University of Cyprus

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

DESIGN AND ANALYSIS OF NOVEL ROUTING PROTOCOLS FOR VEHICULAR DELAY TOLERANT NETWORKS

ANNA SIDERA

A Dissertation Submitted to the University of Cyprus in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

May, 2015

VALIDATION PAGE

Doctoral Candidate: Anna Sidera

Doctoral Thesis Title: Design and Analysis of Novel Routing Protocols for Vehicular Delay Tolerant Networks

*The present Doctoral Dissertation was submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy at the **Department of Electrical and Computer Engineering**, and was approved on 21/05/2015 by the members of the **Examination Committee***

Committee Chair: ....
Christos Panayiotou, Associate Professor, Department of
Electrical and Computer Engineering, University of Cyprus

Research Co-supervisor: ....
Stavros Toumpis, Assistant Professor, Department of
Informatics, Athens University of Economics and Business

Research Co-supervisor: ....
Christoforos Hadjicostis, Professor, Department of
Electrical and Computer Engineering, University of Cyprus

Committee Member: ....
Georgios Ellinas, Associate Professor, Department of
Electrical and Computer Engineering, University of Cyprus

Committee Member: ....
Vasos Vassiliou, Assistant Professor, Department of
Computer Science, University of Cyprus

Committee Member: ....
Fragkiskos Papadopoulos, Lecturer, Department of Electrical
Engineering, Computer Engineering and Informatics, Cyprus
University of Technology

DECLARATION OF DOCTORAL CANDIDATE

The present doctoral dissertation was submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy of the University of Cyprus. It is a product of original work of my own, unless otherwise mentioned through references, notes, or any other statements.

Doctoral Candidate: Anna Sidera

Signature: ....

Περίληψη

Σε αύτη τη διατριβή προτείνουμε δύο πρωτόκολλα δρομολόγησης για ασύρματα δίκτυα με ανοχή στην καθυστέρηση.

Στο πρώτο μέρος της διατριβής παρουσιάζουμε το πρωτόκολλο Delay Tolerant Firework Routing (DTFR), ένα πρωτόκολλο για δίκτυα με ανοχή στην καθυστέρηση, τα οποία αποτελούνται από έναν πολύ μεγάλο αριθμό κόμβων, οι οποίοι έχουν τρόπο να ξέρουν τη θέση τους. Δίκτυα με αυτές τις ιδιότητες υπάρχουν σε πολλές πρακτικές εφαρμογές, για παράδειγμα δίκτυα που αποτελούνται από αυτοκίνητα.

Στο DTFR, το πακέτο ταξιδεύει από την πηγή στην τοποθεσία που εκτιμάται ότι βρίσκεται ο προορισμός, χρησιμοποιώντας εκπομπές ψηλής προτεραιότητας και μια παραλλαγή της γεωγραφικής δρομολόγησης η οποία έχει ανοχή στην καθυστέρηση. Όταν το πακέτο φτάσει στην τοποθεσία που εκτιμάται ότι βρίσκεται ο προορισμός, δημιουργείται ένας αριθμός αντιγράφων και τα αντίγραφα ταξιδεύουν στην γύρω περιοχή προς όλες τις κατευθύνσεις.

Χρησιμοποιώντας προσομοίωση σε ένα σενάριο δικτύου με αυτοκίνητα, συγκρίνουμε το DTFR με δύο βασικά πρωτόκολλα για ασύρματα δίκτυα με ανοχή στην καθυστέρηση, το flooding και το Spray and Wait, δύο πρωτόκολλα που προτάθηκαν πρόσφατα για δίκτυα αυτοκινήτων, το GeoCross και το GeoDTN+Nav, και ένα πρωτόκολλο το οποίο σχεδιάσαμε για να το χρησιμοποιήσουμε σαν άνω φράγμα, το Bethlehem Routing (BR).

Επίσης αναπτύσσουμε ανάλυση βασισμένη σε εργαλεία στοχαστικής γεωμετρίας, ένα αριθμό υποθέσεων που απλοποιούν το πρόβλημα, και έναν μικρό αριθμό προσεκτικά επιλεγμένων προσεγγίσεων. Βρίσκουμε εκφράσεις για τον ρυθμό μετάδοσης δεδομένων και για τον χρόνο που χρειάζονται για να παραδοθούν τα πακέτα, στο DTFR και στο BR.

Στη συνέχεια διερευνούμε την χρήση διάφορων κανόνων για να επιλέγουμε τον επόμενο κόμβο που θα πάρει το πακέτο καθώς το πακέτο ταξιδεύει από κόμβο σε κόμβο για να φτάσει στην τοποθεσία του προορισμού του. Αυτοί οι κανόνες λαμβάνουν υπόψη τη θέση και την ταχύτητα του κόμβου που έχει το πακέτο και του κόμβου που είναι υποψήφιος για

να πάρει το πακέτο και την τοποθεσία που έχει στόχο να πάει το πακέτο. Βρίσκουμε για αυτούς τους κανόνες την μέση καθυστέρηση και το μέσο κόστος που χρειάζεται για να παραδοθεί ένα πακέτο, με προσομοίωση ή και ανάλυση.

Στο δεύτερο μέρος της διατριβής, παρουσιάζουμε το πρωτόκολλο Extended Minimum Estimated Expected Delay (EMEED). Αυτό το πρωτόκολλο είναι σχεδιασμένο για ασύρματα δίκτυα με ανοχή στην καθυστέρηση στα οποία οι κόμβοι συναντούν ορισμένους από τους κόμβους του δικτύου πιο συχνά από άλλους.

Στο EMEED, κάθε δύο κόμβοι που συναντούνται συχνά, είτε άμεσα είτε μέσω άλλων, μεταδίδουν στο δίκτυο την μέση τιμή του χρόνου που πρέπει να περιμένουν μέχρι να συναντηθούν. Οι κόμβοι δρομολογούν πακέτα με βάση πίνακες δρομολόγησης που δημιουργούνται χρησιμοποιώντας αυτές τις μέσες τιμές. Όταν η κύρια του παράμετρος, η ακτίνα επαφής, είναι ίση με τη μονάδα, το EMEED λειτουργεί παρόμοια με το γνωστό πρωτόκολλο Minimum Estimated Expected Delay (MEED).

Όμως, χρησιμοποιώντας χαρακτηριστικά παραδείγματα και προσομοίωση, δείχνουμε ότι για πολλά σενάρια κίνησης, όταν η ακτίνα επαφής είναι μεγαλύτερη από τη μονάδα, το EMEED είναι καλύτερο από το MEED όσον αφορά τον λόγο του αριθμού των πακέτων που φτάνουν στον προορισμό τους εντός της δοθείσας προθεσμίας προς τον αριθμό των πακέτων που δημιουργούνται, με μια μικρή αύξηση στο εύρος ζώνης που σπαταλείται για να εκπεμφθούν πακέτα ελέγχου.

Συγκρίνουμε με προσομοίωση το EMEED με το BUBBLE, ένα πρωτόκολλο που προτάθηκε πρόσφατα για σενάρια στα οποία κάποιοι κόμβοι συναντούν ορισμένους από τους κόμβους του δικτύου πιο συχνά από άλλους. Βρίσκουμε ότι το EMEED έχει συνολικά συγκρίσιμο λόγο αριθμού πακέτων που φτάνουν προς αριθμό πακέτων που δημιουργούνται σε σχέση με το BUBBLE αλλά χρησιμοποιεί λιγότερο εύρος ζώνης για τη λειτουργία του. Επίσης συγκρίνουμε το EMEED με το flooding και το Spray and Wait. Η σύγκριση γίνεται σε δύο σενάρια δικτύων, ένα σχετικό με δίκτυα τσέπης και ανθρώπους που φέρουν μαζί τους τους κόμβους και ένα σχετικό με δίκτυα που αποτελούνται από αυτοκίνητα στο οποίο χρησιμοποιούμε το εργαλείο προσομοίωσης κίνησης SUMO.

Abstract

In this thesis we propose two routing protocols for wireless mobile Delay Tolerant Networks (DTNs).

In the first part of the thesis, we present the Delay Tolerant Firework Routing (DTFR) protocol, a protocol designed for use in DTNs that consist of a very large number of location aware, highly mobile nodes. Networks with these properties appear frequently in many settings, notably in vehicular networks (VANETs).

Under DTFR, each data packet travels from the source to the estimated location of the destination using high priority transmissions and a delay tolerant variant of geographic forwarding. Once there, a number of packet replicas are created, and the replicas proceed to travel through the area where the destination is expected to be.

Using simulations in an urban setting, we compare DTFR with two baseline protocols (flooding and Spray and Wait), two recently proposed state of the art protocols (GeoCross and GeoDTN+Nav), and an idealistic protocol of our design which we term Bethlehem Routing (BR). For a wide range of environmental parameters, DTFR performs significantly better than other realistic protocols, in terms of throughput and delay, and close to the upper performance bounds of BR.

We also develop an analytical framework based on stochastic geometry tools, a number of simplifying assumptions, and a small number of judiciously chosen approximations. Using this framework, we develop approximate closed form expressions for the average end-to-end throughput and delivery delay of DTFR and BR.

We then explore the use of different rules for choosing the next hop, as the packet travels toward its target location. These rules take into account the position and velocity of the current holder and candidates for receiving the packet and the position of the target location. We evaluate these rules, by analysis and simulation, in terms of the average packet delay and the average packet cost they incur per unit of progress.

In the second part of the thesis, we present the Extended Minimum Estimated

Expected Delay (EMEED) protocol which is designed for use in wireless DTNs that consist of a large number of highly mobile nodes with non-uniform correlated mobility patterns.

Under the EMEED protocol, any two nodes that are often in contact, either directly or through a multihop path, disseminate in the network the expected time they have to wait until they come into contact. Nodes route packets according to routing tables created using these expected times. When its main parameter, the *contact radius*, is equal to unity, the EMEED protocol operates similarly to the well known Minimum Estimated Expected Delay (MEED) protocol.

However, using simulations, we show that for many mobility scenarios, when the contact radius is greater than unity, the EMEED protocol performs far better than MEED, in terms of throughput and delay, with only a modest increase in the control overhead.

We compare, using simulations, EMEED with BUBBLE, a state of the art protocol that was also proposed for a scenario where the nodes have preferred locations of movement, and find that EMEED has an overall comparable packet delivery rate but uses the available bandwidth much more judiciously than BUBBLE. We also compare EMEED with flooding and with Spray and Wait. The comparison is made in two mobility scenarios, one related to pocket networks and human levels of mobility, and one related to VANETs, using the SUMO mobility simulation tool.

Acknowledgments

I thank Dr. Stavros Toumpis for giving me the opportunity to do the research I did that is presented in this thesis, for giving me the necessary guidance to do it, and for coming to Cyprus for my quals, my proposal, and my defense. Also, I thank all the professors that accepted to be in the examining committee for doing so. Also, I thank my friends and family for their support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Objectives

Delay/Disruption Tolerant Networks (DTNs) are networks that can tolerate large delays in the delivery of data, to the point that the topology of the network changes substantially during the time a packet is *en route* to its destination [1]. Wireless DTN routing protocols take advantage of changes in the topology induced by node mobility and route packets to their destinations even if there is no end-to-end path from the source to the destination at any time. Many networks may be viewed as wireless DTNs, for example, wildlife tracking networks [2, 3], Vehicular Ad hoc Networks (VANETs) [4, 5], Unmanned Aerial Vehicle (UAV) networks [6, 7], and Pocket Switched Networks (PSNs) [8].

In contrast to traditional networks, in DTNs there are two distinct time scales; this is because there is no end-to-end path from source to destination, therefore the packet needs to wait for the topology to change. While the packet is routed from the source to the destination, it travels by transmission between nodes that are in the same partition; this happens on the usual time scale of a few seconds. Also, it waits for the topology to change and during that time it travels by transport; this transport takes place over a second, much larger time scale. Note that there may be delay when the packet travels by transmission because many nodes need to use the wireless channel, each node has many packets to transmit, and some transmissions are unsuccessful so some packets have to be retransmitted; however the time it takes the packet to wait for the topology to change is much larger than the time it takes the packet to travel by transmission. In DTNs the protocols are designed using the fact that the acceptable delay for the delivery of the packet is so large that the packet will have time to wait for the topology to change. Therefore, a part of the operation takes place in the small time scale, and another part of the operation takes place in the larger time scale.

Delay Tolerant Networks have been extensively studied in the context of network optimization, and in particular dynamic flows [9]. More recently, a number of applications have been proposed in various wireless settings [3, 2, 10] and the Internet [11]. Furthermore, a number of theoretical studies have shown that a tradeoff exists between the packet delay and the throughput [12, 13]. A significant amount of work has also been devoted to the design of practical routing protocols [1, 14, 15, 16]. The current state of the art in wireless DTN research, and the major challenges and approaches facing the research community are summarized, among other works, in [17, 18, 19]. In Chapter 2 we discuss DTN research that is particularly relevant to our work.

As Chapter 2 shows, there are a number of open problems that need to be solved by the research community before DTNs are widely used. Notably, we need better protocols for routing, that take into a more careful account the environment where they are used. As can be seen by the abundance of DTN routing protocols proposed in the literature, no single routing protocol can satisfy all requirements for all DTN network types. Also, we need better analytical tools and better simulation tools. It is difficult to model mobility and, for this reason, more research needs to be done in the area of modeling mobility. Also, it is needed to propose protocols for security and protocols to give motivation to users to help in routing.

When starting work on this thesis, the objective was to design routing protocols that are better than the existing ones, taking into account the above mentioned challenges. Furthermore, the aim was to perform realistic simulation of very large DTNs, with a number of nodes on the order of 10^4, using desktop computers. Indeed, in the literature there were no realistic simulations with such a large number of nodes; this is a significant gap, as many DTNs are expected to have many thousands of nodes. During the course of the research in this thesis, other issues that emerged were, notably, the coupling of DTN routing with geographic routing, the study of DTN routing performance under realistic applications, and exploring the effects of transient nodes.

1.2 Motivation

To provide motivation for why it is important to study wireless mobile DTNs, we present some examples of wireless DTN applications in the following.

An important potential application of wireless mobile DTNs is establishing emergency communication in vehicular scenarios [20]. Consider, for example, the case of a car involved in an accident that immobilizes it in the middle of a road. If all cars carried suitable communication equipment, a distress call could be issued by the car involved in the accident. If the density of the cars is sufficiently large, then the cars will form a vehicular ad hoc network, and the distress call will swiftly reach all nearby cars. If, however, the density of the nodes is not very large, then the resulting wireless network will be partially connected, and any non delay tolerant ad hoc routing protocol will not be of practical use.

A delay tolerant approach, on the other hand, might be life saving. Consider the following delay tolerant flooding approach: as soon as the car is involved in an accident, its radio *continuously* broadcasts a distress signal. Whenever a node receives this signal, it also starts to continuously broadcast it, and this continues until the relaying node is sufficiently away from the location where the distress call originated. If this protocol is used, then any car *leaving* the location of the accident, in either direction, will notify, in a matter of minutes, all cars *approaching* the location of the accident. As this example shows, delay tolerant networking can save lives, in cases where plain ad hoc networking cannot. As a specific example, in the 1999 Mont Blanc Tunnel disaster [20, 21], 38 people died when a truck carrying flammable material caught fire. While the truck was burning, cars started to pile up on both sides of the road; some tried to reverse their direction of travel, but for most this was impossible due to the incoming traffic and low visibility; had the drivers been warned sufficiently well in advance, lives would have been saved.

Another potential application of delay tolerant networks is through their fusion with wireless sensing [22]. Imagine a set of *sensing nodes* moving around in a region of space collecting data and possibly, but not necessarily, exchanging data when they meet. A second set of *collecting nodes* travels the region and collects the data of the sensing nodes. It would be possible to equip all nodes with radios that are powerful enough so that the resulting network is always connected. However, this would imply that the energy of the nodes would be dissipated very fast. If the data collected is delay tolerant, it makes sense to reduce the transmission powers of the nodes, and hence conserve energy, at the cost of increasing the delay with which data arrives at collectors. This idea was explored in [3] and, more recently, in [23, 24]. For example in [23] the sensing nodes are private vehicles, that record videos, detect toxic chemicals, or recognize license plates, and the collecting nodes are police vehicles.

In [25] the authors discuss applications in space-based networking environments. They assume that the contacts are predictable and that their mobility patterns are globally distributed and known.

Many other applications have been proposed for wireless delay tolerant networks; we refer the reader to [26, 27, 28]. Also, many works dealing with specific applications of DTNs are summarized in Chapter 2.

1.3 Organization of the Dissertation

The organization of the dissertation is as follows.

In Chapter 2, we conduct a literature review on the topic of DTNs, in particular connecting the work of others to our own.

In Chapter 3, we present the Delay Tolerant Firework Routing (DTFR) protocol, a protocol for performing routing in Wireless DTNs where the number of nodes is very large (in the orders of thousands and tens of thousands) and where nodes move with large speeds. It requires that nodes are capable of knowing their location, and the location of their destinations (possibly with some error) through the combined use of GPS receivers or a similar technology and a location service. DTFR is superior to other protocols we compare it with when the network is disconnected but not very sparse. A notable domain where all these assumptions frequently hold is VANETs [29, 30, 4].

Broadly speaking, DTFR works as follows: When a packet is created, the source uses information about the location of its destination, provided from a location service and/or previously received data packets, to create an estimate about its current location, which we call the Firework Center (FC). The source sends a single copy of the packet to the FC, using high priority transmissions and a novel delay tolerant geographic forwarding rule, which we term Greedy Lazy Forwarding (GLF). Under GLF, if the current holder of a packet sees another node closer to the destination, it forwards the packet to that node, otherwise the current holder takes no action, and just waits for one such node to appear. Once the packet arrives at the FC, a number of replicas are created, which proceed to propagate in various directions, also using GLF, systematically covering the area where the destination is expected to be.

In Fig. 1.1 we plot an example trajectory of a packet routed using DTFR obtained using simulation. If the packet or a replica traveled from one point to another using wireless transmission we plot a continuous line segment joining the two points. On the other hand, if the packet or a replica traveled from one point to another using physical transport we do not plot anything. We observe that the packet travels from the source to the FC and the replicas travel from the FC to the FEs using a combination of wireless transmission and physical transport. The trajectories of the packet and the replicas trace a pattern similar to the pattern created by a ‘palm tree’ firework [31], hence the name of the protocol. For certain combinations of values of the transmission range, the velocity of the nodes, and the packet arrival rate, when the packet travels using GLF, the velocity of the packet is greater than the velocity of the nodes. This is very important for our protocol to work. Indeed, the packet travels from the source to the FC and then the replicas travel from the FC to the FEs. At the same time the destination, that has smaller velocity than the velocity of the packet, travels from the FC to a point closer to the FC than the FEs or at most at the same distance from the FC as the FEs. Thus the packet is guaranteed to be able to reach the destination.

We develop a simulation tool that is specifically designed for DTNs and is capable of simulating networks of 10^4 nodes while using realistic MAC and physical layers. To the best of our knowledge, this is the first simulation that can simulate 10^4 nodes with this level of detail and it can help significantly in the design of other protocols as well. For this reason, it is publicly available at [32].

Using simulation in an urban setting, we compare DTFR with two baseline protocols (flooding and Spray and Wait), one recently proposed state of the art protocol (GeoDTN+Nav), and an idealistic protocol of our design which we term Bethlehem Routing (BR). For a wide range of environmental parameters, DTFR performs significantly better than other realistic protocols, in terms of throughput and delay, and close to the upper performance bounds of BR.

The chapter is based on the following publications [33, 5]:

- A. Sidera and S. Toumpis, “DTFR: A geographic routing protocol for wireless delay tolerant networks” in Proc. IFIP MedHocNet 2011, Favignana Island, Sicily, Italy, June 2011.

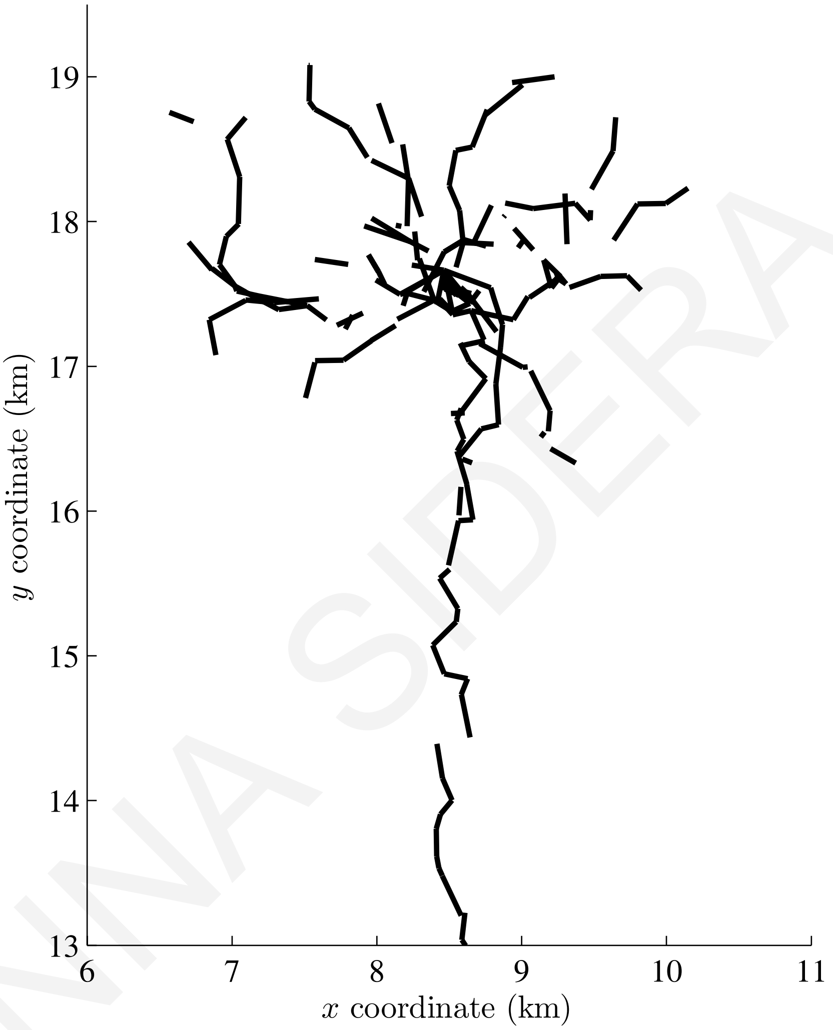


Figure 1.1: Example trajectory of a packet and its replicas, being routed with DTFR. The source is outside the figure. Wireless transmissions are denoted with continuous lines. Physical transports are not denoted. Hence, disconnections correspond to physical transports.

- A. Sidera and S. Toumpis, “Delay tolerant firework routing: A geographic routing protocol for wireless delay tolerant networks,” EURASIP Journal on Wireless Communications and Networking, 2013.

In Chapter 4, we develop an analytical framework for studying GLF based on stochastic geometry tools, a number of simplifying assumptions, and a small number of judiciously chosen approximations. Using this framework, we develop approximate closed form expressions for the average end-to-end throughput and delivery delay of DTFR and BR.

The chapter is based on the following publication [5]:

- A. Sidera and S. Toumpis, “Delay tolerant firework routing: A geographic routing protocol for wireless delay tolerant networks,” EURASIP Journal on Wireless Communications and Networking, 2013.

In Chapter 5, we continue our investigation of delay tolerant routing using tools from stochastic geometry, and in particular we explore the use of different rules for choosing the next hop, as the packet travels toward its target location. These rules take into account the position of the target location and the position and velocity of the current holder and the potential next holders. We evaluate these rules in terms of the average packet delay and the average packet cost they incur, extending the framework we developed in Chapter 4.

The chapter is based on the following publication [34]:

- A. Sidera and S. Toumpis, “On the delay/cost tradeoff in wireless mobile delay tolerant networks,” in WiOpt, Hammamet, Tunisia, May. 2014.

In Chapter 6, we present the Extended Minimum Estimated Expected Delay protocol (EMEED), a protocol for performing routing in Wireless DTNs. EMEED is designed for use in networks where the number of nodes is large and they exhibit non-uniform mobility patterns, for example each node visits some locations more often than others. There are many DTNs for which these assumptions hold, for example, Pocket Switched Networks [8, 35] and vehicular DTNs [4, 33].

Under the EMEED protocol, any two nodes that are in contact often, either directly or through a local multihop path, disseminate in the network the expected time they have to wait until they come in contact. Nodes create routing tables such that the cost of a link between two nodes is related to this expected time, and they use these routing tables to forward packets.

When its main parameter, the contact radius R_C, is set to one, the EMEED protocol approximates the well known MEED [36] protocol that takes into account, when constructing the routing table, only direct contacts between nodes. For values of the contact radius larger than one, EMEED also takes into account *indirect* contacts through multihop paths.

Using simulation, we show that for many important mobility scenarios, EMEED with $R_C > 1$ performs much better than MEED, in terms of throughput and delay, with only a modest increase in the control overhead. Furthermore, we compare, using simulations, EMEED with BUBBLE [37], a state of the art protocol that was proposed for a scenario where the nodes have preferred locations of movement, and find that EMEED has an overall comparable packet delivery rate but uses the available bandwidth much more judiciously than BUBBLE. We also compare EMEED with flooding, which is used as an upper bound, and with Spray and Wait, which is a baseline protocol. The comparison is made in two mobility scenarios, one related to PSNs and human levels of mobility, and one related to vehicular DTNs, using the SUMO mobility simulation tool [38].

The chapter is based on the following publications [39, 40]:

- A. Sidera and S. Toumpis, "Routing using partition-wide information in wireless delay tolerant networks," in Proc. IFIP MedHocNet 2013, Ajaccio, Corsica, France, June 2013.
- A. Sidera and S. Toumpis, "Wireless mobile DTN routing with the extended minimum estimated expected delay protocol," submitted to journal publication, 2015.

In Chapter 7, we provide some concluding remarks and possible future research directions.

1.4 Our Contributions

The contributions of our work can be summarized as follows:

- We propose the DTFR protocol which is a geographic routing protocol for DTNs; we compare it using simulation to other state of the art protocols. A key component of DTFR is Greedy Lazy Forwarding (GLF).

- We develop a simulation tool that is specifically designed for DTNs and is capable of simulating networks of 10^4 nodes while using realistic MAC and physical layers.

- Simulations reveal that DTFR performs much better than the protocols we compare it with. We arrive at the conclusion that, in many scenarios, under GLF, the packet travels toward its target location with larger velocity than the velocity with which the nodes move. Thus, DTFR can be used for applications with smaller acceptable delay for the delivery of the packets, compared to the protocols that rely only on transport to deliver the packets.

- We perform analysis for the version of GLF used in DTFR. The analysis is very general and easily extendable to other scenarios.

- We study different variants of GLF that take into account the velocity of the current holder and the candidate holders. Our work is focused on a fundamental tradeoff that exists between the cost and the delay in all wireless mobile DTNs.

- We propose the EMEED protocol which is a protocol for DTNs in which certain pairs of nodes meet more often than others, mobility patterns are correlated, and many nodes appear in the network only briefly. We compare EMEED, using simulation, to other protocols.

- Simulations show that EMEED performs much better than the protocols we compare it with, in terms of packet delivery ratio and bandwidth usage. We also arrive at conclusions of more general interest. In particular, nodes that appear in the network for limited amounts of time can have a large impact in its performance and should be taken into account; the correlation of the mobility patterns of different nodes can significantly affect the performance of DTN routing protocols; and the distribution of global and community centralities can influence significantly the performance of protocols using them.

Chapter 2

Related Work

In this chapter, we review recent work related to our own on the topic of routing for wireless mobile delay tolerant networks. The works we review fall into the following broad categories:

- Hybrid delay tolerant geographic routing.
- Epidemic routing and related protocols.
- Routing using the history of node encounters.
- Routing using communities.
- Other routing protocols.
- Mobility models and networks with non-uniform mobility

2.1 Hybrid Delay Tolerant Geographic Routing

In this section, we review related work on the topic of hybrid routing protocols that combine geographic routing and delay tolerant routing. Such works are particularly relevant to the first of the two protocols we propose, DTFR.

Geographic routing, also known as position based routing and location aware routing, is a well established method for routing in wireless networks that has excellent scalability and robustness properties. The basic idea is very simple: if a node A has a packet destined for a node D, then A sends the packet to a node in its neighborhood that is closer to node D. In this manner, no routing tables need to be established. The only requirement is that node A has a way of finding the location of node B. If node

B is moving, this is a challenge. Geographic routing has been around for a lot of time. For example, it was used in [41, 42, 43].

There are many different geographic protocols that differ on what decisions they take on various issues, in particular:

- How node *A* selects the neighbor *B* that receives the packet, if multiple such neighbors exist.
- What actions node *A* takes if node *A* has no neighbors that appear to be in a better position than him.
- How the location of the destination is established.

A good overview of early works in geographic routing appears in [44].

GPSR (Greedy Perimeter Stateless Routing) [45] uses a combination of greedy forwarding on the full network graph and perimeter forwarding on a planarized network graph, i.e., a subgraph of the original graph with no crossing links. Initially, the packet is forwarded on the full network graph using the greedy mode; if, at some point, there is no neighbor closer to the destination than the node holding the packet, the packet enters the perimeter mode, traversing the faces of the planarized network graph using the right hand rule [45]. If the packet, while in perimeter mode, reaches a node closer to the destination than the point at which the packet entered the perimeter mode, the packet switches back to the greedy mode.

In [29] the authors propose *GPCR* (Greedy Perimeter Coordinator Routing), a geographic routing protocol designed for use in vehicular networks. GPCR is based on the observation that the road network creates a naturally planar graph that can be exploited for communication purposes. Both greedy routing and perimeter routing are executed using that graph. However, GPCR suffers from the problem that when there is no node at a junction, packets will be forwarded across that junction, and this might lead to a routing loop.

To alleviate this problem, the *GeoCross* protocol is introduced in [30]. GeoCross is similar in its operation to GPCR, but its perimeter mode is enhanced and capable of detecting and removing crossing edges on the road network graph (caused by the lack of nodes at junctions) and creating a planar graph.

GeoDTN+Nav [4] consists of the greedy and perimeter modes of GeoCross and a third mode, termed the DTN mode, which can deliver packets even in the absence of end-to-end routes. In GeoDTN+Nav, packets are first forwarded using the greedy mode and, when this fails, using the perimeter mode. If that mode also fails, the protocol switches to the DTN mode and uses mobility to deliver packets. To decide when to switch to the DTN mode, a node uses a cost function related to network partition detection and to the navigation information of its neighbors. When a packet is in the DTN mode, it returns to the greedy mode whenever it encounters a node that is closer to the expected location of the destination than the point where the perimeter mode started.

DTFR and GeoDTN+Nav are related, as they both employ a geographic routing mode and a DTN mode. However, they have a number of key differences. Firstly, GeoDTN+Nav makes use of a perimeter mode, which DTFR avoids, in order to conserve bandwidth, and in order to avoid the routing loops associated with running a perimeter mode in a network of highly mobile nodes. Secondly, GeoDTN+Nav was designed without taking into account links between nodes that are not on the same road and so makes no use of potentially useful links between nodes lying on different roads. Thirdly, in GeoDTN+Nav, the packet only travels to the expected destination position inserted in the packet by the source, whereas, in DTFR, if the destination is not found when the packet reaches its expected position, replicas are employed to find it. Also, the rules for entering the greedy mode from other modes are different. Finally, DTFR uses a set of priority rules for gaining access to the medium. In Chapter 3 we show using simulation that all these differences lead to significant deviations in the performance of the two protocols.

The *MDDV* (Mobility-centric approach for Data Dissemination in Vehicular networks) protocol [46] is based on two phases. During the *Forwarding Phase*, the message travels to the destination region, and then, in the *Propagating Phase*, it is distributed to all nodes there. In the Forwarding Phase, a group of nodes are forwarding the message along a trajectory consisting of road segments chosen by the protocol. The group consists of the nodes that estimate that they are near the *message head* which is the node closest to the destination region along the trajectory. The members of the group change as the message propagates or the vehicles move. Nodes estimate the position of the message head based on information that is inserted in the copies of the packet, by nodes that estimate they might be the message head. In DTFR, the packet also travels to the location of the destination, during a *Homing Phase*, but using GLF and high priority transmissions. In addition, during the Homing Phase, there is only a single copy of the packet at any time, and if the node that has that copy moves away from the Firework Center, it still has to forward the copy. Finally, our use of replicas is more efficient than the Propagating Phase of MDDV, which distributes the packet to all nodes in the destination region.

More recently, in [6], *LAROD* (Location Aided ROuting for Delay tolerant networks) has been proposed. Like DTFR, LAROD is a delay tolerant geographic routing protocol. In particular, each node carrying a packet (termed a *custodian*) periodically broadcasts it to its neighborhood. Nodes closer to the destination that overhear the transmission set up a timer that depends on their location. After its timer expires, a node broadcasts a reply informing its own neighborhood that it is the new custodian. If the original custodian, or a node that received the packet and waits for its timer to expire, listens to a reply from a node in its neighborhood announcing that it is the new custodian, it discards the packet. This algorithm is related but different from our GLF algorithm, notably allowing the packet to be propagated along multiple paths. Another fundamental difference between LAROD and DTFR is that DTFR creates multiple replicas once the Firework Center is reached. This makes DTFR more robust to destination localization errors.

In [47] the *VADD* (Vehicle Assisted Data Delivery) protocol is proposed. In this work the source is a vehicle and the destination is a fixed point. The goal is to select the forwarding path with the smallest delivery delay. It may be faster for the packet to travel a larger distance by transmissions than to travel a shorter distance carried by a vehicle. The authors assume that vehicles have digital street maps and statistics of traffic density and vehicle speed at different times of the day. They place a boundary including the source and the destination to have a finite number of roads in the system. A large boundary can generally find better paths but induces more computation cost.

In more detail, let D_{ij} be the expected delay to deliver the packet when it is at intersection i and it will travel in road ij. For example consider an intersection i where we have roads ia, ib, ic and id, with $D_{ia} < D_{ib} < D_{ic} < D_{id}$, and the vehicle that carries the packet will travel to road ic. The packet is sent to ia if there is a next hop available in the road, otherwise the packet is sent to ib if there is a next hop available there, otherwise the packet is carried by the vehicle. We have $D_{ij} = d_{ij} + \sum_m P_{jm} D_{jm}$. In this equation d_{ij} is the expected delay to traverse road ij. Therefore, d_{ij} is proportional to the length of the road if the average density of the vehicles is large enough so that the packet is expected to be carried by transmission, otherwise d_{ij} is equal to the mean time needed for vehicles to carry the packet (the average speed of the vehicles is considered to be known), minus a term that increases with the density of the vehicles (because a part of the road may be covered by transmission). P_{jm} is the probability that at intersection j the packet selects road jm. Assuming that the mean time the packet waits at an intersection is known, the authors find the probability that a packet at an intersection i finds an available next hop to go to road ia. Taking the above into account, the authors solve the specified system of linear equations and compute the values of D_{ij}.

2.2 Epidemic Routing and Related Protocols

The Epidemic Routing protocol was first proposed in [48]. It works as follows. When two nodes come within transmission range of each other, they initiate a session with each other. During the session, the two nodes exchange their *summary vectors* to learn which messages each node has. Then, each node requests copies of messages that it does not have. Then, each node sends to the other node the messages it requested. To avoid redundant sessions, each node keeps a list of nodes that it has initiated a session with recently. A session is not initiated with nodes with which a session has been initiated within a configurable time period. Epidemic routing is also called flooding.

There are many protocols that do constrained epidemic routing, where, when a node A that has a copy of a packet comes within transmission range of another node B that does not have a copy of the packet, A gives a copy of the packet to B only if some conditions are satisfied. This approach is also called limited flooding.

In [3] the authors deal with the problem of collecting data from zebras that move in a large area using a network called ZebraNet. The aim is to collect samples of the positions of the zebras and related data. Transceivers equipped with memory and a CPU are attached on some zebras. These form the nodes of the network. The scientists periodically fly or drive through the area with a node that collects the data.

The mobility model used considers some known facts about the movement of the zebras. For example, zebras execute either a grazing, a graze-walking or a fast movement. The data of each zebra is transmitted to other zebras when they meet and stored in their memory. The zebras which come near the scientists transmit the data to them. To save power, two transmission ranges are used: A short transmission range for zebra to zebra transmissions and a larger transmission range for zebra to scientists transmissions.

Two protocols are investigated for transmitting data from one zebra to another: A flooding-based protocol and a history based protocol. In the flooding-based protocol, when two zebras meet, they always exchange data. In the history based protocol, data are transmitted to zebras that, in the past, had greater success in delivering data to the scientists. In both cases, when their memory is full the nodes delete the oldest packets they have from other nodes. This is done because if a packet is old there is large probability that it was already delivered to the scientists by another zebra.

The work in [2] deals with the problem of collecting data from whales. The work proposes the Shared Wireless Infostation Model (SWIM) which integrates the Infostation concept [49] with the ad hoc network technology. In the Infostation model, a user needs to be within the transmission range of an Infostation to communicate. In SWIM, infostations which offer intermittent coverage are used. A packet is given from one node to other nodes and when one of them goes near an Infostation the packet is delivered. This reduces the delay but more transmissions are needed for each packet so the capacity is reduced.

