.2 while Section 6.3 describes the DTMC-CA model along with the concept of lumpability of Markov chains. Section 6.4 presents numerical results of the evaluation of the DTMC-CA model. An approximation method for longer paths is described in Section 6.5. Section 6.6 describes the Connectivity Window model, while several ways of incorporating path estimates in RBVT protocols are presented in Section 6.7 The chapter concludes in Section 6.8.

## 6.1 Motivation

Although it provides better performance than other VANET routing protocols, RBVT still suffers from relatively frequent broken routes. When a disconnection occurs, the sender can either (1) simply drop the packets and create a new route to the destination, or (2) decide to use greedy geographical routing to route data, or (3) wait for a while and re-attempt to transmit on the same route. Dropping packets is avoided whenever possible and using greedy geographical routing has some drawbacks in VANET, which have been discussed in previous chapters. Some protocols, such as RBVT-R, select to wait for a while and re-attempt to use the same route. However, the lack of knowledge of the proper amount of time the node should wait reduces the effectiveness of this approach. Note that if the vehicle knows the expected duration of an RBVT path, it could, alternatively, decide to create (or select from a cache) a new RBVT path before the current one breaks.

The expected duration of connectivity between two communicating endpoints is defined as the average time a multihop path will exist between the endpoints, over all potential paths (Figure 6.1). RBVT and other VANET protocols performance could be

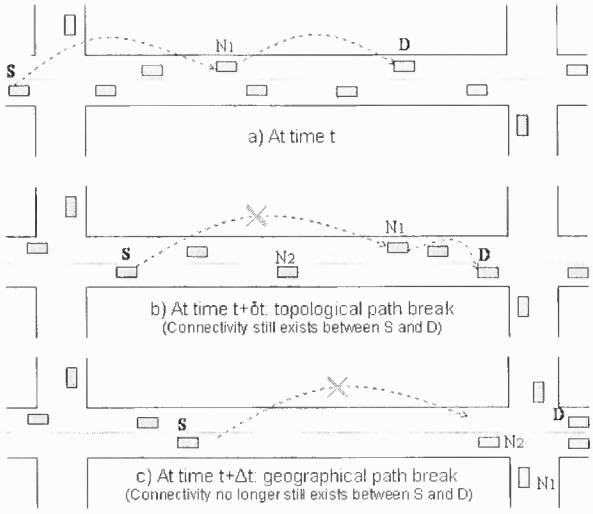



**Figure 6.1** The models presented in this chapter deal with the expected duration of connectivity between a source S and a destination D, (case c), as opposed to the expected duration of a topological path (case b). The topological (or node-centric) path breaks because the distance between $S$ and $N_1$ exceeds the transmission range at time $t + \delta t$. However, the connectivity is maintained because $S$ can still communicate with $D$ through $N_2$.

improved if provided with the ability to determine, dynamically, the duration of continued multihop connectivity. Such knowledge can be used to answer questions such as:

- How long can one expect the currently employed route to last?
- Can a 100Mb file be successfully transferred on the current route?
- Is it possible to estimate, efficiently, the duration of a path disconnection?

Existing approaches for estimating the duration of connectivity in multihop VANET are based on simulations [63]. Using a validated vehicular traffic mobility model [90], the authors collected data by running simulations on an input map. This method provides valuable insight into the connectivity patterns of the study map. However, new simulations will likely be needed for other areas of interest. Analytical approaches, on the contrary,

allow for estimations based on the characteristics of the network and they can easily be applied to new regions. To the best of our knowledge, no analytical model has been proposed to estimate the duration of connectivity in multihop VANET, although a number of models have been proposed to estimate path durations in mobile ad hoc networks (MANETs) [58, 59, 91, 92, 93]. One may argue that MANET models can be applied to VANETs by constraining the movements of the nodes to road structures and increasing the available node speeds. Such approach would create two issues. First, the MANET models consider a path as a succession of predefined links between the endpoints. In general, the duration of the connectivity will be higher than the duration of a given path. As shown on Figure 6.1, when one path breaks, other paths might be available to maintain connectivity. Second, the mobility models used in the MANET approaches do not accurately approximate VANET mobility. Random Way-point [25] is the most common model in MANETs. However, it does not account for vehicular traffic interactions such as acceleration and slow down due to the presence of other vehicles.

## 6.2 Background

This section briefly describes the microscopic one-lane freeway cellular automaton (CA) traffic model [90] used in the analysis. It is a discrete-time and discrete-space stochastic traffic model which was selected because of its computational simplicity. The CA traffic model divides a road in cells of fixed length $L_c$ (Figure 6.2). At any point in time, a cell is either empty or occupied by at most one vehicle. Each vehicle has a speed $v \in \{0, 1, \ldots, v_{max}\}$ where the speed represents the number of cells advanced by a vehicle in one step (one step corresponds to one unit of time). The state of the system is updated at

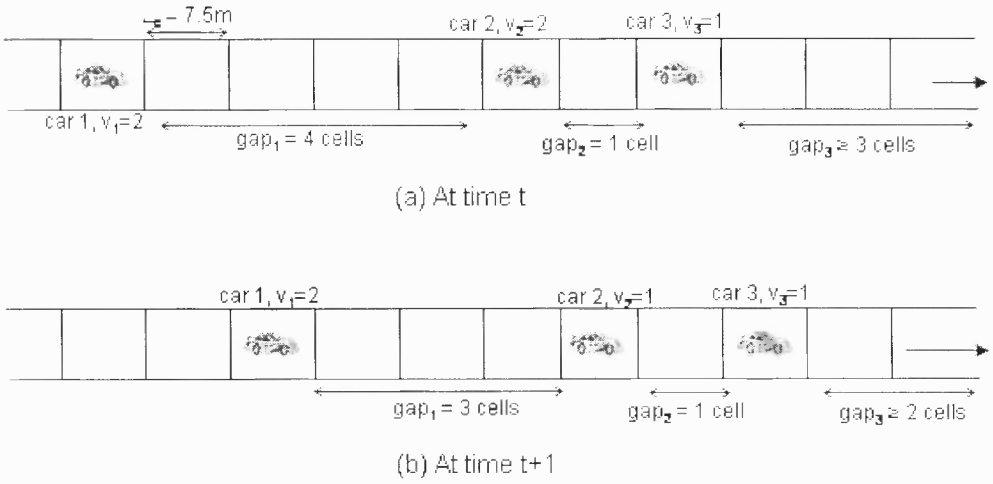



**Figure 6.2** A single step parallel update using the Cellular Automaton freeway traffic model.

discrete time steps by applying, in parallel, the four rules listed below to all vehicles. The space interval between vehicle $i$ and its preceding vehicle is $gap_i$. Then,

1. Rule 1: Acceleration; If $v_i < v_{max}$, $v_i = v_i + 1$.

2. Rule 2: Slow down due to other cars; If $v_i > gap_i$, then $v_i = gap_i$.

3. Rule 3: Stochastic behavior; If $v_i > 0$, then $v_i = v_i - 1$ with probability $p$.

4. Rule 4: Move vehicle; $x_i = x_i + v_i$.

Figure 6.2 illustrates a step of parallel update of the system. At time $t$ (Figure 6.2(a)), vehicles 1, 2 and 3 have speeds of $v_1 = 2$, $v_2 = 2$ and $v_3 = 1$ respectively. At the end of the update period, vehicles 3 and 2 have their speeds equal to 1 while $v_1 = 2$. Vehicle 2 must decrease its speed because of the presence of vehicle 3 (application of rule 2, with $gap_2 = 1$, result in $v_2 = 1$). Note that the above rules apply to a single-lane road. Rules for multilane traffic, including vehicle passing can be found in [94].

## 6.3 DTMC-CA Model

This section presents DTMC-CA, a discrete-time and discrete-space Markov model for estimating the expected connectivity duration in vehicular networks using the Cellular Automaton freeway traffic model. Recall that connectivity between two vehicle nodes is maintained as long as intermediate nodes can be found to form a multihop communication path, independently of the specific intermediate nodes used. A multihop path is a list of nodes such that the distance between any pair of successive nodes in the list is at most equal to the transmission range $r$. The following description assumes a single lane, unidirectional road (traffic moving from left to right) or a two-lane bidirectional road.

### 6.3.1 Description

Consider a road with vehicles moving with velocities from a set $\{0, 1, \ldots, v_{max}\}$. Now consider the section of road located between 2 vehicles in communication. The road section of interest is divided in $k$ cells of fixed length $L_c$. A value from the set $V = \{0, 1, \ldots, v_{max}, \infty\}$ is associated with each cell. A positive value $c_i > 0, c_i \in V$ corresponds to a vehicle in cell $i$ moving with speed $c_i$. A cell value of 0 represents a stationary vehicle while a cell value of $\infty$ corresponds to an empty cell. To simplify the presentation, the following description first assumes that the communication endpoints are static (source and destination have speeds $v_s = 0$ and $v_d = 0$ respectively). Thus, the distance between the endpoints will not change during the communication. This assumption will be relaxed in the later part of this section.

A Markov chain $M = (S, P, s_0)$ is constructed with state space $S = \{s = (c_1, c2, \ldots, c_k) : c_i \in V$, for all $1 \leq i \leq k\}$ and transition probability matrix $P$. No assumption is made on the starting state $s_0$ of the Markov chain as only stationary measures are of interest in this

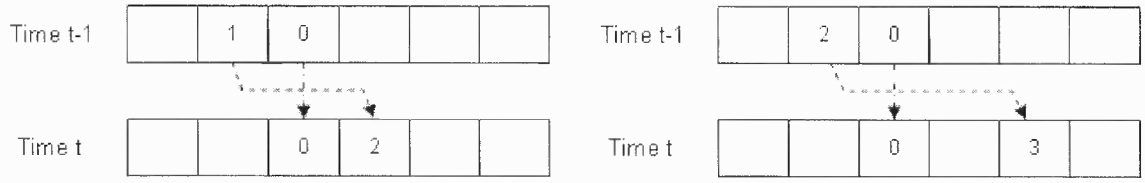



**Figure 6.3** Invalid states in the DTMC-CA model. On a single lane road, the configurations at time t are not possible because they would involve vehicles crossing (or flying above) other vehicles.

study. As described, the size of the state space would be $|S| = |V|^k$. However, it is possible to reduce this size considerably (more than 95%) using the properties of the mobility model as well as techniques such as lumping of Markov chains.

### 6.3.2 State Space Reduction

Two techniques of reduction of the state space are described. Each state $s \in S$ is of the form $s = (c_1, c_2, \ldots, c_k)$ with $c_i \in V$, for all $1 \leq i \leq k$. If one is to consider all possible instances of $s$, the size of the state space $S$ would grow very quickly. However, keeping all those states would be unnecessary because further examination shows that many of those states would violate the rules of the cellular automaton model (Section 6.2). The first reduction technique identifies those instances of $s$ which violate the rules and removes them. The second technique aggregates the remaining states in logically equivalent states using lumping.

**Invalid States** The first reduction technique prevents the inclusion in $S$ of states that violate the update rules of the mobility model used. These are transient states which cannot be reached from any other state in the system. For example, considering a road with $k = 6$ cells and $v_{max} = 2$, one such invalid state is $s = (\infty, \infty, 0, 2, \infty, \infty)$. This state is invalid because of the pair of speeds $(0, 2)$ located on consecutive cells. As shown in

Figure 6.3, such a state is invalid on a single-lane road because it would have resulted from vehicles passing each other on a lane road, which is not allowed in the traffic model. From the configurations in Figure 6.3 at time $t-1$, application of rule 2 would have resulted in the vehicle at the left slowing down to speed 0 at time $t$. Using a similar reasoning, state $s = (\infty, \infty, 1, \infty, 3, \infty)$ is also found to violate the rules of the system (because of the configuration $(1, \infty, 3)$). Algorithm 3 outputs a state space set containing only non-transient states, for a given value of maximum speed $v_{max}$ and number of cells $k$. This algorithm does not work by generating all possible states and then removing the invalid ones because the algorithm would run out of memory very quickly (e.g., for $k = 10$ and $v_{max} = 5$, temporarily generating all possible states would require $7^{10}$ instances of vectors of 10 numbers). Instead, the formation of those states that are known to contain invalid configurations is prevented. Lines $2-7$ show how this is achieved for $k = 1$, 2 and 3. For larger values of $k$ (lines $9-19$), the algorithm uses the set obtained for $k-1$ and adds one position to the left of each vector elements. Note that the invalid states concept is an extension of *Garden of Eden* states described in [95].

Table 6.1 clearly confirms that removing the invalid states greatly decreases the state space size, with reduction of 85% or more when $k = 5$. The greater the maximum speed, the bigger the benefits of removing invalid states from the state space set.

**Lumpability of Markov Chains** The next state-reduction technique applies the concept of lumpability to create an aggregated Markov chain without losing any information for the performance measures of interest. One notes that other aggregation techniques can be found in the literature (e.g., nearly completely decomposability (NCD) [96, 97, 98]). In this study, the choice was made to lump the Markov chain because unlike other methods

**Algorithm 3** Generation of Valid State Space S with length $k$ states and Maximum Speed $v_{max}$

Notation:

$S_m$: State space set with element of the form $(c_1, c_2, ..., c_m)$
$S_{m,i,j}$: Subset of $S_m$ with value $i$ at position $j$
$V = \{0, 1, ..., v_{max}, \infty\}$: Set of possible values of $c_i$
$REP(i, m)$: Replicate value $i$, $m$ times
$(c, S)_r$: Add $c$ to beginning of each element of set $S$
$S[i : j]$: Extract positions $c_i$ through $c_j$ of each element of set $S$

**for** $l \leftarrow 1, k$ **do**
  **if** $l == 1$ **then**
    $S_1 = V$
  **else if** $l == 2$ **then**
    $S_2 = \{(c_1, c_2), c_1, c_2 \in V, c_1 = \infty$ or $c_2 = \infty$ or $c_1 = 0, c_2 = 0\}$
  **else if** $l == 3$ **then**
    $S_3 = \{(c_1, c_2, c_3), c_1, c_2, c_3 \in V, (c_1 = \infty, (c_2, c_3) \in S_2)$ or $(c_1 \in S_1, c_2 = \infty, c_3 = 1)$ or $(c_1 = 0, c_2 = 0, c_3 = 0)\}$
  **else**
    $tmpv0 = (0, S_{l-1,0,1})_r \cup (0, \infty, S_{l-1,1,1}[2 : end])_r \cup (0, \infty, S_{l-1,1,2}[2 : end])_r$
    $tmpv1 = tmpv0$
    **for** $e \leftarrow 2, min(l - 2, v_{max})$ **do**
      $tmpv1 = tmpv1 \cup (1, REP(\infty, e), S_{l-1,e,e+1}[e + 1 : end])_r$
    **end for**
    $S_l = (\infty, S_{l-1})_r$;
    $S_l = S_l \cup tmpv0 \cup tempv1$
    **for** $e \leftarrow 2, v_{max}$ **do**
      $S_l = S_l \cup (V(e), tmpv1[2 : end])_r$
    **end for**
  **end if**
**end for**
Return $S_k$

(e.g., NCD), (ordinary) lumpability allows the exact derivation of stationary and transient measures of the original Markov chain from the resulting lumped Markov chain [99].