One way to collect data from whales is to offload the data from the whales to satellites if the whale is surfacing when the satellite is passing overhead. However, as the considered application is not time critical, the SWIM model is cheaper than the high cost and low data rate satellite solution: packets are given from the whales to SWIM stations through close range, high bandwidth links. The authors do not consider the problem of conveying the data from the SWIM stations to the terrestrial network. A Time To Live (TTL) is used to delete the packets from the whales. This TTL has to be such that a whale delivers the packet to the Infostation with high probability before the TTL expires.

The protocol is studied through simulation. An empirical movement model is used that takes into account the following information about the whales: they tend to do migration movement, they move toward feeding areas and within feeding areas, they tend to move toward other whales, and finally they tend to move together as groups.

Spray and Wait [15] was one of the first delay tolerant routing protocols to be proposed. In the *Spray Phase*, the source distributes L copies to L distinct relays. In the *Wait Phase*, the relays move around the network, until eventually one of them meets the destination and hands over its replica of the packet. DTFR also employs replicas, however, in DTFR, the replicas are created not at the location of the source, but at a location estimated to be close to the destination, in order to conserve bandwidth and buffer space. In addition, in DTFR nodes make use of geographic information.

In [50] the authors study the problem of deriving the optimal spraying policy for spraying based schemes. It is assumed that the buffer size of all nodes is infinite, and simultaneous transmissions do not compete with each other. A random walk mobility model is used. Under this model the expected meeting time $E[M]$ for two nodes starting from the stationary distribution is known: it is a function of the network area and the transmission range. The expected meeting time $E[M(d)]$ for two nodes starting at distance d is also calculated.

In this setting, assume that a node A has l copies of a packet and another node B has none. The distance of the two nodes from the destination is known. The authors propose an algorithm, that finds how many copies A should give to B so that the expected delivery delay is minimized. $E[D(d,l)]$ is the expected time a node needs to deliver a packet to the destination when it is at distance d from the destination and has l copies of the packet. E_1 is the event that the node meets the destination and E_2 is the event that the node meets a relay. $E[D(d,l)]$ can be expressed in terms of the expected time needed for one of the two events to happen and the expected value of the additional time needed to deliver the packet if the event that happened was E_2. The latter depends on the probability that node A and the relay B are at distances dA and dB from the destination when they meet and the minimum for all $0 \leq i < l$ of $E[\min(D(dA, l-i), D(dB, i))]$. Some approximations are done to simplify the equation. Then it is solved using dynamic programming. The value of $E[(d,l)]$ for $l = 1$ is assumed to be known for each spraying protocol.

The algorithm is implemented for Spray and Wait and for Spray and Focus [51]. For both cases the number of copies given to B is plotted as a function of l for different values of $dA - dB$. For all these cases for Spray and Focus nearly half of the copies are given to node B but this does not hold for Spray and Wait. For Spray and Focus the algorithm performs about the same as binary spraying. For Spray and Wait this algorithm performs better than binary spraying.

In [52] the authors find analytically the delay of epidemic routing under contention. The contention model consists of the following:

- Finite bandwidth: Two nodes i and j that meet have many packets to exchange but they have to select only one packet to exchange due to finite bandwidth.

- Scheduling: i and j will be scheduled to exchange a packet according to some probability. If i is transmitting to j all nodes within distance $2K$ from i must not transmit, where K is the transmission range.

- Interference: If i transmits a packet to j there is a probability less than unity that the transmission is successful.

Using the above constraints caused by contention the paper finds $p_{txS}(k)$, which is defined as the probability that nodes i and j, when at distance $k \leq K$, exchange successfully a specific packet A that they need to exchange. $E[d(m)]$ is the mean duration of the m^{th} time epoch, defined as the period of time during which we have m copies of the packet in the network. $p_{txS}(k)$ depends on $E[d(m)]$. When nodes i and j come within transmission range, they stay within transmission range for some time. The probability that packet A is exchanged successfully during that time is $p_{success}$ and depends on $p_{txS}(k)$. $E[d(m)]$ depends on $p_{success}$. To find $p_{txS}(k)$ an approximate value of $E[d(m)]$ is chosen. From this $p_{txS}(k)$ is found and from this $p_{success}$ is found and from this a new value for $E[d(m)]$ is found. The same procedure continues using the new value of $E[d(m)]$ until the value converges. The mean delay is found using $E[d(m)]$ and the fact that it is equally likely for the packet to reach the destination at the end of any time epoch. Analytical results are compared with simulation results.

In our simulation for DTFR we use a model for contention similar to the one used in [52].

The work in [53] presents data collected in a rollerblading tour. Contacts are recorded when two persons get close to each other. This is done with devices given to some of the people in the tour. In vehicular networks, if a car slows down, due to the reaction time of the other drivers, a traffic jam may be caused among the following cars. Similarly, in the rollerblading tour, participants have a delayed reaction to the movement of others, and the leaders have to adapt their speed according to road conditions. Thus we have alternating phases of compression and expansion of the crowd.

Taking these observations into consideration, the authors propose the Density Aware Spray and Wait protocol (DA-SW), which is similar to Spray and Wait but the number of copies n is large when the node density is small and small when the node density is large. DA-SW relies on the node degree, which, in this work, is the number of neighbors the node has had within the last 30 seconds. DA-SW uses curves, derived from numerical analysis, of the average delay as a function of the node degree when the packets were generated, for different values of n.

2.3 Routing Using the History of Node Encounters

In this section we review protocols that perform routing based on the collection and distribution of the statistics of encounters between nodes. The basic idea of all these protocols is that two nodes that have met frequently in the past will likely meet again soon, therefore their meetings can be utilized for routing a packet to its destination. The second protocol we propose, EMEED, falls into this class of protocols.

The Minimum Expected Delay (MED) protocol is perhaps the first in this class of protocols and was proposed in [1]. This protocol performs shortest path routing by assuming that there is a link between each pair of nodes in the network with a cost equal to the expected time the two nodes need to wait until they come in direct contact with each other.

The Minimum Estimated Expected Delay (MEED) protocol is proposed in [36]. It is similar to MED, but, when it creates the routing table, it sets to 0 the costs of the links from the current holder of the packet to the nodes that are currently in direct contact with the current holder. Our EMEED protocol extends the MEED protocol by taking into account *multihop* paths in deciding if two nodes are in contact.

Apart from the expected wait time, other metrics have been used to predict future contacts based on past contacts. In [54, 55] the delivery probability, defined as a metric that is increased whenever the two nodes come into contact, is used. This definition gives larger delivery probability for pairs of nodes that come into and out of contact often with respect to pairs of nodes that are continuously in contact [56]. In [57, 58] the delivery probability, defined as the percentage of time that two nodes are in contact, is used. This definition does not take into account whether the two nodes are in contact for many short or a few large intervals.

In [54] the probabilistic routing (PROPHET) protocol is proposed. The protocol uses the delivery predictability $P(A, B)$ which is a value between 0 and 1 that each node A maintains for each destination B. A high value of $P(A, B)$ indicates that A is a good candidate to deliver a packet to B. The more often A meets B, the larger $P(A, B)$ is. $P(A, B)$ is increased whenever A encounters B. As time passes and A does not encounter B, $P(A, B)$ is decreased. Also, if node A meets node C often and node

B also meets node C often, then A is a good candidate to deliver a packet to B via C. This is also taken into account in computing $P(A, B)$. When two nodes A and B meet and A has a packet for destination D and $P(A, D) < P(B, D)$, then A gives a copy of the packet to B. The protocol is compared to epidemic routing using simulation. For a community based mobility model, PROPHET has better performance than epidemic routing. For a random waypoint model, PROPHET has comparable performance to epidemic routing but lower communication overhead.

In [55] the MaxProp protocol is proposed. Each node i keeps an estimate of the probability of meeting any other node. Initially these probabilities are set to equal values. When i meets node j the probability of i meeting j is increased by 1 and then the values are normalized. The cost of a path to a destination d, is the sum of the probabilities that each connection of the path does not occur. The cost of d is the lowest path cost. Also, priorities are assigned to the packets. Packets with higher priority are transmitted first at transmission opportunities. Packets with lower priority are deleted first when there is no space in the buffer. The protocol is evaluated based on data that are collected from a network of buses that serves a large area between five colleges. The data are traces of when transfer opportunities occur and for how much bandwidth. An artificial DTN is also used in which each peer has a set of peers that it meets with often. It meets with the remaining nodes very rarely or not at all.

In [58] the MV protocol is proposed. The MV protocol learns statistics in the movement of network participants and uses them for routing. There are also agents that adapt their movement to help with routing. The protocol learns the frequency of meetings between nodes and how often they visit specific geographic cells. The authors assume finite buffer size but infinite bandwidth. They assume that the destination nodes are at fixed positions. When a node A meets another node B, and A has messages for which B has larger likelihood of delivery, A gives these messages to B and then discards them. It is assumed that the probability of visiting a region in the future is strongly correlated with the history of visiting the region.

In [59] the PRiortised EPidemic (PREP) protocol is proposed. Each link is associated with an Average Availability (AA). The AA is an estimate of the fraction of time the link will be available for use. It is based on information from the recent past. If a link is down for more than T_g seconds, it is forgotten. When it comes up again it is considered a new link and the AA jumps to nearly one. A cost $(1 - AA) + 0.01$ is assigned to each link and Dijkstra's algorithm is used to find the shortest paths.

When buffer space is needed, packets that have higher cost to reach their destination, are deleted first. Regarding the priorities with which packets are transmitted, the following holds. Packets for which the receiver has lower cost to the destination than the transmitter are in the high priority group and the others are in the low priority group. For packets in the same group, packets that will expire earlier have higher priority and in the case that the expiry time is the same older packets have higher priority.

In [56] the authors assume that the expected delay for two nodes to meet depends on the previous hop of the packet being routed and, based on this observation, define the *expected dependent delay*. They use the expected time a node B needs to meet another node C given that B just met the previous hop A. Other works, for example [60], assume that the delays are independent exponential random variables.

In [61] a graph is used where a vertex, denoted by ij, represents a contact between two nodes and an edge, denoted by $ij \to jk$, represents the delay between the beginnings of the two encounters. Each edge is associated with an average delay denoted by $d(ij \to jk)$ and a delay variance denoted by $s^2(ij \to jk)$. The path to a destination w has a delay and variance denoted by $d(ij \mapsto w)$ and $s^2(ij \mapsto w)$. When a node j meets another node i, it finds for all nodes k, $delay_k = d(ij \to jk) + d(jk \mapsto w)$ and $var_k = s^2(ij \to jk) + s^2(jk \mapsto w)$ and $cost_k = delay_k + 1.65\sqrt{var_k}$. This definition of the cost is chosen so that if the delays follow the normal distribution the probability that the delay is smaller than the cost is 95%. Then j finds which node $k*$ has the smallest $cost_k$, it sends to i the value of $delay_k$ and var_k for that node and i sets $d(ij \mapsto w) = delay_{k*}$ and $s^2(ij \mapsto w) = var_{k*}$.

When a node i that has a packet comes in contact with another node j, it computes for every node k, including j, the delivery probability p_k. p_k is the probability that if the packet is given to k it is delivered before its Time To Live (TTL) expires, where the time elapsed since the packet was created is taken into account. p_k is found using the CDF of the normal distribution, and assuming that the delays of the links follow normal distribution with mean $d(ij \to ik) + d(ik \mapsto w)$ and variance $s^2(ij \to ik) + s^2(ik \mapsto w)$. i puts all nodes in a list, ordered by decreasing delivery probability. Then, i goes through the list, and assigns to each node k in the list $p_k L$ copies of the packet, until all copies have been assigned or all nodes have been considered. Then, if there are copies that have not been assigned, i assigns to j half of these copies. Then i transmits to j the copies assigned to it.

In [62] the authors use the conditional intermeeting time $t_A(B \mid M)$ which is the average time it takes for node A to meet node B after meeting node M. If $B = M$, this is equal to the standard intermeeting time between A and B. They propose a protocol similar to MEED but instead of using the standard intermeeting times, they use the conditional ones.

In EMEED we use unconditional expected wait times because it is simpler to handle them and we expect that the choice of conditional or unconditional expected wait time does not change the performance of the protocol significantly in our setting. We disseminate the expected wait times using a flooding-based protocol instead of using the method proposed in [61] to disseminate information to create the routing tables, because that method is more complicated and while it requires less overhead to create the routing tables initially, when a link breaks or a new link comes up or the cost of a link changes significantly, our protocol will disseminate in the network only the information about that link, while in that method we expect that more control overhead will be needed to update the routing tables.

The Prediction Assisted Single-copy Routing (PASR) protocol [63] uses historical information about the network connectivity. If the mobility shows contact periodicity, the protocol routes packets using the average contact duration and average intercontact duration. If the inter-contact time follows a known distribution, the protocol routes packets using the last contact time to predict after how much time a node will come in contact with another node.

In [64] the authors introduce a multicopy routing protocol for DTNs called Self Adaptive Utility based Routing Protocol (SAURP). The protocol measures the intercontact time between any node pair A and B. When a node A that has more than one copies of a packet for a destination D meets another node B, it gives to it $N_B = N_A WT(A, D)/(WT(A, D) + WT(B, D))$ copies, where N_A is the number of copies A has, and $WT(A, D)$ and $WT(B, D)$ are calculated using the intercontact times of the nodes. If a node has only one copy of the packet, it forwards it to a node it meets according to some rules based on the intercontact times of the nodes. In the calculation of the intercontact times the protocol takes into account what portion of the time the wireless channel is free and the buffer is not full and if the contacts last for enough time to transfer at least one packet.

In [65] the Density Adaptive With Node deadline awareness (DAWN) protocol is proposed. The authors consider a data collection network that consists of M mobile nodes and one or more data gathering base stations. The mobile nodes generate data packets that must be delivered to a base station before a deadline. At each slot t, a node l can transmit $K/\lambda_l(t)$ packets where K is a constant that depends on the available bandwidth and $\lambda_l(t)$ is the number of mobile nodes within the transmission range of node l at time t. The utility of a packet at time t, is the probability that after time t, if no other replication of the packet happens, at least one of the copies of the packet will reach the destination before the deadline. Packets with higher utility gain are transmitted with higher priority. The protocol introduces a distributed way to estimate the utility of each packet.

In [66] a metric called ExMin that is computed using the Expectation of the Minimum delays over all possible routes is proposed. The authors use expected wait times to decide the next hop relay node but they take into account all possible routes. For example if the intermeeting times of the nodes are exponentially distributed and node A meets nodes B and C with expected wait time equal to 2 and nodes B and C meet node D with expected wait time equal to 2, then if the packet is given to A, the expected time for the packet going from A to D is equal to 3, because on average it will take time equal to 1 for A to meet either B or C and then on average it will take time equal to 2 for B or C to meet D. It is assumed that the nodes can learn the expectation and the distribution of the wait times.

2.4 Routing Using Communities

In this section we review a very large class of wireless mobile DTN routing protocols that make routing decisions based on the communities that nodes belong. A community is a set of nodes that a node meets often, and/or a region of space in which a node is frequently found. Clearly, depending on its mobility patterns, a node might belong to one or more communities. The most prominent example of these routing protocol is BUBBLE [37]. Since its publication, many other protocols have been developed using the same basic idea.

Under BUBBLE [37], each node first finds its community, its global popularity, and its popularity within its community. When a node A that has a packet for a destination D meets another node B, A gives a copy of the packet to B in the following cases: 1) both A and B are not in the community of D and B has larger global popularity than A has, 2) A is not in the community of D but B is in the community of D, 3) both A and B are in the community of D and B has larger popularity within the community of D than A has. The authors claim that BUBBLE creates less control overhead than MEED because nodes do not use routing tables, however, under the protocol multiple copies are created for each packet, and no comparison to MEED is offered via simulation or analysis.

In [67] a protocol for multicasting is proposed. The *destination cloud* for a certain destination is defined as the set of nodes that come frequently in contact with that destination. The protocol assumes that if the packet is given to one of these nodes it will be delivered to the destination by that node. A node is in the destination cloud if it comes in contact with the destination with a frequency larger than a *contact frequency threshold*. The protocol consists of a *pre-cloud phase* and an *inside-cloud phase*. In the pre-cloud phase, the current holder holder forwards the multicast message to nodes it encounters. In the inside-cloud phase, the current holder waits until it comes in contact with the destination.

In more detail, routing is based on *forwarding metrics* as follows: Let F_i be the forwarding metric of node i. Let $e(i)$ be the number of destination clouds to which node i belongs. Let $Z(i)$ be the number of message copies node i has. When node i comes in contact with node j the following is done. If j is the destination, i gives the packet to j. If j is not the destination but it belongs to one or more destination clouds i gives to j $\min(Z(i), e(j))$ copies. If j is not in any destination cloud and $F_j > F_i$, i gives to j $Z(i)F_j/(F_j + F_i)$ copies. Otherwise i does not give copies to j. The protocol records in the packet the latency for all nodes in the routing path. When the packet reaches the destination the destination gives this information to other nodes and the information is spread out through the whole network. This information is used to update the forwarding metrics. Initially F_i is set to the sum of the contact frequencies of i with every other node. The authors do analysis for the latency of the protocol.

In [35] the profile based routing for pocket switched networks (PRO) protocol is proposed. The *observation score* of a node depends on the probability that the node will come in contact with the destination node. The *information dissemination score* of a node depends on how good a candidate is the node for forwarding the packet to other nodes. If a node A that has a packet comes in contact with other nodes the following is done: If A is in contact with the destination of the packet, A forwards the packet to the destination. Otherwise, if A is in contact with a node that has higher observation score for the destination of the packet than A has, A forwards the packet to that node. Otherwise, if A is in contact with a node with information dissemination score greater than the *internal threshold* stored in A, A forwards the packet to that node. Each node maintains a list of the information dissemination scores of the nodes it came in contact with. When the packet is at hop k the current holder sets the internal threshold equal to the average value of the larger $1/k$ dissemination scores in its list. The *forwarding quota* is the maximum number of copies of the packet that a node can forward. The authors use a forwarding quota equal to 2.

In [68] the network is partitioned into *zones*. The *home* of a node is the set of zones it visits often. A node is an *activist* if the number of different zones it visits per unit time is greater than a threshold value. For each packet we have a certain number of copies. If a node that has copies of the packet comes in contact with a node that qualifies as a relay it gives to it half of its copies. The proposed protocol has two variants. In the first variant a node qualifies as a relay if the distance between its home and the home of the destination is smaller than a threshold value. In the second variant a node qualifies as a relay if the distance between its home and the home of the destination is smaller than a threshold value or it is an activist.

In [69] the Group Aware cooperative Routing protocol for opportunistic networks (GAR) is proposed. Within a group the nodes encounter each other frequently and their connections are stable compared with those among the nodes from different groups. If a message is delivered into the destination group, it is approximately considered successfully delivered to its destination. Each message initially has a fixed number of replicas. Each node records its meeting intervals with other nodes. Thus, when a node i meets another node j, j knows the probability that it meets one of the nodes in the destination group before the packet TTL expires. Based on this, i decides how many copies of the packet it should give to j. The protocol has rules that specify which transmissions should be given priority when two nodes meet. When a node overhears a packet, it stores it but it transmits it only if it meets the destination.

In [70] the authors propose a metric for measuring friendship and finding direct and indirect close friends for each node from the encounter history of nodes. They use a metric that is the same as the expected wait time in MEED for one hop paths. For two hop paths they use a different metric than MEED, that takes into account the contact history of the three nodes involved in the two hops. They assume that two friends are at most two hops away. Furthermore, the friends of a node are not the same for the whole duration of the day, but they depend on the time of the day. If a node i meets a node j, then i forwards the message to j if and only if the destination of the message is a friend of j and j is a stronger friend of the destination than i is. The authors separate the day in periods and if the current period is near its end, the forwarding decisions are made based on the friendships for the next period of the day.

The protocols in [67, 35, 68, 69, 70] described above are all similar to BUBBLE. Like BUBBLE, they all have the goal to give the packet to nodes that are in the community of the destination and they assume that when the packet reaches these nodes it will be delivered to the destination. Like BUBBLE, most of them [67, 35, 68, 69] use multiple copies of the packet. To make the packet reach nodes that are in the community of the destination, most of them [67, 35, 68] give the packet to nodes with high popularity, like BUBBLE does, while others [69, 70] assume that a node i that has the packet will meet nodes j that often meet nodes k that are in the community of the destination. The existence of these protocols in the literature shows that it is important to compare EMEED with BUBBLE.

2.5 Other Routing Protocols

In this section we discuss a few notable related works that do not fall in the scope of any of the previous sections.

In MobEyes [23] we have a network of private vehicles called *regular nodes* and police agents called *authority nodes*. The regular nodes collect data and deliver them to the authority nodes. Packets are diffused into the network. In particular, MobEyes executes either single hop diffusion (only the source can transmit the packet) or k-hop diffusion (the packet travels at most k hops from the source). An authority node asks his neighbors which packets they have. Then it broadcasts a message requesting the packets he does not have. A regular node sends missing packets to the authority node. The authority node sends an acknowledgment with a list of received packets. The other regular nodes use this message to know which packets the authority node still needs. Then another regular node sends packets to the authority node. This continues until there is no remaining packet.

The Vehicular Cyber Physical System (VCPS), introduced in [71], provides services using the sensing, computing and communication capabilities of vehicles. In [71] a probabilistic framework for studying the performance of epidemic routing using network coding (ERNC) in VCPSs is presented.

In [72] the authors focus on downlink communications from sporadically deployed

BSs/APs to mobile nodes in a setting where nodes form *coalitions*. $r_{ij} = r_{ji}$ is the rate at which nodes i and j meet. The encounter process for each pair of nodes is a Poisson process whose parameter is the encounter rate. Each mobile node i is willing to carry and forward packets for other mobile nodes with probability p_i. The cost for a node i to receive or forward a packet is c_{ij}^r and c_{ij}^f respectively, where j is a BS or a mobile node in the same coalition as i. d_i is the packet delivery delay. Each mobile node calculates its encounter rate with other nodes and sends it to a central coordinator. Given a coalitional structure the central coordinator finds the expected cost and delay for each mobile node. If the solution is unstable the process is repeated.

In [73] the authors perform performance evaluation of DTN protocols with asymmetric channel rates for space communications using a testbed that is comprised of the simulator, a source PC, a relay PC, and a destination PC.

In [74] the authors propose VideoFountain, a service that deploys kiosks at popular venues to store and transmit videos to users. Mobile users distribute the videos received from their sources to the requesting venues. Given a contact between a venue and a user, the protocol first identifies all potential downloads and uploads. It then computes the marginal utility of each potential replication, which is the improvement in the utility function if we complete this replication. Once it has the marginal utility value of all potential uploads and downloads, it sorts them based on the marginal utility divided by the flow size and starts replicating from the ones with the highest marginal per packet utility.

The work in [75] proposes a parking scheme based on vehicular communications. The parking scheme is evaluated using simulation. Many cars are equipped with an On Board Unit (OBU) which allows them to communicate with other cars and Roadside Units (RSUs). In the proposed parking scheme, the parking lot has a number of RSUs, for example three RSUs, each of them covering the whole parking lot. A Trusted Authority (TA) generates a private key for each RSU or OBU that registers to the TA.

The RSUs keep information about the occupancy of each parking space in the lot. When an OBU enters the lot, it receives a ticket key which is known only to the driver. The RSUs choose a vacant parking space and navigate the vehicle to the parking space using encrypted information. The blocking probability of the parking lot is disseminated to the vehicles on the road. Using this information they can decide to look elsewhere for parking.

When the car is parked, the OBU is in sleep mode and sends beacons periodically to the RSUs. When the driver comes, he enters his password and sets the OBU to active mode. The RSUs detect vehicle theft when the position of the vehicle changes without setting the OBU to active mode or if the ticket key is not used. When the vehicle gets stolen, all encountered RSUs and OBUs detect the vehicle.

The work in [76] presents a protocol for Internet access from vehicles that have intermittent access to WiFi access points (APs). The APs can be gateways in apartments, located at bus stops, etc. The authors consider a network where the probability of forming a multi-hop path is very small. The model assumes that vehicles move on a two way street without changing direction and that there is no overtaking. The protocol is simulated using traces of a network of buses in a campus.

When a mobile client wants to initiate a download, instead of waiting to come in contact with the AP, it informs the AP to pre-fetch and cache the data. The WWAN network is used as a control channel.

There are various manners for transmitting data from the AP to a client. The client can get the data directly from the AP (1 hop). A direct relay is a node that moves toward the client and meets the AP before it meets the client (and after the request is generated). A direct relay can get the data from the AP and give it to the client (2 hops). A forerunner is a node that moves in the same direction as the client and meets the AP before the client does (and after the request is generated). An indirect relay is a node that moves toward the client and meets the client before it meets the AP. An indirect relay can get the data from a forerunner and give it to the client (3 hops). Also there are other cases, for example, AP → direct relay → forerunner → indirect relay → client.

When the AP has data it sends it to the forerunners and the direct relays. When two vehicles meet they want to exchange all the data they have that were not yet delivered to the client. Because the volume of data that can be exchanged during a contact is limited, some priority rules are used to decide which data to exchange first. When a direct or indirect relay meets the client, data are transferred from the relay to the client. The client informs, through the WWAN, the AP and forerunners about new blocks it received.

2.6 Mobility Models and Networks with Non-Uniform Mobility

Mobility models are central to our work, and for this reason, in this section, we review some important mobility models that are related to our work and the routing protocols on whose analysis these models have been applied.

The Community Mobility Model is proposed in [77]. In this mobility model each node has a community, i.e., a region where it prefers to move. The node moves within the community (local movement) or not (roaming movement) using the Random Direction Mobility Model or the Random Waypoint Mobility Model. Given that the node is in its community, it has a certain probability to stay there and given that it is not it has a certain probability of not returning. The Time Variant Community Mobility Model is proposed in [78]. It is similar to the Community Mobility Model, but the community of a node changes periodically with time. The main idea is that each node has some preferred locations and that these preferred locations change with time. In the simulations for EMEED we use a mobility model that uses the same concept of nodes having location preferences that are non-uniform in space and time, but is simpler.

Another mobility model related to the mobility model we used for EMEED is *SWIM* (Small World In Motion) [79, 80]. In this model, the authors assign to each node a *home* location. The network area is divided in square cells, where the side length of each cell is equal to the transmission range of the nodes. Each cell represents a destination where a node may choose to go. When the node chooses to go to a specific cell, it goes to a point chosen randomly and uniformly within the cell. A node A has larger probability to go to cells with larger weight, where the weight of a cell C is given by the formula $w(C) = \alpha dist(h_A, C) + (1-\alpha) seen(C)$, where α is a parameter between 0 and 1, $seen(C)$ is the percentage of nodes A saw in C the last time it was there, and the function $dist(h_A, C)$ decays with the distance between the home of node A and the center of cell C. To keep things simple, when a node chooses to go from its home to a cell, the speed of the node is chosen to be equal to the distance the node has to cover.

In [81] the authors present a mobility model that produces synthetic human walk traces. From the analysis of GPS traces of human walks it is found that the waypoints of humans can be modeled by fractal points. Based on this a fractal mobility model is designed.

In [82] the authors deal with the inter-meeting time of nodes in wireless mobile ad hoc networks. The authors use three data sets, the Dartmouth data, the iMode data and the MIT data. They find that the empirical distribution of the inter-meeting times is well approximated by the log-normal distribution.

In [83] the authors provide a classification of ad hoc networks. The classification is useful for choosing an appropriate routing strategy. The SPN group is for networks that are connected most of the time. The U-DTN group is for networks which provide space-time paths between all nodes. If, for example, there is no complete path from node A to node B at any time, but at some time node A is in contact with node C and later node C is in contact with node B, this means there exists a space-time path between A and B. The A-DTN group is for networks in which nodes may be moving locally in specific areas of the network and ferries may be needed to carry the messages from a source to a destination. SPN is a subset of U-DTN and U-DTN is a subset of A-DTN.

The work in [84] explains that when a path breaks data can be forwarded from the intermediate node before the breakage or from the source node. Partial paths are multi-hop paths that allow packet forwarding closer to the destination. DTN protocols typically ignore partial paths of more than one hop.

Four areas are identified for ad hoc networks:

- Area 1: End-to-end paths exist most of the time. If the initial path brakes intermediate forwarding can reroute the message though an alternate path.
- Area 2: End-to-end paths exist for some fraction of the time. Using intermediate forwarding, nodes can wait until a new partial path comes up.
- Area 3: The network is always partitioned. Each node can reach the other nodes in its partition using multi-hop paths.
- Area 4: Multi-hop paths are rare.

This paper deals with Area 2. A r.v. X is stochastically monotonic in r.v. A if $P_r[X \geq x \mid A = a_1] \geq P_r[X \geq x \mid A = a_0]$ if $a_1 > a_0$. The primary route is the route to the destination used. The primary route may break and an alternate route needs to be determined. The authors demonstrate that stochastic monotonicity of the alternate path length (hop count) in the primary path length is sufficient for intermediate forwarding to outperform source forwarding.

In [85] the nodes are classified as *message ferries* or *regular nodes*. Ferries are devices which carry messages among other nodes. Ferries move following known routes. For example buses could carry messages within a city or between a city and a village. An airplane could be used to transport data in a battlefield. The ferries could also be a set of robots moving in a disaster area.

In the Node-Initiated MF scheme the ferry moves according to a specific route which is known by the nodes, and nodes proactively move to meet with the ferry. On the other hand, in the Ferry-Initiated MF scheme the ferry takes proactive movement to meet with the nodes. The ferry uses a long range radio to notify the nodes about its position and the nodes use a long range radio to send service requests to the ferry. The nodes use a short range radio to send data to the ferry.

In [86] the author explains that challenged networks have high latency and low data rate and end-to-end disconnection may be more common than connection. Therefore, messages should be able to be forwarded to alternative next hops if better routes are discovered prior to message transmission. Special-purpose extremely low power radios could be used that are able to monitor communication and wake up a primary radio when data arrives. The paper presents a delay tolerant network separated in different regions that communicate through gateways. The name of a node consists of the name of the region in which it belongs and a name that is uniquely defined within the specified region.

In [1] the authors explain that in Delay Tolerant Networks dynamics may be known in advance. This paper shows that if we use information about the network topology the algorithms can perform better. In one scenario we have a city and a village and we want to connect the village to the city for internet use. There is a dial up connection between the city and the village which is available at certain hours each day. There is a satellite that can communicate with both the village and the city at certain times each day. Also there is a motorbike that travels between the city and the village at scheduled times each day. In the other scenario we have city buses that communicate with each other and form a delay tolerant network.

Knowledge oracles are constructed to avoid the complexities of how routing data is created and propagated. Even if there is an available link, the algorithm may prefer to wait for a lower delay link to become available. In the following algorithms each node has the following information about the data queued at each edge of the network graph. ED: No knowledge, EDLQ: Knowledge of queue length at edges starting from the node. EDAQ: Knowledge of queue length at all edges. The results show that ED performs worse than EDLQ and EDAQ. However we do not need to implement EDAQ because it performs about the same with EDLQ.

In [87] the authors explain that vehicular ad hoc networks are different in rural areas than in cities because in rural areas the networks are more disconnected. In the simulation a source node located at a cross point transmits some information to the information center. The authors propose the BBR protocol which is similar to epidemic routing but instead of giving the packet to all one hop neighbors, a packet is given to the neighbor/neighbors that shares the smallest number of common neighbors with the forwarding node, or the largest number of uncommon neighbors.

In [88] the authors deal with Bidirectionally Coupled simulators. In these simulators, two independent processes, the network simulator and the road traffic simulator, are running concurrently and exchange data. Data like position and speed of vehicles are used by both simulators. Data like radio state are used only by the network simulator and data like planned route are used only by the road traffic simulator.

The authors simulate an evenly spaced grid pattern of roads that have one lane per driving direction. A number of vehicles start at a common source and head to a common destination. A car breaks down on a single lane road, blocking the road for several minutes. Vehicles participate in a VANET and exchange congestion warnings. Based on these warnings they dynamically recalculate the best path to their destination. The simulations show that vehicles are able to reach their destination faster if peers inform them of congested roads.

2.7 Classification of Protocols

Table 2.1 provides a classification of the most significant protocols discussed here based on their basic features.

Protocol	Uses Geographic Routing	Uses History of Encounters	Uses Communities	Uses Packet Replication	Applications	Performs Mathematical Analysis	Uses Traces	Nodes Aware of Future Mobility	Constructs Connectivity Graph
DTFR [33, 5]	Yes	No	No	Yes	VANETs	Yes	No	No	No
Spray and Wait [15]	No	No	No	Yes	Not Mentioned	Yes	No	No	No
GeoDTN+Nav [4]	Yes	No	No	No	VANETs	Yes	No	Yes	No
GeoCross [30]	Yes	No	No	No	VANETs	Yes	No	No	No
EMEED [39, 40]	No	Yes	No	No	Pocket Networks	No	No	No	Yes
MEED [36]	No	Yes	No	No	Not Mentioned	No	Yes [89]	No	Yes
BUBBLE [37]	No	No	Yes	Yes	Pocket Networks	No	Yes [90, 91]	No	No
GPSR [45]	Yes	No	No	No	Not Mentioned	No	No	No	No
GPCR [29]	Yes	No	No	No	VANETs	No	No	No	No
MDDV [46]	Yes	No	No	Yes	VANETs	No	No	No	No
LAROD [6]	Yes	No	No	Yes	UAVs	No	No	No	No
PROPHET [54]	No	Yes	No	Yes	Not Mentioned	No	No	No	No
MaxProp [55]	No	Yes	No	Yes	VANETs	No	Yes [92]	No	Yes
MV [58]	No	Yes	No	Yes	Not Mentioned	Yes	No	No	No
Epidemic Routing [48]	No	No	No	Yes	Not Mentioned	No	No	No	No

Table 2.1: Classification of the most significant protocols discussed.

Chapter 3

The Delay Tolerant Firework Routing Protocol

In this chapter[1] we present the Delay Tolerant Firework Routing (DTFR) protocol and evaluate its performance.

3.1 Basic Network Assumptions

In this section we outline our fundamental assumptions on the networks for which we design DTFR. Although these assumptions are satisfied in a variety of settings, a good example are large vehicular DTNs [4, 93].

Location Awareness: We assume that each node is capable of knowing its own location, either directly (for example, through GPS) or indirectly (for example, using beacons). We assume that there is a mechanism available to the nodes that provides the location of their destination, possibly with error.

Nodes: We assume a very large number of nodes, on the orders of thousands and tens of thousands. We assume that the nodes move in a region independently of their communication needs.

Communication Needs: Nodes are executing one or more applications that depend on the communication between node pairs. (One member of the pair could be an Access Point, or similar entity, communicating with multiple nodes.) The application(s) running at each node are delay tolerant, however there is a maximum acceptable delay for the delivery of the packets.

[1] This work also appears in [33, 5].

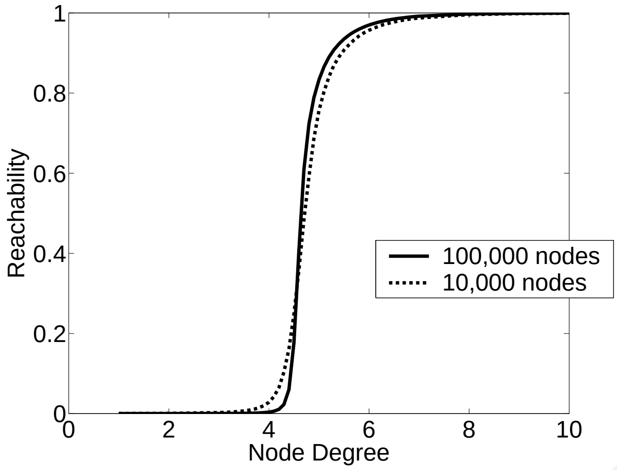


Figure 3.1: Reachability versus node degree.

Connectivity level: As discussed in Chapter 2, in [84] wireless ad hoc networks are classified in four categories: *(i)* End-to-end paths exist almost always, *(ii)* End-to-end paths exist for some fraction of the time, *(iii)* The network is always partitioned, *(iv)* Multi-hop paths are rare. DTFR and EMEED are designed for use in categories *(ii)* and *(iii)* of the above classification.

To clarify this point, Fig. 3.1 shows simulation results for networks of 10,000 nodes and 100,000 nodes. The nodes are placed randomly, according to the uniform distribution, on a square area. Nodes that are separated by a distance of at most R can communicate directly with each other. The nodes are divided randomly and uniformly in pairs and each node is the destination for the packets of its counterpart. In the figure, we plot the average *Reachability*, which is defined as the proportion of nodes that have a path to their destination at a given time instant, versus the expected node degree. (We trace different values of the expected node degree by varying the communication range R.)

As the figure shows, for values of the average node degree below a threshold, an end-to-end path to the destination does not always exist. Therefore, traditional ad hoc routing protocols such as greedy-face-greedy protocols [45] and reactive ad hoc routing protocols [94] are not suitable and DTN solutions are needed. We show that for a large range of average node degrees at this case, for some values of the maximum acceptable delay for the delivery of the packets, DTFR performs much better than Spray and Wait and a number of state of the art protocols we simulate.

Neighborhood Awareness: Each node is aware of the network topology in its local neighborhood. At the very least, this means that the node is aware of all nodes close enough for direct communication with them to be possible. In the more general case, each node might also have location information about some of the nodes a small number of hops away. Neighborhood awareness is achieved through the use of eavesdropping packets and/or beaconing [95], or through the use of a proactive ad hoc routing protocol such as OLSR [96].