A lumpable Markov chain is a Markov chain in which some states can be merged together (their transition probabilities are summed) and the resulting process is also a Markov chain. A more formal definition is provided below, followed by an application of the concept in the context of the DTMC-CA model.

**Definition:** Let $\tilde{M} = (\tilde{S}, \tilde{P}, \tilde{s_0})$ be a finite, irreducible Markov chain with state space $\tilde{S}$, transition matrix $\tilde{P}$ and initial state $\tilde{s_0}$. Let $\tilde{\pi}$ represent the steady state row vector

| $k$ | 1 | 2 | 5 | 7 | 9 | 10 |
|---|---|---|---|---|---|---|
| $\|S\|$, $v_{max} = 5$, Potential | 7 | 49 | 16807 | 823543 | $7^9$ | $7^{10}$ |
| $\|S\|$, $v_{max} = 5$, Actual | 7 | 19 | 321 | 2080 | 13460 | 34242 |
| Percent decrease | 0% | 61% | 98% | 99.7% | 99.97% | 99.99% |
| | | | | | | |
| $\|S\|$, $v_{max} = 2$, Potential | 4 | 16 | 1024 | 16384 | $4^9$ | $4^{10}$ |
| $\|S\|$, $v_{max} = 2$, Actual | 4 | 10 | 156 | 979 | 6102 | 15235 |
| Percent decrease | 0% | 37.5% | 84.76% | 94% | 97.7% | 98.5% |

**Table 6.1** Comparison of state space set sizes. "Potential" sizes represent state space set sizes with invalid states while "actual" sizes represent state space sizes without the invalid states. Removing invalid states greatly decreases the potential computational cost.

($\tilde{\pi}\tilde{P} = \tilde{\pi}$ and $\tilde{\pi}e^T = 1$) and $L = \{L_1, L_2, \ldots, L_m\}$ represent a partition of the state space such that $L_i \subseteq \tilde{S}$, $L_i \neq \emptyset$, $L_i \cap L_j = \emptyset$ and $\bigcup_{i=1}^{m} L_i = \tilde{S}$ for $i, j \in \{1, 2, \ldots, m\}$.

The Markov chain $\tilde{M}$ is (ordinarily) lumpable with respect to L if, for all $L_i, L_j \in L$ and all $i', i'' \in L_i$ [100],

$$\sum_{j' \in L_j} \tilde{P}(i', j') = \sum_{j' \in L_j} \tilde{P}(i'', j').$$

When the Markov chain $\tilde{M}$ is lumpable with respect to a partition $L$, then $L$ can be used to create an aggregated Markov chain $\hat{M} = (L, \hat{P}, \hat{s_0})$ in which each element of $L$ is represented as a single state. The aggregated Markov chain does not introduce any error in the computation of stationary and some transient properties [99]. Then

$$\hat{\pi}(i) = \sum_{j \in L_i} \tilde{\pi}(j).$$

To illustrate this concept, consider for example a Markov chain $\tilde{M} = (\tilde{S}, \tilde{P}, \tilde{s_0})$ with $\tilde{S} = \{1, 2, 3, 4\}$. The partition set is $L = \{(1, 2), (3, 4)\}$ and the transition matrix

$$\tilde{P} = \begin{vmatrix} \frac{3}{10} & \frac{1}{5} & \frac{1}{2} & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{2} \\ \frac{1}{8} & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} \\ 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{vmatrix}.$$

$\tilde{M}$ is lumpable with respect to $L$ and the lumped Markov chain $\hat{M} = (L, \hat{P}, \hat{\pi})$ has the transition matrix

$$\hat{P} = \begin{vmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{4} & \frac{3}{4} \end{vmatrix}.$$

As the example above illustrate, lumpable states are most easily determined using the matrix of transition probabilities of the system. In the DTMC-CA model, the main reasons for lumping the states would be to reduce spatial and computational costs. However, most of the cost is incurred in the computation of the transition matrix $P$. If this matrix is already computed, there would be no benefit from lumping the state space because $P$ would directly be used to compute the desired probabilities measures. For this reason, the approach selected takes advantage of the characteristics of the system to determine the lumpable states.

Considering the system update rules, note that if two states $x, y$ are equal after the first two rules of update (Section 6.2) then row $x$ and row $y$ will be identical in the transition matrix. Hence, the two states can be lumped together because they are *equivalent* from a stochastic view. Note that the connectivity status of each state must be considered when creating the lumps (a state corresponding to disconnected status should not be lumped with a state corresponding to a connected status). To illustrate the method, consider a system with $k = 4$ and $v_{max} = 2$. Then, the following two states result in identical rows in the transition matrix: $s_1 = (\infty, 1, \infty, \infty)$ and $s_2 = (\infty, 2, \infty, \infty)$. This is because after the first two rules are applied both will become $(\infty, 2, \infty, \infty)$.

Table 6.2 shows the reduction in state space size after equivalent states are lumped, taking into account the connectivity status of each state. The transmission range $r = 4$ cells was used for the values in this table. Decreases of more than 80% of the state space size can be achieved using this technique.

| $k$ | 1 | 2 | 5 | 7 | 9 | 10 |
|---|---|---|---|---|---|---|
| $\|S\|$, $v_{max} = 5$, without lumping | 7 | 19 | 321 | 2080 | 13460 | 34242 |
| $\|S\|$, $v_{max} = 5$, with lumping | 7 | 12 | 79 | 363 | 1635 | 3484 |
| Percent decrease | 0% | 36.8% | 75.4% | 82.5% | 87.8% | 89.8% |
| | | | | | | |
| $\|S\|$, $v_{max} = 2$ without lumping | 4 | 10 | 156 | 979 | 6102 | 15235 |
| $\|S\|$, $v_{max} = 2$, with lumping | 4 | 6 | 55 | 246 | 1103 | 2336 |
| Percent decrease | 0% | 40% | 64.7% | 74.87% | 81.9% | 84.7% |

**Table 6.2** Benefits of lumping together equivalent states shown by comparing state space set sizes with and without lumping. The aggregation is performed on the state space set without invalid states.

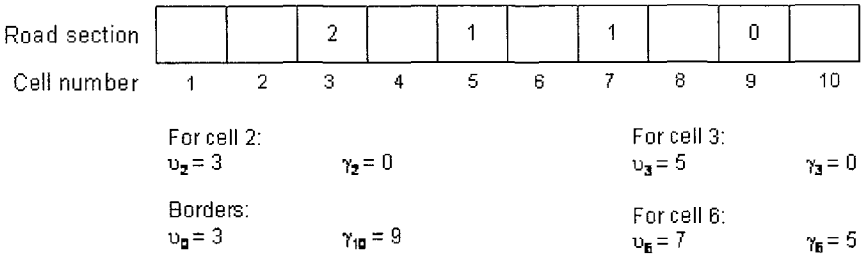



**Figure 6.4** Illustration of expressions used in the computation of the transition probabilities.

### 6.3.3 Transition Matrix P

The aggregated state space is used to compute the matrix of transition probabilities $P = \{P(x,y) : x,y \in S\}$ where $P(x,y)$ represents the probability of moving from state $x$ to state $y$ in one time step. Considering two states $x = (x_1, x_2, \ldots, x_k)$ and $y = (y_1, y_2, \ldots, y_k)$, only a few valid transitions from $x$ to $y$ are possible. Below is the list the allowable transitions from $x$ to $y$. Recall that the stochastic rule of the cellular automaton traffic model (third rule) specifies that a vehicle will choose to decrease its speed, if greater than 0, with probability $p$. Let $\nu_i$ represent the position of next occupied cell at the right of cell $i$ while $\gamma_i$ represents the position of the previous occupied cell at the left of cell $i$. Figure 6.4 shows some examples. Let $\nu_0$ be the position of the first occupied cell from the left border and $\gamma_k$ be the position of the first occupied cell from the right border (last occupied cell from the left).

For $\nu_0 \leq i \leq \gamma_k - 1$:

- If $x_i = j$, $j < v_{max}$ and $\nu_i \geq j + 1$, then $y_{i+j} = j$ with probability $p_i = p$ and $y_{i+j+1} = j + 1$ with probability $p_i = 1 - p$.
- If $x_i = j$, $j < v_{max}$ and $\nu_i < j + 1$, then $y_{i+\nu_i-1} = \nu_i - 1$ with probability $p_i = p$ and $y_{i+\nu_i} = \nu_i$ with probability $p_i = 1 - p$.

- If $x_i = j, j = v_{max}$ and $\nu_i \geq j + 1$, then $y_{i+j-1} = j - 1$ with probability $p_i = p$ and $y_{i+j} = j$ with probability $p_i = 1 - p$.

- If $x_i = j, j = v_{max}$ and $\nu_i < j + 1$, then $y_{i+\nu_i-1} = \nu_i - 1$ with probability $p_i = p$ and $y_{i+\nu_i} = \nu_i$ with probability $p_i = 1 - p$.

For the border cells, additional probabilities must be defined. On the left border, a new car may enter the area of interest (through any of the cells 1 through $min(v_{max}, \nu_0 - 1)$). For example, in Figure 6.4, a new car may enter in cell 1 or 2. On the right border, the vehicle in position $\gamma_k$ may leave the area of observation (if $\gamma_k > k - v_{max}$). The probability that a new vehicle enters the area of observation at cell $e$ with speed $h$ is $p_{e,h}$ and $p_\beta$ is the probability that a vehicle is blocked from exiting the area (because of the presence of another vehicle after cell $k$). For the left border

- If $x_i = \infty$ with $1 \leq i \leq min(v_{max}, \nu_0 - 1)$, then $y_i = h$ and $y_j = \infty, j \neq i$, $1 \leq j \leq min(v_{max}, \nu_0 - 1)$, with probability $p_i = p_{i,h}$.

For the right border

- If $x_{\gamma_k} = j, j < v_{max}, \gamma_k < k$ and $\gamma_k + j > k$, then (1) $y_k = k - \gamma_k, y_i = \infty, \gamma_k \leq i < k$, with probability $p_i = p_\beta(1-p)$; (2) $y_{k-1} = k - \gamma_k - 1, y_i = \infty, \gamma_k \leq i \leq k, i \neq k$ with probability $p_i = p(p_\beta)$; (3) $y_i = \infty, \gamma_k \leq i \leq k$, with probability $p_i = (1 - p_\beta)$.

- If $x_{\gamma_k} = j, j < v_{max}, \gamma_k < k$ and $\gamma_k + j + 1 > k$, then (1) $y_k = k - \gamma_k, y_i = \infty, \gamma_k \leq i < k$, with probability $p_i = p(1 - p_\beta) + p_\beta(1 - p)$; (2) $y_{k-1} = k - \gamma_k - 1, y_i = \infty, \gamma_k \leq i \leq k, i \neq k$ with probability $p_i = p(p_\beta)$; (3) $y_i = \infty, \gamma_k \leq i \leq k$, with probability $p_i = (1 - p)(1 - p_\beta)$.

- If $x_{\gamma_k} = j, j = v_{max}, \gamma_k < k$ and $\gamma_k + j > k$, then (1) $y_k = k - \gamma_k, y_i = \infty, \gamma_k \leq i < k$, with probability $p_i = p(1 - p_\beta) + p_\beta(1 - p)$; (2) $y_{k-1} = k - \gamma_k - 1, y_i = \infty, \gamma_k \leq i \leq k, i \neq k$ with probability $p_i = p(p_\beta)$; (3) $y_i = \infty, \gamma_k \leq i \leq k$, with probability $p_i = (1 - p)(1 - p_\beta)$.

- If $x_{\gamma_k} = j, j > 0, \gamma_k = k$, then (1) $y_k = 0$, with probability $p_i = p_\beta$; (2) $y_i = \infty, \gamma_k \leq i \leq k$, with probability $p_i = (1 - p_\beta)$.

- If $x_{\gamma_k} = j$, $j = 0$, $\gamma_k = k$, then (1) $y_k = 0$, with probability $p_i = p(1 - p_\beta) + p_\beta$; (2) $y_i = \infty$, $\gamma_k \leq i \leq k$, with probability $p_i = (1 - p)(1 - p_\beta)$.

- If $x_{\gamma_k} = j$, $\gamma_k < k$ and $\gamma_k + j + 1 \leq v_{max} \leq k$, then probabilities are computed as in the case of $\nu_0 \leq i \leq \gamma_k - 1$.

From these individual cells probabilities, the transition matrix entry $P(x, y)$ is computed as $P(x, y) = \prod_{i=1}^{k} p_i$.

### 6.3.4 Probabilistic Measures

The transition matrix is used to compute various probabilistic measures of interest. The state space is divided in two subsets, $S_1$ and $S_2$, such that $S_1 \cup S_2 = S$ and $S_1 \cap S_2 = \emptyset$. The subset $S_1$ contains the states in which the status of the system is connected while $S_2$ contains the states corresponding to a disconnected status. The first measure is the expected time to disconnection, for which two variations are computed: with $s_0 \in S_1$ and $s_0 \in S$. The second measure is the expected time to connection, for which two variations are also computed, with $s_0 \in S_2$ and $s_0 \in S$. The final measure is the probability of maintaining connectivity for a period longer than a value $t$. Below is a brief description of how each of these measures is derived from the transition matrix.

**Expected Time to Disconnection:** Given the transition $P$ and for each $(x, y) \in SxS$, set $P_r(x, y) = P(x, y)$ if $y \in S_1$, and $P_r(x, y) = 0$ if $y \in S_2$. Then define $h = (1, 1, \ldots, 1)$, a vector of size $|S|$ containing all $1's$ and $g = (I - P_r)^{-1}h$.

Then, the expected time to disconnection given that the system is in a connected state in steady state is given by [101]

$$E[T_{conn}|s_0 \in S_1] = \frac{\sum_{x \in S_1} g(x)\pi(x)}{\sum_{y \in S_1} \pi(y)}. \tag{6.1}$$

And the expected time to disconnection independent of the current status in steady state is given by

$$E[T_{conn}] = \sum_{x \in S_1} g(x)\pi(x). \tag{6.2}$$

**Expected Time to Connection:** Given the transition $P$ and for each $(x, y) \in S\text{x}S$, set $P_{r_2}(x, y) = P(x, y)$ if $y \in S_2$, and $P_{r_2}(x, y) = 0$ if $y \in S_1$. $h$ is defined as in the case above and $g_2 = (I - P_{r_2})^{-1}h$.

Then, the expected time to connection given that the system is in a disconnected state in steady state is given by

$$E[T_{disconn}|s_0 \in S_2] = \frac{\sum_{x \in S_2} g_2(x)\pi(x)}{\sum_{y \in S_2} \pi(y)}. \tag{6.3}$$

Then, the expected time to connection independent of the current status in steady state is given by

$$E[T_{disconn}] = \sum_{x \in S_2} g_2(x)\pi(x). \tag{6.4}$$

**Probability of Connection Duration:** The last measure outputs the probability of maintaining connectivity for more than a period of time t. In this case too, for $(x, y) \in S\text{x}S$, set $P_r(x, y) = P(x, y)$ if $y \in S_1$ and $P_r(x, y) = 0$ if $y \in S_2$. Then,

$$P[T > t] = \sum_{x \in S_1} \pi(x) \sum_{y \in S} P_r^t(x, y). \tag{6.5}$$

where $P_r^t$ represents matrix $P_r$ raised to the power $t$.

### 6.3.5 DTMC-CA Example

The computation of the transition matrix $P$ is illustrated through a simple example. Assume $k = 4$, $v_{max} = 2$ and the threshold for connectivity is $r = 2$. The straightforward computation of the state space $S$ would lead to a size of $|S| = 4^4 = 256$ states. By preventing invalid states, the size becomes $|S| = 63$, which reduces to $|S| = 26$ after the states are lumped together. Table 6.3 lists the final set S of states involved in the computation of $P$.

| | | | |
|---|---|---|---|
| (∞ ∞ 0 ∞) | (∞ 2 ∞ ∞) | (∞ ∞ 2 0) | (∞ 0 ∞ ∞) |
| (∞ 2 ∞ 0) | (∞ 2 ∞ 1) | (2 ∞ ∞ 0) | (2 ∞ ∞ 2) |
| (∞ ∞ 2 ∞) | (∞ 2 0 ∞) | (∞ 2 0 0) | (2 ∞ 0 ∞) |
| (0 ∞ ∞ 0) | (0 ∞ ∞ 1) | (2 ∞ 1 0) | (2 ∞ 1 ∞) |
| (2 0 ∞ ∞) | (2 0 ∞ 0) | (2 0 ∞ 1) | (2 0 0 ∞) |
| (2 0 0 0) | (∞ ∞ ∞ ∞) | (∞ ∞ ∞ 0) | (∞ ∞ ∞ 2) |
| (2 ∞ ∞ ∞) | (0 ∞ ∞ ∞) | | |

**Table 6.3** Sample state space S after state reduction steps. Set S reduced from potentially a size $|S| = 256$ states to a size $|S| = 26$ states.

Next, the transition matrix $P$ is calculated. In this example, $p = 0.5$; $p_{1,1} = 0.0773$; $p_{1,2} = 0.0697$; $p_{2,2} = 0.0697$ and $p_\beta = 0.216$. The transition matrix obtained is a sparse matrix, with 116 non-zero entries out of the $26^2 = 676$ total entries. Table 6.4 list the probabilities from the first 3 states in the first row of Table 6.3. The symbol ∞ has been replaced by a (.) in each state.

| (. . 0 .) | | (. 2 . .) | | (. . 2 0) | |
|---|---|---|---|---|---|
| To State | $P(x, y)$ | To State | $P(x, y)$ | To State | $P(x, y)$ |
| (. . 0 .) | 0.3916 | (2 . . 2) | 0.0773 | (. . 0 .) | 0.3071 |
| (. 2 . 1) | 0.0348 | (. . 2 .) | 0.4227 | (. . 2 0) | 0.4762 |
| (2 . . 2) | 0.0735 | (2 . 1 .) | 0.0773 | (. 2 0 .) | 0.0273 |
| (. 2 0 .) | 0.0348 | (. . . 2) | 0.4227 | (. 2 0 0) | 0.0424 |
| (2 . 0 .) | 0.0735 | | | (2 . 0 .) | 0.0576 |
| (. . . 2) | 0.3916 | | | (2 . 1 0) | 0.0894 |

**Table 6.4** Three rows extracted of the transition matrix P corresponding to state space S.

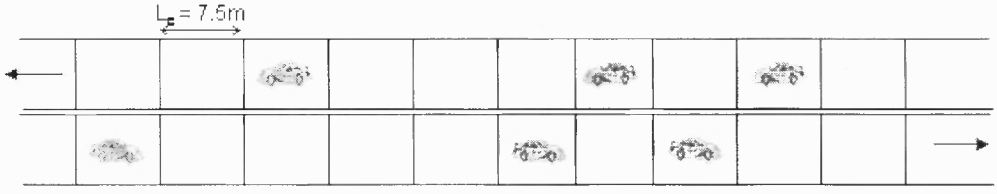



**Figure 6.5** Bidirectional traffic: the road interval is divided in juxtaposed cells of fixed length on each traffic lane of a bidirectional road.

Each of the columns in Table 6.4 represents a row in the transition matrix $P$. As expected, they each sum up to exactly 1. Also recall that each of the transitions represents the sum of (typically) more than one transition, as each of the state in $S$ is a representative from an aggregated group of states.

### 6.3.6 Bidirectional Traffic

This subsection considers a stretch of a road with traffic moving in both directions. The source and destination endpoints are still assumed static.

Each lane of the road section of interest is divided in $k$ cells of fixed length $L_c$, as in the single-lane case. The cells are juxtaposed on the lanes (see Figure 6.5). A value from the set $\acute{V} = \{0, 1, \ldots, v_{max}, \infty\}$ is associated with each cell.

A Markov chain $\acute{M} = (\acute{S}, \acute{P}, \acute{s_0})$ is constructed with state space

$\acute{S} = \{\acute{s} = (c_{1,1}, c_{1,2}, \ldots, c_{1,k}, c_{2,1}, \ldots, c_{2,k}) : c_{i,j} \in \acute{V}$, for $i = 1, 2$ and $1 \leq j \leq k\}$

and transition probability matrix $\acute{P}$. In this case too, no assumption is made on the starting state $\acute{s_0}$ of the Markov chain as only stationary measures are of interest.

The movement in each traffic lane is independent of the other lane. Hence, a state $\acute{s} \in \acute{S}$ can be considered as a vector of two states $\acute{s} = (s_{l_1}, s_{l_2})$ with $s_{l_1}, s_{l_2} \in S$, the state space of the single-lane traffic described in Section 6.3.1. $s_{l_1}$ is the state of the system on lane 1 while $s_{l_2}$ is the system state on lane 2.

The matrix of transition probabilities $\acute{P} = \{\acute{P}(x, y) : x, y \in \acute{S}\}$ where $\acute{P}(x, y)$ represents the probability of moving from state $x$ to state $y$ in one time step is computed as

$\acute{P}(x, y) = \acute{P}((x_{l_1}, x_{l_2}), (y_{l_1}, y_{l_2}))$ with $x_{l_1}, x_{l_2}, y_{l_1}, y_{l_2} \in S$. And from the independence of the lanes movements,

$$\acute{P}(x, y) = P(x_{l_1}, y_{l_1}) P(x_{l_2}, y_{l_2})$$

where $P(x_{l_1}, y_{l_1})$ represents the transition probability on lane 1 from state $x_{l_1} \in S$ to $y_{l_1} \in S$. The decomposition of the state space set $\acute{S}$ in two disjoint sets $\acute{S_1}$ and $\acute{S_2}$ representing, respectively, the states with connected status and those with disconnected status, is a little different in this case. Indeed, a state of disconnection on one lane does not translate in a state of disconnection when both lanes are considered. Thus, $\acute{S_1} = \{\acute{s} = (s_{l_1}, s_{l_2}) \in \acute{S}$ : not $(s_{l_1} \in S_2$ *and* $s_{l_2} \in S_2)\}$ and $\acute{S_2} = \{\acute{s} = (s_{l_1}, s_{l_2}) \in \acute{S} : (s_{l_1} \in S_2$ *and* $s_{l_2} \in S_2)\}$

### 6.3.7 Moving Endpoints and Lane Changes

The final configuration considered is a multilane road with $\ell$ lanes in which lane changes are enabled. In this case, the endpoints are moving on the road and the length of the portion of road between the source and the destination may change even as the connectivity is maintained.

The road portion between the endpoints is initially divided in $k$ cells of fixed length, $L_c$. The number of cells $k$ may change over time, with $0 \leq k \leq \check{K}$, where $\check{K}$ is the maximum distance between endpoints considered. The speed of the source vehicle $v_{src}$ is used as the base speed. Thus, for each vehicle $u$, the relative speed $v_{u,src} = v_u - v_{src}$ is used by the model (rather than $v_u$ as previously done). The set of relative speeds is $\check{V} = \{-v_{max}, \ldots, -1, 0, 1, \ldots, v_{max}, \infty\}$. A value from $\check{V}$ is associated with each cell. A negative cell value corresponds to a vehicle $u$ moving at a speed $v_u < v_{src}$ while a positive cell value corresponds to a vehicle $u$ moving at a speed $v_u > v_{src}$.

A Markov chain $\check{M} = (\check{S}, \check{P}, \check{s_0})$ is constructed with state space

$$\check{S} = \{\check{s} = (v_{1,1}, v_{1,2}, \ldots, v_{1,k}, \ldots, v_{1,\check{K}}, v_{2,1}, \ldots, v_{2,k}, \ldots, v_{2,\check{K}}, v_{i,1}, \ldots, v_{i,k}, \ldots, v_{i,\check{K}}) : v_{i,j} \in \check{V}, \; for \; i = 1, 2, \ldots, \ell \; and \; 1 \leq j \leq \check{K}\}$$

and transition probability matrix $\check{P}$. No assumption is made on the starting state $\check{s_0}$ of the Markov chain as only stationary measures are of interest.

To compute the transition matrix, four additional rules are added to the rules set to regulate lane changes and passing of vehicles [94]. For vehicle $i$, let $gap_i$ represent the distance to the vehicle ahead on the same lane; let $gap_{i,f,o}$ represent the distance from vehicle $i$ to the first vehicle forward on the other lane and $gap_{i,b,o}$ represent the distance

from vehicle $i$ to the first vehicle backward on the other lane. Then, vehicle $i$ can change to another lane if the all the following rules are satisfied:

- $gap_i < \zeta$
- $gap_{i,f,o} > \zeta_{f,o}$
- $gap_{i,b,o} > \zeta_{b,o}$
- With all above conditions satisfied, change lane with probability $p_{change}$

The parameters $\zeta$, $\zeta_{f,o}$ and $\zeta_{b,o}$ specify how far ahead and back the vehicle looks on the current lane and the other lane before deciding it is safe to make a lane change. The remaining computations are performed in a fashion similar to the single-lane unidirectional traffic described in Section 6.3.1. The cells on all lanes are juxtaposed and the connected states set $\check{S}_1$ as well as the disconnected states set $\check{S}_2$ are determined in a manner similar to Section 6.3.6.

## 6.4 DTMC-CA Performance Evaluation

This section presents the evaluation of the DTMC-CA model. In the following, the evaluation methodology, the metrics used for assessing the performance, and an analysis of the results is presented.

### 6.4.1 Evaluation Methodology

The evaluation of the analytical model on a road segment verifies the correctness of its outputs against simulations results obtained using validated traffic generators on traffic on a ring topology (Figure 6.6). A road in the form of a closed ring was selected to maintain a

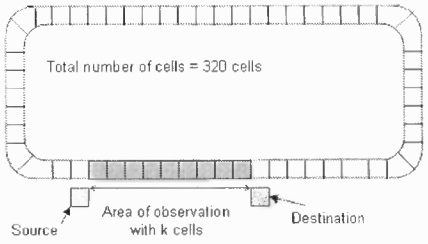



**Figure 6.6** Map used in the simulation scenario

| Parameter | Value |
|---|---|
| Simulation area | 320 cells. Cell length $\geq$ 7.5m |
| Number of vehicles | 35-50-75 |
| Transmission range | 4-11 cells |
| Vehicle velocity | 2 - 5 cells per second |

**Table 6.5** Simulation Setup

constant density during the simulations. The ring layout contains a large number of cells, from which a portion is selected as the interval between source and destination. The large system (ring) has an impact on the connectivity of the selected interval but DTMC-CA considers only the selected interval. However, this evaluation will show that it is able to account for the impact of the ring as a whole.

The road layout used in the simulations is shown on Figure 6.6. It is a closed loop road with $L = 320$ cells. A fixed source and destination pair is communicating using the vehicular traffic moving on the road. There are $k$ cells between the endpoints in communication. The length of a cell is often considered equal to $7.5m$ in the literature. This value is used, in part, because the average vehicle length is about $5m$, thus the distance $7.5m$ accounts for about $2.5m$ safety distance for both front and rear of vehicle (cell length could be set to $10m$ or higher if longer safety distance are considered). The

number of vehicles $N$ on the loop is varied to measure the effects of density $\rho = \frac{N}{L}$. The first 3600 seconds of simulation time are discarded to eliminate initial transient effects of movement. Next the traffic on the shaded portion of road is captured and analyzed to obtain the measures of interest. The simulation parameters are summarized in Table 6.5.