3.2 DTFR Overview

The DTFR protocol consists of four mechanisms: 1) a *Dissemination Rule*, responsible for disseminating a number of replicas in the vicinity of the destination, 2) a forwarding rule, responsible for node to node packet forwarding, which we term *Greedy Lazy Forwarding (GLF)*, 3) a *Priorities Policy*, for assigning priorities to nodes contending for access to the wireless medium, and 4) a *Buffer Policy*. Next, we discuss each of these four mechanisms. We stress that some implementation details of these mechanisms will depend on the details of the application, and so are left undefined here.

3.3 Dissemination Rule

The dissemination rule of DTFR consists of four phases:

1. *Homing Phase*: The packet travels to a point called the *Firework Center (FC)*, at the center of a region where the source estimates the destination to be.

2. *Explosion Phase*: Then, the packet is replicated and given to L relays.

3. *Spread Phase*: Then, the packet replicas travel, using GLF, to L different points called the *Firework Endpoints (FEs)*, that are symmetrically placed around the FC, at a distance D from it. Once there, the replicas are discarded. The distance D is chosen to be such that the destination will be between the FC and the FEs with high probability.

4. *Lock Phase*: At any time during the first three phases, if a packet comes near enough to the destination to discover a multihop route, it enters the *Lock Phase* in

which it is forwarded to the destination using that route, in the usual, non DTN manner, using a non delay tolerant ad hoc routing protocol.

3.4 Greedy Lazy Forwarding (GLF)

GLF is used in the *Homing* and *Spread* phases, when packets travel toward the FC and FEs respectively. Consider a node A holding a packet P, destined for some distant location D. Let the *forwarding area* F be the set of points closer to the destination D than point A, and also with a distance of at most R from A. (Note that when D is far away from A, F becomes a semicircle.) Node A uses the following rule for choosing the next relay of P:

1. (Greedy part) If there is at least one node within F, then A forwards the packet to one of the nodes in F. (Details of the choice are left to the particular implementation.)

2. (Lazy part) If there is no node within F, then A waits until a node appears on the boundary of F, and then immediately sends the packet to that node.

This rule is being executed continuously: when a packet arrives at a node, the node checks its forwarding area for potential relays. If the area contains one or more nodes, the packet gets forwarded to one of them. Otherwise, the packet waits for one such node to appear. Observe that, as with all other geographic routing protocols, our forwarding protocol is greedy, in the sense that it provides an immediate improvement if this is possible. However, in contrast to them, it is also lazy: upon failure to achieve an immediate improvement, it just waits for the topology to get better.

3.5 Firework Center and Firework Edges Calculation

The locations of the FC and the FE are specified by the packet source when the packet is created, and inserted to the packet's header.

As with other aspects of DTFR, the precise method for calculating the FC depends on the details of the environment and the application, and notably the localization mechanism that is assumed to exist. The overriding principle is that the FC should be as close as possible to the destination, when the packet arrives at the FC. Therefore, if the localization mechanism provides to the packet source an estimate of the location of the destination at a recent time instant, then this estimate should be used as a FC. If, however, the localization mechanism also provides estimates of the location of the destination at future instances (for example, by revealing the route of the destination and its basic travel characteristics), then the FC can be selected so that it approximately intercepts the destination some time in the future [97]. If the source and destination participate in a regular exchange of packets, then they can provide to each other all the information about their location and future trajectory that is available to them.

The L FEs are placed uniformly on the circumference of a circle of radius D centered at the FC. D is selected to be, with a high degree of confidence, large enough so that one of the replicas will have the opportunity to overtake the destination. Clearly, the larger D is selected, the larger L should also be, in order for the circular region bounded by the FEs to be covered adequately. Note that using excessively large values for L and D means that bandwidth will be wasted. The precise rule for selecting the values of L and D will depend on the environment, and notably on the mobility model, and so is omitted here.

3.6 Priorities Policy

In order to access the medium, packets are given different priorities, depending on the phase they are in. *Lock Phase* transmissions have priority over transmissions of all other phases. This is because when a packet goes near its destination we do not want to loose the opportunity to deliver it, given the changing topology. *Homing Phase* transmissions have priority over *Spread Phase* transmissions and *Explosion Phase* transmissions, as we do not want to delay the only copy of a packet from reaching the FC and so delay the search in the whole region near its destination. *Explosion Phase* transmissions have priority over *Spread Phase* transmissions, as we want to create all replicas quickly.

3.7 Buffer Policy

The buffer of each node has a finite size B. Once a buffer is full, the node cannot receive any packet unless it is destined for that node, and must discard the packets its user creates. Packets are discarded when they reach the FEs. Also, the packets have a Time To Live (TTL) equal to the maximum acceptable delay for the delivery of the packets.

3.8 Bethlehem Routing

Under *Bethlehem Routing (BR)*, each packet is continuously aware of the location of its destination, and moves towards it by continuously staying in the *Homing Phase* with the actual location of the destination chosen as the FC. Once near enough to the destination to discover a route, the packet will enter the *Lock Phase*. Excluding this (crucial) modification, BR is identical to DTFR. Clearly, BR can only work when an extremely effective location service is available to the system. In relatively small networks, such a location service has been shown to exist: the authors of [6] found no significant difference between an oracle location service and the practical location service LoDiS introduced there, for networks with approximately 100 nodes. This indicates that BR might be a practical and better routing protocol than DTFR, when the number of nodes is on that order. Note, however, that networks with $10,000$ nodes are within the scope of this work, and the performance of BR we report here, for networks of this size, should be viewed as an upper bound only. Indeed, we are interested in BR mostly as a means of evaluating the cost of not precisely knowing the location of the destination. This cost is the gap between the respective performance metrics of DTFR and BR.

3.9 Simulation Setting

Mobility Model: Nodes move on a square grid composed of vertical and horizontal roads. Initially, each node is placed at random on the grid, and then proceeds to travel, using the road network, to a randomly chosen location, using a constant speed, uniformly distributed between 0 and $v_{\max}$. Then, it chooses another random location, and another speed, moves to that location, and so on.

Traffic Pattern: All nodes are divided in pairs, each node communicating with its counterpart. Pairs do not change for the whole duration of the simulation.

Channel Model: We assume an urban environment where both Line Of Sight (LOS) and Non LOS (NLOS) communication are possible, however the power of signals received through a LOS attenuates slower with distance. In particular, LOS communication is only possible between nodes lying on the same road. However, if a node is within a threshold distance R_T from the intersection between two roads, we assume that this node belongs to both roads. (R_T is essentially the radius of the junction.)

In the case of LOS transmissions, the signal power P_r received at distance d from a transmitter is

$$P_r = P_0 \left(\frac{d_0}{d}\right)^{\alpha_{\text{LOS}}}$$

and, for NLOS transmissions,

$$P_r = P_0 \left(\frac{d_0}{d}\right)^{\alpha_{\text{NLOS}}}$$

where P_0 is the received power at a small reference distance d_0 from the transmitter and α_{LOS} and α_{NLOS} are exponents that describe the environment, typically 2-6 [98], with $\alpha_{\text{LOS}} < \alpha_{\text{NLOS}}$, so that NLOS signals attenuate faster.

Transmitter Model: While transmitting, a node cannot listen to the transmissions of other nodes. If node k is not a transmitter, a packet from node i is received successfully at node k if

$$\frac{P_{ik}}{\sum\limits_{j \in S, j \neq i} P_{jk} + N} > \gamma_T, \tag{3.1}$$

where N is the background noise, γ_T is the minimum Signal to Interference plus Noise Ratio (SINR) required at the receiver, P_{jk} is the received power at node k from node j, and S is the set of all transmitters.

Slotted Time: We slot time, and at the start of each timeslot each node creates a packet with a predefined probability λ. The packet is immediately stored in the buffer of that node if it is not full. The transmission of each packet takes one timeslot. Timeslots are assumed to be so short, that the topology cannot change appreciably for a timeslot duration, and hence channel gains are constant throughout each of them. In practice, in our simulations nodes are stationary during timeslots and move abruptly to their new positions, according to the underlying mobility model, during timeslot transitions. As the networks we simulate have many thousands of nodes, for such a slotted system to work it is necessary to employ GPS receivers or, alternatively, a sophisticated distributed clock synchronization system [99]. Further discussion on this topic goes beyond the scope of this work.

Medium Access Control: At the start of each slot, nodes employ a MAC scheme to decide who will transmit at that slot, what packet, and to whom. At any given time during the execution of this scheme, the state of a node can be either *available* or *reserved*. At the start of the slot, all nodes are available, but progressively attempt to make reservations, according to their priorities (See Section 3.6). For a node A to be able to send a packet to another node B, both A and B must not be reserved. If this is the case, nodes A, B and all the nodes within distance $K \cdot d_{AB}$ from A or B become *reserved*. K is a constant greater than 1, which we term the *Reservation Radius Constant*. As we are not interested in the evaluation of the MAC layer, we assume that the reservations are all arranged instantaneously, at the start of each slot, and no MAC control messages are simulated. As our focus is on routing, we refrain from using a more detailed MAC protocol. We note, however, that our MAC protocol allows the use of priorities and realistically captures the capabilities of the wireless channel, notably modeling congestion.

Power Control: If node A has decided to transmit a packet to another node B, A uses a power level P_t such that the transmission will be successful if the interference from competing transmissions turns out to be at most $(I_f - 1)$ times the thermal noise, where I_f is a constant we call the *Power Control Safety Margin*. Also, there is a maximum allowed transmission power $P_{0\max}$.

Local Routing Table: As already discussed, nodes maintain a routing table that can be used for routing in their immediate neighborhood. To conserve bandwidth and improve robustness, nodes do not use local routes that minimize the number of hops. Rather, a link cost is introduced, and nodes try to use paths with minimum total link cost. In particular, each LOS transmission from a node A to another node B is associated with a cost d_{AB}^2, where d_{AB} is the distance between nodes A and B. Each NLOS transmission is associated with a cost $d_{AB}^{\frac{2\alpha_{\text{NLOS}}}{\alpha_{\text{LOS}}}}$. The routing table includes destinations for which there is a path with total link cost at most equal to a threshold value C_T, which we term the *Local Routing Threshold*.

We do not simulate control messages for the creation of the local routing table. Therefore, interference experienced by data packets comes only from data packets. We believe that, as we are interested in the more challenging case of communication across large distances and large stretches of time, these assumptions, that essentially remove local routing issues from the picture, are justified.

Firework Center: In the case of DTFR, GeoDTN+Nav, and GeoCross, we assume that when a packet is created the source is informed of the location of its destination and (in the case of DTFR) uses that as the FC. Unless stated otherwise, we assume that this location is reported with no error.

Routing Protocols: We simulate DTFR, BR, GeoDTN+Nav, GeoCross, Spray and Wait, and flooding. We also simulate a protocol of our own design that we call Bethlehem GeoDTN+Nav (BetGeo), which is identical to GeoDTN+Nav except from one point: whenever a routing decision is made that involves the location of the destination, instead of using the position that the destination occupied at the time of the packet's creation, its current position is used. As with the BR protocol, this is an idealization, however the performance of this protocol allows us to evaluate the cost on the performance of GeoDTN+Nav of using location information that is not current. Under the flooding protocol, each node sends copies of all packets it has in its buffer to all nodes it meets, and all transmissions have the same priority.

In our implementation of Spray and Wait, and in order to have a more fair comparison to DTFR, nodes make use of the local routing table. *Lock Phase* transmissions have priority over *Spray Phase* transmissions.

In our implementation of GeoCross and GeoDTN+Nav, nodes are given access to the local routing table. *Lock Phase* transmissions have priority over all other transmissions. Greedy mode transmissions, perimeter mode transmissions, and DTN mode transmissions are equal in priority, but transmissions from junction nodes have priority over transmissions from street nodes.

Unless otherwise stated in each particular case, the parameters used are those of Table 3.1. For each point in the plots, we simulate each protocol for different values of its various parameters, and select the values that produce the best results. The results shown are for the steady state of the simulation. We also run the mobility model until it reaches its steady state, before starting creating packets.

3.10 Simulation Tool

In order to evaluate our protocol, we have developed VL-DTN-S (Very Large DTN Simulator), a simulation tool specifically designed for DTNs, and written in C. The tool is available online [32].

We have refrained from using NS-3 [100], OMNeT++ [101], or a similar general-purpose simulation tool, because such tools were designed for routing in traditional networks and so are not best adapted to the unique challenges appearing in DTNs (for example, the need for very large buffers), particularly in the case where there are many

PARAMETER	NUMERICAL VALUE
Slot Duration	0.01 sec
Packet arrival rate	$\lambda = 0.02$ packets/sec/node
Number of nodes	$n = 5000$
Side of the grid in which the nodes move	7 km
Distance between junctions	200 m
Junction radius	$R_T = 10$ m
Maximum node speed	$v_{\max} = 10$ m/sec
Packet TTL	6 min
LOS exponent	$\alpha_{\text{LOS}} = 3$
Non LOS exponent	$\alpha_{\text{NLOS}} = 5$
Propagation model reference distance	$d_0 = 1$ m
SINR Threshold	$\gamma_T = 10$
Power Control Safety Margin	$I_f = 10$
Thermal Noise over Transmitter Power	$\frac{N}{P_{0\max}} = 1.25 \cdot 10^{-9}$
Local Routing Threshold	$C_T = 4 \cdot 10^4$ m^2
Simulation duration	1 hour
Buffer size	$B = 10^4$ packets
Reservation Radius Constant	$K = 2.5$
Maximum Location Error	$E = 0$ m

Table 3.1: Default simulation parameters.

thousands of nodes. We also refrained from using ONE [102], DTNSim2 [103], or any other JAVA-based DTN simulation tool, as the use of JAVA necessarily slows down the execution of the simulation when the number of nodes is very large. Discussions on the relative merits of the various simulators for use in DTN environments can be found in [102, 104], and references therein.

Efforts have been made to make VL-DTN-S as accurate as possible. Among others, *(i)* full buffer information for all nodes is kept, *(ii)* realistic physical layers are used, and *(iii)* contention in the channel is taken into account. At the same time, efforts have been made so that the simulator is as fast as possible and, as a result, the tool is capable of detailed simulations of networks of more than 10^4 nodes on a desktop computer, and for a variety of routing protocols. Challenging simulations with 10^4 nodes take at most a few hours.

In the rest of this section we explain some of the techniques we used to make the program faster.

As already described, for a node to receive a packet successfully Equation (3.1) must hold, i.e., the SINR at its receiver must be greater than a threshold value γ_T. We use the following approximation for verifying that Equation (3.1) holds. Assume that a node A wants to receive a packet that another node B is transmitting. The power of the signal is the power received at node A from node B. The power of the interference is the total power received at node A from all other nodes that transmit packets during that slot and are at distance at most R_{ii} from A, where R_{ii} is a parameter of the simulation. We choose a large value for R_{ii} so that the total interference at A from all transmitters at distance more than R_{ii} from A is very small compared to the total interference at A from all transmitters at distance less than R_{ii} from A. We use the parameter R_{ii} to avoid spending time on calculations that do not change the value of the denominator of Equation (3.1) significantly.

Also, in order to make the program faster, we divide the area in which the nodes move in square cells. The side of the cells is equal to the reservation radius constant K multiplied by the transmission range. The cells are numbered from 1 to no_cells. The number of nodes in the network is num_nod. $cellc$, $cell1$, $cell2$, and $cell3$ are one-dimensional arrays that have num_nod elements each. $cellidx$ is an one-dimensional array that has $no\_cells + 2$ elements. At each slot the following is done. First the program finds the cell of each node i and stores it in $cellc[i]$. Then for $i = 1$ to $i = no\_cells$, the following is done.

- The program stores in $cellidx[i]$ the address of the first empty position of $cell1$, $cell2$, and $cell3$. For $i = 1$ this is position 0.

- For $j = 0$ to $j = num\_nod - 1$ if $cellc[j] == i$ the program stores the node id of node j and its x and y coordinates in the first empty position of $cell1$, $cell2$, and $cell3$ respectively.

Then $cellidx[no\_cells + 1]$ is set to num_nod. After the above are done, the element $cellidx[i]$, for $i = 1, 2, \ldots, no\_cells$ stores the position in the one-dimensional arrays $cell1$, $cell2$, and $cell3$ where the information for cell i begins. This information ends at position $cellidx[i + 1] - 1$.

Using the cells and the associated data structures described in the previous paragraph, we accelerate our simulation as follows:

- In the part of the program where each node finds its neighbors, each node checks if it is within the transmission range of each node in its cell and the 8 cells around its cell. It does not check for the rest of the nodes in the network and this makes the program faster.

- Also, the ids and coordinates of the nodes in a cell are at continuous places in $cell1$, $cell2$, and $cell3$, and this also makes the program faster.

- In the part of the program that finds which nodes will transmit packets, when a node A decides to transmit a packet to another node B, all nodes within distance K times the transmission range from A or B that are not reserved become reserved. This means that they are not allowed to decide to transmit or receive a packet during that slot. To find which nodes have to become reserved the program checks only in the cell of A and the 8 cells around it and the cell of B and the 8 cells around it.

In the part of the program that finds which packets are received successfully, we divide the area in cells of side R_{ii} and we use one-dimensional arrays to store for each transmitter the node id, x and y coordinates, and transmitted power, and using these makes the program faster.

In order to speed up the handling of packets in the various parts of the simulation, each packet in the buffer of a node i for a destination j, consists of a row, say k, in each of the following one-dimensional arrays. If the packet uses the k^{th} row in one of the one-dimensional arrays, then it uses the k^{th} row in all of them.

- $buffer0[k]$ stores the destination of the packet.
- $buffer1[k]$ stores the creation time of the packet.
- $buffer2[k]$ and $buffer3[k]$ store the row of the next and previous element respectively in the list of packets node i has for destination j.
- $buffer11[k]$ and $buffer12[k]$ store the row of the next and previous element respectively in the list of packets node i has.
- $buffer9[k]$ stores 1 if the destination has received the packet and 0 if not.
- $buffer4n[k]$ and $buffer4p[k]$ store the row of the next and previous element respectively in the list of copies of the specific packet in the network.
- bf_f and bf_l are arrays of num_nod rows and num_nod columns. $bf\_f[i][j]$ and $bf\_l[i][j]$ store the row of the first and last element respectively in the list of packets node i has for destination j.
- bf_f_fir and bf_l_fir are one-dimensional arrays of num_nod elements. $bf\_f\_fir[i]$ and $bf\_l\_fir[i]$ store the row of the first and last element respectively in the list of packets node i has.

The program inserts packets in the buffers and deletes packets from the buffers easily by updating the linked lists described above.

- When a node i checks if it has a packet to transmit for a destination j in the Lock Phase, it checks the list of packets it has for destination j and this is faster than checking the list of all the packets it has.
- When i checks if it has a packet to transmit in the Homing Phase, it checks the list of all the packets it has and this is faster than checking the lists it has for each destination j.
- When a copy of a packet is received by the destination, the packet is deleted from the buffer of the node that has given it to the destination. However other nodes may still have copies of the packet. When a packet is received by the destination, the program sets the value of $buffer9$ of all these copies to 1. The program does this with the help of $buffer4n$ and $buffer4p$. When a node is checking if it can transmit a packet in the Lock Phase, say the packet is at row k, it checks the

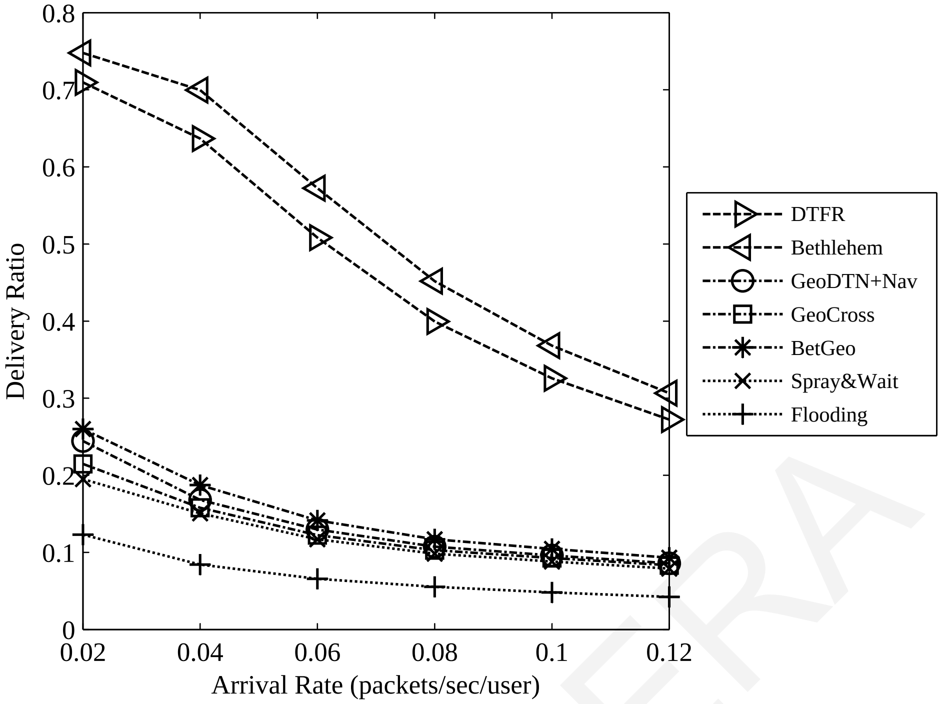


Figure 3.2: Delivery ratio versus arrival rate.

value of $buffer9[k]$ and it does not transmit packets that the destination has already received.

3.11 Results

In this section we present simulation results. Unless otherwise stated in each particular case, the parameters used are those of Table 3.1.

In Fig. 3.2 we plot the packet delivery ratio versus the packet arrival rate for all protocols. Even with very small arrival rates, no protocol manages to deliver all packets within the TTL. This is due to the fact that the network is often partitioned for periods of time comparable to or larger than the TTL. In addition, quite often the network is not partitioned but bottlenecks are formed due to the topology, leading to queuing delays.

Observe that the delivery ratio of GeoDTN+Nav is significantly smaller than the delivery ratio of DTFR. There is a number of reasons for this. First of all, DTFR uses the *Explosion*, *Spread*, and *Lock Phases* to counter the fact that the destination is moving. No similar mechanisms exist in GeoDTN+Nav. (Note, however, that even with Bethlehem GeoDTN+Nav, where GeoDTN+Nav is enhanced so that the packets

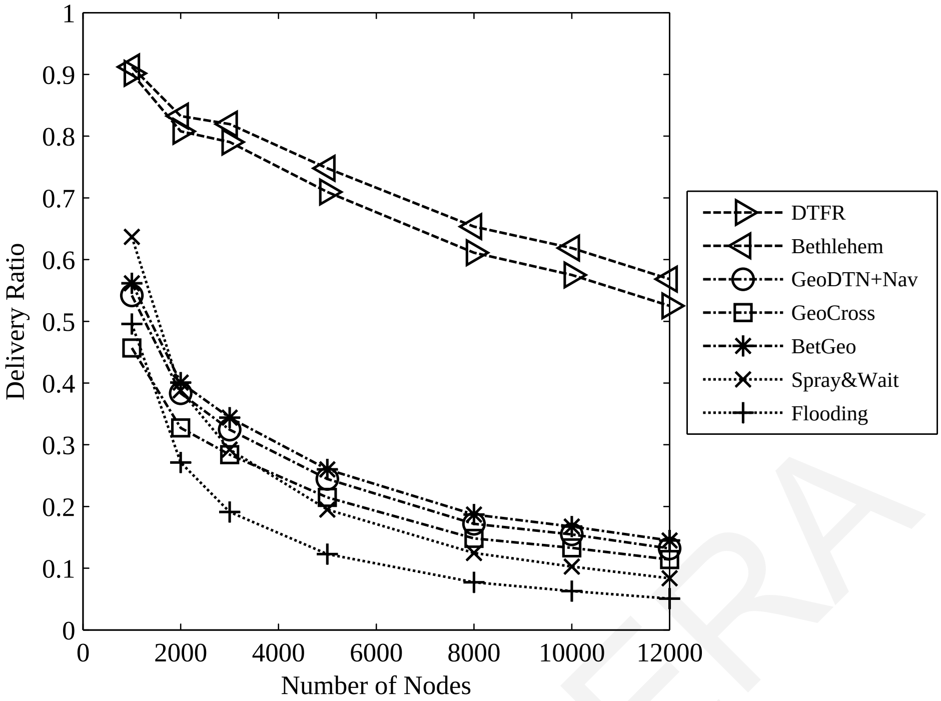


Figure 3.3: Delivery ratio versus number of nodes.

have continuous perfect knowledge of the position of their destination, the delivery ratio improves modestly over GeoDTN+Nav.) Secondly, under GeoDTN+Nav it is possible that packets leave the perimeter mode and enter the DTN mode at a node that is further away from the destination than the node they were when they entered the perimeter mode. In between, they were transmitted multiple times, wasting precious bandwidth in the process. DTFR, on the other hand, never transmits a packet away from the destination. Thirdly, under GeoDTN+Nav packets stay in the DTN mode even when there are neighbors of the current holder closer to the destination, because their distance to the destination is greater than the distance between the destination and the point where the packet entered the perimeter mode. Under DTFR, on the other hand, nodes always send packets to neighbors closer to the destination than themselves.

In Fig. 3.3 we plot the packet delivery ratio versus the network size. We change the network size by changing the number of nodes and the dimensions of the area, keeping the number of nodes per unit road length constant. Note that the performance of all protocols diminishes with the network size. This is due to the facts that *(i)* the TTL counter remains fixed, *(ii)* with larger network sizes partitions are more frequent, and

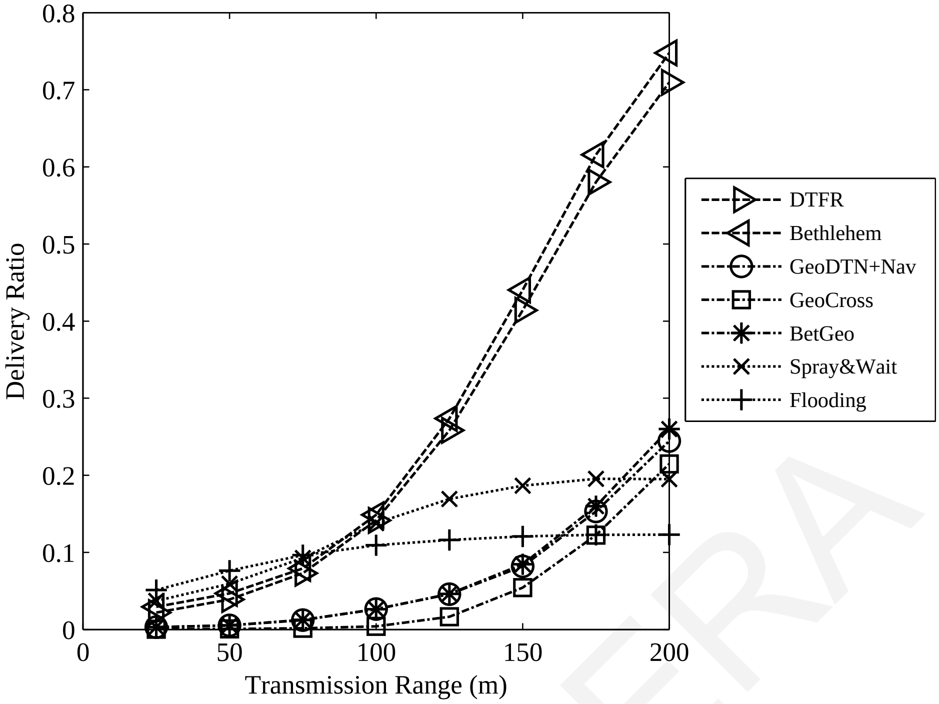


Figure 3.4: Delivery ratio versus transmission range.

(iii) with larger network sizes more transmissions are needed for the delivery of each packet. However, the relative performance of all protocols remains the same with that of the previous plot.

In Fig. 3.4 we plot the packet delivery ratio versus the transmission range. We change the transmission range by changing the value of $\frac{N}{P_{0\max}}$. All protocols gain by an increase in the transmission range, however the two protocols that do not depend on the fast forwarding of the packets to the area where the destination is expected to be, Spray and Wait and flooding, benefit the least. On the other hand, Spray and Wait is slightly superior to the rest (except flooding) in the case of small transmission ranges. This is due to the fact that, in this regime, packets using our GLF mechanism travel with very small speeds towards the destination. We explore this issue in great detail later on. The performance of GeoCross and GeoDTN+Nav increases fast as the transmission range increases, because the perimeter mode becomes more efficient for larger transmission ranges.

In Fig. 3.5 we plot the packet delivery ratio versus the maximum speed of the nodes. For high speeds, Spray and Wait gives good performance. Clearly, when the node mobility is too high, the best strategy for the source is to get out as many

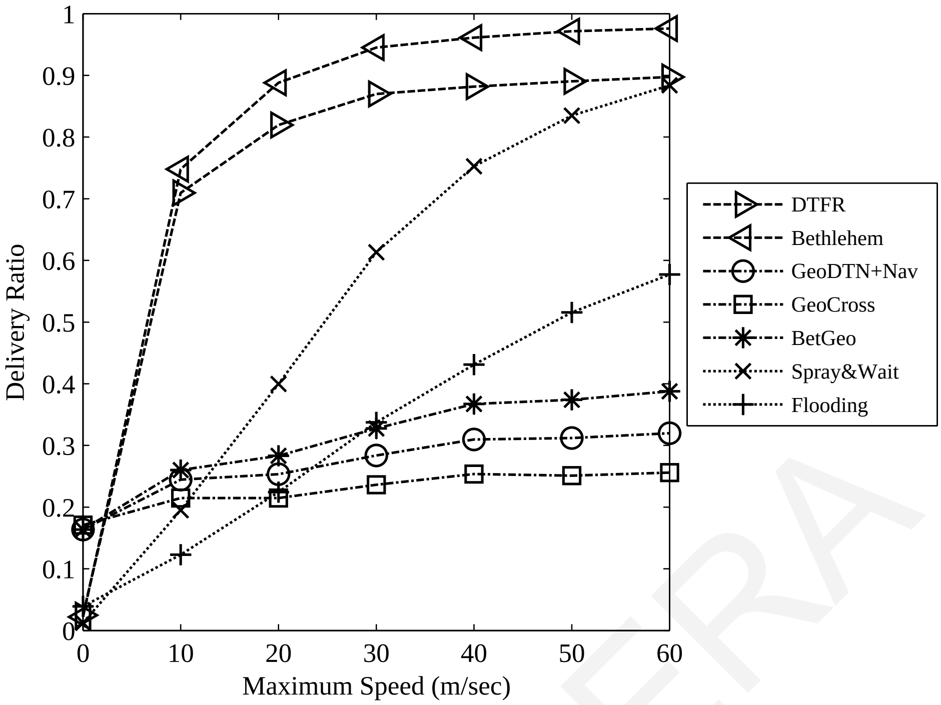


Figure 3.5: Delivery ratio versus maximum speed.

replicas as possible. Also observe that, in the other extreme, when nodes are immobile, GeoCross gives better results than DTFR. This is expected: when a packet reaches a local optimum, and nodes are immobile, waiting is futile, and the only alternative is going into perimeter mode. However, for all the cases in the middle, DTFR is surpassed only by BR. Note that to obtain the points in Fig. 3.5 for 0 velocity, we averaged multiple runs of the simulation, each of them for a different network topology chosen randomly from steady state topologies.

In Fig. 3.6 we plot the packet delivery ratio versus the maximum speed of the nodes and the transmission range in a three dimensional plot. As in the previous plots for each point in the plot we simulate DTFR using different values of the number of copies produced in the explosion phase and the distance between the firework center and the firework endpoints and we choose the values that give the best delivery ratio. We observe that as the maximum speed of the nodes increases the delivery ratio increases. Likewise, we also observe that as the transmission range increases the delivery ratio increases. This plot provides to network designers an estimate of how transmission range should change in order to keep the delivery rate fixed, in response to changes in the speed with which nodes move, or vice versa.

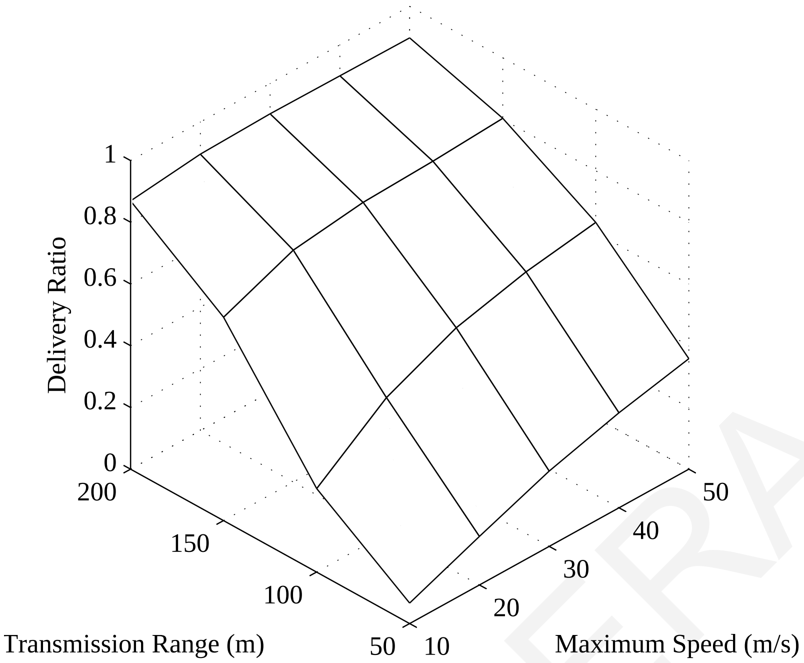


Figure 3.6: Delivery ratio versus maximum speed and transmission range.

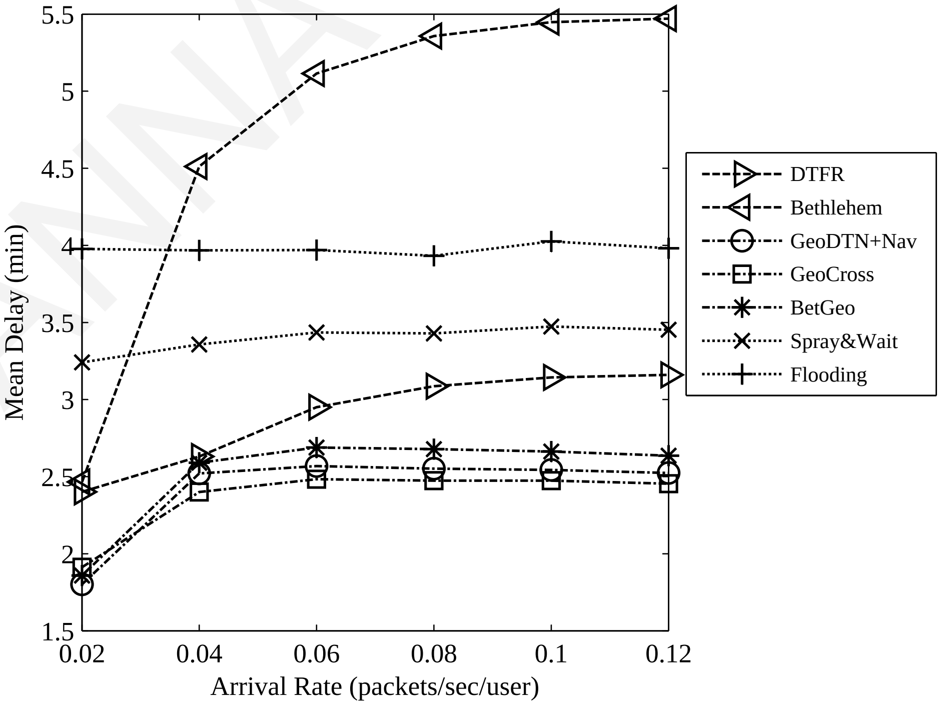


Figure 3.7: Mean delay versus arrival rate.

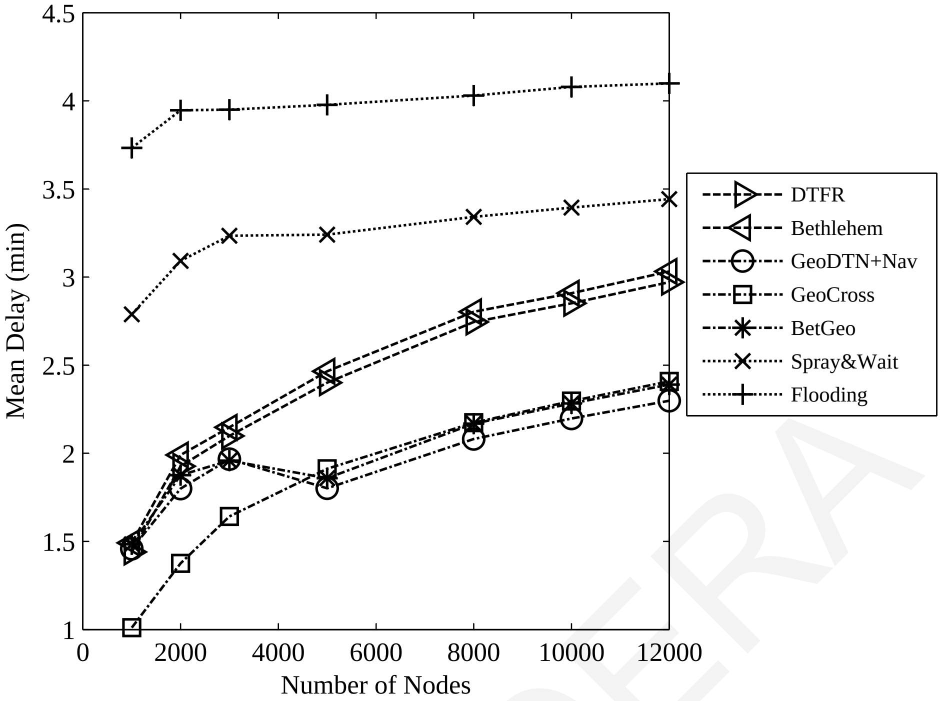


Figure 3.8: Mean delay versus number of nodes.