The parameters of the DTMC-CA model are calibrated to the simulations. The parameters used are: CA randomization factor $p = 0.5$; parameters function of the density: for 35 vehicles $p_1 = 0.012$; $p_2 = 0.073$; $p_3 = 0.0067$; $p_4 = 0.0322$; $p_5 = 0.0285$; $p_\beta = 0.10$; for 50 vehicles $p_1 = 0.026$; $p_2 = 0.0130$; $p_3 = 0.0096$; $p_4 = 0.0276$; $p_5 = 0.023$; $p_\beta = 0.3060$; for 75 vehicles $p_1 = 0.048$; $p_2 = 0.0211$; $p_3 = 0.0131$; $p_4 = 0.0202$; $p_5 = 0.0148$; $p_\beta = 0.446$.

In the expressions above, $p_i$ is the probability that a vehicle enters the interval of observation (shaded area in Figure 6.6) with speed $i$.

### 6.4.2 Metrics

DTMC-CA is evaluated by varying the threshold connectivity range, the network density and the time needed for connectivity. The metrics used to assess the performance are the following:

- **Expected duration of connectivity.** This metric defines the amount of time a source can expect to communicate with a destination multiple hops away. Two values of this metric are reported, function of the existence (or not) of at least one communication path at initial time. These values correspond to equation 6.1 and equation 6.2 respectively. The expected duration of connectivity measures the accuracy of the predictions of connectivity duration of the analytical model.

- **Expected duration of disconnectivity.** This metric defines the amount of time a source can expect not to be able to communicate with a destination multiple hops away. Two values of this metric are reported, function of the non-existence (or not) of at least one communication path at initial time. These values correspond to

equation 6.3 and equation 6.4 respectively. The expected duration of disconnectivity measures the accuracy of the predictions of disconnect duration of the analytical model.

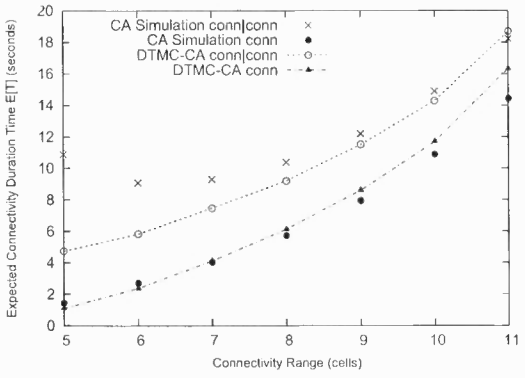
- **Probability of connectivity duration.** This metric defines, in steady-state, the probability of uninterrupted connectivity lasting at least for a time duration $t$. It corresponds to equation 6.5. Using this metric, one can evaluate the independence of connectivity estimations on adjacent portions of road.

### 6.4.3 Numerical and Simulation Results

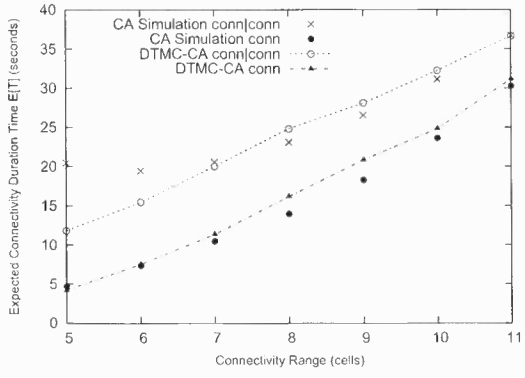
To obtain the data points of the CA simulation, 1000 simulation runs of 3200 simulation steps are generated. The results presented here are averaged over all the simulations. Within a simulation, only one of five consecutive steps is considered. This is done to eliminate the correlation between consecutive samples. Additionally, a random number uniformly generated from the range $(1, 100)$ is used to determine the start step within the 3200 steps. This is used to add randomness between two simulation runs.

**Expected Duration of Connectivity:** Figure 6.7 shows that the expected duration of connectivity values obtained using the DTMC-CA model match well with the results from the CA simulations. Comparing analytical and simulation results for a given density, one observes that the analytical predictions are more exact for ranges greater than 5. This is, in part, due to the fact that the maximum speed for a vehicle was set to 5 cells per step. For ranges of 7 cells and greater, the output from the analytical results are within one second of the simulation results.

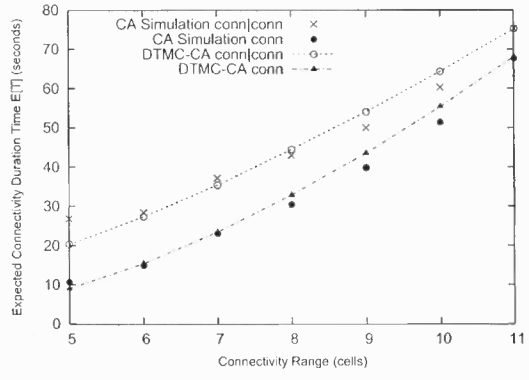
Another observation from this figure is, as expected, an increase in the transmission (or connectivity) range leads to an increase in the expected duration of connectivity for a fixed value of $k$. Additionally, the denser the network, the more accurate the outputs from the DTMC-CA model, when all ranges are considered. Overall, the difference between the




(a) 35 vehicles




(b) 50 vehicles




(c) 75 vehicles

**Figure 6.7** Expected duration of connectivity when the system is initially in a connected state (or not) with different node densities and different transmission ranges. $v_{max} = 5$ cells per time step and $k = 12$ cells.

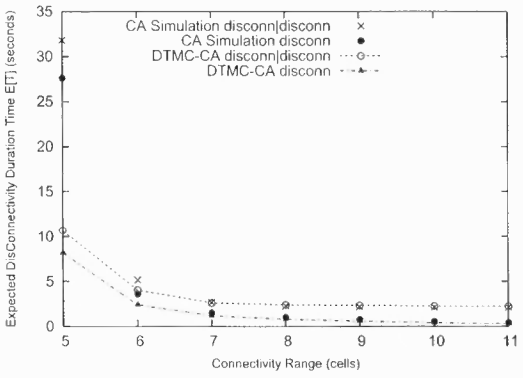
simulation and analytical results is very small (within 3 seconds for transmission ranges greater or equal with 6 cells). Also there is a greater precision for predictions of expected duration of connectivity in general than for expected duration of connectivity knowing that the communication instantiated when the system was in a connected state.

An interesting point to note is the stochastic nature of the CA model, a characteristic which is seen in Figure 6.7(c). With a density of 0.23, one sees an expected duration of connectivity of about 80 seconds on an interval of 12 cells and with a connectivity range of 11 cells. This result is not likely to appear with a deterministic or near-deterministic traffic model (duration of connectivity would likely go to infinity using comparable parameters).

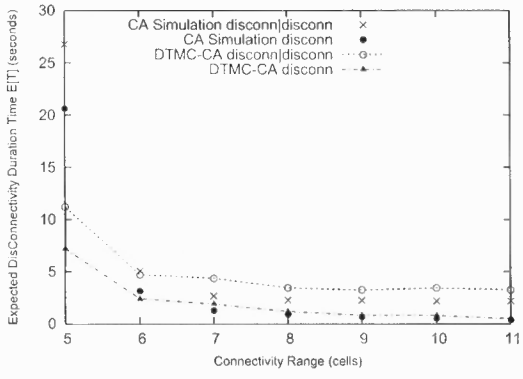
**Expected Duration of Disconnectivity:** Figure 6.8 shows a very good agreement between expected duration of disconnectivity values obtained using the DTMC-CA model and the CA simulation results. The accuracy of the predictions does not vary with changing values of transmission range, except for the case of $r = 5\ cells$, which is influenced by border effects.

Over different densities, the outputs from DTMC-CA are within 2 seconds of the values obtained through simulations, for transmission ranges of 6 cells and higher. Combined with the expected connectivity duration results, these results show that the DTMC-CA model provides a good level of accuracy and it can be used by protocols at different layers of the protocol stack. Note that as expected, increasing the transmission range leads to a decrease in the expected duration of disconnectivity.

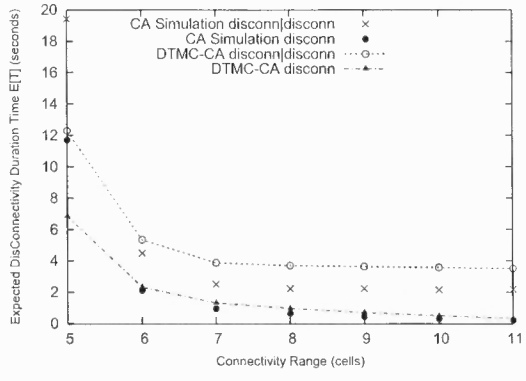
Interestingly, this figure also shows that the expected duration of disconnectivity, given that the system is in disconnected state, does not decrease as the density increases. This contrasts with Figure 6.7 where a clear difference in expected connectivity duration values can be observed as the density increases. The impact of density on the expected




(a) 35 vehicles




(b) 50 vehicles




(c) 75 vehicles

**Figure 6.8** Expected duration of disconnectivity when the system is initially in a disconnected state (or not) with different node densities and different transmission ranges. $v_{max} = 5$ cells per time step and $k = 12$ cells.

duration of disconnectivity appears reduced, when compared to its impact on expected duration of connectivity. Thus, longer periods of network connectivity do not necessarily translate in shorter periods of network disconnection using the CA traffic model. Further tests are required with other vehicular traffic models to identify whether this is an artifact of the CA model.

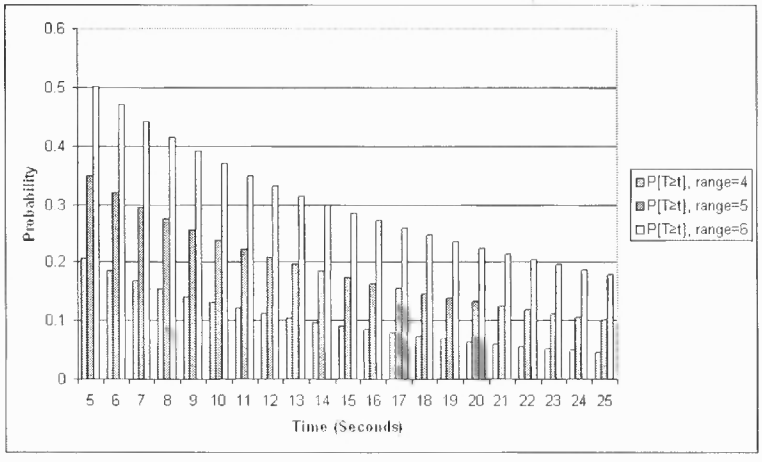
**Probability of Duration of Connectivity:** The probability of connectivity duration represents the likelihood of uninterrupted connectivity lasting at least $t$ steps between the endpoints. A step is a unit of time. Figure 6.9 shows the probabilities corresponding to different transmission range values and different number of cells $k$.

As expected, longer uninterrupted connectivity periods are less likely than shorter periods independently of the number of cells $k$ between the endpoints. Additionally, the probabilities decrease when the number of cells, $k$, increases.

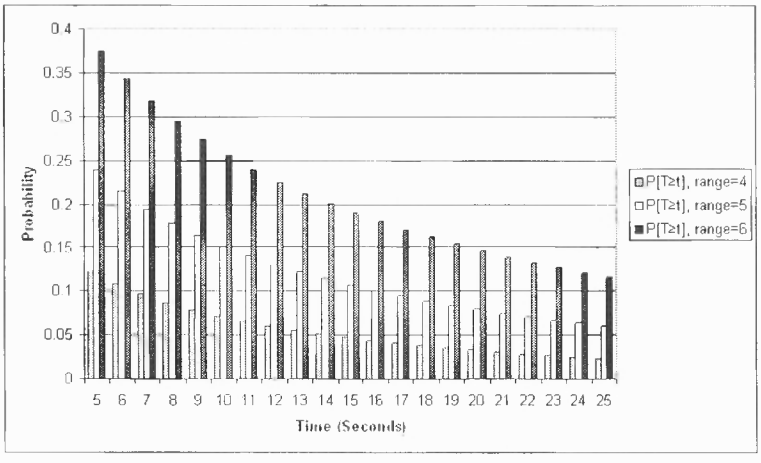
Figure 6.9 also indicates that the probabilities for proportional settings are similar, though not equal. For example, $P[T > 5]$ for $k = 8$ cells and range $r = 4$ equals $0.21$ while the same probability for $k = 10$ cells and range $r = 5$ equals $0.24$.

## 6.5 Approximating Connectivity for Longer Paths

As previously shown, the DTMC-CA model outputs expected duration of connectivity as well as expected duration of disconnectivity with good accuracy when compared with simulation measures. This accuracy is obtained without making any assumption of probabilistic independence between any pairs of communication links between the source, destination and any intermediate vehicles on the observed segment. The cost is high memory and computation requirements. Thus the results presented so far were produced for short paths of 2 or 3 hops.

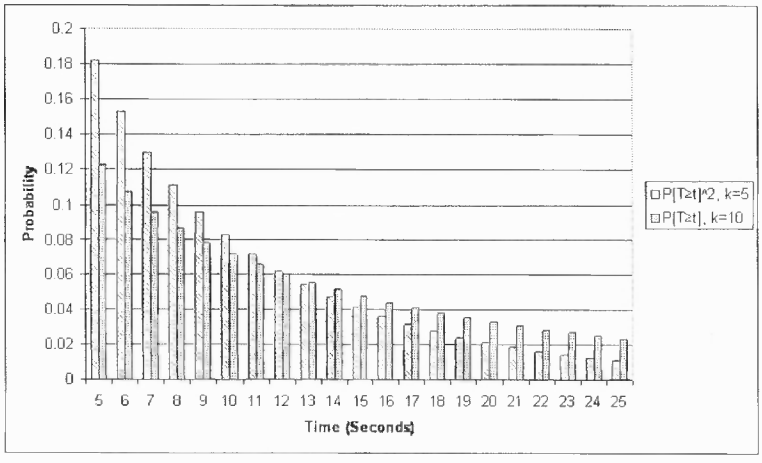



(a) k = 8 cells

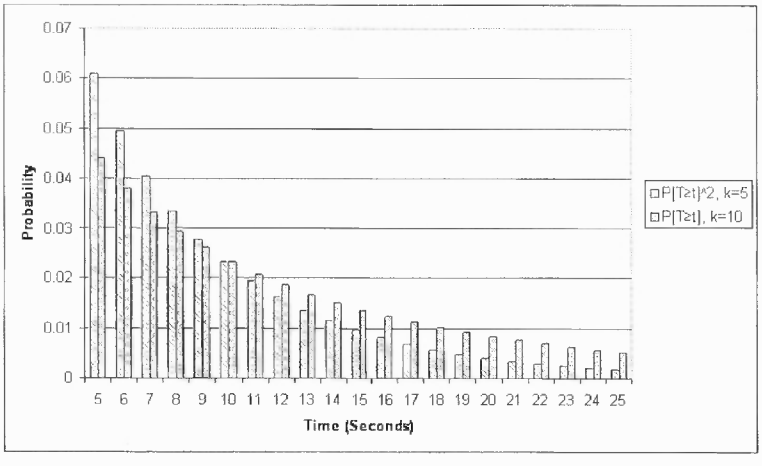



(b) k = 10 cells

**Figure 6.9** Probability of connectivity duration greater or equal to different time limits, for different connectivity range thresholds using the DTMC-CA model.