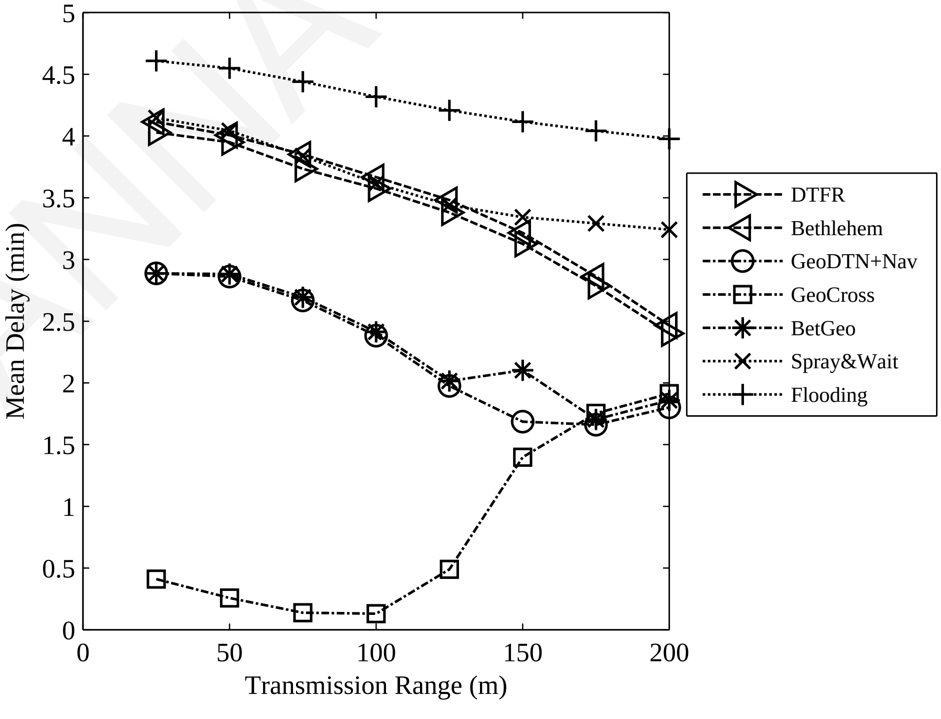


Figure 3.9: Mean delay versus transmission range.

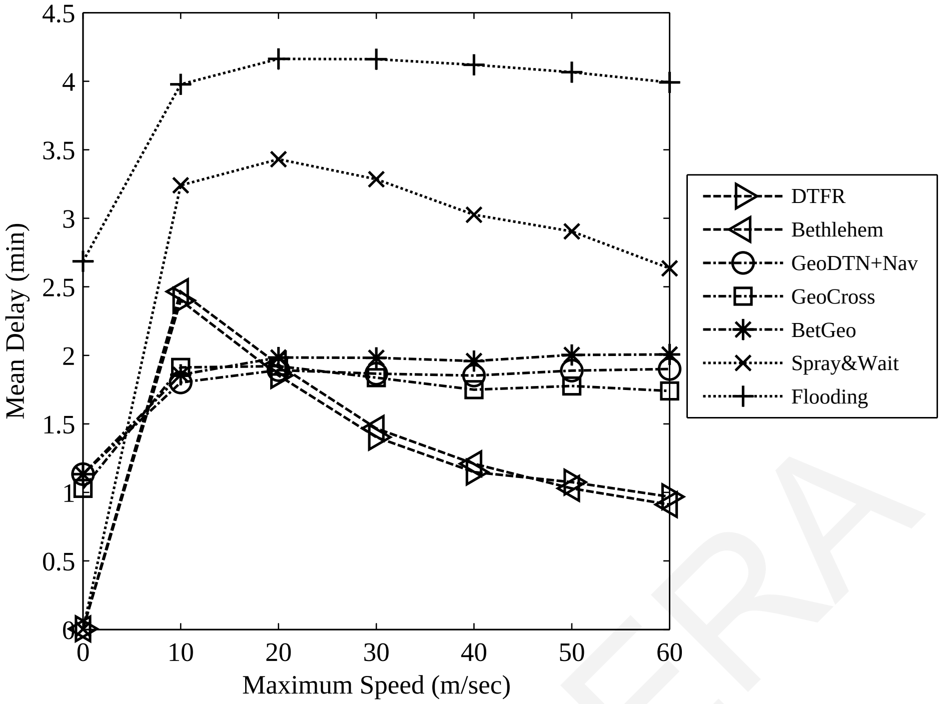


Figure 3.10: Mean delay versus maximum speed.

Figs. 3.7, 3.8, 3.9, and 3.10 show the mean delay versus the arrival rate, number of nodes, transmission range, and maximum speed of nodes, respectively. Note that protocols with small delivery rates, for example GeoCross, have small mean delays because those packets that do get delivered under them are easy to deliver (for example, the source and the destination are nearby), and so are delivered fast. In other words, their low delays are an artifact of their low delivery rates.

In our simulations, we place the FC at the location of the destination at the time of the creation of the packet. By the time the packet arrives at the FC, the destination has moved away. In order to keep packet losses at small levels, it is important that the distance D between the FC and the FEs is large enough. To verify that this is indeed the case, in Fig. 3.11 we plot the empirical cumulative distribution function of the distance x between the FC and the destination at the time of the arrival of a packet at the FC. The plot was created using a simulation with the parameters of Table 3.1 and 10^4 packets. Packets that entered the *Lock Phase* or expired before arriving at the FC were disregarded. The average distance is a little over 200 m, and the distance is less than 1000 m for around 95% of the packets. Most importantly, the distribution of the distance had a thin tail, which means that D does not have to be set excessively

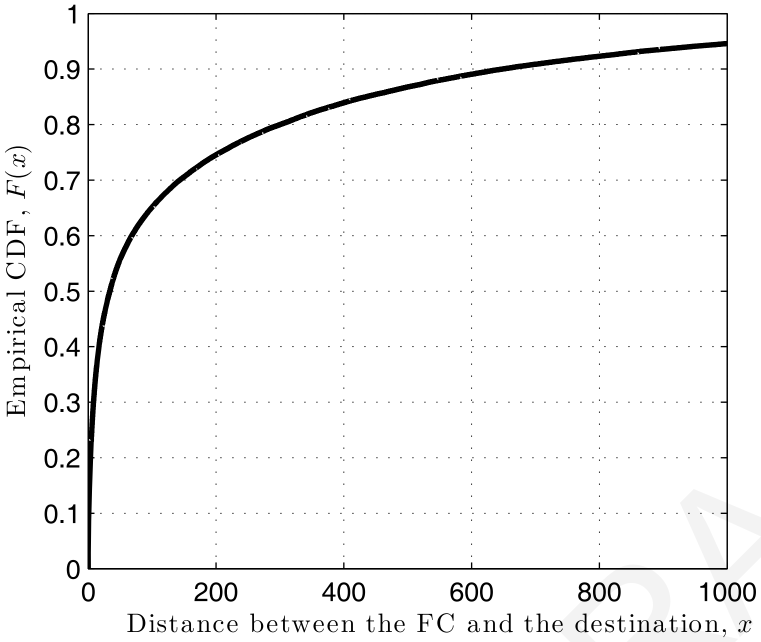


Figure 3.11: Empirical cumulative distribution function of the distance between the FC and the destination at the time of the arrival of the packet at the FC.

large to accomodate a wide range of distances x. In this simulation, the number of FEs was 16, and they were placed 2000 m from the FC.

Until now, we have assumed that the source obtains the exact location (x_0, y_0) of the destination at the time the packet is created using a location service, and inserts that location in the packet. Assume now that the X and Y coordinates inserted in the packet by the source are uniformly distributed in the intervals $[x_0 - E, x_0 + E]$ and $[y_0 - E, y_0 + E]$ respectively, where E is called the *Maximum Location Error.* The performance metrics, i.e., the delivery ratio and the mean delay, are shown in Figs. 3.12 and 3.13 respectively, as a function of E. Note that the performance of all protocols using the location of the destination decreases as E increases, but the performance of DTFR is superior to that of the others even for large values of E.

We also simulated GeoCross using the parameters of Table 3.1 but with a very low arrival rate of 10^{-4} packets/sec/node, immobile nodes, and a very large number of permitted hops in the perimeter mode, $h_{\max} = 10^4$. It was found that only 36% of the packets reached their destination, although an end-to-end path existed for 45% of the node pairs. Therefore, although GeoCross is a major breakthrough over GPCR, as discussed in Chapter 2, it does not altogether eradicate routing loops.

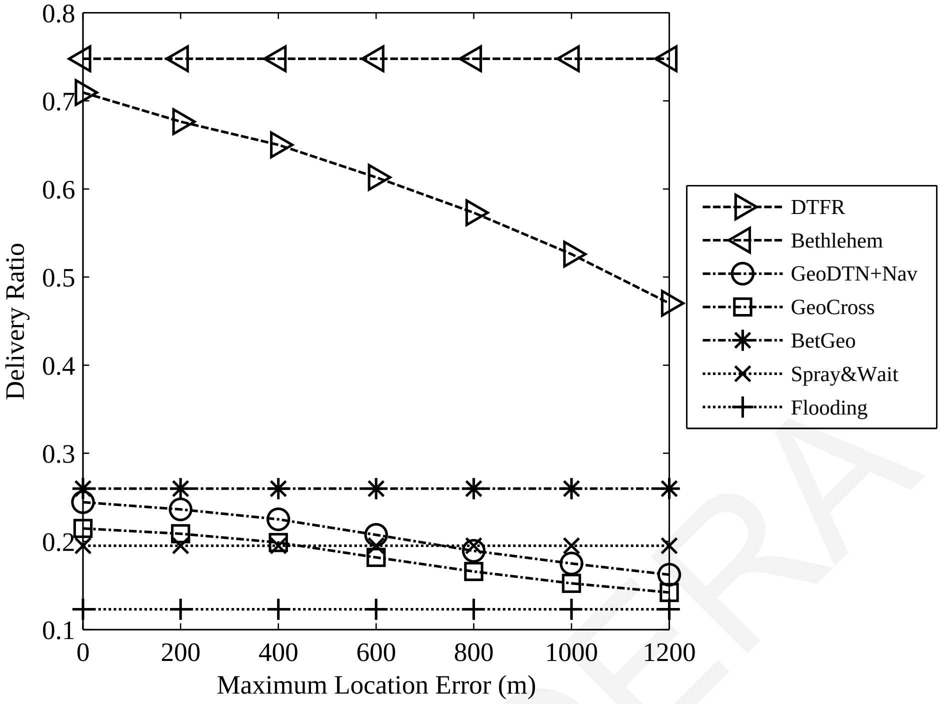


Figure 3.12: Delivery ratio versus maximum location error.

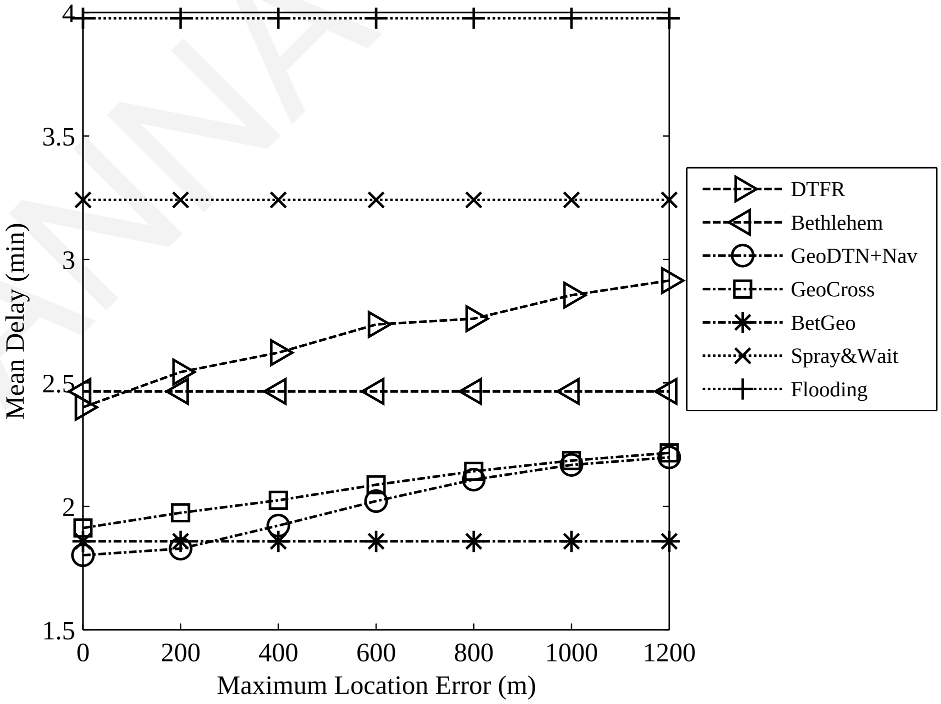


Figure 3.13: Mean delay versus maximum location error.

3.12 Conclusions

In this chapter we propose DTFR, a hybrid geographic delay tolerant routing protocol for wireless DTNs. DTFR consists of the homing, explosion, spread, and lock phases. In the homing phase, the packet travels to a location where the destination is estimated to be, called the Firework Center (FC). When it reaches a certain distance from the FC it enters the explosion phase, in which the packet is replicated to a number of copies. Then we have the spread phase, in which the copies travel toward different directions up to a certain distance from the FC. The copies travel to points called the Firework Endpoints (FEs). Each copy is assigned to one FE. When a copy reaches a certain distance from its FE it gets discarded. We assume that each node knows which nodes are a few hops away from it and knows routes to these nodes. If at any time during the homing, explosion, or spread phase the packet reaches a node that knows a route to the destination, the packet enters the lock phase and it gets delivered using that route.

The packet travels from the source to the FC and from the FC to the FEs using a delay tolerant version of geographic routing which is termed Greedy Lazy Forwarding (GLF) and works as follows. Let the forwarding area F be the set of points within the transmission range of the current holder and also closer to the destination than the current holder. If there is at least one node within F, the current holder forwards the packet to one of the nodes in F. If there are more than one nodes in F, the next hop is chosen at random. If there is no node within F, the current holder waits until a node appears in F, and then forwards the packet to that node.

As we establish by simulation (and also by analysis, in Chapter 4), the average velocity with which the packet travels from the source to the FC and from the FC to the FEs increases as the transmission range increases. Furthermore, it is greater than the velocity of the nodes for a wide range of transmission ranges for which there is no end-to-end path from source to destination; this fact is crucial to the success of our protocol. The distance between the FC and the FEs is a protocol parameter. It is chosen so that the time the packet needs to travel from the source to the FC and then from the FC to the FEs is the same or smaller than the time it takes for the destination to travel from the FC to the FEs with high probability. Thus, most packets reach the destination.

We compare DTFR with flooding and Spray and Wait, which are two baseline protocols, with GeoCross and GeoDTN+Nav, which are two state of the art protocols for vehicular networks, and with Bethlehem and Bethlehem GeoDTN+Nav which are two idealized protocols that provide upper bounds. We compare the performance of these protocols in terms of their packet delivery ratio and mean packet delivery delay versus the arrival rate, number of nodes, transmission range, and velocity of nodes. We use only transmission ranges for which there is no end-to-end path from source to destination.

GeoCross which uses traditional geographic routing fails because there is no end-to-end path from source to destination. In Spray and Wait in the wait phase the packets travel toward the destination with velocity at most equal to the velocity of the nodes. Thus for many values of the velocity of the nodes in Spray and Wait the packets cannot cover the distance between the source and the destination before the TTL expires. Simulations reveal that DTFR performs better than the other protocols we simulated, except in the following cases:

- When the nodes are immobile or move very slowly, GeoCross and GeoDTN+Nav perform better than DTFR because in this case in DTFR the packets get stuck in the wait phase.
- For very large node velocities Spray and Wait performs as well as DTFR because in this case in Spray and Wait the packets can cover the distance between the source and the destination by transport before the TTL expires.
- For small transmission ranges Spray and Wait and flooding perform a little better than DTFR because in this case under DTFR the packets can cover only a small part of the distance between the source and the destination by transmission, hence their speed of travel is small, and thus many packets cannot reach the destination before the TTL expires.

Chapter 4

Analysis for DTFR and BR

In this chapter[1] we present a succinct analysis of DTFR and BR, and in particular calculate the average delay and per node throughput achieved by DTFR and BR. Due to the complexity of these protocols, it is necessary to make a number of simplifying assumptions and approximations. Therefore, the aim of this chapter is not to arrive at accurate values for the performance metrics, as was done using simulations. Rather, our analysis has the following goals: *(i)* to verify the fundamental effects of the basic parameters of the environment (such as the node density and node speed) on the performance of the protocols, that were observed in the simulations, and *(ii)* to shed light on the fundamental reasons for its superior performance to protocols such as Spray and Wait for a wide range of node degrees.

4.1 Network Model

Node placement and mobility: The nodes are placed on an infinite region according to a spatial Poisson process with density λ (therefore, there is an infinity of nodes). All nodes move with a velocity of magnitude v_0, each one on its own direction, which is kept constant. Movement directions are independent and uniformly distributed in $[-\pi, \pi]$.

Data traffic: Each node sends data to another node chosen randomly among the rest, so that the distance between a source and its destination is a random variable with first moments $E(D_{od})$ and $E(D_{od}^2)$.

Channel access: All nodes are equipped with a transceiver of data rate r_D bps,

[1] This work also appears in [5].

PARAMETER	SYMBOL
Node Density	λ
Transceiver data rate	r_D
Delay of hop i	D_i
Cost of hop i	C_i
Event that F is eMpty	M
Normalized transmission cost	c_p
Average cost of protocol x	$E(C^x)$
Radius of explosion phase	R_X
Node speed	v_0
Distance between OD pair	D_{od}
Maximum communication range	R
Progress of hop i	X_i
Forwarding area	F
Packet speed	v_p
Average delay of protocol x	$E(D^x)$
Maximum per node throughput of protocol x	T^x

Table 4.1: Notation of Chapter 4.

and the maximum distance of direct communication is R. As we are interested in modeling very large delays, comparable to the time needed for the topology to change substantially, we assume that the packet transmission times are 0, i.e., packet transmissions are instantaneous.

To capture the contention among the nodes for the shared channel we assume that, for a transmission from a transmitter A to a receiver B to be successful, there must be no transmitter or receiver C closer to receiver B than transmitter A. Therefore, we associate with each successful transmission across distance d_i a disk-shaped footprint of radius d_i centered at the receiver. The footprints are not allowed to overlap, in order for the transmissions not to interfere, hence the condition above. This model for channel access contention is simple, and ignores many aspects of wireless communication, notably the fact that interference is additive. However, it captures the fact that there is a tradeoff between the number of transmissions and the distances they cover [12, 13, 105]. A similar model was used in [105]. We define the **cost of a transmission** across distance d to be πd^2.

Other assumptions: Under BR, each packet is constantly aware of the destination location. Under DTFR, each packet becomes aware of the destination location at the moment of its creation, but receives no update after that point. Nodes are equipped with buffers of infinite size. Also, nodes do not maintain any local routing protocol, and only know the location of the nodes currently within their communication range R.

4.2 Delay, Progress, and Cost of First Hop

Let node A create, at time $t = 0$, a packet destined for location Z. We assume that A is at the origin, and the destination Z on the positive x-axis, and sufficiently far away so that the forwarding region F is a semicircle. We also assume that if there are more than one nodes in F the next hop is chosen at random. (This choice of next hop was used in [106].)

Let D_1 be the delay until the packet is forwarded to its first relay, B, and let (X_A, Y_A) and (X_1, Y_1) be the coordinates of A and B at time D_1. Finally, let $X_T = X_1 - X_A$, $Y_T = Y_1 - Y_A$ and $C_1 = \pi(X_T^2 + Y_T^2)$ be the transmission cost. Observe that X_1, which we will call **progress**, represents the net reduction of the distance to the destination achieved at the conclusion of the first hop.

Let the event M that when A creates the packet, F is empty. Conditioning on M,

$$\begin{aligned} E(D_1) &= E(D_1|M)P(M) + E(D_1|M')(1 - P(M)), && (4.1)\\ E(X_1) &= E(X_1|M)P(M) + E(X_1|M')(1 - P(M)), && (4.2)\\ E(C_1) &= E(C_1|M)P(M) + E(C_1|M')(1 - P(M)). && (4.3) \end{aligned}$$

As F has an area $\pi R^2/2$, it follows that $P(M) = \exp[-\lambda\pi R^2/2]$.

Observe that if F is empty, the first relay B is the first node to enter it. In Section 4.7, we show that

$$E(D_1|M) = \frac{I_1}{v_0 R\lambda}, \qquad E(X_1|M) = I_2 R, \qquad E(C_1|M) = I_3 R^2, \tag{4.4}$$

where $I_1 \simeq 0.4817,\ I_2 \simeq 0.3890,\ I_3 \simeq 2.3317$.

Now assume that M' holds. We first note that, as transmission do not take time, we have $E(D_1|M') = 0$. As the first relay B is chosen randomly among those available, it follows that its location is uniformly distributed in F. Therefore:

$$\begin{aligned} E(X_1|M') &= \frac{1}{\pi R^2/2}\iint_F x\,dA = \frac{2}{\pi R^2}\int_0^R \left(\int_{-\pi/2}^{\pi/2} r^2\cos\theta\,d\theta\right) dr = \frac{4}{3\pi}R,\\ E(C_1|M') &= \frac{1}{\pi R^2/2}\iint_F \pi r^2\,dA = \frac{2}{R^2}\int_0^R \left(\int_{-\pi/2}^{\pi/2} r^3\,d\theta\right) dr = \frac{\pi R^2}{2}. \end{aligned}$$

Therefore, we know all quantities appearing on the right hand sides of Equations (4.1), (4.2), (4.3).

4.3 Packet Speed and Normalized Cost in Greedy Lazy Routing

Let P be a packet traveling from node A to node B, along a sequence of hops $i = 1, \dots$ Let D_i, X_i, and C_i be the delay, progress, and cost of hop i. We make the following assumption[2]:

Basic Assumption: The $\{D_i\}$ are i.i.d., the $\{X_i\}$ are i.i.d., and the $\{C_i\}$ are i.i.d.

The law of large numbers applies and we have, as $n \to \infty$,

$$\frac{1}{n}\sum_{i=1}^{n} D_i \to E(D_1), \quad \frac{1}{n}\sum_{i=1}^{n} X_i \to E(X_1), \quad \frac{1}{n}\sum_{i=1}^{n} C_i \to E(C_1).$$

[2] The interested reader is referred to [107, 108], where the packet speed in a DTN is investigated without the use of this assumption (although various others are made). There, however, the focus is exclusively on the topic of the packet speed in DTNs.

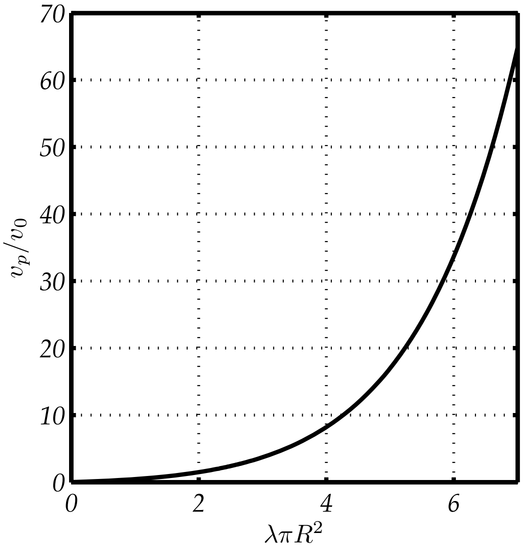


Figure 4.1: Normalized packet speed, v_p/v_0, versus the average number of neighbors, $\lambda\pi R^2$.

Taking quotients,

$$\frac{\sum_{i=1}^{n} X_i}{\sum_{i=1}^{n} D_i} \to \frac{E(X_1)}{E(D_1)}, \quad \frac{\sum_{i=1}^{n} C_i}{\sum_{i=1}^{n} X_i} \to \frac{E(C_1)}{E(X_1)}.$$

The first limit expresses the fact that the speed of the packet, averaged over the whole duration of its journey, will converge to the **packet speed**

$$v_p \triangleq \frac{E(X_1)}{E(D_1)} = v_0 \lambda R^2 \left[\frac{I_2}{I_1} + \frac{4\,(1 - \exp(-\lambda\pi R^2/2))}{3\pi I_1 \exp(-\lambda\pi R^2/2)} \right]. \tag{4.5}$$

The packet speed equals the rate with which the packet approaches the destination, and its value is not affected by any move that the packet does perpendicularly to the direction to the destination. In other words, it is the magnitude of the projection of the average velocity vector of the packet on the line connecting the current location of the packet to the destination. It is taken to be positive when the packet moves toward the destination.

The second limit expresses the fact that the total cost over the total progress converges to the **normalized cost**

$$c_p \triangleq \frac{E(C_1)}{E(X_1)} = R \times \frac{I_3 \exp(-\lambda\pi R^2/2) + \frac{\pi}{2}\,(1 - \exp(-\lambda\pi R^2/2))}{I_2 \exp(-\lambda\pi R^2/2) + \frac{4}{3\pi}\,(1 - \exp(-\lambda\pi R^2/2))}.$$

We make the approximation that the speed with which packets move is v_p, and the cost per unit of distance is c_p, even when the number of hops n does not approach infinity.

Observe that v_p is proportional to v_0, and depends on the node density λ and communication range R only through the average number of neighbors, $\lambda\pi R^2$. In Fig. 4.1 we plot v_p/v_0 as a function of $\lambda\pi R^2$. The speed v_p is an increasing function of $\lambda\pi R^2$ and becomes larger than v_0 for $\lambda\pi R^2 \simeq 1.64$. The plot reveals that the packet speed is significantly larger than the node speed for a wide range of node degrees for which the network is not connected (compare Fig. 4.1 with Fig. 3.1).

4.4 Delay and Throughput of DTFR

We have defined D_{od} as the random distance between an origin and a destination of a packet. Let R_X be the distance covered by the destination during the time it takes the packet to reach the destination. Clearly,

$$\frac{R_X}{v_0} = \frac{D_{od} + R_X}{v_p} \Rightarrow R_X = D_{od}\frac{v_0}{v_p - v_0}.$$

Let D^{DTFR} be the delay in the delivery of the packet. It follows that

$$D^{\text{DTFR}} = \frac{R_X}{v_0} = \frac{D_{od}}{v_p - v_0} \Rightarrow E(D^{\text{DTFR}}) = \frac{E(D_{od})}{v_p - v_0}. \tag{4.6}$$

Next, we calculate approximately the maximum throughput per node pair T^{DTFR} that DTFR can support. To this end, we first calculate the average of the total cost C^{DTFR} (in square meters) for the delivery of a packet to the destination. C^{DTFR} is comprised of two terms: the cost C_A up until the delivery of the packet to the FC, and the cost C_B due to the transmissions taking place during the explosion and spread phases. The first term equals c_pD_{od}. To calculate the second term, we first assume that the distance between the FC and the FEs is set to R_X, i.e., to the minimum that guarantees delivery of the packet given that nodes move with speed v_0 and the packet moves with speed v_p. Therefore, the cost of transmitting a single replica is c_pR_X. The cost of transmitting L replicas *consecutively* is Lc_pR_X. However, the replicas are not transmitted consecutively, but simultaneously. Therefore, many transmissions, particularly at locations close to the FC, can be combined, as they involve replicas of the same packet being transmitted from the same transmitter to the same receiver. Therefore, a more accurate approximation for the total cost during the spread and explosion phases is the total area that the replicas must cover during these phases, i.e., πR_X^2. Combining the two terms,

$$C^{\text{DTFR}} = C_A + C_B = c_pD_{od} + \pi R_X^2 = c_pD_{od} + \pi D_{od}^2\frac{v_0^2}{(v_p - v_0)^2}.$$

Taking expectations,

$$E(C^{\mathrm{DTFR}}) = c_p E(D_{od}) + \pi E(D_{od}^2) \frac{v_0^2}{(v_p - v_0)^2}. \tag{4.7}$$

This average cost represents the average aggregate area of the footprints needed for the transmission of a single packet from the source to the destination.

To convert the average cost to the maximum throughput per node pair T^{DTFR}, we proceed as follows: as the node density is λ, each node is allocated on the average an area $\frac{1}{\lambda}$. As the area required for the transmission of a packet is $E(C^{\mathrm{DTFR}})$, it follows that each node can occupy the channel for a percentage of time equal to $\left(\frac{1}{\lambda}\right) / E(C^{\mathrm{DTFR}})$, therefore

$$T^{\mathrm{DTFR}} = \frac{r_D}{\lambda E(C^{\mathrm{DTFR}})} = \frac{r_D}{\lambda} \left[c_p E(D_{od}) + \pi E(D_{od}^2) \frac{v_0^2}{(v_p - v_0)^2} \right]^{-1}. \tag{4.8}$$

4.5 Delay and Throughput of Bethlehem Routing

Bethlehem Routing operates similarly to DTFR, with the exception that the packet travels toward the destination, and never enters the explosion phase. Therefore, the trajectory that the packet follows is not a straight line. Finding its precise average length goes beyond the scope of this work. Noting that if $D_{od} \gg R_X$ then this average length is approximately equal to $E(D_{od})$, we approximate it as $E(D_{od})$. It follows that

$$E(D^{\mathrm{BR}}) = \frac{E(D_{od})}{v_p}, \quad E(C^{\mathrm{BR}}) = c_p E(D_{od}), \quad T^{\mathrm{BR}} = \frac{r_D}{\lambda c_p E(D_{od})}.$$

4.6 Discussion

The effects that the basic environmental parameters have on the average delay and throughput of DTFR, as these were evaluated in the simulations, are consistent with the basic results of the analysis, i.e., (4.6) and (4.8). Indeed, (4.8) predicts that the average throughput will decrease when the size of the network, and hence $E(D_{od})$ and $E(D_{od}^2)$, increase. These predictions are consistent with Fig. 3.3. Equation (4.8) shows that increasing the transmission range R increases the average throughput, because the second term in the brackets diminishes, due to the increase in the packet speed; this is verified by Fig. 3.4. Finally, (4.6) predicts that the delay decreases as the node speed increases, and this is consistent with Fig. 3.5 which shows that, as the speed increases, the delivery rate increases, because more nodes arrive at their destinations before the TTL expires.

Furthermore, the analysis, in particular (4.5) and Fig. 4.1, shows that there is a wide range of node degrees for which the network is not connected, and so traditional routing protocols cannot be used, but for which GLF ensures that the packet travels to the destination with a speed much larger than the node speed, and so can catch up with the destination quickly, while the cost of the *Spread Phase* remains manageable.

This last finding explains why the performance of DTFR is superior to that of Spray and Wait. In more detail, for some values of the distance between the source and the destination, the maximum acceptable delay for the delivery of the packets, and the velocity of the nodes, the distance between the source and the destination cannot be covered by transport before the packet expires. Spray and Wait spreads a number of copies in the area in which the nodes move, and one of the relays has to travel near the destination and transfer the packet there. A part of the distance between the source and the destination has to be covered by transport. A part of the distance between the source and the destination is covered by transmission, but if the packet travels distance, say x, by transmission, in one direction, due to symmetry the packet covers distance x by transmission in all directions, and this has a large cost, at least πx^2, regardless if the distance x is covered using many small or a few large hops. If the distance between the source and the destination is s_1, the delay in the above case is at least $(s_1 - x)/v_0$. This bound on the delay-cost tradeoff of Spray and Wait is very weak, but it gives an intuition as to why Spray and Wait does not perform well for certain environment parameters. Spray and Wait is designed to give a number of copies to relays in order to maximize the probability that a relay goes near the destination and delivers the packet, not to cover distance by transmission. In the above case DTFR has delay $s_1/(v_p - v_0)$ and cost $c_p s_1 + \pi s_1^2 \frac{v_0^2}{(v_p - v_0)^2}$. Assume that the delay constraint is equal to the delay that DTFR can achieve. If we set $s_1/(v_p - v_0) = (s_1 - x)/v_0$ it follows that $x = s_1(v_p - 2v_0)/(v_p - v_0)$. In this case the cost of Spray and Wait is at least $\pi s_1^2 (v_p - 2v_0)^2/(v_p - v_0)^2$, while the cost of DTFR is less than this, for a number of values of the node degree. (For example $\lambda \pi R^2 = 4.7$, which gives $v_p = 13.75 v_0$ and $c_p = 0.4294R$.) To conclude, DTFR achieves a delay cost tradeoff that is superior to that of Spray and Wait for a wide range of settings.

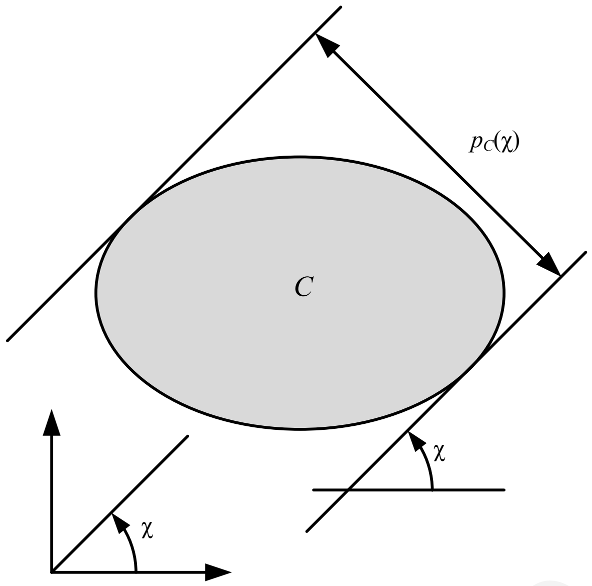


Figure 4.2: The definition of the projection function $p_C(\chi)$ of a set C.

4.7 Statistics of the First Stage Given that the Forwarding Area is Empty

Let C be a non empty, convex, and compact subset of $\mathbb{R}^2$. We define the **projection function** $p_C : \mathbb{R} \to (0, \infty)$ as follows: if $\chi \in \mathbb{R}$, then $p_C(\chi) > 0$ is the minimum width that an infinite closed strip, inclined with respect to the x axis by an angle χ, can have and at the same time have C as its subset, as depicted in Fig. 4.2. Observe that $p_C(\chi) = p_C(\chi + \pi)$ for all $\chi \in \mathbb{R}$. For the semicircular forwarding region F of our analysis, straightforward geometry shows that

$$p_F(\chi) = R(1 + |\cos\chi|). \tag{4.9}$$

Lemma 1. *Let C be a non empty, convex, and compact subset of $\mathbb{R}^2$ with projection function $p_C(\chi)$. Let $\mathbb{R}^2$ be uniformly covered by nodes distributed, at time 0, according to a spatial Poisson process of density λ, all moving with a common speed $\boldsymbol{v}$ of magnitude v_0 and direction with respect to the x axis equal to χ. Then, nodes enter C through its boundary according to a Poisson process with rate equal to $\gamma(\chi) = v_0 p_C(\chi)\lambda$.*

Proof. Consider Fig. 4.3 and focus on a time instant t_0 and a time interval $[t_0, t_0 + T]$. The nodes that enter the lightly shaded set C during this interval are exactly those that at time t_0 are on the darker shaded set D, whose nonlinear boundaries are parallel

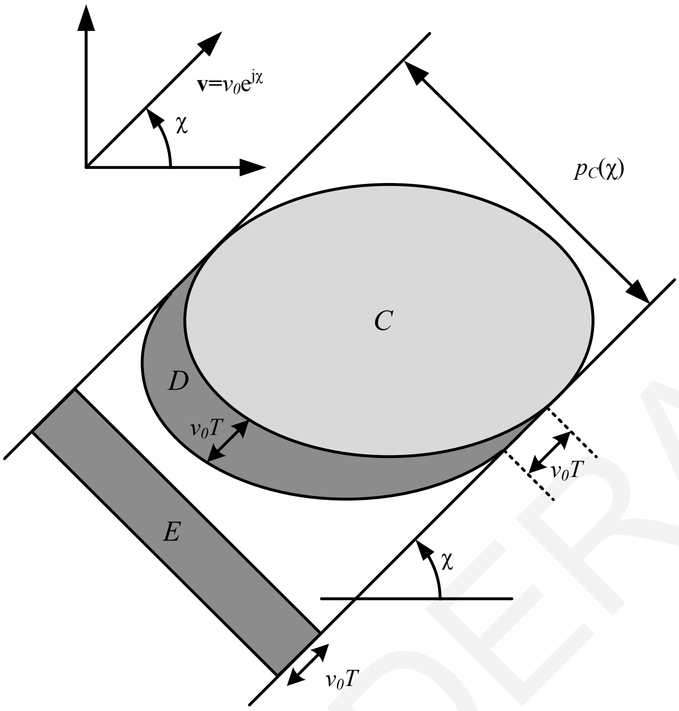


Figure 4.3: Proof of Lemma 1.

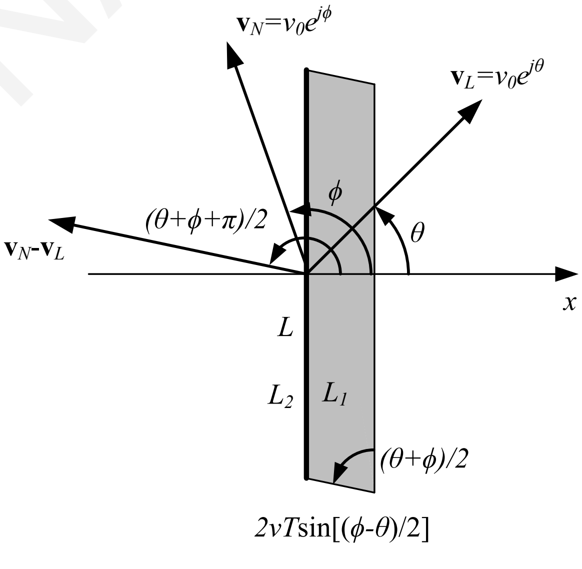


Figure 4.4: The setting of Lemma 2.

to each other and are at a distance of v_0T from each other. Since the nodes follow a spatial Poisson distribution at time $t = 0$, they will also follow a spatial Poisson distribution, with the same density, at time t_0. Therefore, the nodes within D are Poisson distributed, with mean equal to λ multiplied by the area of D. Note, however, that this area equals the area $p_C(\chi)v_0T$ covered by the rectangle E. Therefore, the average number of nodes entering C in the time interval $[t_0, t_0 + T]$ is $v_0p_C(\chi)\lambda T$. Finally, also note that the numbers of nodes arriving at non overlapping time intervals are independent, because they are due to the existence of nodes, at time $t = 0$, at non overlapping subsets of $\mathbb{R}^2$. It follows that the arrival process of nodes in C satisfies the definition of a Poisson process [109] with rate $\gamma(\chi) = v_0p_C(\chi)\lambda$. □

Lemma 2. *Let L be a linear segment of length l, parallel to the y axis. Let L move with a velocity $\mathbf{v}_L$ of magnitude v_0, forming an angle $\theta \in [0, \pi]$ with the positive x axis. Let $\mathbb{R}^2$ be covered with nodes placed, at time $t = 0$, according to a spatial Poisson process of density λ, and moving with velocity vectors of magnitude v_0 and directions uniformly distributed in $[0, 2\pi]$, independently of each other. Then, the process with which nodes cross L through its side L_1 looking at the positive x axis is Poisson, with rate*

$$\gamma_L(\theta) = \frac{\lambda l v_0}{\pi}\left[\sin\theta + (\pi - \theta)\cos\theta\right]. \tag{4.10}$$

Proof. Assume, for now, that all nodes have the same velocity vector $\mathbf{v}_N$, and travel toward the same angle $\phi \in [0, 2\pi]$. Using phasor notation, $\mathbf{v}_N = v_0e^{j\phi}$ and $\mathbf{v}_L = v_0e^{j\theta}$. The setting appears in Fig. 4.4. We will specify the process with which nodes cross L (entering from side L_1) in this case.

Observe, first, that we must have $\phi \in [\theta, 2\pi - \theta]$, otherwise the nodes arrive at L from the other side, L_2. Also observe that the relative velocity of the nodes with respect to L is

$$\begin{aligned}\mathbf{v}_N - \mathbf{v}_L &= v_0e^{j\phi} - v_0e^{j\theta} = v_0e^{j\phi} + v_0e^{j(\theta+\pi)}\\ &= 2v_0\cos\left(\frac{\pi - \phi + \theta}{2}\right)e^{j\left(\frac{\phi+\theta+\pi}{2}\right)} = 2v_0\sin\left(\frac{\phi - \theta}{2}\right)e^{j\left(\frac{\phi+\theta+\pi}{2}\right)}.\end{aligned}$$

Let us move to the coordinate system where L remains stationary and parallel to the y axis. Consider a time interval $[t_0, t_0 + T]$. The nodes crossing L from L_1 are exactly those that at t_0 are in the shaded rectangle of Fig. 4.4. The number of those is a Poisson random variable with average λ multiplied by the area of the rectangle $\left(2v_0T\sin\left(\frac{\phi-\theta}{2}\right)\right) \times l \times \sin\left(\frac{\phi+\theta}{2}\right)$. To conclude, the nodes arriving in the interval $[t_0, t_0+$

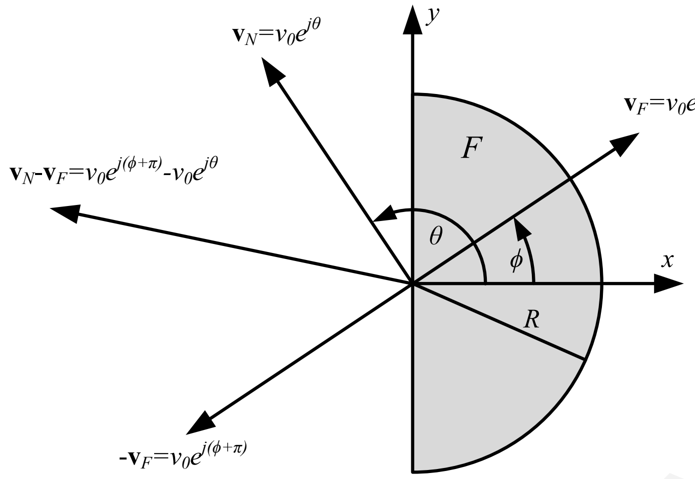

$\mathbf{v}_N = v_0 e^{j\theta}$
y
$\mathbf{v}_F = v_0 e^{j\phi}$
F
$\mathbf{v}_N - \mathbf{v}_F = v_0 e^{j(\phi+\pi)} - v_0 e^{j\theta}$
θ
ϕ
x
R
$-\mathbf{v}_F = v_0 e^{j(\phi+\pi)}$

Figure 4.5: The semicircle used in the proof of Equations (4.4).

$T]$ are Poisson distributed with average $2v_0\lambda l \sin\left(\frac{\phi-\theta}{2}\right)\sin\left(\frac{\phi+\theta}{2}\right)T$. Furthermore, the arrivals at non overlapping time intervals are independent, because they are caused by the existence of nodes, during time t_0, at non overlapping subsets of $\mathbb{R}^2$. It follows [109] that the arrival process is Poisson with rate $2v_0\lambda l \sin\left(\frac{\phi-\theta}{2}\right)\sin\left(\frac{\phi+\theta}{2}\right)$.

However, nodes do not have a fixed direction ϕ, but rather the direction of each is uniformly distributed and independent of the directions of the rest. It follows from the previous case that the arrivals of nodes with direction in the incremental range $[\phi, \phi + d\phi]$ form a Poisson process with incremental rate $2v_0\frac{\lambda d\phi}{2\pi} l \sin\left(\frac{\phi-\theta}{2}\right)\sin\left(\frac{\phi+\theta}{2}\right)$. As the summation of multiple Poisson processes is a Poisson process with rate equal to the sum of the rates, it follows that the arrival process from all angles is a Poisson process with rate given by the integral

$$\gamma_L(\theta) = \int_{\theta}^{2\pi-\theta} \frac{\lambda l v_0}{\pi} \sin\left(\frac{\phi-\theta}{2}\right)\sin\left(\frac{\phi+\theta}{2}\right) d\phi.$$

Calculating the integral, we arrive at (4.10). □

Assume, for now, that F is moving with a constant velocity $\mathbf{v}_F$ of magnitude v_0 and direction (with respect to the x axis) equal to ϕ, where $\phi \in [-\pi, \pi]$. Let also $\mathbb{R}^2$ be filled with nodes, all with a common velocity $\mathbf{v}_N$ of magnitude v_0 and direction, with respect to the x-axis, equal to θ, where $\theta \in [-\pi, \pi]$. At time $t = 0$, nodes are placed on $\mathbb{R}^2$ according to a spatial Poisson process with density λ. The setting appears in Fig. 4.5. In this setting, the arrival process of nodes in F through its boundary is a

Poisson process with rate equal to

$$\gamma(\theta,\phi) = 2v_0R\lambda\left|\sin\left(\frac{\theta-\phi}{2}\right)\right|\left(1+\left|\sin\left(\frac{\phi+\theta}{2}\right)\right|\right). \qquad (4.11)$$

Indeed, in phasor notation, $\mathbf{v}_N = v_0e^{j\theta}$ and $\mathbf{v}_F = v_0e^{j\phi}$, and the relative velocity of the nodes as perceived by F is

$$\mathbf{v}_N - \mathbf{v}_F = v_0e^{j\theta} + v_0e^{j(\phi+\pi)} = 2v_0\sin\left(\frac{\theta-\phi}{2}\right)e^{j\frac{\phi+\theta+\pi}{2}}.$$

Therefore, the magnitude of the relative velocity is $2v_0\left|\sin\left(\frac{\theta-\phi}{2}\right)\right|$, and the angle of incidence of the nodes on F is either $\frac{\phi+\theta+\pi}{2}$ or $\frac{\phi+\theta+\pi}{2}-\pi=\frac{\phi+\theta-\pi}{2}$, depending on the sign of $\sin\left(\frac{\theta-\phi}{2}\right)$. As the two possibilities for the angle of incidence differ by π, they give the same value of the projection function. It follows from Lemma 1 that

$$\gamma(\theta,\phi) = 2\lambda v_0\left|\sin\left(\frac{\theta-\phi}{2}\right)\right|p_F\left(\frac{\phi+\theta+\pi}{2}\right),$$

and substituting $p_F(\cdot)$ from (4.9) the result (4.11) follows.

We now modify the setting to assume that each node is moving toward a direction Θ that is uniformly chosen in $[-\pi,\pi]$, independently of the directions of all other nodes. In this setting, the arrival process is again a Poisson process with rate equal to

$$\gamma(\phi) = \frac{v_0R\lambda}{\pi}\Big[4+(\pi-2|\phi|)\cos\phi+2\sin|\phi|\Big]. \qquad (4.12)$$

This is due to the facts that, firstly, each incremental range of node velocity angles, $[\theta,\theta+d\theta]$ creates a Poisson arrival process and, secondly, the process consisting of the arrivals of any number of Poisson processes is still a Poisson process, with a rate equal to the sum of the incremental rates, i.e., the integral

$$\gamma(\phi) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\gamma(\theta,\phi)\,d\theta = \frac{v_0R\lambda}{\pi}\int_{-\pi}^{\pi}\left|\sin\left(\frac{\theta-\phi}{2}\right)\right|\left(1+\left|\sin\left(\frac{\phi+\theta}{2}\right)\right|\right)d\theta. \quad (4.13)$$

After straightforward calculations, (4.12) follows.

Let us now move to the setting of Equations (4.4): Let F be empty of nodes and at time $t=0$ centered at the origin. Let the directions of both F and all nodes be randomly and uniformly distributed in $[-\pi,\pi]$, and let Φ be the random direction of F. In this setting, the arrival process of nodes at the boundary of F is a conditional Poisson process [109]. Indeed, if we condition on Φ, the node arrival process is Poisson, with rate given by (4.12). Observe that conditioning on M does not affect the statistics of new arrivals, as these are coming from regions that do not overlap with the interior of F at $t=0$.

To calculate $E(D_1|M)$, $E(X_1|M)$, and $E(C_1|M)$, we condition on Φ. We start with $E(D_1|M)$, noting that

$$\begin{aligned} E(D_1|M) = E(E(D_1|\Phi, M)) = \frac{1}{2\pi}\int_{-\pi}^{\pi} E(D_1|\Phi = \phi, M)\, d\phi = \\ = \frac{1}{\pi}\int_{0}^{\pi} E(D_1|\Phi = \phi, M)\, d\phi, \end{aligned}$$

where in the last equation we used the fact that, due to symmetry, the function $E(D_1|\Phi = \phi, M)$ is even. As discussed, conditioned on $\Phi = \phi$, the node arrival process is a Poisson process with rate $\gamma(\phi)$ given in (4.12), and hence $E(D_1|\Phi = \phi, M) = \frac{1}{\gamma(\phi)}$. Combining everything, it follows that

$$E(D_1|M) = \frac{I_1}{v_0 R\lambda}, \quad I_1 \triangleq \int_0^{\pi} \frac{d\phi}{4 + (\pi - 2\phi)\cos\phi + 2\sin\phi} \simeq 0.4817.$$

Next, we calculate $E(X_1|M)$, noting that $E(X_1|M) = E(X_T|M) + E(X_A|M)$. We first note that

$$\begin{aligned} E(X_A|M) = E(E(X_A|\Phi, M)) = \frac{1}{2\pi}\int_{-\pi}^{\pi} E(X_A|\Phi = \phi, M)\, d\phi = \\ = \frac{1}{\pi}\int_{0}^{\pi} E(X_A|\Phi = \phi, M)\, d\phi, \end{aligned}$$

where the last equality is due to symmetry. Now observe that $\gamma(\phi) = \gamma(\pi - \phi)$ for all $\phi \in [0, \pi]$. It follows that $E(X_A|\Phi = \phi, M) = -E(X_A|\Phi = \pi - \phi, M)$, and therefore $E(X_A|M)$ is zero. It follows that $E(X_1|M) = E(X_T|M)$.

Furthermore,

$$\begin{aligned} E(X_T|M) = E(E(X_T|\Phi, M)) = \frac{1}{2\pi}\int_{-\pi}^{\pi} E(X_T|\Phi = \phi, M)\, d\phi = \\ = \frac{1}{\pi}\int_{0}^{\pi} E(X_T|\Phi = \phi, M)\, d\phi, \end{aligned}$$

where in the last equation we used symmetry. As discussed, conditioned on $\Phi = \phi$, the node arrival process is a Poisson process with rate $\gamma(\phi)$ given in (4.12). This Poisson process can be broken down to a set of independent, incremental Poisson processes, each one corresponding to the node arrivals in the semicircle through an incremental arc $[\chi, \chi + d\chi]$ along its circumference, where $\chi \in [-\pi/2, \pi/2]$, and a last one, independent of the rest, corresponding to the arrivals through the linear part. It follows that the probability that there is an arrival through such an arc $\chi \in [-\pi/2, \pi/2]$ is equal to the incremental rate of arrivals there, $d\gamma(\chi)$ over the aggregate arrival rate $\gamma(\phi)$. Therefore, and noting that arrivals through the linear part do not contribute to $E(X_T|\Phi = \phi, M)$, we have

$$E(X_T|\Phi = \phi, M) = \int_{-\pi/2}^{\pi/2} R\cos\chi \frac{d\gamma(\chi)}{\gamma(\phi)}.$$

Observe, however, that $d\gamma(\chi)$ is equal to the arrival rate through a linear segment of length $Rd\chi$ and moving towards an angle $|\chi - \phi|$, with respect to its vertical. Therefore, Lemma 2 applies. Taking into account that $|\chi - \phi|$ might be greater than π, it follows that

$$d\gamma(\chi) = \frac{R\lambda v_0}{\pi} h(|\chi - \phi|)d\chi, \quad \text{where} \quad h(x) \triangleq \begin{cases} \sin x + (\pi - x)\cos x, & 0 \leq x \leq \pi, \\ h(2\pi - x), & \pi < x \leq 2\pi. \end{cases}$$

Putting everything together, it follows that

$$E(X_1|M) = I_2 R, \quad I_2 \triangleq \int_0^{\pi} \frac{\frac{1}{\pi}\int_{-\pi/2}^{\pi/2} h(|\chi - \phi|)\cos\chi\, d\chi}{4 + (\pi - 2\phi)\cos\phi + 2\sin\phi}\, d\phi \simeq 0.3894.$$

To calculate $E(C_1|M) = \pi E(X_T^2 + Y_T^2|M)$, note that

$$E(X_T^2 + Y_T^2|M) = E(E(X_T^2 + Y_T^2|\Phi, M)) = \frac{1}{\pi}\int_0^{\pi} E(X_T^2 + Y_T^2|\Phi = \phi, M)\, d\phi,$$

The value of $X_T^2 + Y_T^2$ depends on whether the arriving node comes through the linear segment of the semicircle or not. The probability $P(S)$ of the event S that the arrival will be through the linear segment equals the rate of arrivals through the linear segment, as specified by Lemma 2, over the total rate of arrivals $\gamma(\phi)$, as specified by (4.12). Therefore,

$$P(S) = \frac{\frac{2\lambda R v_0}{\pi}\left[\sin(\pi - \phi) + \phi\cos(\pi - \phi)\right]}{\frac{\lambda R v_0}{\pi}\left[4 + (\pi - 2\phi)\cos\phi + 2\sin\phi\right]} = \frac{2(\sin\phi - \phi\cos\phi)}{4 + (\pi - 2\phi)\cos\phi + 2\sin\phi}.$$

Having $P(S)$, we note that

$$E(X_T^2 + Y_T^2|\Phi = \phi, M) = [1 - P(S)]R^2 + P(S)\int_{-R}^{R} \frac{y^2\, dy}{2R} = R^2\left(1 - \frac{2}{3}P(S)\right).$$

Combining everything, it follows that

$$E(C_1|M) = I_3 R^2, \quad I_3 \triangleq \int_0^{\pi} \left(1 - \frac{2}{3} \times \frac{2(\sin\phi - \phi\cos\phi)}{4 + (\pi - 2\phi)\cos\phi + 2\sin\phi}\right) d\phi \simeq 2.3317.$$

Chapter 5

Alternative Variants of GLF

In this chapter[1] we propose different variants of Greedy Lazy Forwarding (GLF) and evaluate their performance using simulation and analysis. The network model and definitions we employ partially overlap with those of Chapter 4, but for reasons of clarity we present them fully in this chapter as well.

5.1 Network Model and the Delay-Cost Plane

Network Model: At time $t = 0$, an infinite number of nodes are placed on the infinite plane $\mathbb{R}^2$ according to a Poisson distribution with density λ. Beginning at $t = 0$, each node moves with a fixed velocity vector, of magnitude v_n common for all nodes, and a direction chosen uniformly, and independently of the rest. It is straightforward to show that, under these assumptions, nodes are Poisson distributed with density λ for all $t \geq 0$ (cf. Section 1.3.3 of [110]). Furthermore, by a standard thinning argument, the nodes whose direction of travel forms an angle with the x axis within the interval $[\chi, \chi + \Delta\chi]$ are also Poisson distributed with density $\lambda\frac{\Delta\chi}{2\pi}$.

Two nodes separated by a distance d can exchange a packet at a cost $c(d) = d^2$. For simplicity, we do not specify an upper bound on the distance that a transmission can cover. Finally, packet exchanges are instantaneous, errorless, and not subject to interference and media access constraints.

We focus on a specific packet that must be delivered at a destination located at an infinite distance which, for simplicity, is taken to be in the direction of the positive x-axis. The packet can travel to its destination through a combination of physical

[1] This work also appears in [34].

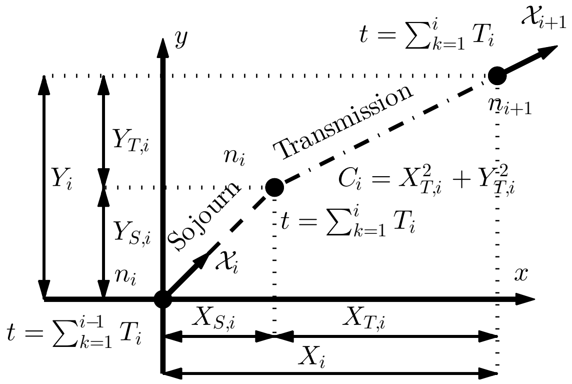


Figure 5.1: The i-th stage of a journey.

transports (involving only delay) and wireless transmissions (involving only cost). Note that there is a clear tradeoff between delay and cost: the more a packet moves towards its destination using wireless transmissions instead of physical transports, the higher the cost, but the lower the delay, and vice versa.

We assume that the packet is following a **forwarding rule** under which the resulting **journey** is comprised of **stages**, which we index by $i = 1, 2, \ldots$. Each stage i consists of two parts: a **sojourn** at the buffer of a node n_i, that lasts for a **sojourn time** T_i, and a wireless **transmission**, from node n_i to node n_{i+1}. Observe that stage i lasts from time $\sum_{k=1}^{i-1} T_k$ until time $\sum_{k=1}^{i} T_k$. Let $\mathcal{X}_i$ be the direction of travel of node n_i. Let $X_{S,i}$ and $Y_{S,i}$ be the change in the x and y coordinates of the packet due to its i-th sojourn. Let $X_{T,i}$ and $Y_{T,i}$ be the change in the x and y coordinates of the packet due to its i-th transmission. Let $C_i = X_{T,i}^2 + Y_{T,i}^2$ be the wireless transmission cost of the i-th stage. We also define the net total changes in the x and y coordinates during the i-th stage as $X_i = X_{S,i} + X_{T,i}$ and $Y_i = Y_{S,i} + Y_{T,i}$ respectively. These definitions are summarized in Fig. 5.1.

Note that some stages will consist only of a transmission, i.e., a node will retransmit a packet the moment it receives it, and therefore for these stages $X_{S,i} = Y_{S,i} = T_i = 0$.

Normalized Delay and Cost: We define the **(normalized packet) delay** D_p and the **(normalized packet) cost** C_p of the forwarding rule as the following limits, provided they exist:

$$D_p \triangleq \lim_{n\to\infty} \frac{\sum_{i=1}^{n} T_i}{\sum_{i=1}^{n}(X_{S,i} + X_{T,i})}, \tag{5.1}$$

$$C_p \triangleq \lim_{n\to\infty} \frac{\sum_{i=1}^{n} C_i}{\sum_{i=1}^{n}(X_{S,i} + X_{T,i})} = \lim_{n\to\infty} \frac{\sum_{i=1}^{n} X_{T,i}^2 + Y_{T,i}^2}{\sum_{i=1}^{n}(X_{S,i} + X_{T,i})}. \tag{5.2}$$

The normalized delay D_p is the limit, as $n \to \infty$, of the total time it takes for n

stages to complete divided by the progress towards the destination during these stages, whereas the normalized cost is the limit, as $n \to \infty$, of the total cost incurred during the first n stages divided by the progress made during these stages.

A few comments are in order. First of all, the two limits may not exist for some forwarding rules, for example forwarding rules that vary with time. However, we expect that the limits will exist for most simple, time invariant rules. Establishing general conditions for their existence is outside the scope of this work. Secondly, the y-coordinates $Y_{S,i}$ and $Y_{T,i}$ do not appear in the denominator of the two fractions that measures progress towards the destination. This is because offsets in the direction of the y-axis do not have an effect on the progress made. Thirdly, observe that the normalized delay is simply the inverse of the average speed, which might perhaps be a more intuitive figure of merit. We opt to use normalized delay for reasons of mathematical convenience and uniformity.

The pair (D_p, C_p) describes the efficiency of the forwarding rule, and our primary task in this chapter is to calculate it approximately for a few forwarding rules, by both analysis and simulation. The 'holy grail' problem coming out of this chapter is finding the Pareto optimal combinations (D_p, C_p) and the forwarding rules that achieve them. Based on the preliminary investigation conducted here, and related works [107, 111], we believe that this task is formidable, and we leave it for future work.

5.2 First Forwarding Rule

Whenever a packet is created at or relayed to a node A, node A scans for other nodes that could act as relays for the packet within an area termed the **forwarding region** $\mathcal{F}$. $\mathcal{F}$ is a subset of $\mathbb{R}^2$ that is defined relative to node A, and moving with it. We assume that it is closed, bounded, and convex. For the forwarding rule to perform well, we expect that $\mathcal{F}$ must be placed, with respect to the current holder, towards the destination of the packet. We assume, therefore, that A is outside $\mathcal{F}$, or at most on its boundary.

Node A surveys the nodes within $\mathcal{F}$ and takes the following action:

1. If there are nodes in $\mathcal{F}$ whose directions of travel form an angle with the positive x-axis that is within the limits $[-\chi_m, \chi_m]$, where $\chi_m \in (0, \pi]$ is a parameter termed the **maximum (angular) deviation**, then node A immediately transmits the

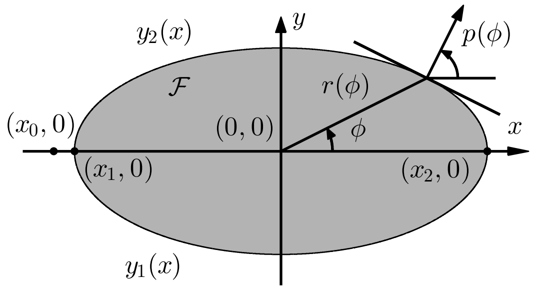


Figure 5.2: Parametrization of the forwarding region $\mathcal{F}$.

packet to that node among them with the smallest quotient C/x of transmission cost C over progress x.

2. If there is no node in $\mathcal{F}$ whose direction of travel falls within the limits $[-\chi_m, \chi_m]$, then A will keep the packet until one such node B appears on the boundary of the forwarding region. A will then transmit the packet to node B and the process will start over.

The motivation for restraining the direction of travel of a receiving node in the range $[-\chi_m, \chi_m]$ is clear: we want to avoid using nodes whose direction of travel is not sufficiently close to the direction of the destination. The motivation for selecting the node with the smallest C/x quotient is also clear: as we want to minimize the long term quotient of total cost over total progress, it makes sense to greedily minimize it on a stage by stage basis.

Different instances of this forwarding rule differ on the choices of $\mathcal{F}$ and χ_m. We expect to be able to trade off D_p with C_p by tuning these two parameters. For example, we expect that as the area of $\mathcal{F}$ becomes larger, D_p becomes smaller but C_p becomes larger.

We describe the forwarding region $\mathcal{F}$ in terms of a coordinate system whose origin $(0, 0)$ lies in the interior of $\mathcal{F}$, and such that the current packet holder A is located in the position $(x_0, 0)$. We also use two functions $r(\phi)$, $p(\phi)$. The function $r(\phi)$ is such that $(r(\phi), \phi)$ traces the boundary of $\mathcal{F}$, in polar coordinates, whereas $p(\phi)$ is the angle formed between the positive x-axis and the vector that is perpendicular to the boundary of $\mathcal{F}$ at the location $(r(\phi), \phi)$. We assume that $r(\phi)$ is continuous and $p(\phi)$ is increasing in ϕ (but not necessarily continuous). Although it would suffice to define $r(\phi)$ and $p(\phi)$ for $\phi \in [-\pi, \pi]$, for mathematical convenience we define them for $\phi \in \mathbb{R}$, therefore $r(\phi)$ is periodic (with period 2π). We also require that $p(\phi + 2\pi) = p(\phi) + 2\pi$

for all $\phi \in \mathbb{R}$. Also, let x_1 and x_2, with $x_1 < x_2$, be the locations on the x-axis where $\mathcal{F}$ intersects it. Note that we already assumed that $x_0 \leq x_1$. Finally, let $y_1(x)$ and $y_2(x)$ be the functions describing the boundary of $\mathcal{F}$ below and above the x-axis respectively, in Cartesian coordinates. See Fig. 5.2.

5.3 Statistics of the First Stage

As a preliminary to the approximate calculation of the (D_p, C_p) pairs achievable by the forwarding rule of Section 5.2, which will be given in Section 5.4, in this section we focus on the *first* stage, i.e., the period between the creation of the packet and its first transmission to a relay. So let a packet be created at time $t = 0$, and let A be its source node. According to our mobility model of Section 5.1, A, and hence also the forwarding region $\mathcal{F}$, are moving with speed v_n towards a random direction $\mathcal{X}_1$ which is uniformly distributed in $[-\pi, \pi]$.

We will calculate the mean values of the random variables $X_{S,1}$, $X_{T,1}$, C_1, T_1 conditioned on the event $\mathcal{X}_1 = \theta$, for $\theta \in [-\pi, \pi]$. We will calculate these by further conditioning on the events M and M', where M is the event that the forwarding region $\mathcal{F}$ is initially, i.e., at time $t = 0$, empty of nodes with a direction of travel within the $[-\chi_m, \chi_m]$ interval. We have

$$E(*|\mathcal{X}_1 = \theta) = E(*|\mathcal{X}_1 = \theta, M)P(M) + E(*|\mathcal{X}_1 = \theta, M')(1 - P(M)), \tag{5.3}$$

where the asterisk $*$ can be any of $X_{S,1}$, $X_{T,1}$, C_1, T_1. Observe that

$$P(M) = \exp\left(-\lambda'|\mathcal{F}|\right), \tag{5.4}$$

where we define $\lambda' \triangleq \frac{\chi_m}{\pi}\lambda$ and where $|\mathcal{F}|$ is the area of the forwarding region $\mathcal{F}$.

We will also calculate the conditional pdf $f_{\mathcal{X}_2|\mathcal{X}_1=\theta}(\chi)$ of the direction $\mathcal{X}_2$ of the *second* node to receive the packet, also using conditioning on the event M, as follows:

$$f_{\mathcal{X}_2|\mathcal{X}_1=\theta}(\chi) = f_{\mathcal{X}_2|\mathcal{X}_1=\theta,M}(\chi)P(M) + f_{\mathcal{X}_2|\mathcal{X}_1=\theta,M'}(\chi)(1 - P(M)). \tag{5.5}$$

The aforementioned quantities that are conditional on the event M' are derived in Section 5.3.1, and those conditional on the event M are derived in Section 5.3.2.

5.3.1 $\mathcal{F}$ is Initially Not Empty

In this case, we first note that the direction of travel of the first relay is uniformly distributed in the range $[-\chi_m, \chi_m]$, and independent of the value of $\mathcal{X}_1$, therefore

$$f_{\mathcal{X}_2|\mathcal{X}_1=\theta,M'}(\chi) = \begin{cases} \frac{1}{2\chi_m}, & |\chi| \leq |\chi_m|, \\ 0, & |\chi_m| < |\chi| \leq \pi. \end{cases} \tag{5.6}$$

Furthermore, in the event M', $X_{S,1} = T_1 = 0$, therefore we have

$$E(X_{S,1}|\mathcal{X}_1 = \theta, M') = E(T_1|\mathcal{X}_1 = \theta, M') = 0. \tag{5.7}$$

To calculate the remaining two expectations needed, i.e., $E(X_{T,1}|\mathcal{X}_1 = \theta, M')$ and $E(C_1|\mathcal{X}_1 = \theta, M')$, we first observe that $X_{T,1}$ and $Y_{T,1}$, and hence also $C_1 = X_{T,1}^2 + Y_{T,1}^2$, are independent of $\mathcal{X}_1 = \theta$. This significantly simplifies the calculations.

We will first consider the statistics of the quotient

$$Q = \frac{C_1}{X_{T,1}} = \frac{X_{T,1}^2 + Y_{T,1}^2}{X_{T,1}}.$$

Let $A(q)$ be the area of the subset of the forwarding region $\mathcal{F}$ for which $Q \leq q$. Note that the locus of the points on the plane for which $Q \leq q$ is a disk with center at $\left(\frac{q}{2}, 0\right)$ and radius $\frac{q}{2}$. Therefore, $A(q)$ is increasing with q, with $A(0) = 0$ and $A(\infty) = \lim_{q\to\infty} A(q) = |\mathcal{F}|$. Note that

$$P\left(q \leq Q \leq q + dq|M'\right) = \frac{P\left(q \leq Q \leq q + dq, M'\right)}{P(M')} = \frac{\exp\left(-\lambda' A(q)\right)\lambda' A'(q)dq}{1 - \exp\left(-\lambda'|\mathcal{F}|\right)},$$

where $A'(q)$ is the derivative of $A(q)$. It follows that the conditional distribution of Q is

$$f_Q(q) = \frac{\lambda' A'(q)}{1 - \exp\left(-\lambda'|\mathcal{F}|\right)} \exp\left(-\lambda' A(q)\right).$$

To find the conditional expectations $E[X_{T,1}|\mathcal{X}_1 = \theta, M', Q = q]$ and $E[C_{T,1}|\mathcal{X}_1 = \theta, M', Q = q]$ we consider the locus of the points within $\mathcal{F}$ for which we have $\frac{x^2+y^2}{x} = q$, which is the intersection of $\mathcal{F}$ with a circle of radius $\frac{q}{2}$ centered at $\left(\frac{q}{2}, 0\right)$. Let $L(q)$ the length of the locus, and let $x(s)$ and $y(s)$ be the parametrized coordinates of the locus where s is the length of the locus, with $s \in [0, L(q)]$. Therefore,

$$E[X_{T,1}|\mathcal{X}_1 = \theta, M', Q = q] = \int_0^{L(q)} \frac{x(s)}{L(q)} ds,$$

$$E[C_1|\mathcal{X}_1 = \theta, M', Q = q] = \int_0^{L(q)} \frac{x^2(s) + y^2(s)}{L(q)} ds.$$

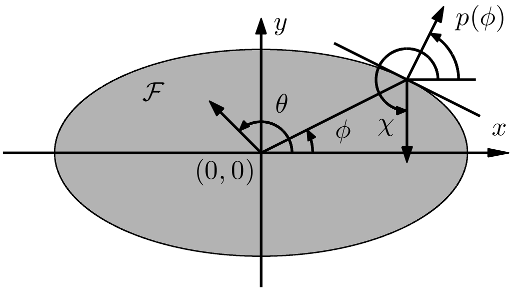


Figure 5.3: An empty forwarding region $\mathcal{F}$ travels towards direction $\mathcal{X}_1 = \theta$.

The required conditional expectations can then be calculated using

$$E[X_{T,1}|\mathcal{X}_1 = \theta, M'] = \int_0^\infty E[X_{T,1}|\mathcal{X}_1 = \theta, M', Q = q] f_Q(q)\, dq, \tag{5.8}$$

$$E[C_1|\mathcal{X}_1 = \theta, M'] = \int_0^\infty E[C_1|\mathcal{X}_1 = \theta, M', Q = q] f_Q(q)\, dq. \tag{5.9}$$

5.3.2 $\mathcal{F}$ is Initially Empty

Next, we calculate the expected values of $X_{S,1}$, $X_{T,1}$, C_1, T_1, and the distribution of $\mathcal{X}_2$, subject to $\mathcal{X}_1 = \theta$ and to the event M that $\mathcal{F}$ is initially empty.

To this end, consider the following counting process $\{N_{\chi,\phi;\theta}(t),\ t \geq 0\}$: there is an arrival whenever a node enters $\mathcal{F}$ through the part of its boundary described by the range of angles $[\phi, \phi + d\phi]$ and that node has a direction of travel within the range $[\chi, \chi + d\chi]$, where $\phi, \chi \in [-\pi, \pi]$. All angles appear in Fig. 5.3.

Observe that arrivals of this process at non overlapping time intervals are independent, as they are caused by nodes that at the start of time existed in non overlapping regions of $\mathbb{R}^2$. Furthermore, the number of arrivals in a time interval $[t_0, t_1]$ is Poisson distributed, with a rate proportional to the duration of the time interval $[t_1 - t_0]$. Indeed, the number of arrivals equals the number of nodes that existed at time t_0 in a region of space adjacent to $\mathcal{F}$ whose area is proportional to $t_1 - t_0$. (We elaborate on this region in Appendix A.) The number of nodes in that region is Poisson distributed, with a parameter proportional to the area of the region, and therefore proportional to the duration of the interval $t_1 - t_0$.

To conclude, *(i)* the number of arrivals in a time interval is Poisson distributed with a rate proportional to the duration of the time interval, and *(ii)* arrivals at non overlapping time intervals are independent. It follows that the counting process $\{N_{\chi,\phi;\theta}(t), t \geq 0\}$ is Poisson [109]. Observe that the (yet unknown) arrival rate is incremental, due to the fact that we consider an incremental part of the boundary $[\phi, \phi+d\phi]$, and an incremental part of the node directions $[\chi, \chi + d\chi]$. Let $\gamma(\chi, \phi; \theta)d\chi d\phi$ be this rate.

Next, consider the counting process $\{N_{\chi;\theta}(t),\ t \geq 0\}$ of all nodes arriving with a direction within $[\chi, \chi + d\chi]$ at *any* part of the boundary of $\mathcal{F}$. By the additive property of Poisson processes (i.e., the summation of independent Poisson processes is also a Poisson process, with a rate equal to the sum of the rates of the constituent Poisson processes [109]) it follows that this process is also Poisson, with a rate $\gamma(\chi; \theta)d\chi$ such that

$$\gamma(\chi; \theta) = \int_{-\pi}^{\pi} \gamma(\chi, \phi; \theta)\, d\phi. \tag{5.10}$$

Also consider the counting process $\{N_{\phi;\theta}(t),\ t \geq 0\}$ of all nodes arriving at the boundary $[\phi, \phi + d\phi]$, but with *any* direction in $[-\chi_m, \chi_m]$. By the additive property of Poisson processes, this process is also Poisson. Let its rate be $\gamma(\phi; \theta)d\chi$. We must have

$$\gamma(\phi; \theta) = \int_{-\chi_m}^{\chi_m} \gamma(\chi, \phi; \theta)\, d\chi. \tag{5.11}$$

Finally, consider the counting process $\{N_\theta(t),\ t \geq 0\}$ of all nodes arriving at *any* point of the boundary with *any* direction in $[-\chi_m, \chi_m]$. Again, the new process is also Poisson, with some rate $\gamma(\theta)$. We must have

$$\gamma(\theta) = \int_{-\chi_m}^{\chi_m} \gamma(\chi; \theta)\, d\chi = \int_{-\pi}^{\pi} \gamma(\phi; \theta)\, d\phi. \tag{5.12}$$

We will refer to all rates $\gamma(\chi, \phi; \theta)$, $\gamma(\chi; \theta)$, $\gamma(\phi; \theta)$, and $\gamma(\theta)$ as **incidence rates**. Observe that, in order to keep the notation simple, we have used the same symbol, i.e., γ, for all of them. We will differentiate them by their arguments. The incidence rates are calculated in the Appendix, where they are given by Eqns. (A.2), (A.4), (A.7), and (A.8) respectively.