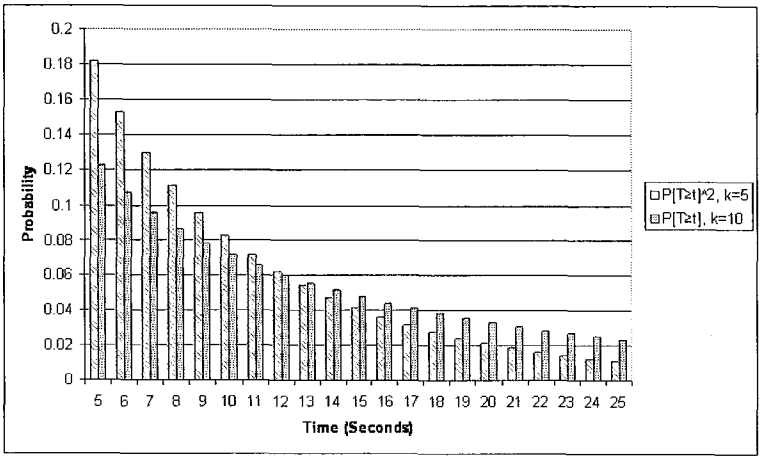



(a) Range 4 cells

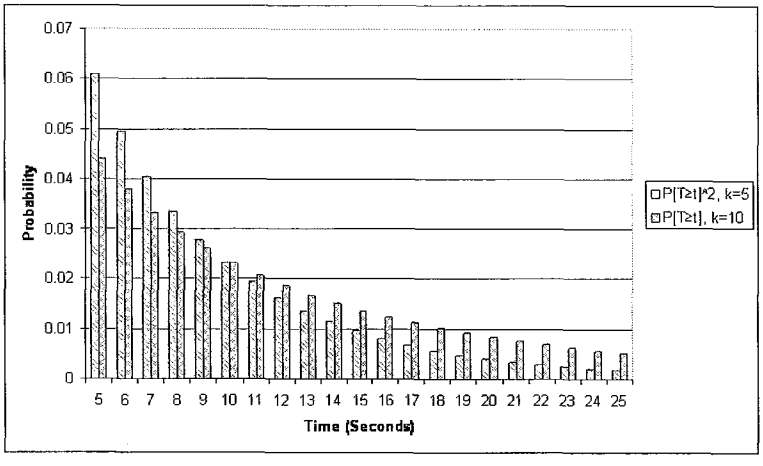



(b) Range 5 cells

**Figure 6.10** Probability of connectivity duration greater or equal to different time limits, for different source-destination lengths. For a distance $d$ and a range threshold $r$, the probability of connectivity duration $\geq t$ is comparable to the square of the same probability for a distance $\frac{d}{2}$ and a range $r$ using the DTMC-CA model.




(a) Range 4 cells




(b) Range 5 cells

**Figure 6.10** Probability of connectivity duration greater or equal to different time limits, for different source-destination lengths. For a distance $d$ and a range threshold $r$, the probability of connectivity duration $\geq t$ is comparable to the square of the same probability for a distance $\frac{d}{2}$ and a range $r$ using the DTMC-CA model.

In this section, the question addressed is whether probability measures from adjacent sub-stretches of a road segment can be considered as independent. If sub-stretches are found independent or weakly correlated, then the probability of connectivity duration of a long road stretch can be obtained by simple multiplication of probabilities of smaller parts.

To verify independence between adjacent sub-stretches of a road segment, two ranges, $r = 4$ cells and $r = 5$ cells, and two number of cells, $k = 5$ cells and $k = 10$ cells are considered. The $k$ values were selected so that one is half of the other. For a given transmission range $r$, the values $P[T > t]_{k=10,r}$ and $P[T > t]^2_{k=5,r}$ are computed and the verification of whether $P[T > t]_{k=10,r} \approx P[T > t]^2_{k=5,r}$ is performed.

Figure 6.10 shows the results of the comparison of $P[T > t]_{k=10,r}$ and $P[T > t]^2_{k=5,r}$. Both sets of probability values are comparable, though not equal. Certain values of time steps $t$ lead to closer match between values of $P[T > t]_{k=10,r}$ and $P[T > t]^2_{k=5,r}$. For example, for $t = 12$ seconds (assuming 1 step is 1 second) and $r = 5$ cells, the expected values for a larger stretch can be approximated from a smaller sub-stretch (about $0.001$ or $2\%$ difference in the probability values). This suggests that although not exactly independent, smaller sub-stretches can be used to approximate longer road stretches with a small error. This can be used to approximate DTMC-CA estimates of duration of connectivity on longer road segments.

## 6.6 Connectivity Window Model

The Connectivity Window (CW) model is a generalization of DTMC-CA (Section 6.3) for any microscopic vehicular traffic model. Connectivity patterns can vary greatly across different traffic generator models, even when the models are validated in the literature. Thus the need for a model which can be used with many traffic generators. In the following, a

comparison of the CA model against another vehicular traffic model is shown to illustrate the differences between different models. Then, a description of the CW model is provided.

### 6.6.1 SUMO Mobility

SUMO is a microscopic, continuous-space, and discrete-time vehicular traffic generator package commonly used for assessing the performance of vehicular networks protocols. The goal here is to assess whether the DTMC-CA model (based on the CA traffic model) matches the results of other vehicular models. Since DTMC-CA results are in good agreement with the CA simulations, the following will compare just the simulation results of SUMO and the CA model. Note that SUMO was used in the evaluation of the RBVT protocols in the previous chapters.

The collision-free car-following model implemented in SUMO is described in [74, 102]. Expected values derived using DTMC-CA (or measured from CA) may not match simulation measurements from SUMO. Nevertheless, both CA and SUMO are validated traffic models used in vehicular networks experiments and it would be helpful to understand their behaviors when applied on an identical input map.

Using the map previously introduced (Figure 6.6), vehicles movements are generated with SUMO. The initial transient steps are discarded (first 3600 seconds of simulation) then measurements are taken on the shaded area of the map. Table 6.6 shows the expected duration of connectivity corresponding to $k = 12$ cells and different values of transmission range $r$ for both SUMO and the CA model.

The simulation measures obtained with SUMO (Table 6.6) differ from the results obtained with the CA model. The reason is that both models do not handle uninterrupted traffic in a similar fashion, as illustrated on Figure 6.11. On this figure, the CA model

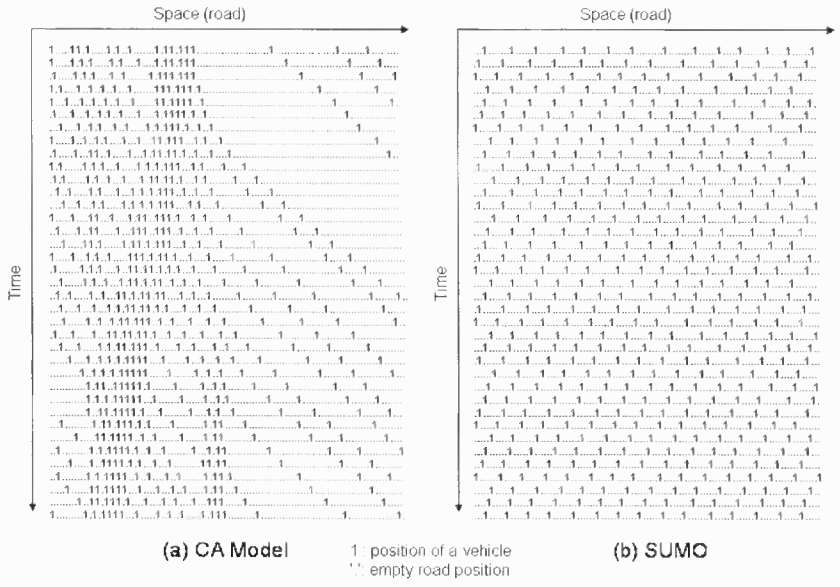



**Figure 6.11** Space-time diagrams using the CA traffic model and SUMO. The stochastic nature of CA model leads to stop-and-go waves of traffic while SUMO traffic stabilizes to near-constant speed for all vehicles in the scenario considered.

| Range $r$ | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|
| $SUMO$ $E[T_{conn}]$ (sec) | 0.002 | 2.876 | 14.338 | 149.042 | 1856.392 |
| $CA$ $E[T_{conn}]$ (sec) | 7.3906 | 10.4578 | 13.987 | 18.3252 | 23.6212 |

**Table 6.6** Expected duration of connectivity using SUMO and the input map. Number of cars = 50. Length of road stretch observed = 12 cells (equivalent to 90m if $L_c = 7.5m$).

creates waves which traverse the road in direction opposite to the traffic. SUMO on the other hand stabilizes to a close to unchanging gap length between consecutive vehicles. The variations in the gap length with SUMO build very slowly compared to the CA model.

### 6.6.2 CW Model Description

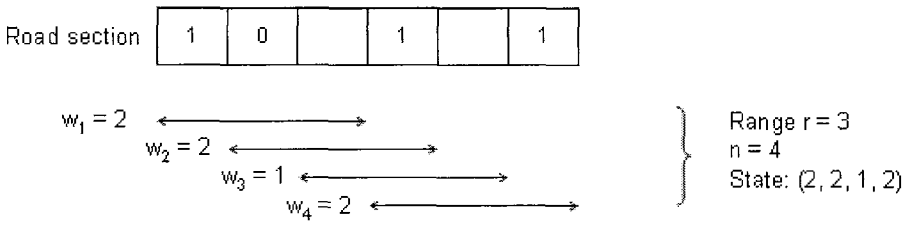
The system observed is identical to the one in Section 6.3.1. The transmission range $r$ represents the maximum number of consecutive empty cells for successful connectivity.

The main idea of the CW model is to abstract the actual vehicle movements and focus on the number of vehicles in each transmission range. For each physical arrangement of vehicles on the observed cells, a vector is created with the counts of the number of vehicles $w_i$ in $r$ consecutive cells starting at cell $i$. The set $W = \{0, 1, \ldots, r\}$ contains the possible values of $w_i$. Note that $w_i$ and $w_{i+1}$ have $r - 1$ overlapping cells.

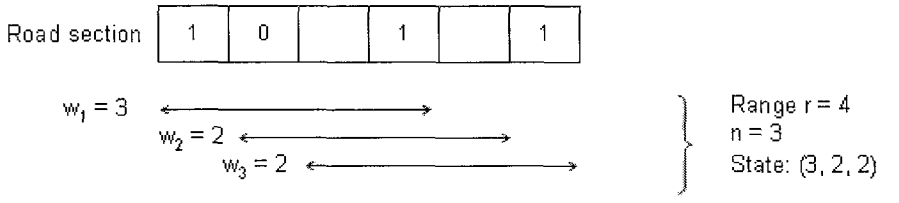
A Markov chain $M' = (S', P', w'_0)$ is constructed with state space $S' = \{w = (w_1, w_2, \ldots, w_n) : w_i \in W$, for all $1 \leq i \leq n\}$ where $n = k - r + 1$. The size of $S$ is function of the parameter $r$, the connectivity range and the total number of cells $k$. Figure 6.12 illustrates the CW concept. A physical disposition of vehicles on a stretch with $k = 6$ cells and 4 cars positioned as shown on the figure. When $r = 3$ cells, the state $w \in S'$ will be $(2, 2, 1, 2)$. For $r = 4$ cells, the corresponding state in $S$ is $w = (3, 2, 2)$. Because CW abstracts the actual movements of vehicles and focuses on the count of vehicle in each connectivity range, it can be used in conjunction with different traffic mobility models.

### 6.6.3 Transition Matrix P

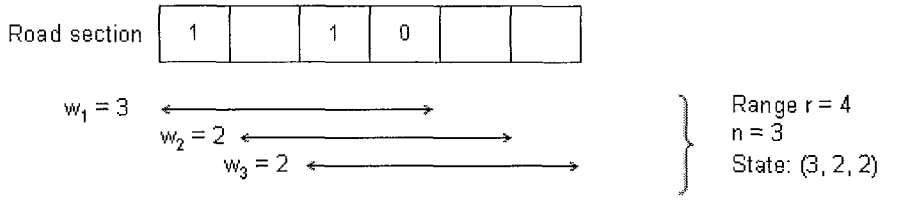
A two-step method is described to compute the matrix of transition probabilities $P' = \{P'(w, y) : w, y \in S'\}$ where $P'(w, y)$ represents the probability of moving from state $w$ to state $y$ in one time step. The following description assumes the CA model as microscopic traffic model. First, each state of $S'$ is translated into states of $S$, (i.e. from the count of the number of vehicles to actual location of vehicles). The transition probabilities of physical states are computed and the output transitions are translated into states of $S'$. A detailed description is presented below.




(a) Using a range r = 3 cells on the road section, the resulting state has 4 cells




(b) Using a range r = 4 cells on the same road section, the resulting state has 3 cells




(c) Different physical arrangement of vehicles (from case b) can lead to same resulting state with 3 cells

**Figure 6.12** Moving window of size $r$ is used to generate states in the CW model. In each subgroup of size $r$, the count of the number of vehicles is recorded.

First, a state $w = (w_1, w_2, \ldots, w_n) \in S'$ is translated to *physical states* of the form $s = (c_1, c_2, \ldots, c_k)$, where $k$ represents the number of cells and $c_i \in V$. A state $w \in S'$ may correspond to multiple state $s \in S$ (see Figure 6.12). Let $A(w)$ represent the set of all physical states corresponding to $w$. Algorithm 4 is used for this conversion. For each state $w$, it identifies the cells which are occupied cells and the ones that are empty. Then the occupied cells are replaced with the vehicles' speeds to produce valid states according to the rules of the mobility model (Section 6.2).

Next, the physical states transition probabilities are computed as in Section 6.3, with the exception that the output transition states are translated in their representations in $S'$. Let $\tau(s, y)$ be the transition probability from the physical state $s \in A(w)$ to the state $y \in S'$ such that $\tau(s, y) = \sum_{\alpha \in A(y)} P(s, \alpha)$. Let $\chi(s|w)$ be the probability of state $w$ representing the physical state $s$. The steady state probability of finding a vehicle with speed $i$ is given by $\kappa_i$, $0 \leq i \leq v_{max}$. Let $\rho$ represent the vehicle density, with $d = 1 - \rho$ and $q = 1 - p$. The individual speed probabilities in steady-state, which are given in [103], are

$$\kappa_i = \begin{cases} \rho^2 \frac{1+pd}{1-pd^2} & \text{if } i = 0 \\ q\rho^2 d \frac{1+d+pd^2}{(1-pd^3)(1-pd^2)} & \text{if } i = 1 \\ \frac{1+(q-p)d^i}{(1-pd^{i+2})} d\kappa_{i-1} - \frac{qd^i}{(1-pd^{i+2})}\kappa_{i-2} & \text{if } 1 < i \leq v_{max} - 2 \\ \frac{1-qd^{v_{max}}}{1-d^{v_{max}-1}(q+pd)} qd^{v_{max}-1}\kappa_{v_{max}-2} & \text{if } i = v_{max} - 1 \\ \frac{qd^{v_{max}}}{1-qd^{v_{max}}}\kappa_{v_{max}-1} & \text{if } i = v_{max} \end{cases}$$

Then, $\chi(s|w) = \frac{\prod_{j \in s} \kappa_j}{|A(w)|}$. and $P'(w, y) = \sum_{s \in A(w)} \chi(s|w)\tau(s, y)$.

### 6.6.4 Probabilistic Measures

The computation of probabilistic measures is similar to the DTMC-CA model. Sets $S'_1$ and $S'_2$ are determined as $S'_1 = \{w = (w_1, w_2, \ldots, w_n) \in S' : w_i \neq 0, \forall i = 1, \ldots, n\}$ while $S'_2 = \{y = (y_1, y_2, \ldots, y_n) \in S' : \exists y_i, y_i = 0\}$. The expected time to disconnection, expected time to connection and the probability of connection duration have the same expressions as in Section 6.3.

### 6.6.5 Example

The computation of the state space $S'$ is illustrated through a small example. Assume $k = 4$ and $v_{max} = 2$ and the threshold for connectivity is $r = 2$. The straightforward computation of the state space $S$ would lead to a size of $|S| = 4^4 = 256$ states. The connectivity window model leads to $|S'| = 15$ states, with $n = 3$. Table 6.7 lists the set $S'$ of states involved in the computation of $P'$.

| | | | |
|---|---|---|---|
| (0 0 0) | (0 0 1) | (0 1 1) | (0 1 2) |
| (1 0 0) | (1 0 1) | (1 1 0) | (1 1 1) |
| (1 1 2) | (1 2 1) | (1 2 2) | (2 1 0) |
| (2 1 1) | (2 2 1) | (2 2 2) | |

**Table 6.7** Sample state space $S'$ using connectivity window model. Set $S'$ reduced from potentially a size $|S| = 256$ states to a size $|S'| = 15$ states.

## 6.7 On Incorporating Path Estimates in RBVT Protocols

This section identifies components of RBVT protocols which can benefit from the knowledge of path estimates and, for each one, briefly describes how these estimates can be incorporated

in RBVT protocols to improve their performance. But first, applicability requirements are presented. They specify possible methods to employ in determining the parameters of the analytical models.

### 6.7.1 Applicability Requirements

The main requirement needed to ensure adequate usage of the DTMC-CA/CW analytical route predictions in RBVT is the determination of the traffic density and the probability of speed of entry on road segments. A practical method to obtain daily values for the first requirement is to use existing transportation sensors placed on the roads. For example, the analysis of data collected by such sensors can generate hourly traffic densities and average speeds. These values can then be uploaded in the cars and used on-demand by the models. This method, however, cannot adapt to the actual real-time traffic conditions. A better method would employ wireless sensor networks on the road-side [104], which can compute dynamically the required values and share them with the passing cars.

### 6.7.2 Improving Route Selection

The predictions of expected route durations can be used as a metric in evaluating the quality of routes in RBVT protocols. In RBVT-R, route request packets would not only store the intersections on the path, but also contain the densities and other probability measures of the traversed road segments. As mentioned above, these values are stored at the sensors along the roads. For example, RBVT-R could be modified to wait for more than one route reply and select the best route based on its analytically estimated duration. In RBVT-P, the Connectivity Packet (CP) visits the network and, for each visited road segment, will store not only the connectivity state but also the densities and probability measures from

the sensors. These values are then included in the edge weights for computing the shortest paths in the network.

### 6.7.3 Enhancing RBVT-R Route Maintenance

The expected duration of disconnectivity can be used to enhance the maintenance phase of RBVT-R. When a source node receives a route error in RBVT-R, it waits for a while and attempts to use the same route again. The waiting time is determined, at present, through heuristics. This can be improved by replacing the heuristic wait time with the expected duration of disconnectivity determined by DTMC-CA. Furthermore, the knowledge of the expected connectivity duration of an RBVT-R route can be used to prevent route failures. One possible method would have the source node generate a new route discovery before the currently active route fails. A similar method would start with the first route received in response to a route discovery packet, but switch to better routes which may arrive with subsequent route reply packets.

### 6.7.4 Determining RBVT-P CP Generation Interval

The next improvement considers the interval of generation of new Connectivity Packet (CP) in RBVT-P. Determining the proper inter-CP time interval would consider the expected duration of connectivity on the roads. For areas with short expected connectivity durations, CPs should be generated more often than in dense vehicular traffic.

### 6.7.5 Reducing Overhead Network Traffic

Incorporating the outputs from DTMC-CA can reduce the network overhead of RBVT protocols. Indeed, the expected time of connection and disconnection can inform the source

of the likelihood of success in transmitting a file of a certain of size (e.g., 100MB) at a given instant. The source may then decide to delay the transmission or to transmit just a chunk of the file on the current route. In addition, selecting better routes which have better connectivity, as mentioned above, can further reduce the overhead traffic in the network.

## 6.8 Chapter Summary

The lifetime of RBVT paths represents the duration of network connectivity between nodes of a vehicular network. In this chapter, two analytical models were introduced to estimate the duration of connectivity as well as the duration of disconnectivity between two endpoints in a vehicular network. The models use discrete-time and discrete-space Markov chains to track the evolution of the connectivity status. The first model, DTMC-CA, is based on the Cellular Automaton freeway traffic model. The second model, CW, generalizes DTMC-CA to any traffic model by abstracting the microscopic vehicle movements in the Markov chains. Simulation results showed that DTMC-CA is able to estimate, with high accuracy, the duration of connectivity and disconnection in a vehicular network.

Future work will consider integrating the results presented here in RBVT-P to determine the proper size of the region to be covered by a connectivity packet, and in RBVT-R to better assess the wait time after a path breaks. Protocols at other layers (application layer, transport layer), can also make use of the results of the models presented here to improve their performance in highly dynamic environments such as vehicular networks.

**Algorithm 4** Translate from CW state to physical state

Notation:

$S = \{(w_1, \ldots, w_n), 1 \leq w_i \leq r\}$: State space set of CW matrix
$k$: Number of cells in each physical state
$n = k - r + 1$: Number of cells in each element of $S_{CW}$
$r$: Connectivity range
$w = \{(w_1, \ldots, w_n\}$: State to translate to physical states
$w_i$: Cell $i$ of $w$
$x = \{(x_1, \ldots, x_k\}$: Physical state corresponding to $w$
$x_i$: Cell $i$ of $x$
$column(x, i)$: Column $i$ of vector/matrix $x$. Equivalent to $x_i$ when $x$ is a vector

Initialization

1: $x \leftarrow -1 * [1, 2, \ldots, k]$

Determine constraints

2: **for** $i \leftarrow 1, n-1$ **do**
3: **if** $w_i == w_{i+1}$ **then**
4: $x_{r+i} = x_i$
5: **else if** $w_i == w_{i+1} + 1$ **then**
6: $x_i = 1$
7: $x_{r+i} = 0$
8: **else if** $w_i == w_{i+1} - 1$ **then**
9: $x_i = 0$
10: $x_{r+i} = 1$
11: **else**
12: Return Error $w_i - w_{i+1} > 1$
13: **end if**
14: **end for**

Determine corresponding physical states

15: $z \leftarrow (y(i), y(i) >= 0)$
16: $validity \leftarrow w(1) - \sum_i z(i)$
17: **if** $validity < 0$ **then**
18: Return Invalid state $w$
19: **else if** $validity == 0$ **then**
20: **if** $x(i) < 0$ **then**
21: $x(i) = 0$
22: **end if**
23: Return $x$
24: **end if**
25: $l = length(y) - length(z)$
26: $comb \leftarrow \{c = (c_1, \ldots, c_l), c_i \in y - z\}$
27: $y \leftarrow$ replicate $y$ $|comb|$ rows
28: $y(comb) = 1$
29: $y(not\ in\ comb) = 0$
30: $x \leftarrow$ replicate $x$ $|comb|$ rows
31: $column(x, [1, 2, \ldots, r]) \leftarrow y$
32: **for** $i \leftarrow r+1, k$ **do**
33: **if** $column(x, i) < 0$ **then**
34: $column(x, i) = -1 * column(x, x(1, i))$
35: **end if**
36: **end for**
37: Return $x$

# CHAPTER 7

# CONCLUSION AND FUTURE WORK

This dissertation has addressed the problem of efficient routing and forwarding in VANETs. Through its contributions, it has demonstrated that the integration of VANET features (road topology, real-time road traffic flow, presence of building, etc) in the design of VANET protocols can lead to better performance. In the following, a summary of these contributions and several directions for future work are presented.

## 7.1 Main Contributions

A new class of routing protocols, called Road-Based using Vehicular Traffic (RBVT) routing, was proposed. RBVT improves the end-to-end delivery ratio and the end-to-end delay with as much as 40% for delivery ratio and 85% for delay when compared to existing protocols. These results are due to the low-sensitivity to changes in network topology of the RBVT routes. This low-sensitivity comes from the route representation as sequences of network-connected road intersections reachable through geographic forwarding.

Three optimizations were proposed to further improve the performance of data transfers in VANETs. First, a beaconless MAC layer technique enables geographical data forwarding with a significantly reduced network overhead. The selection of the next-hop to forward a packet is done through a distributed self-election algorithm between nodes. This method is particularly useful in vehicular networks where the speed of moving vehicles quickly render prior position information obsolete. Evaluations showed delivery ratios as much as 3 times higher, while the network overhead was mostly eliminated.

Next, theoretical and simulation studies were performed on the impact of queuing disciplines on end-to-end delay in ad hoc networks. FIFO versus LIFO, and Taildrop versus Frontdrop were considered in all four possible combinations. LIFO was rarely studied in networking literature due to its perceived unfairness. Yet, the analysis and simulations confirm that LIFO with Frontdrop queuing has a level of fairness comparable to traditional FIFO with Taildrop, while reducing the end-to-end delay to less than half that of FIFO with Taildrop.

Finally, the analytical characterization of road-based path durations was derived to help improve RBVT performance. The path duration estimates can be used to select better paths or to decide when to start data transfers in VANETs. Two analytical models were introduced to derive the estimate of RBVT path expected lifetime as well as the expected duration of path disconnectivity. The evaluation showed a high level of accuracy in the prediction of expected duration of RBVT paths, while taking into account the complex and dynamic interactions between intermediate vehicles.

## 7.2 Future Work

**Adaptive queuing mechanism to dynamically select the best queuing policy:** It was shown in this dissertation that the performance of delay sensitive but loss tolerant applications can be improved in multihop ad hoc networks if the queuing strategy was switched from the traditional FIFO with Taildrop queuing discipline to a LIFO with Frontdrop discipline. An extension of this study would determine which parameters to use in deciding whether and when a node should switch its queuing policy. Another extension would perform an implementation of this approach in actual networked devices, which will allow for verification of the results on real-life systems.

**Evaluation of the Connectivity Window (CW) model:** The CW was proposed in this dissertation to estimate the expected duration of connectivity and disconnectivity between nodes of a VANET independently of the microscopic vehicular traffic model used. An extension of the work presented here would evaluate the CW model using different traffic models proposed in the literature and assess whether an independence assumption can be made on the connectivity durations of adjacent stretches of a road. Additionally, the model could be extended to allow for entry and exit from multiple cells.

**Incorporating route lifetime predictions in RBVT:** Two models for estimating the expected duration of connectivity and disconnectivity were proposed in this dissertation. An extension of the work presented here would add the route lifetimes prediction estimates in the selection of RBVT routes and the maintenance phase of reactive RBVT.

**Security in RBVT protocols:** The security level of a system and its vulnerability to potential attacks are important components of an ad hoc communication system. An extension of the work presented here would study the security implications of the RBVT approach and develop methods to safeguard the authenticity and confidentiality of the data transmitted.

To conclude, this dissertation highlighted the performance benefits of including the characteristics of VANETs in the network protocols. This main result and the individual dissertation contributions can be leveraged in future real-world deployment of VANETs to enable a large set of applications ranging from dynamic route planning to file sharing between moving vehicles.

# APPENDIX

## EXPECTED HITTING TIME

In this section, the expressions of $\sigma_{k,N}(.)$, $\phi_{k,N}(.)$ as well as $\zeta_{k,k,S}(s)$ are derived. The notation used is the same as introduced in Section 5.3.

### A Deriving Expressions of $\sigma_{k,N}(.)$, $\phi_{k,N}(.)$

A host queue is modeled as a Markov Chain $X(t)$ on the state space $\{-1, 0, \cdots, N\}$, where $X(t)$ is the state at time $t$. For all disciplines considered, if a packet arrives while $X(t^-) = (k)$ and $k < N$ then the state changes to $X(t^+) = (k+1)$, all packets already in the system keep their position, and the new packet goes into position $k+1$.