Having the incidence rates, we will now calculate the conditional expectations $E(*|\mathcal{X}_1 = \theta, M)$ (where the asterisk $*$ is any of the random variables $X_{S,1}$, $X_{T,1}$, C_1, and T_1) as well as $f_{\mathcal{X}_2|\mathcal{X}_1=\theta,M}(\chi)$.

Let the random angle Φ be defined so that the location on the boundary of $\mathcal{F}$ where the first relay appears is $(r(\Phi), \Phi)$. Observe that the density of Φ conditional on $\mathcal{X}_1 = \theta$ is equal to

$$f_{\Phi|\mathcal{X}_1=\theta,M}(\phi) = \frac{\gamma(\phi; \theta)}{\gamma(\theta)}. \tag{5.13}$$

This is due to the fact that the minimum of a number of exponential random variables is equal to one of them with probability equal to its rate over the sum of all rates [109]. It follows that

$$E(X_{T,1}|\mathcal{X}_1 = \theta, M) = \int_0^{2\pi} [r(\phi)\cos\phi - x_0] \frac{\gamma(\phi;\theta)}{\gamma(\theta)}\, d\phi, \tag{5.14}$$

$$E(C_1|\mathcal{X}_1 = \theta, M) = \int_0^{2\pi} \left[(r(\phi)\cos\phi - x_0)^2 + (r(\phi)\sin\phi)^2\right] \frac{\gamma(\phi;\theta)}{\gamma(\theta)}\, d\phi. \tag{5.15}$$

Regarding the time T_1, observe that, as the counting process $\{N_\theta(t),\ t \geq 0\}$ is Poisson with rate $\gamma(\theta)$, T_1 is exponentially distributed with mean

$$E(T_1|\mathcal{X}_1 = \theta, M) = \frac{1}{\gamma(\theta)}. \tag{5.16}$$

Furthermore, since $X_{S,1} = v_n T_1 \cos\mathcal{X}_1$, we have

$$\begin{aligned} E(X_{S,1}|\mathcal{X}_1 = \theta, M) &= E(v_n T_1 \cos\mathcal{X}_1 | \mathcal{X}_1 = \theta, M) \\ &= v_n \cos\theta E(T_1|\mathcal{X}_1 = \theta, M) \\ &= \frac{v_n \cos\theta}{\gamma(\theta)}. \end{aligned} \tag{5.17}$$

Finally, by an argument similar to that applied for deriving (5.13), observe that

$$f_{\mathcal{X}_2|\mathcal{X}_1=\theta,M}(\chi) = \frac{\gamma(\chi;\theta)}{\gamma(\theta)}. \tag{5.18}$$

Wrap-up: We can now calculate numerically the conditional averages $E(*|\mathcal{X}_1 = \theta)$ of the random variables $X_{S,1}$, $X_{T,1}$, C_1, T_1, using the equations (5.3) and (5.4) and the conditional averages in Eqns. (5.7), (5.8), (5.9) and Eqns. (5.14), (5.15), (5.16), (5.17).

We can also calculate the conditional pdf $f_{\mathcal{X}_2|\mathcal{X}_1=\theta}(\chi)$ using Eqns. (5.4) and (5.5) together with Eqns. (5.6) and (5.18).

5.4 Approximate (D_p, C_p) Calculations

If the random vectors $(X_{S,i}, X_{T,i}, C_i, T_i)$ describing each stage i were independent and identically distributed, then it would be straightforward to calculate the normalized delay and cost using their definitions (5.1) and (5.2) along with the Strong Law of Large Numbers (SLLN). Indeed, by the SLLN it would follow that

$$\begin{aligned} \lim_{n\to\infty} \frac{\sum_{i=1}^n T_i}{n} &= E(T_1), \\ \lim_{n\to\infty} \frac{\sum_{i=1}^n C_i}{n} &= E(C_1), \\ \lim_{n\to\infty} \frac{\sum_{i=1}^n (X_{S,i} + X_{T,i})}{n} &= E(X_{S,1}) + E(X_{T,1}), \end{aligned}$$

from which we would have

$$D_p = \frac{E(T_1)}{E(X_{S,1}) + E(X_{T,1})}, \ C_p = \frac{E(C_1)}{E(X_{S,1}) + E(X_{T,1})}. \tag{5.19}$$

However, for our forwarding rule, we do not expect these random vectors to be either independent or identically distributed.

For example, whereas $\mathcal{X}_1$ is uniformly distributed in $[-\pi, \pi]$, the distribution of $\mathcal{X}_2$ (given by integrating the conditional distribution (5.5) over the uniform distribution of $\mathcal{X}_1$) is zero outside $[-\chi_m, \chi_m]$. Likewise, the distribution of $\mathcal{X}_3$ will not in general be the same as the distribution of $\mathcal{X}_2$, and so on. As the statistics of $X_{S,i}$, $X_{T,i}$, C_i and T_i all depend strongly on the distribution of $\mathcal{X}_{i-1}$, we expect that the vectors $(X_{S,i}, X_{T,i}, C_i, T_i)$ are not identically distributed. These vectors are also not independent; for example, a long sequence of zero sojourn times, i.e., $T_k = T_{k+1} = \cdots = T_{k+m} = 0$ for some $k, m > 0$ with m large, suggests that there are many nodes in the vicinity of node n_{k+m+1}, which means that the conditional expected time $E[T_{k+m+1}|T_k = T_{k+1} = \cdots = T_{k+m} = 0]$ will be smaller than the unconditional expected time $E[T_{k+m+1}]$.

The fact that the vectors $(X_{S,i}, X_{T,i}, C_i, T_i)$ are neither independent nor identically distributed significantly complicates the analysis. Intuitively, we expect that for most reasonable selections of the forwarding region and maximum deviation χ_m, the correlation across the stages diminishes fast enough so that the SLLN approximately holds. Even if this is indeed the case, we need to find the expected values of the components $(X_{S,i}$, $X_{T,i}$, C_i, $T_i)$, which is also not a simple task.

Motivated by these observations, we introduce the following approximation:

First Order Approximation: *We take the vectors $(X_{S,i}, X_{T,i}, C_i, T_i)$ to be iid, and so we use Eqns. (5.19) to calculate D_p and C_p. However, these vectors are now distributed according to the vector $(X_{S,1}, X_{T,1}, C_1, T_1)$ when the direction $\mathcal{X}_1$ that the first node travels is distributed according to the limiting distribution $g(\chi) = \lim_{i\to\infty} f_{\mathcal{X}_{i+1}}(\chi)$ arrived at by the following iteration, provided the limit exists:*

$$f_{\mathcal{X}_{i+1}}(\chi) = \int_{-\chi_m}^{\chi_m} f_{\mathcal{X}_2|\mathcal{X}_1=\theta}(\chi) f_{\mathcal{X}_i}(\theta)\, d\theta. \tag{5.20}$$

Intuitively, this approximation can be explained as follows: assume that whenever a hop is made, the process describing the movement of the nodes restarts, with the exception of a single piece of information (hence the name of the approximation), which is the direction of travel of the node that received the packet. Under this approximation, the distributions $f_{\mathcal{X}_i}(\chi)$ of the directions of travel $\mathcal{X}_i$ will evolve according to the given formula (5.20). We use their limit, $g(\chi) = \lim_{i\to\infty} f_{\mathcal{X}_{i+1}}(\chi)$, to find the statistics of all other random variables of interest.

As the simulation results of the next section show, the First Order Approximation leads to numerical results for the normalized delay and cost that closely match those results found by simulating the network. In any case, finding a better approximation, or avoiding approximations altogether, is clearly important.

5.5 Results

In this section we calculate the (D_p, C_p) pairs achieved by our forwarding rule, using the analysis (coupled with the First Order Approximation) of the previous sections as well as simulations.

We limit ourselves to the case where the forwarding region $\mathcal{F}$ is a circular disk of radius R with the current holder of the packet lying on its circumference opposite to the direction of the packet destination. Therefore, we have two parameters available for trading off cost with delay, the radius R of the disk $\mathcal{F}$, and the maximal deviation χ_m. For each pair of values (R, χ_m), there is a corresponding pair (D_p, C_p).

We consider circular disks because, due to their symmetry, the analysis of the previous sections involves relatively simple calculations and also because in a related, purely geographic, non DTN routing setting [112], it was shown that they are the optimal shape for the forwarding region, given that $c(d) = d^2$. (We stress that the sense of optimality used in [112] is not compatible to our current work.)

In Fig. 5.4 the parameters chosen are $v_p = 1$ and $\lambda = 1$. The radius of the forwarding region ranges from $R = 0$ to $R = 5$, and the maximum deviation ranges from $\chi_m = 0$ to $\chi_m = \pi$. In the plot, we have drawn a total of 30 dotted lines, each line showing the evolution of the delay-cost pair as χ_m is fixed but R increases from $R = 0$ to $R = 5$. The values of χ_m used are $\chi_m = \frac{i\pi}{30}$, $i = 1, \ldots, 30$.

As expected, increasing the radius R leads to a decrease of the normalized delay (as sojourns become shorter), and an increase of the normalized cost (as transmissions become more frequent). The effects of increasing the maximum deviation χ_m are mixed. When the radius is large, it is best to use large values of χ_m. The intuitive explanation is that, since the packets mostly rely on wireless transmissions, the direction a node is moving is not so important, and so it makes sense to use all nodes available, so that the transmission costs are minimized. On the other hand, when the radius R is small,

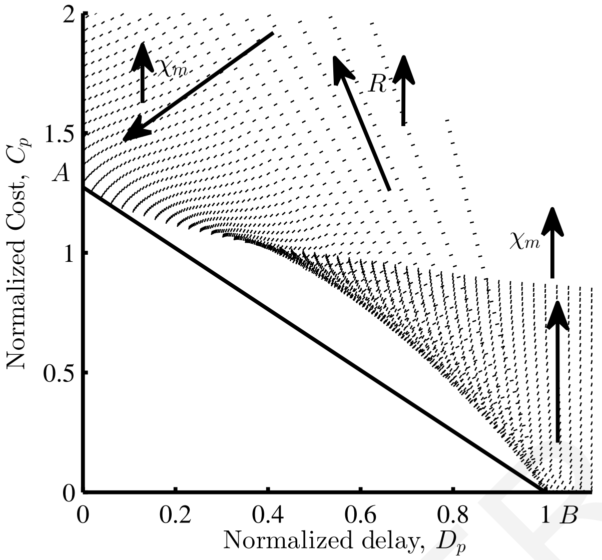


Figure 5.4: Delay-Cost plots for the case of the circular disk.

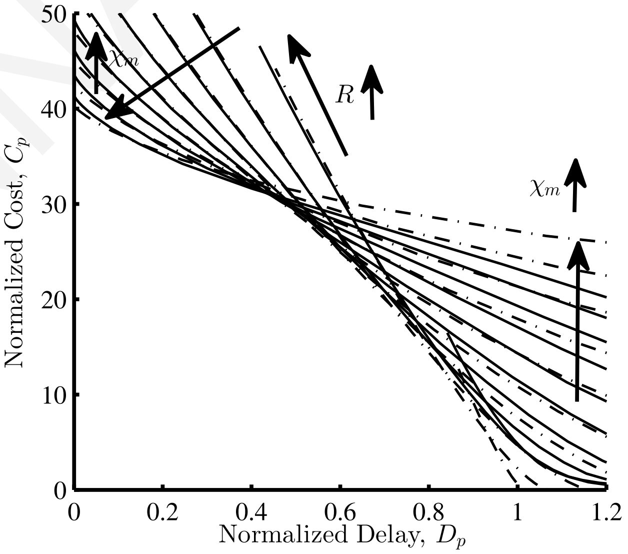


Figure 5.5: Simulation (continuous lines) vs. analysis (dash-dotted lines).

it is best that small values of χ_m are used. Indeed, when R is small, packets travel to their destination mostly by physical transport, spending (on the average) a lot of time at each relay, so it is best to avoid relays traveling in the wrong direction.

Two points on the delay-cost plane are of particular interest. The first is the point $B = (1, 0)$ achieved when $R \to 0$ and $\chi_m \to 0$. In this limiting case, our forwarding rule dictates that the packet should find a node moving in the exact direction of the destination, and stay with that node forever, thus traveling with an average cost $C_p = 0$ and an average delay $D_p = \frac{1}{v_n} = 1$. The second point is the one at which the pair (D_p, C_p) converges as the radius $R \to \infty$, for $\chi_m = \pi$. It is numerically established that this point $A \simeq (0, 1.2732)$. Intuitively, in this limit the packet aggressively moves from node to node, without being transported physically at all.

A somehow surprising result is that no combination of parameters R and χ_m leads to a delay-cost pair below the line connecting these two points. This line is also plotted in Fig. 5.4, and describes the set of delay-cost pairs achievable by a time division between the two extreme strategies associated with each of these points. Therefore, if we limit ourselves to the forwarding rules of Section 5.2, it is optimal to use time division between the two extreme strategies.

In order to evaluate the effects of the First Order Approximation, and make sure the analysis remains relatively accurate, we also evaluated our forwarding rules by simulation. We placed N nodes in a square torus of side L, moving along straight lines with speed v_n as in the network model, for time T. In Figure 5.5 we plot, with continuous lines, 10 delay-cost curves. We set $N = 1000$ and $L = 1000$, so that $\lambda = 0.001$, and also $v_n = 1$, $T = 1000$. Each curve is created by keeping χ_m fixed and increasing R from $R = 5$ to $R = 100$. The values of χ_m used were $\frac{\pi}{10}$, $\frac{2\pi}{10}, \ldots, \pi$. We also plot the delay-cost curves derived using the analysis, for the same parameters. The two sets of curves match particularly well in the low delay regime. In the high delay regime, the match is not as good, but we attribute this mostly to the fact that the simulation statistics are based on smaller numbers of hops, and so are not as accurate. However, the overall match between simulation and analysis is encouraging.

Finally, we present simulation results for another forwarding rule. We are motivated by the observation that the forwarding rules described in Section 5.2 attempt to minimize the quotient of the cost over the progress, without taking into account the direction of travel of either the current holder, or the potential packet receivers.

Therefore, we modify our forwarding rule as follows. We select as forwarding area a circle of diameter R, and radius $\frac{R}{2}$, centered at the location $\left(\frac{R}{2}, 0\right)$, with the current holder placed at the origin. As with the previous forwarding rule, the current packet holder A places all the nodes in the forwarding area in the order of increasing quotients Q of Cost C over progress X. However, it does not give the packet to the node with the smallest quotient. Rather, it goes through the list of these nodes, going from the one with the smallest quotient and moving upwards, and transmits the packet to the first node B that *also* satisfies the following condition:

$$Q_B = \frac{c_B}{x_B} \leq \frac{R}{2} \times \left[1 + \frac{bv_{x,B} - av_{x,A}}{2v_n}\right].$$

In the above, x_B is the x-coordinate of B, i.e., the progress achieved by the packet towards the destination by its reception by node B, and $c_B = x_B^2 + y_B^2$ is the cost of transmitting the packet to B. Also, $v_{x,B}$ and $v_{x,A}$ are the x-components of the velocity of nodes B and A respectively, and v_n is the node speed. The parameters $b, a \geq 0$ are used to tune the effects that the speeds $v_{x,B}$ and $v_{x,A}$ have on the decision, respectively.

Under this rule, the faster node A travels towards the destination, i.e., the larger $v_{x,A}$ is, the smaller is the right hand bound, and so the harder it becomes for a node in the forwarding region to satisfy the inequality. Similarly, the faster a node B travels towards the destination, i.e., the larger $v_{x,B}$ is, the larger the right hand side will be, and the more appealing that node becomes to the packet.

Henceforth, we will refer to the rule of Section 5.2 as **Rule I** and the rule described here as **Rule II**.

In Fig. 5.6 we plot the following three curves, for the setting of Fig. 5.5: the Pareto optimal curves of Rule I, over all combinations of the parameters R, χ_m, using analysis, the same Pareto optimal curves using simulations, and finally the Pareto optimal curves of Rule, using simulations. We note that it is also possible to arrive at analytical results for the second rule, but the derivations are lengthy, and we omit them due to space constraints. As expected, Rule II performs better than Rule I, due to its more refined relay selection. It is also interesting to note that part of the Pareto optimal curve of Rule II lies below the time division curve connecting the points A and B. Therefore, there is a choice of parameters of Rule II such that the rule performs better than any time division between the two extreme strategies corresponding to points A and B.

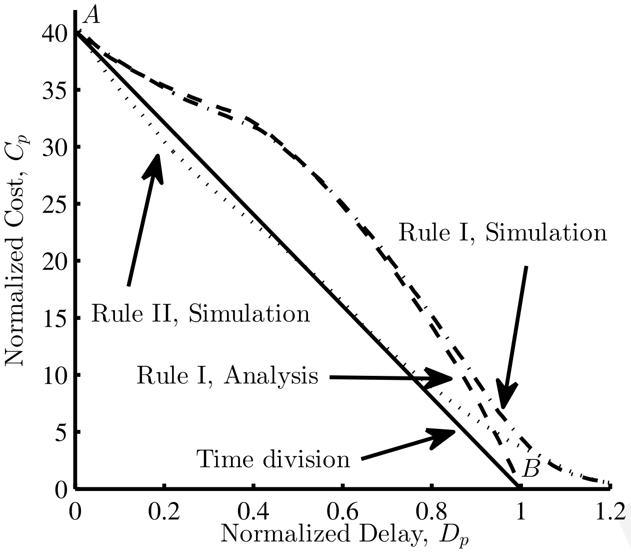


Figure 5.6: Simulation (continuous lines) vs. analysis (dash-dotted lines) results.

5.6 Delay/Cost Tradeoff when Future Topology is Known

In this section we present simulation results when our setting is modified in the following manner: all nodes have complete information about the future trajectories of all other nodes. This short, concluding section is an example of what other problems can be studied with our framework.

Regarding the simulation setup, we simulate a network of 1000 nodes that move on a square torus of side 1000 m for 1000 slots. The nodes have velocity of magnitude 1 m/s and transmission range 100 m. The direction of the velocity of each node is chosen randomly and uniformly over all possible directions at the beginning of the simulation and it does not change during the simulation. The initial position of each node is chosen randomly and uniformly on the square torus at the beginning of the simulation. At the beginning of the simulation each node creates a packet that travels to a direction chosen randomly and uniformly over all possible directions.

We assume that all nodes know the trajectories of all nodes. Each packet follows the space-time path that has the smallest $C_o - c \cdot P_o$, where C_o is the transmission cost of the space-time path, P_o is the progress of the space-time path, and c is a parameter of the protocol. As explained earlier, in our simulation the time is slotted and the nodes move on a torus. The set of the space-time paths over which we find the best path is the set of all space-time paths that exist during the simulation, with the following constraint that we set for simplicity. A packet is allowed to go at slot t from node i to node j, for $t = 0, 1, ..., 999$ and $i, j = 0, 1, ..., 999$, only following the path that has the smallest transmission cost, regardless if there is a path at slot t from node i to node j that has larger transmission cost but has smaller transmission cost minus c times progress.

For each packet we find the path that has the smallest $C_o - c \cdot P_o$ using Algorithm 2 that is described in [113], which is an algorithm for finding optimal space-time paths, where in our case we define as optimal the space-time path that has the smallest $C_o - c \cdot P_o$. We run the simulation for $c = 1, 2, ..., 50$. For each value of c, we find the total cost and total progress done by all packets in all slots, and from this we find a point in the normalized cost normalized delay plane. We refer to this rule as Rule III and we compare it with Rule I and Rule II that were defined previously in this chapter.

Figure 5.7 shows our simulation results for the cost per progress versus the delay per progress that can be achieved using Rule I, Rule II, and Rule III. Rule II performs better than Rule I because it takes into account not only the velocity of the next hop but also the velocity of the current holder. Rule III performs better than Rule I and Rule II because it uses information about the future trajectories of the nodes that is not available in Rule I and Rule II.

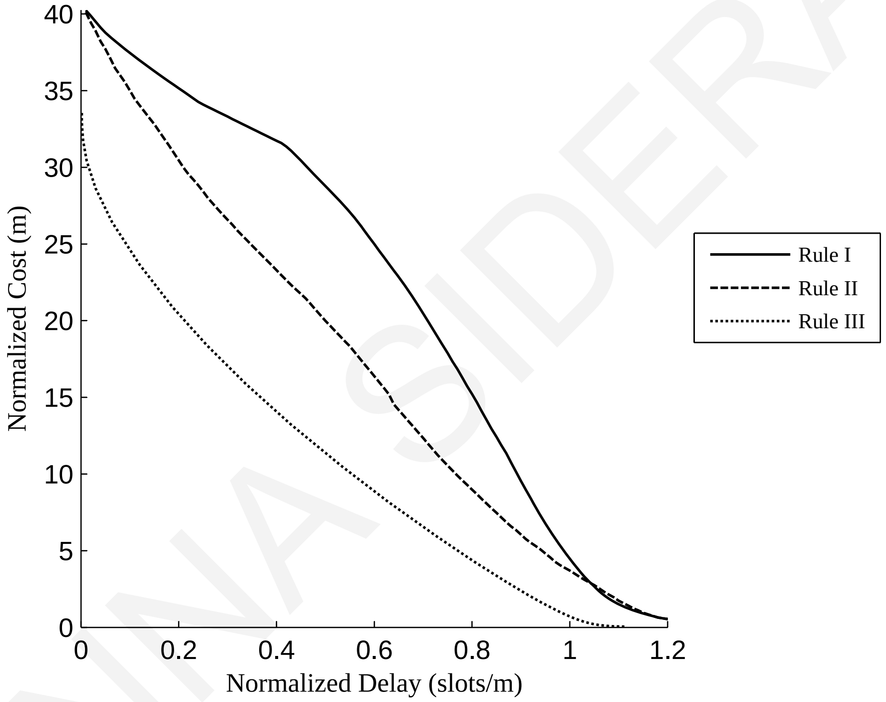


Figure 5.7: Normalized Cost versus Normalized Delay for Rule I, Rule II, and Rule III.

Chapter 6

The Extended Minimum Estimated Expected Delay Protocol

In this chapter[1] we present the Extended Minimum Estimated Expected Delay (EMEED) protocol and evaluate its performance in two settings, a Pocket Switched Network (PSN) and a vehicular DTN. The EMEED protocol may be thought of as a nontrivial generalization of the Minimum Estimated Expected Delay (MEED) protocol [36], from which it has been inspired.

6.1 Basic Network Assumptions

EMEED is designed for use in networks that satisfy the following fundamental assumptions:

Nodes: We have a very large number of nodes that move in a region independently of their communication needs.

Non-uniform, correlated mobility: The mobility patterns of the nodes are not uniform, so that some pairs of nodes systematically meet more often than others, and also the mobility patterns of different nodes are correlated.

Communication Needs: The applications running at each node are delay tolerant, however there is a maximum acceptable delay for the delivery of the packets. The applications depend on the communication between node pairs.

Connectivity level: EMEED is suitable for networks where there is no end-to-end path from source to destination, but there are multihop paths.

[1] This work also appears in [39, 40].

Neighborhood Awareness: Each node knows which nodes are a few hops away from it and also knows routes to these nodes, through the use of a proactive routing protocol such as OLSR [96].

6.2 EMEED Protocol Specification

The main parameter of the EMEED protocol is the **contact radius** R_C. When, according to the current topology, two nodes i and j are separated by at most R_C hops, we say that i and j are in contact. The parameter R_C can take the following values: (a) $R_C = 1, 2, 3, \ldots$, (b) $R_C = \infty$, in which case two nodes are in contact if they are in the same partition. When $R_C = 1$, the EMEED protocol operates similarly to the MEED protocol, which assumes that two nodes are in contact if they can communicate directly. We define the R_C neighborhood of a node to be all the nodes that are at most R_C hops from the node.

Estimation of Expected Delays: Every node j maintains, for every other node k, an estimate of the expected value $E[WT(j,k)]$ of the time it will have to wait until it comes in contact with node k. These estimates are calculated as follows: assume that during the time interval $[0,T]$ node j is not in contact with node k for m intervals of durations $d_1, d_2, \ldots, d_m$ and that for the rest of the time in this interval nodes j, k are in contact. Then j estimates $E[WT(j,k)]$ using the formula

$$E[WT(j,k)] = (d_1^2 + d_2^2 + \cdots + d_m^2)/(2T).$$

This method for estimating $E[WT(j,k)]$ was used in [36], and its use is justified there.

Creation of Expected Delay Routing Table: The nodes disseminate the estimates of the expected delays in the network and so each node i stores the estimates of $E[WT(j,k)]$ for different pairs of nodes (j,k). Nodes forward packets according to a routing table they create, called the **expected delay routing table**. Every node i creates its expected delay routing table performing shortest path routing on a graph called the **expected delay graph** of node i. This graph consists of links of cost $E[WT(j,k)]$ for each pair (j,k) for which i has a value of $E[WT(j,k)]$ in its memory, but we set to 0 the costs of the links from node i to every node that is currently within R_C hops of node i.

Dissemination of Expected Delays: In order to keep a check on the amount of routing overhead used for the dissemination of the expected delays, the protocol uses

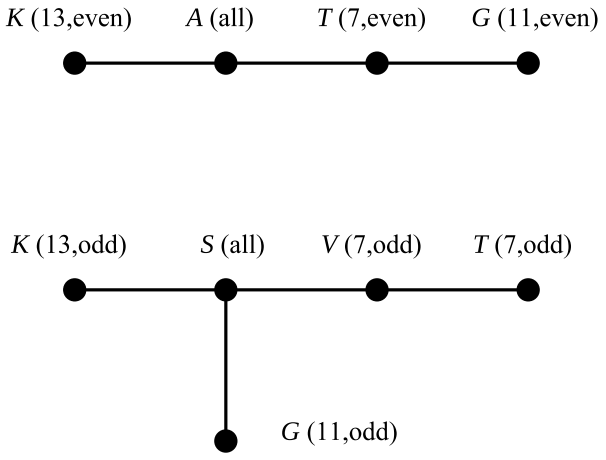


Figure 6.1: Example 1.

two parameters, the **routing table cost threshold** C_T and the **number of friends** N_F. When the expected delay routing table of a node is created, only paths of total cost C_T or less are discovered. Therefore, expected delays greater than C_T are not broadcast. If a node estimates fewer than N_F expected wait times to be smaller than C_T it disseminates in the network all the expected wait times it estimates to be smaller than C_T. If a node estimates more than N_F expected wait times to be smaller than C_T, it disseminates in the network only the N_F smallest expected wait times it has estimated. We call the nodes for which a node i disseminates expected wait times it estimates the *friends* of node i.

6.3 Examples

To motivate the advantages of using the parameter R_C, this section illustrates the operation of the EMEED protocol using two examples.

Example 1: Consider the scenario shown in Fig. 6.1. Time is measured in slots. Nodes that can communicate directly during some of the slots are connected by lines. Nodes A and S are at the positions shown in the figure at all slots. The other nodes are not always at a fixed position. Node K is at the position shown near node A at slots that are divisible by 13 and 2. Node K is at the position shown near node S at slots that are divisible by 13 but not 2. Node K is isolated at slots that are not divisible by 13. Node G is at the position shown near node A at slots that are divisible by 11 and 2. Node G is at the position shown near node S at slots that are divisible

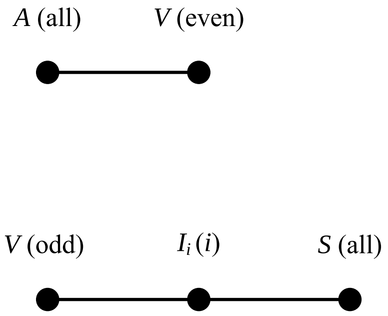


Figure 6.2: Example 2.

by 11 but not 2. Node G is isolated at slots that are not divisible by 11. Node T is at the position shown near node A at slots that are divisible by 7 and 2. Node T is at the position shown near node S at slots that are divisible by 7 but not 2. Node T is isolated at slots that are not divisible by 7. Node V is at the position shown near node S at slots that are divisible by 7 but not 2. Node V is isolated at all other slots.

We simulate our protocols using this network scenario. We run the simulation without creating packets, until a slot that is large enough for the nodes to have good estimates of the expected wait times, and that is divisible by 2, 7, 11, and 13 so that at this time all nodes are present at the positions shown near node A. At this time we create a packet which has node A as source and node S as destination. We continue running the simulation and check when the packet is delivered.

When $R_C = 1$ the delay is 13 slots and when $R_C = \infty$ the delay is 7 slots.

Example 2: Consider the scenario shown in Fig. 6.2. Time is measured in slots. Nodes that can communicate directly during some of the slots are connected by lines. Nodes A and S are at the positions shown in the figure at all slots. The other nodes are not always at a fixed position. Node V is at the position shown near node A at slots that are divisible by 2. Node V is at the position shown near node S at slots that are not divisible by 2. At slot i, for $i = 0, 1, 2, \ldots$, there is a node I_i near node S. That is, at each slot i there is a node I_i that appears in the example only once and then it disappears.

Assume that we run our protocol for $R_C = 1$. Node A estimates that $E[WT(A, V)] = 2$. Node V does not send to node A any value of $E[WT(V, I_i)]$, as nodes V and I_i, for $i = 0, 1, 2, \ldots$, are in contact only once. Node V does not send to node A any value of $E[WT(V, S)]$, as when $R_C = 1$, nodes V and S are never in contact. Thus node A does not have any path to node S in its expected delay routing table. Thus, if node A

creates packets for node S the packets cannot be delivered.

Assume that we run our protocol for $R_C = 2$. Node A estimates that $E[WT(A, V)] = 2$. Node V estimates that $E[WT(V, S)] = 2$, as nodes V and S are in contact every 2 slots, via another node. Node V sends the value of $E[WT(V, S)]$ it estimates to node A. Thus, in the expected delay routing table of node A there is a path to node S where the next hop is node V and the cost is 4. Thus, if node A creates packets for node S the packets can be delivered.

Observe that the two examples differ in that in the second case new nodes are constantly introduced. In reality, a scenario similar to that of Example 2 can happen, for example, if a bus comes several times a day near an access point, and has a multi-hop path to it that consists of cars, and the cars are different every time. Therefore, using EMEED with $R_C > 1$ allows us to take advantage of nodes that appear in the network very infrequently, or even only once.

6.4 Performance Evaluation Setting

In this section we present the setting on which we evaluate the performance of EMEED.

Slotted Time: We divide time in slots and we assume that the positions of the nodes are fixed during a slot.

Transmitter-Receiver Model: All nodes have the same transmission range R. We assume that the nodes can only communicate directly with each other if they are at a distance R or smaller from each other.

MAC layer: We assume that transmissions are always successful, and there is no contention. In other words, if two nodes are in contact and the routing protocol instructs them to exchange a packet, the packet exchange is always successful. In order to have a fair comparison with EMEED, we assume that in all the protocols we simulated, two nodes are in contact when they are at most R_C hops away from each other.

Packet Forwarding: At each slot each packet can travel more than one hops. Specifically, it is forwarded from node to node, according to the routing decisions of its consecutive holders. The forwarding stops when the packet reaches its destination or a node that decides to wait for the topology to change instead of forwarding the packet to one of the nodes that are currently in contact with it. Similarly, for the protocols that use replication instead of forwarding, we assume that during a slot the packet is replicated until no more replications are possible.

Buffer Policy: All nodes have sufficiently large buffer spaces so that packets are discarded only when their TTL elapses.

Traffic: In EMEED, initially, for some time that we term the **estimation time** T_E, we send only control packets for each node to know which nodes are currently in contact with it, and then, for some time which we term the **dissemination time** T_D, we send only control packets for the dissemination of the expected wait times and control packets for each node to know which nodes are currently in contact with it. In BUBBLE, initially, for time $T_E + T_D$ we send only control packets for each node to know which nodes are currently in contact with it. In flooding and Spray and Wait, initially, for time $T_E + T_D$ we do not send any packets. Then, in all the protocols, at time equal to $T_E + T_D$, every node creates one packet for one other node, and then, the simulation runs for time equal to the Time To Live (TTL) of the packets, and during that time the nodes send only data packets and control packets for each node to know which nodes are currently in contact with it.

MEED Implementation: We simulate it by running EMEED for $R_C = 1$.

BUBBLE Implementation: There are different versions of BUBBLE. Also we had to adjust BUBBLE to our setting. For these reasons we describe next how we implemented BUBBLE. The global centrality of a node is the average number of different nodes it comes in contact with during an interval of duration T_C, where T_C is a parameter of the simulation that we call **centrality time**. The local centrality of a node, for a certain community, is the average number of different nodes it comes in contact with during an interval of duration T_C, that belong to that community. The average takes into account all intervals of duration T_C from the beginning of the simulation until time $T_E + T_D$. When a node A that has a packet comes in contact with a node B that does not have the packet, A gives a copy of the packet to B in the following cases. *(i)* There is at least one of the communities of the destination node in which B belongs but A does not belong. *(ii)* There is at least one of the communities of the destination node in which both A and B belong and B has larger local centrality for that community than A. *(iii)* B has larger global centrality than A. *(iv)* B is the destination node. Node A does not discard the packet unless its TTL expires.

Spray and Wait Implementation: In Spray and Wait the packet source transmits as soon as possible L packet replicas to L different nodes. We set $L = 10$, because we found using simulation that in this case, in the PSN setting, we have approximately the same number of transmissions per created packet as with EMEED. Therefore, there is a good basis for comparing the two protocols.

6.5 Simulation Tool

In order to evaluate our protocol, we have developed a simulation tool, specifically designed for DTNs, and written in C. The tool is available online [32]. As in the case of DTFR, we refrain from using off the shelf DTN simulation tools like ONE [102] because we are interested in studying very large networks, for which a lean, customized simulator based in C is ideally suited. We evaluate EMEED and compare it to MEED, BUBBLE, flooding, and Spray and Wait, using two settings, a Pocket Switched Network (PSN) setting and a vehicular DTN setting.

In this chapter we use mobility models in which the nodes belong to communities. We selected these in order to be able to accurately compare EMEED with BUBBLE which assumes that each node belongs to one or more communities. As BUBBLE is an extremely popular protocol, comparing EMEED with it is very useful. In [39] we compare EMEED with MEED using a mobility model in which the nodes do not belong to communities, but results are omitted here.

The simulator for EMEED is written recycling many parts of the simulator for DTFR. However, although the simulator for DTFR takes contention into account, the simulator for EMEED does not. In the simulator for EMEED when two nodes are within transmission range of each other they can exchange packets regardless of what other transmissions are taking place during that slot. For this reason, in order to gain a sense of how much traffic the simulated protocols create, in our simulations we count the numbers of all kinds of transmitted packets.

The simulator for EMEED is written to simulate smaller networks than the simulator for DTFR. The part of the simulator for EMEED that takes the most time to execute is the part that creates the expected delay routing tables of the nodes. If we want to simulate EMEED in larger networks we have to work to make this part of the simulator faster. In the simulator for EMEED the buffers of the nodes are implemented in a simpler way than in the simulator for DTFR. This is because the simulator for EMEED is designed for smaller number of nodes and smaller number of created packets compared to the simulator for DTFR.

6.6 Performance Evaluation in Pocket Switched Network

6.6.1 Mobility Model

We have $n_1 = 1000$ nodes, that we call **persistent nodes**, moving as follows: at the beginning of the simulation each node selects, a location called **home** (H), a location called **destination 1** (D_1), and a location called **destination 2** (D_2). These locations do not change during the simulation. At the beginning of each day the node is at its H. It stays there for some time. Then it selects randomly either to go to its D_1 or to its D_2. It stays there for some time. Then it returns to its H. Specifically, we use the parameters **minimum depart time** DT_{min}, **maximum depart time** DT_{max}, **minimum return time** RT_{min}, and **maximum return time** RT_{max}. Each day the node selects randomly a time between DT_{min} and DT_{max} to leave its home and a time between RT_{min} and RT_{max} to return to its H.

We assume that when the node leaves H it appears immediately at D_1 or D_2 and when the node leaves D_1 or D_2 it appears immediately at H. This approximation is made in order to speed up the simulation. Indeed, it would take too long to run a simulation that accurately simulates the movement of the nodes in detail for a large number of days. We believe that this approximation is reasonable, taking into account that delays are on the order of hours, and that while nodes are on the move, they cannot establish long lasting, high capacity links with their neighbors.

The positions of the home Hs and the destinations D_1s and D_2s are chosen as follows. At the beginning of the simulation we separate randomly the 1000 Hs the 1000 D_1s and the 1000 D_2s in communities of either 5 Hs or 5 D_1s or 5 D_2s. Thus in total we have 600 communities. For each community we have a square area where we place randomly and uniformly the 5 Hs or the 5 D_1s or the 5 D_2s. The square areas of the 600 communities are in such a distance from each other so that two nodes that are in different square areas are never within transmission range of each other.