Given $X(0^+) = k$, $1 \leq k \leq N$, what is the probability that after time 0 state $(0)$ will be reached before state $(N+1)$ is reached? and what is the probability that state $(N+1)$ will be reached before state $(0)$ is reached? The shorthand "Success" stands for "the state $(0)$ is reached before the state $(N+1)$ is reached", and "Failure" stands for "the state $(N+1)$ is reached before the state $(0)$ is reached".

Define

$$\sigma_{k,N}(s) = E[e^{-sT_{k,N}}\chi(success)], \quad \phi_{k,N}(s) = E[e^{-sT_{k,N}}\chi(failure)]. \tag{A.1}$$

The Difference Equation solved here was considered in (at least) Feller II [105] page 454 etc, but with boundary conditions different from (A.2) below. It may have been considered over the years in other places in Applied Probability, possibly with other or similar Boundary Conditions as in (A.2).

Conditional First Passage Time Distributions for Birth–Death Processes have been studied extensively. Most interest has been in Birth–Death processes with State–Dependent arrival rates $\lambda_n$ and departure rates $\mu - n$. The simple case studied in this Section was not found.

One has $\sigma_{k,N}(0) = \sigma_{k,N}$, $\phi_{k,N}(0) = \phi_{k,N}$, and $\sigma^*_{k,N}(s) = \frac{\sigma_{k,N}(s)}{\sigma_{k,N}}$ is the Laplace Transform of the time until the waiting room is empty for the first time, given one starts in state $(k)$ and given that the waiting room goes empty before the first drop occurs. Similarly $\phi^*_{k,N}(s) = \frac{\phi_{k,N}(s)}{\phi_{k,N}}$ is the Laplace Transform of the time until the first drop occurs, given one starts in state $(k)$ and given a drop occurs before the first time the waiting room goes empty.

The boundary conditions are

$$\sigma_{0,N}(s) \equiv 1, \ \sigma_{N+1,N}(s) \equiv 0, \ \ \phi_{0,N}(s) \equiv 0, \ \ \phi_{N+1,N}(s) \equiv 1. \tag{A.2}$$

Considering the boundary conditions in A.2, one gets

$$\sigma_{k,N}(s) = \frac{\lambda+\mu}{\lambda+\mu+s}\left(\frac{\mu}{\lambda+\mu}\sigma_{k-1,N}(s) + \frac{\lambda}{\lambda+\mu}\sigma_{k+1,N}(s)\right),$$

$$\phi_{k,N}(s) = \frac{\lambda+\mu}{\lambda+\mu+s}\left(\frac{\mu}{\lambda+\mu}\phi_{k-1,N}(s) + \frac{\lambda}{\lambda+\mu}\phi_{k+1,N}(s)\right). \tag{A.3}$$

Solving

$$(\lambda+\mu+s)x = \mu + \lambda x^2 \tag{A.4}$$

gives

$$x_1(s) = \frac{(\lambda+\mu+s) + \sqrt{(\lambda-\mu)^2 + 2(\lambda+\mu)s + s^2}}{2\lambda},$$

$$x_2(s) = \frac{(\lambda+\mu+s) - \sqrt{(\lambda-\mu)^2 + 2(\lambda+\mu)s + s^2}}{2\lambda}. \tag{A.5}$$

Then

$$\sigma_{k,N}(s) = C_{1,S,N}(s)(x_1(s))^k + C_{2,S,N}(s)(x_2(s))^k,$$

$$\phi_{k,N}(s) = C_{1,F,N}(s)(x_1(s))^k + C_{2,F,N}(s)(x_2(s))^k, \tag{A.6}$$

where ($S$ and $F$ stand for "success" and "failure") $C_{1,S,N}(.), C_{2,S,N}(.), C_{1,F,N}(.), C_{2,F,N}(.)$ are computed from the boundary conditions (A.2):

$$C_{1,S,N}(s) + C_{2,S,N}(s) = 1, \;\; C_{1,S,N}(s)(x_1(s))^{N+1} + C_{2,S,N}(s)(x_2(s))^{N+1} = 0,$$

$$C_{1,F,N}(s) + C_{2,F,N}(s) = 0, \;\; C_{1,F,N}(s)(x_1(s))^{N+1} + C_{2,F,N}(s)(x_2(s))^{N+1} = 1. \tag{A.7}$$

This gives:

$$C_{1,S,N}(s) = -\frac{(x_2(s))^{N+1}}{(x_1(s))^{N+1} - (x_2(s))^{N+1}}, \;\; C_{2,S,N}(s) = +\frac{(x_1(s))^{N+1}}{(x_1(s))^{N+1} - (x_2(s))^{N+1}},$$

$$C_{1,F,N}(s) = +\frac{1}{(x_1(s))^{N+1} - (x_2(s))^{N+1}}, \;\; C_{2,F,N}(s) = -\frac{1}{(x_1(s))^{N+1} - (x_2(s))^{N+1}}. \tag{A.8}$$

This finally gives

$$\sigma_{k,N}(s) = \frac{(x_1(s))^{N+1}(x_2(s))^k - (x_2(s))^{N+1}(x_1(s))^k}{(x_1(s))^{N+1} - (x_2(s))^{N+1}},$$

$$\phi_{k,N}(s) = \frac{(x_1(s))^k - (x_2(s))^k}{(x_1(s))^{N+1} - (x_2(s))^{N+1}}. \tag{A.9}$$

Hence

$$\sigma_{k,N}(0) = \frac{1-\rho^{N+1-k}}{1-\rho^{N+1}}, \;\; \phi_{k,N}(0) = \frac{\rho^{N+1-k} - \rho^{N+1}}{1-\rho^{N+1}}. \tag{A.10}$$

If $\rho = 1$ this reduces to

$$\sigma_{k,N}(0) = \frac{N+1-k}{N+1}, \quad \phi_{k,N}(0) = \frac{k}{N+1}. \tag{A.11}$$

## B Deriving Expression of $\zeta_{k,k,S}(.)$

Consider

$$\zeta_{0,L,S}(s) \equiv 1, \quad \zeta_{0,L,F}(s) \equiv 0, \tag{B.12}$$

$$\zeta_{1,N,S} = \frac{\lambda+\mu}{\lambda+\mu+s}.\frac{\mu}{\lambda+\mu} = \frac{\mu}{\lambda+\mu+s}. \tag{B.13}$$

Hence, $\zeta_{k,L,S}(.)$ is known as long as $k = 1$ and $L = N$.

For $1 = k \leq L < N$ one has

$$\zeta_{1,L,S}(s) = \frac{\lambda+\mu}{\lambda+\mu+s}\left(\frac{\mu}{\lambda+\mu} + \frac{\lambda}{\lambda+\mu}\eta_{1,L+1,S}(s)\right) = \frac{\mu}{\lambda+\mu+s} + \frac{\lambda\zeta_{1,L+1,S}(s)}{\lambda+\mu+s}. \tag{B.14}$$

By starting with (B.13) and then iterating (B.14) backward one computes $\zeta_{k,L,S}(.)$ for $k = 1$ for $L = N, N-1, \cdots, 1$.

Suppose $\zeta_{\kappa,L,S}(s)$ is known for all $\kappa < k$ and all $L$, with $k \geq 2$. Then

$$\zeta_{k,N,S}(s) = \frac{\lambda+\mu}{\lambda+\mu+s}\left(\frac{\mu}{\lambda+\mu}\zeta_{k-1,N-1,S}(s) + \frac{\lambda}{\lambda+\mu}\zeta_{k-1,N,S}(s)\right). \tag{B.15}$$

This allows computation of $\zeta_{k,N,S}(s)$. Next for $k \leq L < N$:

$$\zeta_{k,L,S}(s) = \frac{\lambda+\mu}{\lambda+\mu+s}\left(\frac{\mu}{\lambda+\mu}\zeta_{k-1,L-1,S}(s) + \frac{\lambda}{\lambda+\mu}\zeta_{k,L+1,S}(s)\right). \tag{B.16}$$

By iterating (B.16) backward ($L = N-1, N-2, \cdots, k$) one computes $\zeta_{k,L,S}(s)$ for all $1 \leq k \leq L \leq N$.

## REFERENCES


[1] J. Nzouonta, N. Rajgure, G. Wang, and C. Borcea, "VANET routing on city roads using real-time vehicular traffic information," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, 2009.

[2] National Institute of Standards and Technology (NIST), "NIST debuts new approach to ad hoc networks for first responders," http://www.itl.nist.gov/breadcrumb2.pdf. Last accessed April 2009.

[3] The National Oceanic and Atmospheric Administration (NOAA), "NOAA unmanned aircraft system," http://uas.noaa.gov/. Last accessed April 2009.

[4] US Northern Command, "Military airborne imagery supported immediate hurricane recovery efforts," http://www.northcom.mil/news/2008/091908I.html. Last accessed April 2009.

[5] "Wireless: The new backseat driver?" http://news.com.com/2100-11389-5933641.html?tag=tb. Last accessed April 2009.

[6] Cars get PCs despite safety concerns, http://www.post-gazette.com/pg/06026/644908.stm. Last accessed April 2009.

[7] "Chrysler plans in-car web access this year," http://www.washingtonpost.com/wp-dyn/content/article/2008/03/19/AR2008031903447.html. Last accessed April 2009.

[8] Task 3 Final Report, "Identify intelligent vehicle safety applications enabled by DSRC, DOT HS 809 859," http://www-nrd.nhtsa.dot.gov/pdf/nrd-12/1665CAMP3web/index.html. Last accessed April 2009.

[9] S. Dashtinezhad, T. Nadeem, B. Dorohonceanu, C. Borcea, P. Kang, and L. Iftode, "Trafficview: A driver assistant device for traffic monitoring based on car-to-car communication," in *Proceedings 59th IEEE Semiannual Vehicular Technology Conference*, Milan, Italy, May 2004, pp. 2946–2950.

[10] P. Zhou, T. Nadeem, P. Kang, C. Borcea, and L. Iftode, "EZCab: A Cab Booking Application using Short-range Wireless Communication," in *Proceedings 3rd IEEE International Conference on Pervasive Computing and Communications (PERCOM)*, Kauai Island, HI, USA, March 2005, pp. 27–38.

[11] O. Riva, T. Nadeem, C. Borcea, and L. Iftode, "Context-aware migratory services in ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 12, pp. 1313–1328, December 2007.

[12] H. Luo, R. Ramjee, P. Sinha, L. Li, and S. Lu, "UCAN: a unified cellular and ad-hoc network architecture," in *Proceedings 9th Annual International Conference on Mobile*

*Computing and Networking (MOBICOM)*, San Diego, CA, USA, September 2003, pp. 353–367.

[13] J. Nzouonta and C. Borcea, "STEID: A protocol for emergency information. dissemination in vehicular networks," Technical Report, Computer Science Department, NJIT, December 2006, http://www.cs.njit.edu/ borcea/papers/steid.pdf. Last accessed April 2009.

[14] CDMA Development Group, "3G - CDMA2000 1xEV-DO Technologies," http://www.cdg.org/technology/3g_1xEV-DO.asp. Last accessed April 2009.

[15] "Edge introduction of high-speed data in gsm/gprs networks," http://www.ericsson.com/solutions/tems/library/tech_papers/tech_related/edge_wp_technical.pdf. Last accessed April 2009.

[16] Autonet Mobile, http://www.autonetmobile.com. Last accessed April 2009.

[17] E. C. Baig, "Autonet car-based wi-fi system was neat but a bit bumpy," http://www.usatoday.com/tech/columnist/edwardbaig/2008-10-29-baig-autonet-review_N.htm. Last accessed April 2009.

[18] CarTel, MIT, http://cartel.csail.mit.edu.

[19] O. Riva, J. Nzouonta, and C. Borcea, "Context-aware fault tolerance in migratory services," in *Proceedings 5th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous)*, Dublin, Ireland, July 2008.

[20] Q. Sun, S. Y. Tan, and K. C. Teh, "Analytical formulae for path loss prediction in urban street grid microcellular environments," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 4, pp. 1251–1258, July 2005.

[21] X. Zhao, T. Rautiainen, K. Kalliola, and P. Vainikainen, "Path-loss models for urban microcells at 5.3 ghz," *IEEE Antennas and Wireless Propagation Letters*, vol. 5, pp. 152–154, April 2006.

[22] C. Lochert, H. Hartenstein, J. Tian, H. Füßler, D. Hermann, and M. Mauve, "A routing strategy for vehicular ad hoc networks in city environments," in *Proceedings IEEE Intelligent Vehicles Symposium*, Columbus, OH, USA, June 2003, pp. 156–161.

[23] V. Naumov and T. Gross, "Connectivity-aware routing (CAR) in vehicular ad hoc networks," in *Proceedings IEEE International Conference on Computer Communications*, Anchorage, AK, USA, May 2007, pp. 1919–1927.

[24] C. E. Perkins and E. M. Royer, "Ad hoc on-demand distance vector routing," in *Proceedings 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, USA, February 1999, pp. 90–100.

[25] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing*, vol. 353, no. 5, pp. 153–161, 1996.

[26] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings 6th International Conference on Mobile Computing and Networking*, Boston, MA, USA, August 2000, pp. 243–254.

[27] T. V. Lakshman, A. Neidhardt, and T. Ott, "The drop from front strategy in TCP and TCP over ATM," in *Proceedings IEEE International Conference on Computer Communications (INFOCOM)*, Los Angeles, CA, USA, March 1996, pp. 1242–1250.

[28] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, August 1993.

[29] J. Nzouonta, T. Ott, and C. Borcea, "Impact of queue disciplines on congestion in wireless ad hoc networks," November 2008, Submitted to Elsevier Performance Evaluation.

[30] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proceedings ACM Conference on Communications Architectures, Protocols and Applications*, London, United Kingdom, September 1994, pp. 234–244.

[31] V. Namboodiri and L. Gao, "Prediction-based routing for vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 4, pp. 2332–2345, July 2007.

[32] T. Taleb, E. Sakhaee, A. Jamalipour, K. Hashimoto, N. Kato, and Y. Nemoto, "A stable routing protocol to support ITS services in VANET networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3337–3347, November 2007.

[33] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *ACM Wireless Networks*, vol. 7, no. 6, pp. 609–616, November 2001.

[34] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric ad-hoc routing: Of theory and practice," in *Proceedings 22nd Annual Symposium on Principles of Distributed Computing*, Boston, MA, USA, July 2003, pp. 63–72.

[35] L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, J.-P. Hubaux, and J.-Y. L. Boudec, "Self-organization in mobile ad-hoc networks: the approach of terminodes," *IEEE Communications Magazine*, vol. 39, no. 6, pp. 166–174, June 2001.

[36] L. Blazevic, S. Giordano, and J.-Y. L. Boudec, "Self organized terminode routing," *Journal of Cluster Computing*, vol. 5, no. 2, pp. 205–218, April 2002.

[37] J. Tian, L. Han, K. Rothermel, and C. Cseh, "Spatially aware packet routing for mobile ad hoc inter-vehicle radio networks," in *Proceedings IEEE Intelligent Transportation Systems*, Shanghai, China, October 2003, pp. 1546–1551.

[38] B.-C. Seet, G. Liu, B.-S. Lee, C.-H. Foh, K.-J. Wong, and K.-K. Lee, "A-STAR: A mobile ad hoc routing strategy for metropolis vehicular communications," *NETWORKING, Networking Technologies, Services, and Protocols*, vol. 3042, pp. 989–999, April 2004.

[39] M. Jerbi, R. Meraihi, S.-M. Senouci, and Y. Ghamri-Doudane, "GyTAR: improved greedy traffic aware routing protocol for vehicular ad hoc networks in city environments," in *Proceedings 3rd ACM International Workshop on Vehicular Ad Hoc Networks (VANET)*, Los Angeles, CA, USA, September 2006, pp. 88–89.

[40] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter, "MDDV: A Mobility-Centric Data Dissemination Algorithm for Vehicular Networks," in *Proceedings 1st ACM International Workshop on Vehicular Ad Hoc Networks (VANET)*. Philadelphia, PA, USA: ACM, October 2004, pp. 47–56.

[41] J. Zhao and G. Cao, "Vadd: Vehicle-assisted data delivery in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1910–1922, May 2008.

[42] S. Savasta, M. Pini, and G. Marfia, "Performance assessment of a commercial GPS receiver for networking applications," in *Proceedings IEEE International Consumer Communications and Networking Conference*, Las Vegas, NV, USA, January 2008, pp. 613–617.

[43] H. Füßler, J. Widmer, M. Käsemann, M. Mauve, and H. Hartenstein, "Contention-based forwarding for mobile ad hoc networks," *Elsevier Ad Hoc Networks*, vol. 1, no. 4, pp. 351–369, November 2003.

[44] M. Heissenbuttel and T. Braun, "A novel position-based and beacon-less routing algorithm for mobile ad-hoc networks," in *Proceedings 3rd IEEE Workshop on Applications and Services in Wireless Networks*, Bern, Switzerland, July 2003, pp. 197–210.

[45] M. Zorzi and R. R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Multihop performance," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 337–348, October–December 2003.

[46] K. Egoh and S. De, "Priority-based receiver-side relay election in wireless ad hoc sensors networks," in *Proceedings International Conference on Wireless Communications and Mobile Computing*, Vancouver, BC, Canada, July 2006, pp. 1177–1182.

[47] ——, "A multi-criteria receiver-side relay election approach in wireless ad hoc networks," in *Proceedings Military Communications Conference (MILCOM)*, Washington DC, District of Columbia, USA, October 2006, pp. 1–7.

[48] M. Chawla, N. Goel, K. Kalaichelvan, A. Nayak, and I. Stojmenovic, "Beaconless position-based routing with guaranteed delivery for wireless ad hoc and sensor networks," *Acta Automatica Sinica*, vol. 32, no. 6, pp. 847–855, November 2006.

[49] S. Cen, P. C. Cosman, and G. M. Voelker, "End-to-end differentiation of congestion and wireless losses," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 703–717, October 2003.

[50] M. Chen and A. Zakhor, "Rate control for streaming video over wireless," *IEEE Wireless Communications*, vol. 12, no. 4, pp. 59–65, August 2005.

[51] Y. Yi and S. Shakkottai, "Hop-by-hop congestion control over a wireless multi-hop network," *IEEE/ACM Transactions on Networking*, vol. 15, no. 1, pp. 133–144, February 2007.

[52] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification," 2003.

[53] C. Sarr, C. Chaudet, G. Chelius, and I. Guerin-Lassous, "Bandwidth estimation for IEEE 802.11-based ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 8, pp. 1228–1241, October 2008.

[54] E. Setton, T. Yoo, X. Zhu, A. Goldsmith, and B. Girod, "Cross-layer design of ad hoc networks for real-time video streaming," *IEEE Wireless Communications*, vol. 12, no. 4, pp. 59–65, August 2005.

[55] E. Setton, J. Noh, and B. Girod, "Congestion-distorsion optimized peer-to-peer video streaming," in *Proceedings IEEE International Conference on Image Processing (ICIP)*, Atlanta, GA, USA, October 2006, pp. 721–724.

[56] A. Majumdar, D. G. Sachs, I. V. Kozintsev, K. Ramchandran, and M. M. Yeung, "Multicast and unicast real-time video streaming over wireless LANs," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 524–534, June 2002.

[57] E. Kohler, M. Handley, and S. Floyd, "Designing DCCP: Congestion control without reliability," in *Proceedings of ACM SIGCOMM*, Pisa, Italy, September 2006, pp. 27–38.

[58] Y.-C. Tseng, Y.-F. Li, and Y.-C. Chang, "On route lifetime in multihop mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 366–376, October–December 2003.

[59] Y. Han, R. J. La, A. M. Makowski, and S. Lee, "Distribution of path durations in mobile ad-hoc networks: Palm's theorem to the rescue," *Elsevier Computer Networks*, vol. 50, no. 12, pp. 1887–1900, August 2006.

[60] O. Dousse, P. Thiran, and M. Hasler, "Connectivity in ad-hoc and hybrid networks," in *Proceedings IEEE International Conference on Computer Communications (INFOCOM)*, New York, NY, USA, June 2002, pp. 1079–1088.

[61] M. Khabazian and M. K. Mehmet, "A performance modeling of connectivity in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 4, pp. 2440–2450, July 2008.

[62] M. Fiore and J. Härri, "The networking shape of vehicular mobility," in *Proceedings 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Hong Kong, China, January 2008, pp. 261–272.

[63] M. M. Artemy, W. Robertson, and W. J. Phillips, "Connectivity in inter-vehicle ad hoc networks," in *Proceedings IEEE ECE Canadian Conference (CCECE)*, Niagara Falls, Canada, May 2004, pp. 293–298.

[64] A. Nandan, S. Das, G. Pau, and M. Gerla, "Co-operative downloading in vehicular ad-hoc wireless networks," in *Proceedings 2nd IEEE Conference on Wireless On-demand Network Systems and Services (WONS)*, St. Moritz, Switzerland, January 2005, pp. 32–41.

[65] T. Li, S. K. Hazra, and W. Seah, "A position-based routing protocol for metropolitan bus networks," in *Proceedings IEEE 61st Vehicular Technology Conference VTC-Spring*, Stockholm, Sweden, June 2005, pp. 2315–2319.

[66] U.S.Census Bureau - TIGER/Line, http://www.census.gov/geo/www/tiger/.

[67] The Institute of Electrical and Electronic Engineers (IEEE), "Wireless LAN medium access control (MAC) and physical layer scpecifications," http://standards.ieee.org/getieee802/802.11.html. Last accessed April 2009.

[68] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proceedings 5th ACM/IEEE International Conference on Mobile Computing and Networking*, Seattle, WA, USA, August 1999, pp. 151–162.

[69] L. Briesemeister and G. Hommel, "Role-based multicast in highly mobile but sparsely connected ad hoc networks," in *Proceedings 1st Annual Workshop on Mobile and Ad Hoc Networking and Computing (MOBIHOC)*, Boston, MA, USA, August 2000, pp. 45–50.

[70] N. Rajgure, "Road-based proactive routing protocols for vehicular networks," 2008, Master's Thesis, New Jersey Institute of Technology.

[71] The network simulator: NS-2, http://www.isi.edu/nsnam/ns. Last accessed April 2009.

[72] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," 2003, Request for Comments (RFC) 3626.

[73] Centre for Applied Informatics (ZAIK) and the Institute of Transport Research German Aerospace Centre, "SUMO - Simulation of Urban MObility," http://sumo.sourceforge.net. Last accessed April 2009.

[74] S. Krauß, P. Wagner, and C. Gawron, "Metastable states in a microscopic model of traffic flow," *Physical Review E*, vol. 55, no. 304, pp. 55–97, April 1997.

[75] N. Eude, B. Ducourthial, and M. Shawky, "Enhancing ns-2 simulator for high mobility ad hoc networks in car-to-car communication," in *Proceedings 7th IFIP International Conference on Mobile and Wireless Communications Networks*, Marrakesh, Morocco, September 2005.

[76] T. S. Rappaport, "Wireless communications principles and practice," 1996, Prentice Hall.

[77] B. Karp, "Greedy perimeter stateless routing (GPSR)," http://www.icir.org/bkarp/gpsr/gpsr.html. Last accessed April 2009.

[78] HIPERCOM, INRIA, "Optimized link state routing protocol (OLSR)," http://hipercom.inria.fr/olsr. Last accessed April 2009.

[79] M. L. Huson and A. Sen, "Broadcast scheduling algorithms for radio networks," in *Proceedings IEEE Military Communications Conference (MILCOM)*, San Diego, CA, USA, November 1995, pp. 647–651.

[80] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Impact of radio irregularity on wireless sensor networks," in *Proceedings 2nd International Conference on Mobile Systems, Applications, and Services (MOBISYS)*, Boston, MA, USA, June 2004, pp. 125–138.

[81] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *Proceedings 1st International Conference on Embedded Networked Sensor Systems (SENSYS)*, Los Angeles, CA, USA, November 2003, pp. 1–13.

[82] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part II–the hidden terminal problem in carrier sense multiple-access and the busy-tone solution," *IEEE Transactions on Communications*, vol. COM-23, no. 12, pp. 1417–1433, 1975.

[83] P. G. Gipps, "A behavioural car-following model for computer simulation," *Transportation Research Board*, vol. 15, pp. 105–111, 1981.

[84] ——, "A model for the structure of lane-changing decisions," *Transportation Research Board*, vol. 20B, no. 5, pp. 403–414, 1986.

[85] R. S. Prasad, C. Dovrolis, and M. Thottan, "Router buffer sizing revisited: the role of the output/input capacity ratio," in *Proceedings 2007 ACM CoNEXT Conference*, New York, NY, USA, December 2007, pp. 1–12.

[86] R. Chertov, S. Fahmy, and N. B. Shroff, "A device-independent router model," in *Proceedings 27th IEEE Conference on Computer Communications (INFOCOM)*, Phoenix, AZ, USA, April 2008, pp. 1642–1650.

[87] Juniper Networks, "Configuring the ATM1 queue length," https://www.juniper.net/techpubs/software/junos/junos92/swconfig-network-interfaces/configuring-the-atm1-queue-length.html. Last accessed April 2009.

[88] G. Holland and N. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," *Wireless Networks*, vol. 8, no. 2, pp. 275–288, March 2002.

[89] T. Razafindralambo and I. Guerin-Lassous, "Increasing fairness and efficiency using the MadMac protocol in ad hoc networks," *Elsevier Ad Hoc Networks*, vol. 6, no. 3, pp. 408–423, May 2008.

[90] K. Nagel and M. Schreckenberg, "A cellular automaton model for freeway traffic," *Journal de Physique I*, vol. 2, no. 12, pp. 2221–2229, 1992.

[91] A. Trivino-Cabrera, J. G. de-la Nava, E. Casilari, and F. J. Gonzalez-Caete, "Application of path duration study in multihop ad hoc networks," *Telecommunication Systems*, vol. 38, no. 1-2, pp. 3–9, June 2008.

[92] D. Yu, H. Li, and I. Gruber, "Path availability in ad hoc network," in *Proceedings 10th International Conference on Telecommunications*, Tahiti, Papeete, French Polynesia, February 2003, pp. 383–387.

[93] F. Bai, N. Sadagopan, and A. Helmy, "Important: a framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks," in *Proceedings IEEE International Conference on Computer Communications (INFOCOM)*, San Francisco, CA, USA, April 2003, pp. 825–835.

[94] M. Rickert, K. Nagel, M. Schreckenberg, and A. Latour, "Two lane traffic simulations using cellular automata," 1995, http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/9512119. Last accessed April 2009.

[95] A. Schadschneider and M. Schreckenberg, "Garden of Eden states in traffic models," *Journal of Physics A: Mathematical and General*, vol. 31, no. 11, pp. 225–231, March 1998.

[96] P.-J. Courtois, "Decomposability: Queueing and Computer System Applications," Academic Press. 1977.

[97] T. Dayar and W. J. Stewart, "Quasi-lumpability, lower bounding coupling matrices, and nearly completely decomposable markov chains," *SIAM Journal on Matrix Analysis and Applications*, vol. 18, no. 2, pp. 482–498, April 1997.

[98] P. Buchholz, "Lumpability and nearly-lumpability in hierarchical queueing networks," in *Proceedings IEEE International Computer Performance and Dependability Symposium*, Los Alamitos, CA, USA, April 1995, pp. 82–91.

[99] P. Buchholz, "Exact and ordinary lumpability in finite markov chains," *Journal of Applied Probability*, vol. 31, no. 2, pp. 59–75, 1994.

[100] J. G. Kemeny and J. L. Snell, "Finite Markov Chains," 1976, Springer-Verlag, New York.

[101] J. R. Norris, "Markov Chains," Cambridge University Press. 1996.

[102] S. Krauß, "Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics," ISSN 1434-8454; April 1998.

[103] M. Schreckenberg, A. Schadschneider, K. Nagel, and N. Ito, "Discrete stochastic models for traffic flow," *Physical Review E (Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics)*, vol. 51, no. 4, pp. 2939–2949, April 1995.

[104] J.-M. Bohli, A. Hessler, O. Ugus, and D. Westhoff, "A secure and resilient WSN roadside architecture for intelligent transport systems," in *Proceedings 1st ACM Conference on Wireless Network Security (WiSec)*, Alexandria, VA, USA, April 2008, pp. 161–171.

[105] W. Feller, "An Introduction to Probability Theory and its Applications, Vol II," November 1966, Second Corrected Printing, Wiley.