Also, at each slot we have $n_2 = 1000$ **transient nodes**, that exist in the network only for that slot, and then disappear. They are placed randomly and uniformly on points on the 600 square areas. In a vehicular DTN scenario these nodes would correspond to cars that are at a location that they do not visit often, whereas persistent nodes would be cars that are parked outside their owner's home, or office, or any other

PARAMETER	NUMERICAL VALUE
Number of persistent nodes	$n_1 = 1000$
Number of transient nodes	$n_2 = 1000$
Transmission range	$R = 10$ m
Average node degree	$N = 2$
Estimation time	$T_E = 1920$ slots
Dissemination time	$T_D = 1920$ slots
Minimum depart time	$DT_{\text{min}} = 24$ slots
Maximum depart time	$DT_{\text{max}} = 36$ slots
Minimum return time	$RT_{\text{min}} = 60$ slots
Maximum return time	$RT_{\text{max}} = 88$ slots
Duration of a day	96 slots
Packet TTL	1920 slots
Number of persistent nodes per community	5
Buffer size	$B = \infty$
Routing table cost threshold in EMEED	$C_T = 1920$ slots
Number of friends in EMEED	$N_F = 10$
Centrality time in BUBBLE	$T_C = 6$ hours
Number of replicas in Spray and Wait	$L = 10$

Table 6.1: Default environment and protocol parameters for the PSN setting.

location the owner frequents.

The simulation takes as input the transmission range and the average node degree and it calculates the dimensions of the square areas.

Unless otherwise stated, the parameters used in the simulations are those of Table 6.1.

6.6.2 Results

Fig. 6.3 shows the delivery ratio versus the TTL. Fig. 6.4 shows the delivery ratio versus the average node degree. We observe that EMEED performs significantly better than BUBBLE. As expected, increasing R_C increases the delivery ratio of both EMEED and BUBBLE. This comes at an increased control overhead. However, the gains of going from $R_C = 2$ to $R_C = 3$ are small, which means that, in this setting, most of the

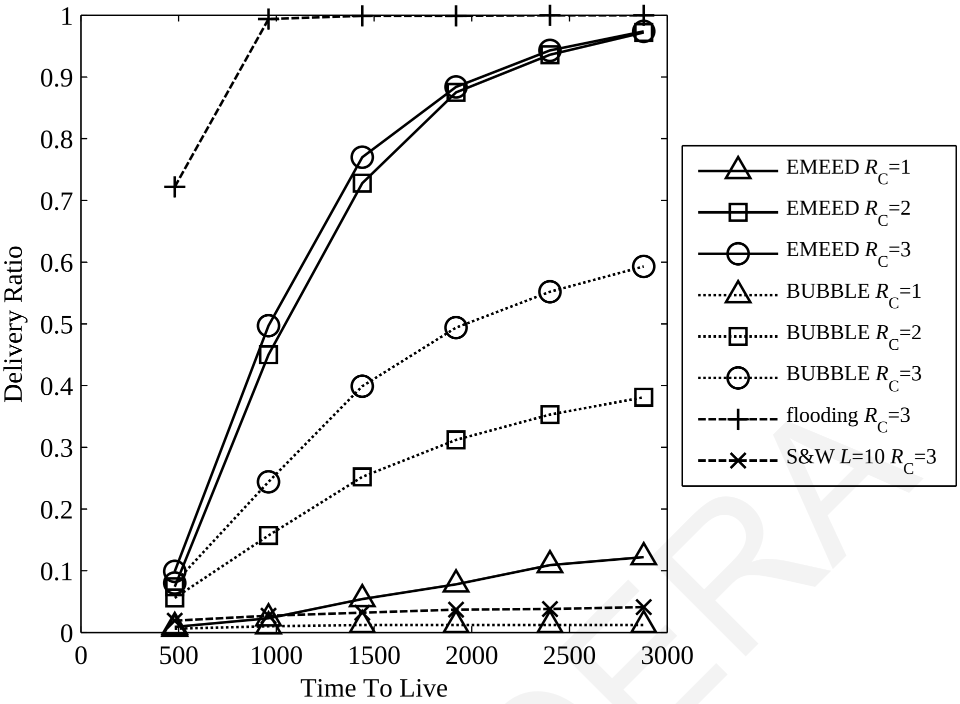


Figure 6.3: Delivery ratio versus TTL for the PSN setting.

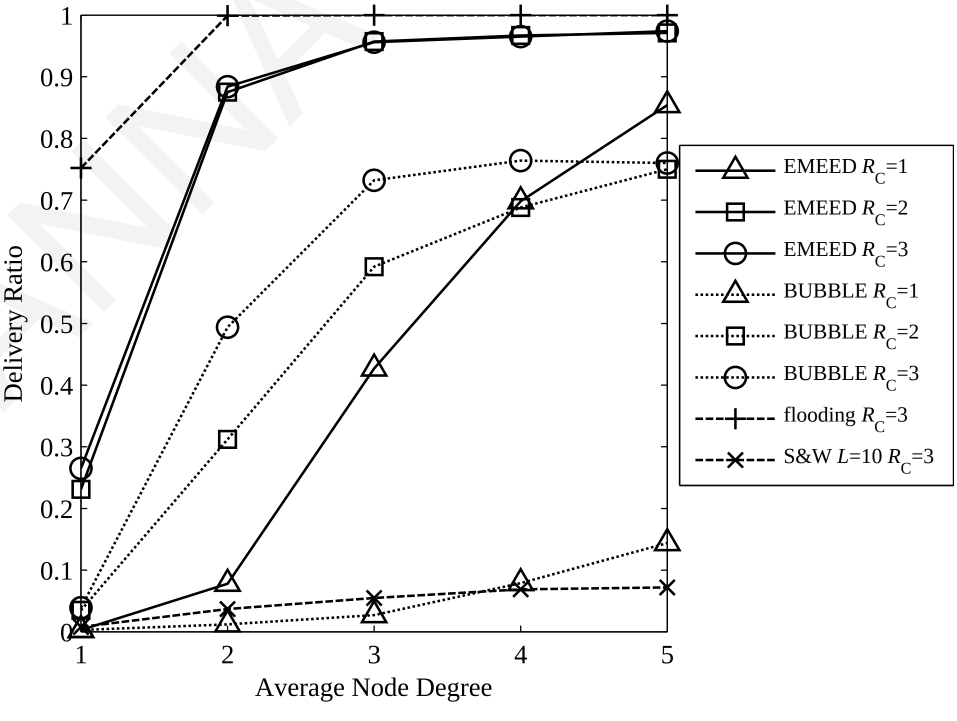


Figure 6.4: Delivery ratio versus average node degree for the PSN setting.

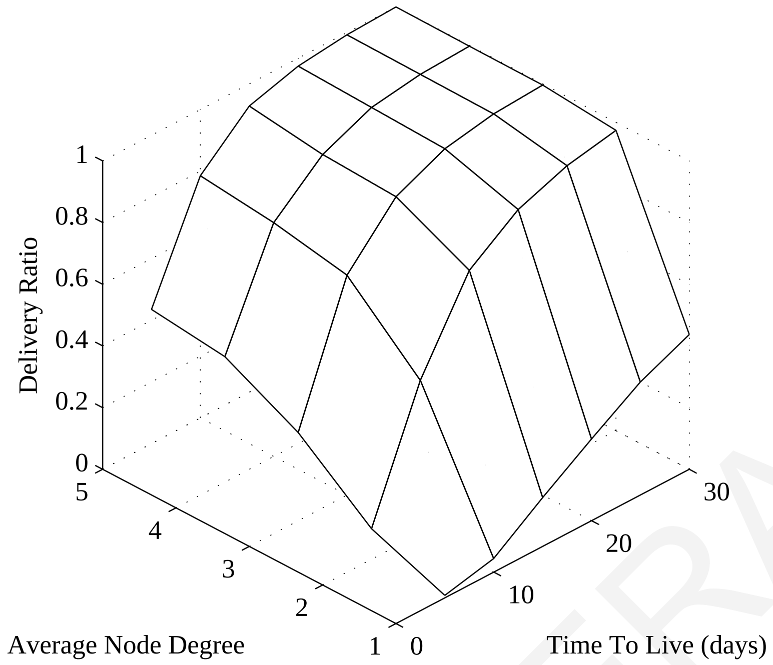


Figure 6.5: Delivery ratio versus TTL and average node degree for EMEED with $R_C = 3$ for the PSN setting.

gains can be attained with a modest increase in the control overhead. flooding has the best performance in terms of delivery ratio, but, as we show later, the worst in terms of the transmissions needed. Both EMEED and BUBBLE perform much better than Spray and Wait, as this protocol does not make an attempt to take into account the fact that the mobility patterns are not uniform.

Fig. 6.5 shows the delivery ratio versus the TTL and the average node degree for EMEED with $R_C = 3$. We observe that for a fixed value of the average node degree, as the TTL increases the delivery ratio increases. We also observe that for a fixed value of the TTL, as the average node degree increases the delivery ratio increases. This figure is useful to designers that want to trade off TTL (by modifying the tolerance of the application to delays) with the average node degree (by changing the transmission power). This figure quantifies this tradeoff.

Fig. 6.6 shows the delivery ratio versus the number of transient nodes, where the number of persistent nodes is the same as in Table 3.1. We observe that when there are no transient nodes, our protocol performs as well as MEED. When there are transient nodes, however, our protocol achieves a much higher delivery ratio. This shows that

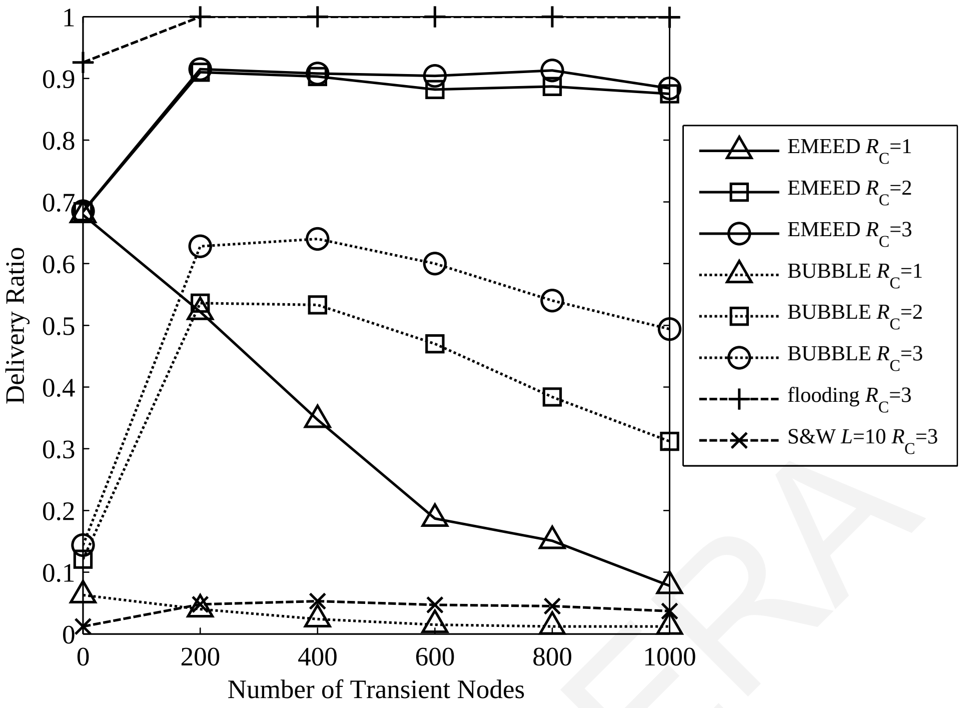


Figure 6.6: Delivery ratio versus number of transient nodes for the PSN setting.

the main advantage of our protocol over MEED is that our protocol makes use of paths connecting a source and a destination that are partially comprised of transient nodes. By its construction, MEED ignores such paths, whereas our protocol utilizes them extensively. This leads to notable improvements over MEED when the number of transient nodes is substantial.

We observe that when there are transient nodes EMEED with $R_C = 2$ performs better than EMEED with $R_C = 1$. To understand why, assume that there are two persistent nodes A and B that are never within transmission range of each other but are very often at distance one and a half transmission ranges from each other. Also, assume that there are transient nodes in the network. When A and B are at distance 1.5 transmission ranges from each other there will be very often a transient node that will be within transmission range of both A and B. Thus in the expected delay routing table of EMEED with $R_C = 2$ there will be a link between A and B. However in the expected delay routing table of EMEED with $R_C = 1$ there will be no link between A and B. This explains why when there are transient nodes EMEED with $R_C = 2$ performs better than EMEED with $R_C = 1$.

For the parameters of Table 6.1, we observed that in almost all the cases that

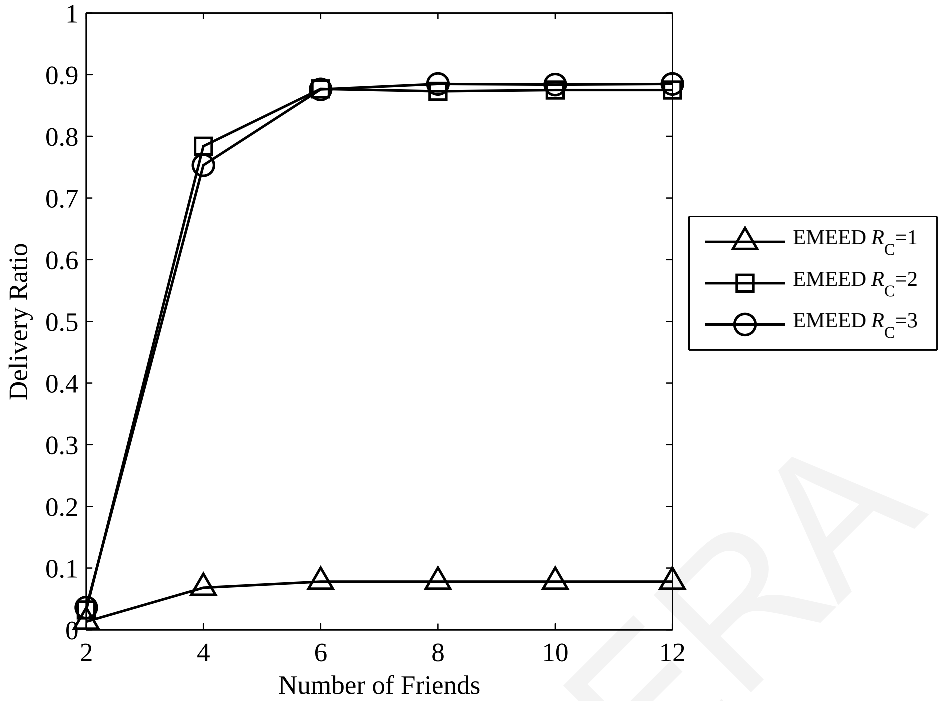


Figure 6.7: Delivery ratio versus number of friends for the PSN setting.

EMEED delivered a packet but BUBBLE did not deliver it, in the path that the packet followed in EMEED, the packet was forwarded from a node with larger global centrality to a node with smaller global centrality and this happened before the packet reached a node in one of the communities of the destination. BUBBLE would not use that link and this is why in the setting we used EMEED performs better than BUBBLE.

BUBBLE assumes that if each node that is not in the community of the destination and has a copy of the packet gives a copy to each node it comes in contact with that is not in the community of the destination and has larger global centrality or is in the community of the destination, the packet eventually reaches a node in the community of the destination. BUBBLE also assumes that if each node that is in the community of the destination and has a copy of the packet gives a copy to each node it comes in contact with that is in the community of the destination and has larger local centrality, the packet reaches the destination. If we have a few large communities, for example 9 communities of equal size, these assumptions hold. However if we have many small communities, as in the scenario used to obtain the results presented above, the first assumption does not hold.

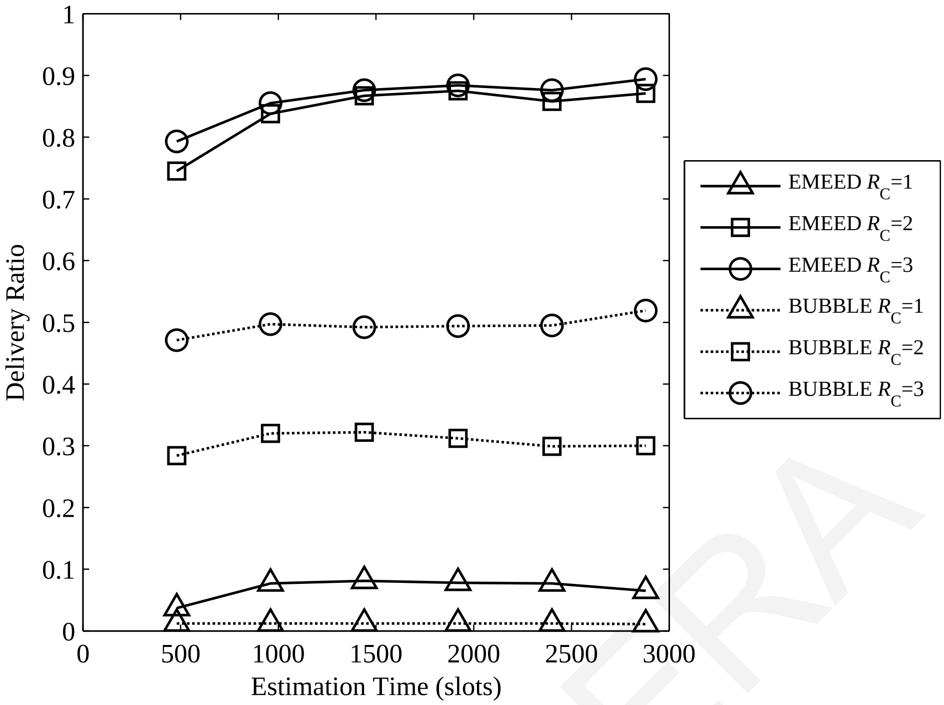


Figure 6.8: Delivery ratio versus estimation time for the PSN setting.

Fig. 6.7 shows results for the delivery ratio versus the number of friends N_F. We see that we do not need a large value of N_F to get best results. Therefore we can achieve good delivery rates with small overheads.

Fig. 6.8 shows the delivery ratio versus the estimation time T_E. Fig. 6.9 shows results for the delivery ratio versus the dissemination time T_D. We observe that the estimation and the dissemination of the expected wait times takes less time than the time needed to deliver the data packets using EMEED. The dissemination of the expected wait times is done using a flooding-based protocol and thus it is done faster than the data packet delivery using EMEED.

Fig. 6.10 shows the delivery ratio versus the estimation time and the dissemination time for EMEED with $R_C = 3$. We observe that when the estimation time increases the delivery ratio increases. We also observe that when the dissemination time increases the delivery ratio increases. When the dissemination time is only 5 days (480 slots) the delivery ratio is small. The dissemination time needs to be at least 10 days (960 slots) to get high delivery ratio. However when the estimation time is only 5 days the delivery ratio is not small. To conclude, of the two processes, i.e., estimation and dissemination, the second one is the most time consuming.

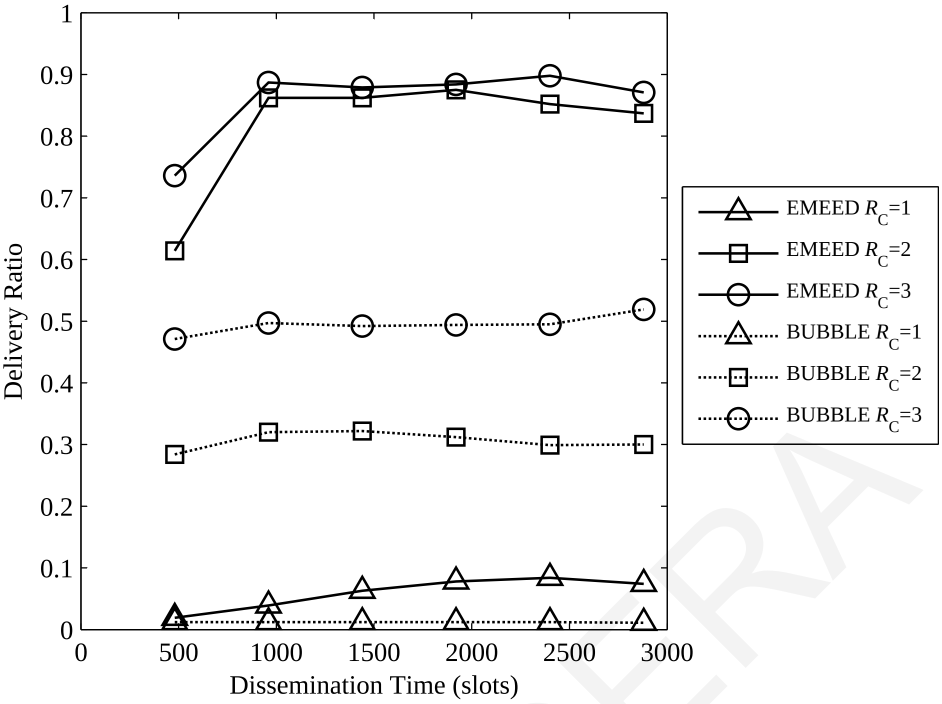


Figure 6.9: Delivery ratio versus dissemination time for the PSN setting.

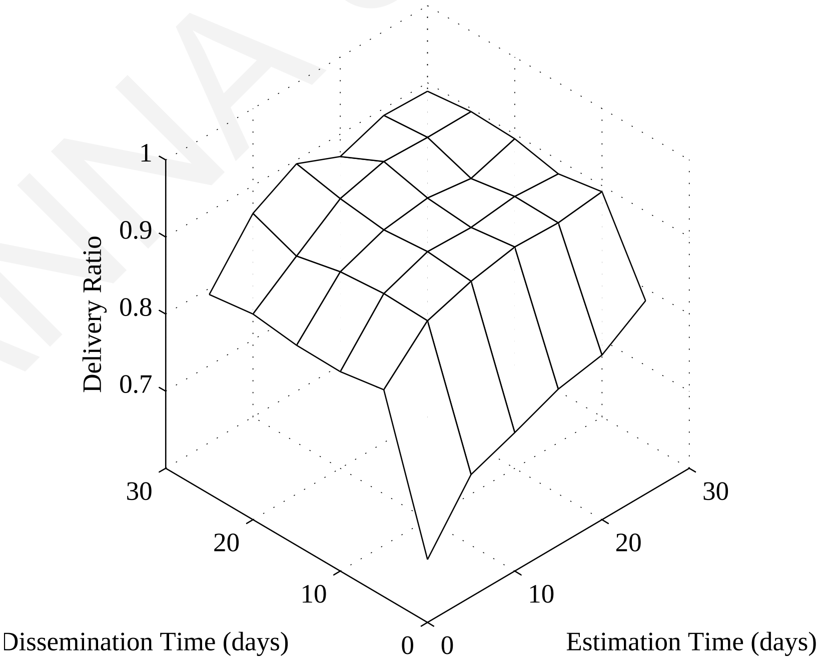


Figure 6.10: Delivery ratio versus estimation time and dissemination time for EMEED with $R_C = 3$ for the PSN setting.

6.7 Performance Evaluation in Vehicular DTN

We also present simulation results using SUMO [114]. SUMO is an open source road traffic simulation tool capable of simulating very large road networks. It is developed by the Institute of Transportation Systems at the German Aerospace Center.

We use a map of Dublin, Ireland that has dimensions 4078.81 m and 3830.50 m. This map was used in [113].

6.7.1 Mobility Model

We have 200 *persistent nodes* that are separated in communities of 4 nodes. For each community we select randomly and uniformly on the map 2 junctions for which there is a route from each of the two junctions to the other one and that the shortest route from each of the two junctions to the other one has length at least 1000 m. The nodes in the community travel non stop from one junction to the other, using shortest paths. Also at each slot we have 200 *transient nodes* that appear in the system at positions chosen randomly and uniformly on the area of the map only for that slot and then they disappear. The simulation takes as input the area of the map and the average node degree and it calculates the transmission range.

Unless otherwise stated, the parameters used in the simulations are those of Table 6.2.

6.7.2 Results

Fig. 6.11 shows the delivery ratio versus the average node degree. We observe that the delivery ratio of BUBBLE is actually superior to that of EMEED. As expected, flooding performs best (at the cost of extensive packet replication) and Spray and Wait performs the worst, among all protocols. The explanation for this is that, in this setting, some nodes have large global centrality, due to the fact that the routes they follow take them through much frequented areas of the map. BUBBLE correctly identifies these nodes as potentially useful in delivering packets to their destinations. Comparing the situation with that of the PSN, we draw the conclusion that the distribution of the values of the global centralities is crucial to the performance of protocols such as BUBBLE that employ them. However, as BUBBLE makes use of packet replication, it will congest the channel sooner, and hence its performance will be much diminished in

PARAMETER	NUMERICAL VALUE
Number of persistent nodes	$n_1 = 200$
Number of transient nodes	$n_2 = 200$
Area dimension	4078.81 m
Average node degree	$N = 2$
Estimation time	$T_E = 3600$ slots
Dissemination time	$T_D = 3600$ slots
Slot duration	1 sec
Packet TTL	3600 slots
Number of persistent nodes per community	4
Buffer size	$B = \infty$
Routing table cost threshold in EMEED	$C_T = 3600$ slots
Number of friends in EMEED	$N_F = 10$
Centrality time in BUBBLE	$T_C = 1$ sec
Number of replicas in Spray and Wait	$L = 10$

Table 6.2: Default environment and protocol parameters for the vehicular DTN setting.

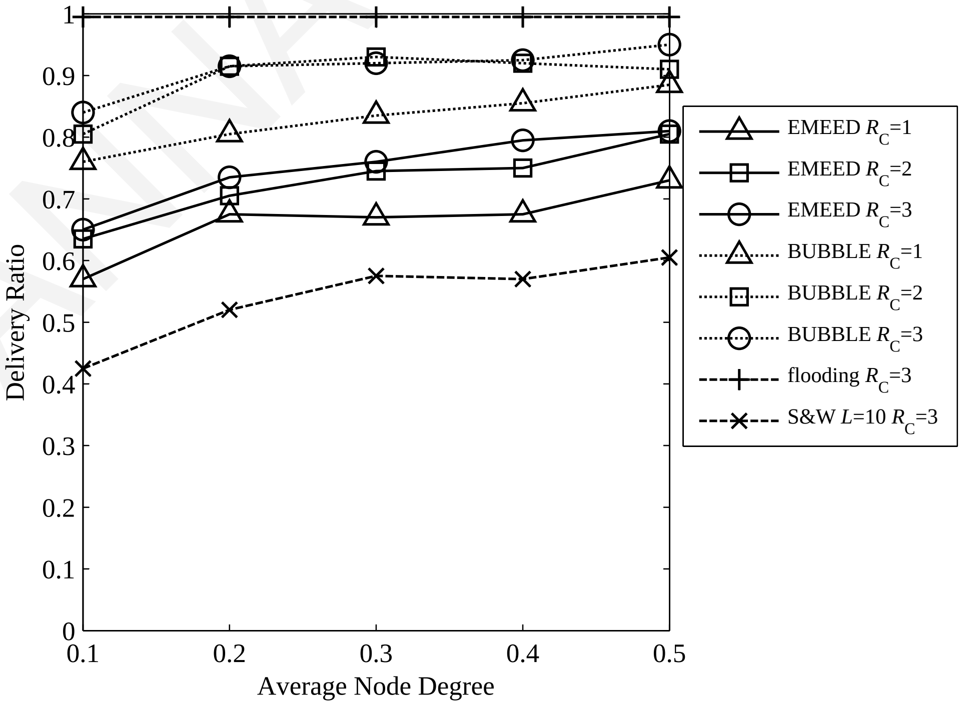


Figure 6.11: Delivery ratio versus average node degree for the vehicular DTN setting.

comparison to the performance of EMEED for a wide range of settings. We elaborate on this issue in the subsequent sections.

6.8 Performance Evaluation in Terms of Transmitted Packets

Comparing the control overhead in EMEED and BUBBLE we see that in EMEED each node i needs to know which nodes are R_C hops away from it at any time to estimate for every other node j the expected wait time $E[WT(i,j)]$. In BUBBLE each node needs to know which nodes are R_C hops away from it at any time to estimate its centrality within its community and its global centrality. Therefore, in both EMEED and BUBBLE, each node needs to inform all nodes that are R_C hops away from it about its presence at fixed time intervals. This is the only control overhead that BUBBLE needs. However, in EMEED the nodes also need to disseminate in the network some of the expected wait times they estimate as explained earlier. So the extra overhead that EMEED needs compared to BUBBLE is the control overhead for the dissemination of the expected wait times. However EMEED creates only one copy of each data packet that makes a number of hops to reach the destination while BUBBLE creates many copies of each data packet.

We plot the number of data packet transmissions divided by the number of delivered data packets versus the average node degree in Figs. 6.12 and 6.13. We use the same simulation scenarios and parameters we used for Figs. 6.4 and 6.11 respectively. We observe that BUBBLE needs more transmissions per data packet than EMEED. We see that in the vehicular DTN setting, to achieve its superior performance in terms of delivery ratio, BUBBLE resorts to aggressive packet replication. The figure reveals that BUBBLE needs a little more than one order of magnitude more transmissions, with respect to EMEED, and so will saturate the wireless channel much earlier than EMEED. Therefore, we find that, if the volume of data that each node creates is large, it is better to use EMEED and have the extra control overhead needed by EMEED compared to BUBBLE, that does not increase with the volume of the created data, than to use BUBBLE and have the extra data packet transmissions needed by BUBBLE compared to EMEED, that increases with the volume of the created data.

Fig. 6.14 shows the number of data packet transmissions divided by the number of

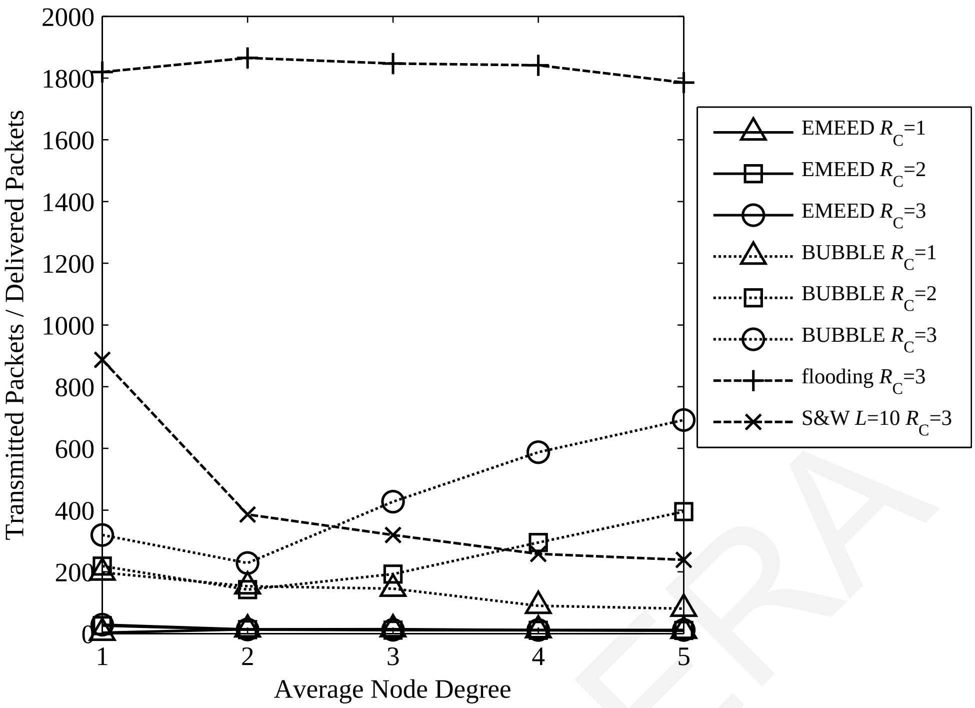


Figure 6.12: Number of data packet transmissions divided by number of delivered data packets versus average node degree for the PSN setting.

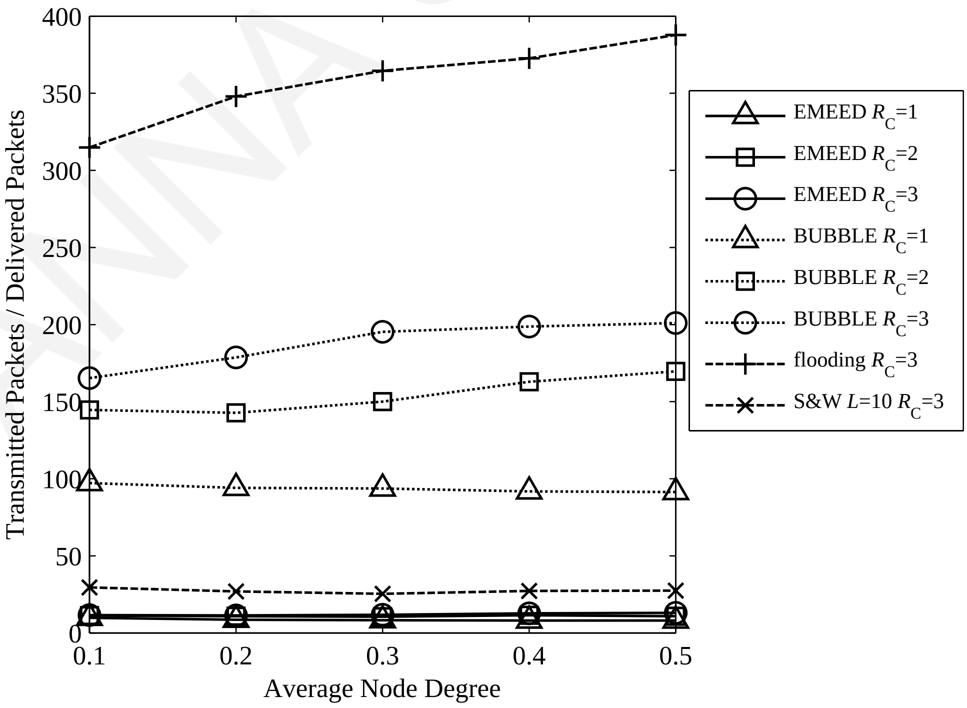


Figure 6.13: Number of data packet transmissions divided by number of delivered data packets versus average node degree for the vehicular DTN setting.

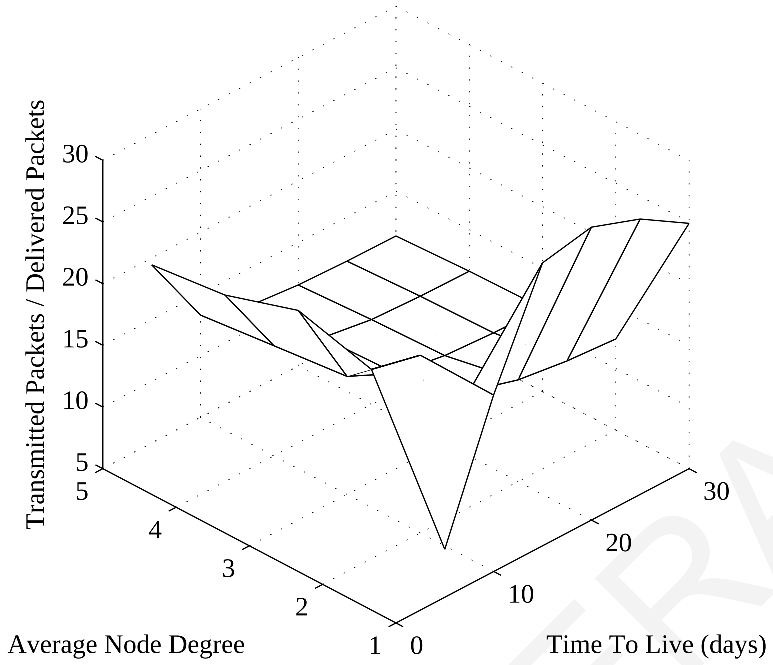


Figure 6.14: Number of data packet transmissions divided by number of delivered data packets vs TTL and average node degree for $R_C = 3$ for the PSN setting.

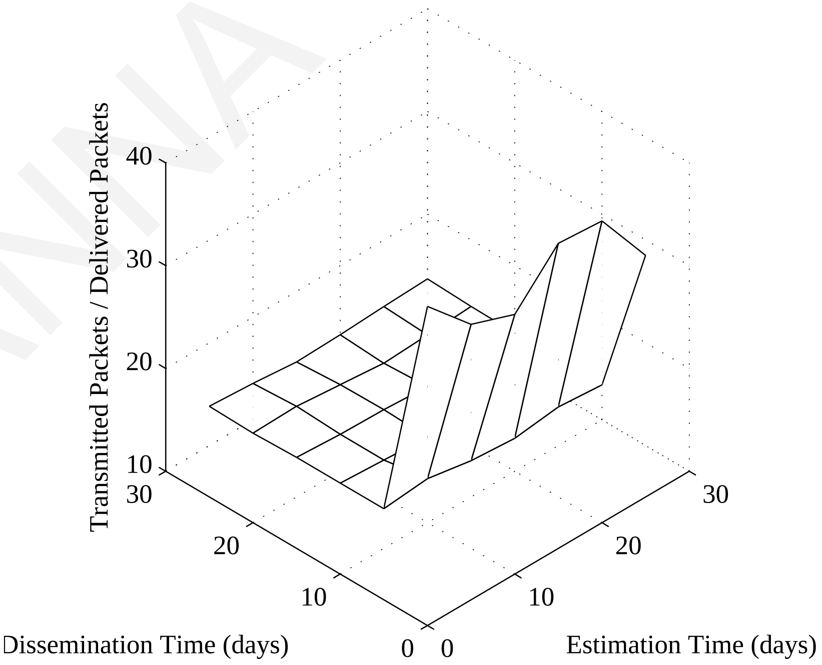


Figure 6.15: Number of data packet transmissions divided by number of delivered data packets vs estimation time and dissemination time for $R_C = 3$ for the PSN setting.

delivered data packets versus the TTL and the average node degree for EMEED with $R_C = 3$ for the PSN setting. We observe that when the TTL is at least 10 days and the average node degree is at least 2 we get good results. Further increasing either the TTL or the average node degree does not lead to any perceptible gain. Fig. 6.15 shows results for the number of data packet transmissions divided by the number of delivered data packets versus the estimation time and the dissemination time for EMEED with $R_C = 3$ for the PSN setting. We observe that when the estimation time is at least 5 days and the dissemination time is at least 10 days we get good results. Increasing these times further does not lead to any gain.

6.9 Evaluation of the Control Overhead

In this section we evaluate the EMEED protocol in terms of the control overhead needed to estimate the expected wait times and disseminate them in the network.

6.9.1 Estimation of Expected Wait Times

As already discussed, at predefined time intervals, each node i sends a control message to each node j that is at most R_C hops away from i, notifying it that i and j are in contact. Node j uses this information to calculate $E[WT(i,j)]$. These control messages are also used so that the nodes know which links are up at any time and so which costs to set to 0 in the creation of their expected delay routing table.

We assume unicast transmission. Each node i sends the control message to each node that is 1 hop away, with a unicast packet. In turn, each of these nodes sends the control message to each node that is 1 hop away from itself except i. Then each node that is 2 hops away from i sends the control message to each node that is 1 hop away from itself except from the node that has sent the message to it, and so on.

Fig. 6.16 shows simulation results for the number of control messages per node per slot needed for the estimation of the expected wait times versus the contact radius R_C for various values of the average node degree N. We use the same simulation scenario and parameters that we used to produce Fig. 6.4. We see that, as the value of R_C increases, the number of control messages also increases. The number of control messages increases modestly for $N = 1$. For example, for $R_C = 1$ and for $R_C = 2$ the difference in control messages needed is not large and the difference in data packet

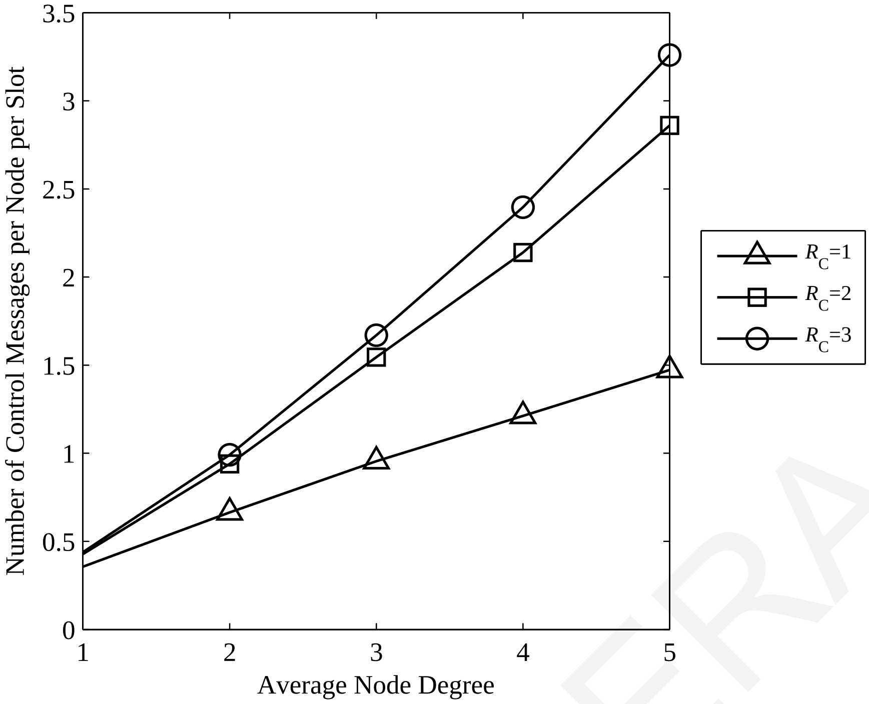


Figure 6.16: Number of control messages per node per slot transmitted for the estimation of the expected wait times vs average node degree N for the PSN setting.

delivery ratio is significant.

Observe that the volume of these messages per node remains constant as the number of nodes in the network increases. For this reason, unless the topology of the network changes very fast, we expect that only a modest fraction of the bandwidth will be consumed for their propagation.

6.9.2 Dissemination of Expected Wait Times

The control overhead needed for the dissemination of the expected wait times increases with the network size, and hence is expected to be more substantial.

In order to keep this control overhead at low levels, the expected wait times that a node A estimates at a certain time are always sent as a set. The set contains the id of node A, a timestamp saying when node A estimated these expected wait times, the node ids of the friends of node A, and the expected times node A has to wait until it comes in contact with its friends. In our simulation each node creates one set of expected wait times at time T_E. In practice, each node creates a set of expected wait times some time after it joins the network, and then, whenever the expected wait times

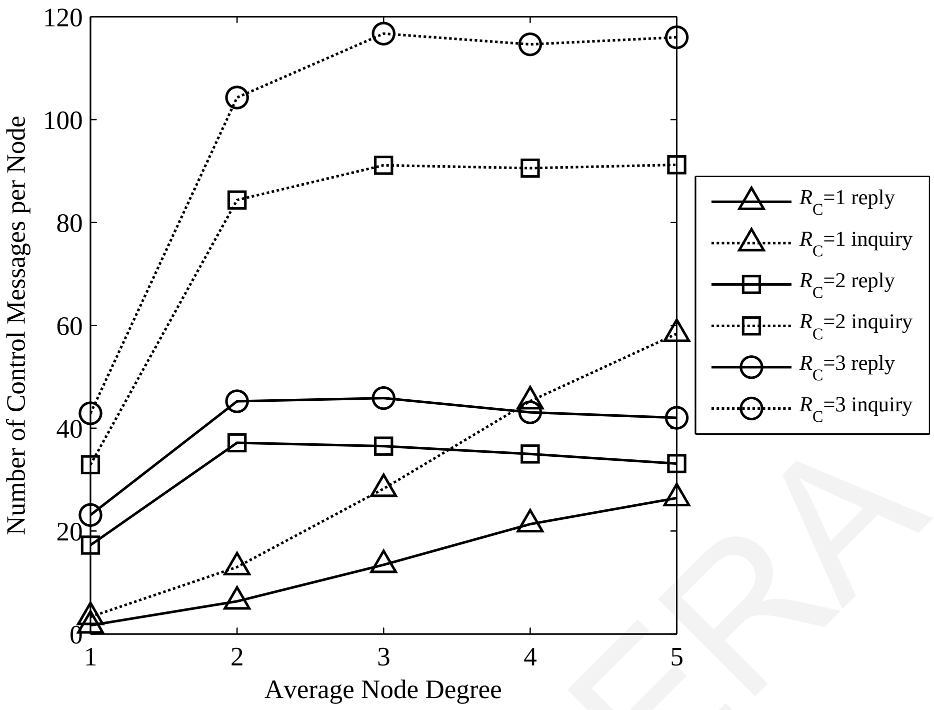


Figure 6.17: Number of control messages transmitted per node for the dissemination of the expected wait times versus the average node degree N for the PSN setting.

it estimates change significantly or its friends change it creates a new set of expected wait times.

The sets of expected wait times the nodes create are disseminated in the network using the following algorithm. Each node i stores in its memory the time it met every other node j for the last time. When a node i receives a set of expected wait times that a node k created it stores it and stores the time it received it. When a node i meets another node j, i sends to j an inquiry packet that says which sets of expected wait times it received after its last meeting with j. Also this packet says if i itself created a set of expected wait times after its last meeting with j. If there are any of these sets that j does not have, j sends a packet to i that says which of these sets it does not have. Then i sends to j a reply packet containing them. The inquiry packet that i sends to j says only which are the sets that i received or created after its last meeting with j because for the rest of the sets it has it assumes that it gave them to j the last time they met or j already had them the last time they met.

Fig. 6.17 shows the number of control messages transmitted per node for the dissemination of the expected wait times versus the average node degree for the PSN setting.

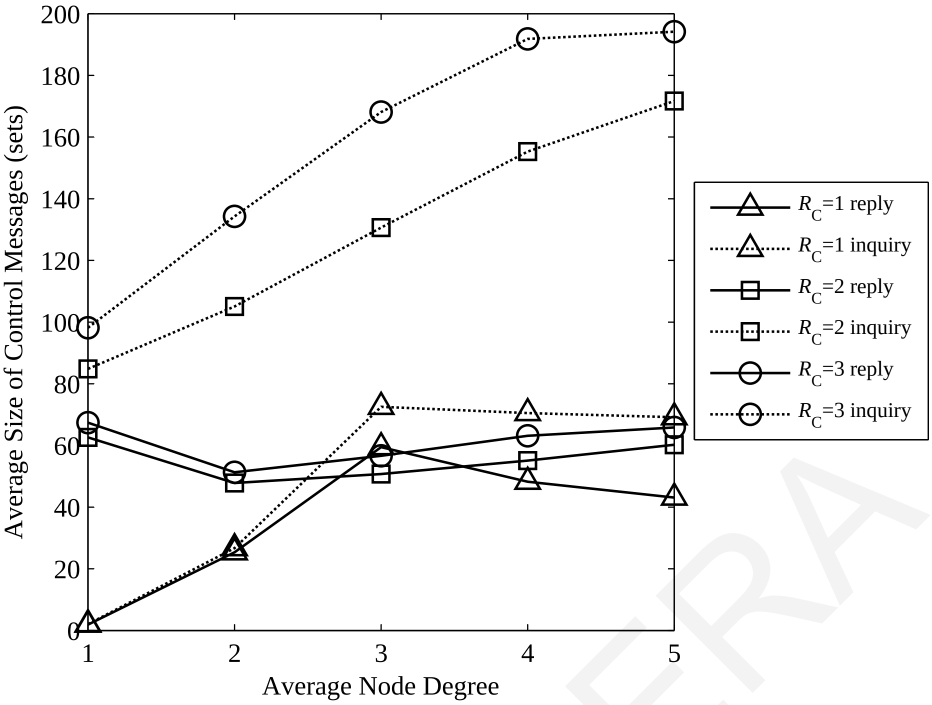


Figure 6.18: Average size of control messages transmitted for the dissemination of the expected wait times versus the average node degree N for the PSN setting.

Fig. 6.18 shows the average size of control messages transmitted for the dissemination of the expected wait times versus the average node degree for the PSN setting. We use the same simulation scenario and parameters that we used to produce Fig. 6.4. If a message is sent from a node A to a node B that is h hops away from A, the message is counted h times. The size of the inquiry packets is in sets of expected wait times they inquire about. The size of the reply packets is in sets of expected wait times they reply about. We observe that the average node degree does not have a strong effect on the size of the dissemination overhead.

Fig. 6.19 shows the number of control messages transmitted per node for the dissemination of the expected wait times versus the dissemination time for the PSN setting. Fig. 6.20 shows the average size of control messages transmitted for the dissemination of the expected wait times versus the dissemination time for the PSN setting. We observe that the number of control messages per node converges to an upper bound, as the dissemination duration increases. This is because, the expected delays do not change, and once they are propagated, they do not need to be propagated again. In realistic environments, the statistics of the topology change, with a rate that depends

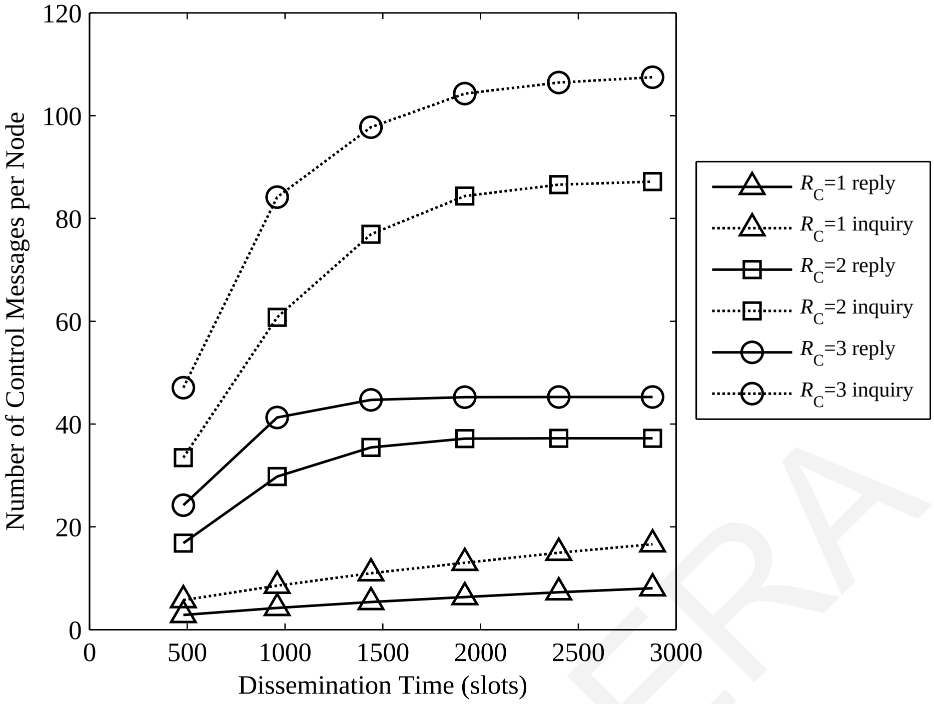


Figure 6.19: Number of control messages transmitted per node for the dissemination of the expected wait times versus the dissemination time T_D for the PSN setting.

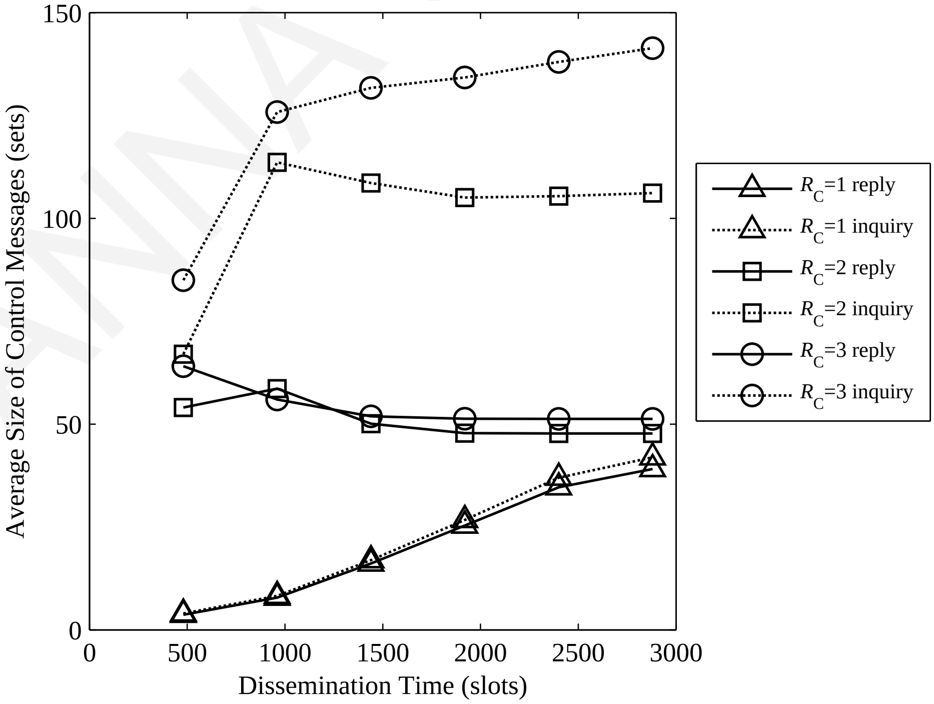


Figure 6.20: Average size of control messages transmitted for the dissemination of the expected wait times versus the dissemination time T_D for the PSN setting.

on the environment. The total dissemination overhead depends tightly on this rate. These plots provide us with a sense of the total dissemination overhead required every time the topology changes.

6.10 Conclusions

In this chapter we propose EMEED, a routing protocol for wireless DTNs in which some pairs of nodes systematically meet more often than others. EMEED is a non-trivial extension of the well known MEED protocol. The main parameter of EMEED is the contact radius R_C. We assume that two nodes are in contact if they are at most R_C hops away from each other. This means that each node knows which nodes are at most R_C hops away from it and also knows routes to these nodes. When $R_C = 1$, EMEED operates similarly to MEED which assumes that two nodes are in contact only if they can communicate directly. When $R_C > 1$ EMEED achieves better performance at the expense of higher control cost.

EMEED operates as follows: Each node i continuously monitors whether it is in contact with every other node j. Based on this information, i finds the expected value of the time $E[WT(i,j)]$ it has to wait until it comes into contact with j. The nodes disseminate the expected wait times in the network using a flooding-based protocol. Thus, each node k learns estimates of expected wait times for different pairs of nodes i and j. Then each node i sets the cost of the link between each pair of nodes j and k equal to $E[WT(j,k)]$ and then it sets the cost of the link between i and each node that is currently in contact with i equal to 0. Then it uses this graph and creates a routing table, which is used to make routing decisions.

We do simulations using two settings, one representing a PSN and one representing a vehicular DTN. We compare EMEED with MEED which is similar to EMEED with $R_C = 1$, with BUBBLE, which is a recently proposed state of the art protocol for DTNs in which some pairs of nodes systematically meet more often than others, with flooding, which is used as upper bound, and with Spray and Wait which is a baseline protocol. In both mobility models we have two kinds of nodes: The persistent nodes and the transient nodes. The persistent nodes stay in the network for the whole duration of the simulation. Each persistent node has some preferred locations of movement. Thus some pairs of persistent nodes systematically meet more often than others. Each persistent node creates a packet that has as destination another persistent node. Also at each slot we have transient nodes which are nodes that appear in the network only for a single slot and then disappear. The persistent nodes are separated in communities to be able to compare with BUBBLE that uses communities.

We observe that when there are transient nodes in the network EMEED with $R_C > 1$ performs better than EMEED with $R_C = 1$. This is because EMEED with $R_C > 1$ uses multihop paths that are partially comprised of transient nodes. EMEED with $R_C = 1$ ignores such paths.

In the PSN setting we observed that in almost all the cases that EMEED delivered a packet but BUBBLE did not deliver it, in the path that the packet followed in EMEED, the packet was forwarded from a node with larger global centrality to a node with smaller global centrality and this happened before the packet reached one of the communities of the destination. This means that EMEED used a link that BUBBLE would not use. BUBBLE has a mechanism to send the packet from the source to the community of the destination and a mechanism to send the packet from the community of the destination to the destination. In our setting the first mechanism does not work because there are no nodes with large global centrality. In the vehicular DTN setting BUBBLE performs a little better than EMEED in terms of delivery rates, because there are nodes with large global centrality, but the number of data packet transmissions that BUBBLE needs is much larger than this of EMEED.

In all the protocols we simulated, we assume that each node knows which nodes are at most R_C hops away from it. This needs control packets. Furthermore, EMEED also needs control packets for the dissemination of the expected wait times. However, EMEED creates only one copy of each data packet that does a number of hops to reach the destination, while BUBBLE creates many copies of each data packet. If the volume of data that each node creates is large, it is better to use EMEED and tolerate the extra control packet transmissions needed by EMEED compared to BUBBLE, that do not increase with the volume of the created data, than to use BUBBLE and suffer the extra data packet transmissions needed by BUBBLE compared to EMEED, that do increase with the volume of the created data.

Chapter 7

Conclusions

7.1 Contributions and Scope of this Thesis

The topic of this thesis is the design and study of routing protocols for DTNs. We focus mostly on vehicular DTNs and to a lesser extent PSNs. Although technology is changing fast and new applications and protocols appear all the time, these settings appear to be two of the most promising for the future commercial large scale deployment of DTNs.

In the first part of this thesis, we propose the DTFR protocol, which is a hybrid geographic delay tolerant routing protocol for wireless DTNs. We simulate DTFR in large networks and compare it to other state of the art protocols. We do analysis for GLF, which is the main forwarding mechanism of the DTFR protocol, and then we do analysis and simulation for different variants of GLF. The major result of our work is that it is possible to combine very successfully two basic mechanisms for routing, i.e., geographic routing and store carry and forward, into hybrid routing protocols that combine both mechanisms. Another outcome of this work is a very general framework for studying such routing protocols using tools of stochastic geometry.

DTFR is better than the other protocols we compare it with when *(i)* there is no end-to-end path from a packet's source to its destination but there are multihop paths and *(ii)* the nodes are mobile but their velocity is not exceedingly large.

Indeed, under DTFR, the average velocity with which the packet travels from the source to the Firework Center (FC) and from the FC to the Firework Endpoints (FEs) increases as the transmission range increases, and it is greater than the velocity of the nodes for a wide range of transmission ranges for which there is no end-to-end path from source to destination. Remember that the distance between the FC and the FEs is chosen so that the time the packet needs to travel from the source to the FC and then from the FC to the FEs is the same or smaller, with high probability, than the time it takes for the destination to travel from the FC to the FEs. Therefore, DTFR delivers the packet with high probability. On the other hand, when there is no end-to-end path from source to destination, traditional geographic routing fails. Also, under the Spray and Wait protocol, in the wait phase the packets travel toward the destination with velocity at most equal to the velocity of the nodes; thus for many node velocities, under Spray and Wait the packets cannot cover the distance between the source and the destination before their TTL expires.

On the other hand, for large node velocities Spray and Wait performs well because the packets can cover the distance between the source and the destination by transport before their TTL expires. Furthermore, for small transmission ranges DTFR does not perform well because in this case the velocity with which the packet travels is small, thus many packets cannot reach the destination before their TTL expires. Finally, when the nodes are immobile or move very slowly DTFR again does not perform well because the packets get stuck in the wait phase for excessive amounts of time.

A possible application of DTFR is cooperative content sharing [93, 115]. In [93], a node A disseminates information about data it has via k-hop broadcasting. A node B that receives this information can send a query to A about data it needs. Then A can send the data to B. The query and the data are sent using AODV. This setting could be a possible application of DTFR. In the case of first time communication, no knowledge of the destination location is needed, as the message is send using limited flooding. After this, nodes have an estimate of the location of their destinations, and so can exchange the rest of the messages using DTFR. As another example, in [47] the authors observe that a moving vehicle may want to query an infostation about the availability of a parking space or a conference room, etc. In these cases the moving vehicle can use a database to learn the position where it wants to sent the query. It can send its position in the query so that the node that will reply will know where to send the reply, using DTFR. As a third example, a police vehicle can send a query at a certain location using a flooding-based protocol to find out if any private vehicle has recorded any video at that point some time before. The private vehicles can use DTFR to reply to the police vehicle while it is moving.

In the second part of this thesis, we propose the EMEED protocol, for use in wireless

DTNs in which the nodes have preferred locations of movement and the movement patterns of different nodes are correlated, so that the statuses of different links are also correlated. We simulate EMEED and compare it to other state of the art protocols in two settings: a vehicular DTN and a PSN. We find that EMEED is overall much better than MEED, from which EMEED has been inspired, and BUBBLE, a well known state of the art protocol. More importantly, we make a number of observations of wider interest. First of all, transient nodes, i.e., nodes that are present to the network for limited amounts of time so that no statistics involving them have been collected, have a very large effect to the performance of protocols; regrettably, they have typically been ignored in the literature until now; EMEED solves this problem. Secondly, the correlation of the mobility patterns of different nodes can lead to the wrong routing decisions in many cases. Lastly, in the case of protocols, such as BUBBLE, that use the global and local centralities of nodes for routing decisions, the distribution of these centralities is crucial to the performance of the protocol.

Overall, EMEED is better than the other protocols we consider when *(i)* some pairs of nodes systematically meet more often than others, *(ii)* there are transient nodes, and *(iii)* there are no nodes with large global centrality or the volume of data is large.

In more detail, first observe that, when there are transient nodes in the network EMEED with $R_C > 1$ performs better than MEED. This is because EMEED with $R_C > 1$ uses multihop paths that are partially comprised of transient nodes. MEED ignores such paths. Regarding BUBBLE, this protocol has a mechanism for sending each packet from the source to the community of its destination and a mechanism for sending that packet from the community of its destination to its destination. In the case of the first mechanism, BUBBLE assumes that the packet will be given to nodes with large global centrality and that these nodes will give the packet to a node in the community of the destination. In the PSN setting we simulated, the first mechanism of BUBBLE does not work because there are no nodes with large global centrality. In the vehicular DTN setting we simulated, BUBBLE performs a little better than EMEED, because there are nodes with large global centrality, but, on the other hand, the number of transmissions that BUBBLE requires is much larger than the number of transmissions that EMEED requires. This is because EMEED does not create data packet copies, while BUBBLE creates many copies of each data packet. However, EMEED needs control packets for the dissemination of the expected wait times. Therefore, if the volume of data that each node creates is large, it is better to use EMEED and tolerate the extra control packet transmissions needed by EMEED compared to BUBBLE, that do not increase with the volume of the created data, than to use BUBBLE and tolerate the extra data packet transmissions needed by BUBBLE compared to EMEED, that increase with the volume of the created data.

In a vehicular DTN scenario the transient nodes would correspond to cars that are at a location that they do not visit often, whereas the persistent nodes would be cars that are parked outside their owner's home, or office, or any other location the owner frequents. If a bus comes several times a day near an access point, and has a multi-hop path to it that consists of cars, and the cars are different every time, the persistent nodes would be the bus and the access point and the transient nodes would be the cars. Furthermore, there are scenarios in which it is possible that the nodes have small global centrality. An example is a network where the persistent nodes are cars that are parked outside their owner's home, or office, or other places the owner visits often and the cars exchange packets only when they are parked. Another example is a network where the persistent nodes are smart phones in the pockets of people. Another scenario in which EMEED can be used is the following. A police vehicle can send a query at a certain point using a flooding-based protocol to find out if any private vehicle has recorded any video at that point some time before. Assume that a number of police vehicles do routes that are always the same. Each private vehicle has a few places where it is parked often. Some of these places are in the routes of police vehicles, so that private vehicles parked at these places come in contact with police vehicles, and some are not. The private vehicles can use EMEED to send the videos to the police vehicles.

7.2 Future Work

Our work opens up numerous new directions for future research. In concluding this thesis, we list some of them.

Firstly, our study by simulation of the Delay Tolerant Firework Routing (DTFR) could be extended in numerous directions:

- One such direction is a more careful incorporation of a location service and its interaction with DTFR.
- Another possibility is the evaluation of its performance in realistic road mobility

scenarios employing SUMO [114].

- Also we can simulate DTFR in a scenario representing a specific application and develop a simulation that simulates the routing protocol, the location service, and this application as they interact with each other.

Secondly, our analytical investigation of Cost/Delay tradeoffs of DTNs on the infinite plane (Chapters 4 and 5) could be extended in numerous promising directions:

- Other aspects of the network model can be considered, for example alternative cost models and metrics other than the normalized cost and normalized delay.

- Alternative forwarding rules can be studied, both by simulation and also by analysis. For example, we could investigate forwarding rules that take into account more extensive information about the neighborhood of the packet holder, but not complete information, as was done with Rule III.

- More accurate approximations could be adopted, with respect to the First Order Approximation. For example, we could consider approximations that maintain more information about the stochastic process than simply the direction of travel of the current packet holder. For example, information about the fact that some parts of the forwarding region are empty could also be maintained.

- An important open issue is the establishment of the Pareto Optimal delay/cost curves, under various assumptions on the level of information known by the current packet holder. Although ideally these should be established analytically, as a first tentative step, these could be established numerically. One possibility is to assume that each node has complete knowledge about the whole network, and use the network optimization formulation of Tasiopoulos [113]. A tentative step in this direction was taken in this thesis. It might be possible to employ multi armed bandit techniques [116]: the packet tries different strategies, monitoring for each the combination of normalized cost and delay achieved, eventually converging to the optimal one.

- Also we can simulate the different forwarding rules we studied using mobility models different from the one we used so that we will have results that do not hold only for the specific mobility model.

- We can investigate how the nodes can choose the optimal forwarding rule after estimating information about the environment, for example node density and velocity of nodes, and taking into account the demands of the application, for example delay per progress or cost per progress.

- We can investigate how the fact that we use a torus for the simulation introduces errors in the results and if it is possible to write a simulation that does not have these errors or at least approximates better the results we would have if the nodes where moving on an infinite plain.

Finally, our study and performance evaluation of the EMEED protocol can also be extended in different directions:

- The primary challenge of a successful implementation of EMEED is its control overhead, in the form of connectivity packets and, mostly, dissemination packets. Therefore, an important issue remaining is the development of more mechanisms for keeping the control overhead on manageable levels.

- Alternative mechanisms for evaluating expected delays are also needed. One major challenge is the following: in reality, expected delays are functions of time. This is because human mobility changes with time. Furthermore, when calculating minimum expected delay paths, we cannot use the correct values for the expected delays of all links other than the first ones in the paths we consider, because we do not know when exactly we will arrive in all other links.

- Simulations in more realistic settings are needed. In the case of the Pocket Switched Network, we resorted to using our own model, due to the lack of realistic traces that are freely available. We note that there are numerous traces involving large numbers of nodes, however typically these traces do not contain information on the connectivity between all pairs of nodes, but only connectivity information between the nodes and a set of access points or base stations. It is possible that utilizing these traces, after a suitable modification, will increase the accuracy of our simulations.

- We should investigate more in which settings BUBBLE works well or not and why, as EMEED would be more useful for the settings for which BUBBLE does not work well. It is also of independent interest to investigate in which settings BUBBLE works well because it is a very well known protocol.

- We can change our simulation for EMEED and the protocols we compared it with, so that we model congestion like we did in the simulation for DTFR. Thus, instead of having plots for the delivery ratio and the number of transmitted packets, as we now have for EMEED and the protocols we compared it with, we will have only plots for the delivery ratio. Both approaches are useful in the simulation, the one currently used because it is simpler, and the proposed one because it is closer to reality and because it allows the measurement of the control overhead per type of control packet. We expect that modeling congestion will not affect the speed of the simulation significantly; what affects the speed of the simulation significantly is increasing the number of nodes in the network.

- Finally, we could change our mobility model for the PSN setting so that instead of using communities it uses clouds, where the cloud of a node is the set of nodes it comes in contact with often [67]. It remains to be seen which approach is more realistic.

Irrespective of the issues listed above, the field of DTN routing is changing fast. Any future work will have to take into account emerging applications, available technologies (for example UAVs), and competing paradigms (for example, satellite communications, WiMax, etc. that lead to connected networks).

Appendix A

Calculation of Incidence Rates

With respect to Chapter 5, here we assume that $\chi \in [\theta, \theta{+}2\pi]$ instead of $\chi \in [-\pi, \pi]$. This is adopted for reasons of mathematical convenience, as we show later on. As all incidence rates are periodic, with a period 2π, with respect to χ, this assumption is not in any way restrictive.

A.1 Calculation of $\gamma(\chi, \phi; \theta)$

Note that the relative speed with which a node B with direction χ moves, as viewed by $\mathcal{F}$, is, using phasor notation,

$$\begin{aligned}
v_n e^{j\chi} - v_n e^{j\theta} &= v_n e^{j\chi} + v_n e^{j(\theta+\pi)} \\
&= 2v_n \exp\left(j\left[\frac{\chi+\theta+\pi}{2}\right]\right) \times \left[\frac{\exp\left(j\left[\frac{\chi-\theta-\pi}{2}\right]\right) + \exp\left(j\left[\frac{\theta+\pi-\chi}{2}\right]\right)}{2}\right] \\
&= 2v_n \exp\left(j\left[\frac{\chi+\theta+\pi}{2}\right]\right) \cos\left(\frac{\pi}{2} - \frac{\chi-\theta}{2}\right) \\
&= \left[2v_n \sin\left(\frac{\chi-\theta}{2}\right)\right] \exp\left(j\left[\frac{\chi+\theta+\pi}{2}\right]\right).
\end{aligned}$$

A graphical explanation of this result appears on the left of Fig. A.1. The two vectors $v_n e^{j\chi}$ and $-v_n e^{j\theta} = v_n e^{j(\theta+\pi)}$, as well as their sum, are drawn with thick arrows. Observe that, due to the range $[\theta, \theta+2\pi]$ that we selected for the angle χ, $\sin\left(\frac{\chi-\theta}{2}\right)$ is never negative, therefore the direction of the relative speed is always $\frac{\chi+\theta+\pi}{2}$.

As the angle with which nodes seem to be approaching $\mathcal{F}$, as viewed from an observer sitting on $\mathcal{F}$, is $\frac{\chi+\theta+\pi}{2}$, it is important that the part of the boundary corresponding to the angles $[\phi, \phi+d\phi]$ (through which we want to calculate the arrival rate

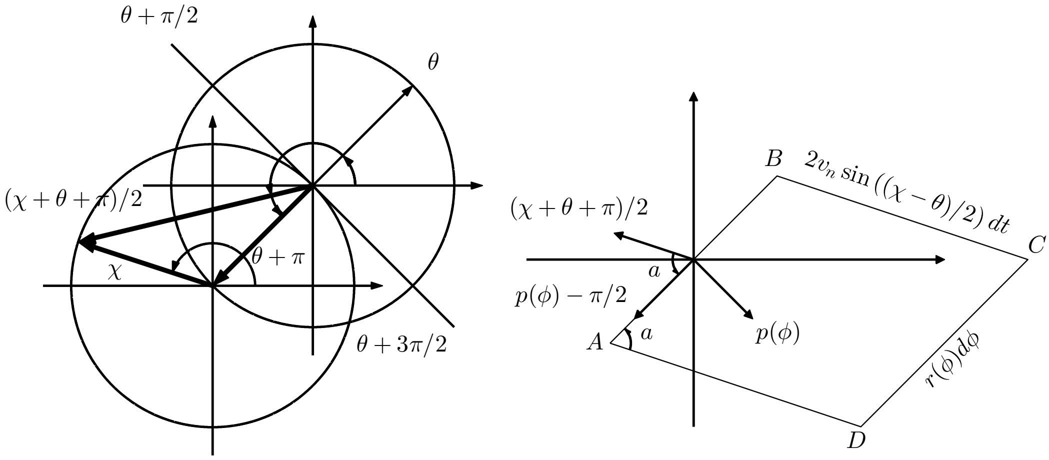


Figure A.1: Plots used in the calculation of the incidence rate $\gamma(\chi, \phi; \theta)$.

in $\mathcal{F}$) is facing them. Therefore, we must have

$$p(\phi) + \frac{\pi}{2} \leq \frac{\chi+\theta+\pi}{2} \leq p(\phi) + \frac{3\pi}{2} \Leftrightarrow \frac{\chi+\theta+\pi}{2} + \frac{\pi}{2} \leq p(\phi) \leq \frac{\chi+\theta+\pi}{2} + \frac{3\pi}{2}$$
$$\Leftrightarrow \frac{\chi+\theta}{2} - \pi \leq p(\phi) \leq \frac{\chi+\theta}{2} \Leftrightarrow p^{-1}\left(\frac{\chi+\theta}{2} - \pi\right) \leq \phi \leq p^{-1}\left(\frac{\chi+\theta}{2}\right). \quad \text{(A.1)}$$

Note that the last expression should be interpreted with care when $p(\phi)$ is not continuous. Also note that the above inequalities should be interpreted in a modulo 2π sense. This means that if they are satisfied for one set of angles, they are also satisfied for any other set of angles derived from the previous set by adding/subtracting to each of these (possibly distinct) multiples of 2π. Observe that we are using the fact that $p(\phi)$ is increasing, therefore it has an inverse function $p^{-1}(\cdot)$.

The setting that applies to the calculation of the incidence rate appears on the right of Fig. A.1. The boundary of $\mathcal{F}$ through which nodes enter $\mathcal{F}$ is the line segment AB of length $r(\phi)d\phi$. Over an infinitesimal time period dt, the nodes to cross AB are exactly those that are within the parallelogram $ABCD$, whose side BC has length equal to $2v_n \sin\left(\frac{\chi-\theta}{2}\right) dt$, and whose angle BAD is equal to

$$a = p(\phi) - \frac{\pi}{2} - \frac{\chi+\theta+\pi}{2} = p(\phi) - \frac{\chi+\theta}{2} - \pi.$$

Taking into account that the number of nodes within this parallelogram is Poisson distributed with incremental rates $\lambda\frac{d\chi}{2\pi}$, it follows that the average number of nodes to cross AB from time t until time $t + dt$ is

$$\begin{aligned}
2v_n \sin\left(\frac{\chi - \theta}{2}\right) r(\phi) d\phi (\sin a) \lambda \frac{d\chi}{2\pi} dt = \\
&= -\left(\frac{v_n \lambda r(\phi)}{\pi}\right) \sin\left(\frac{\chi - \theta}{2}\right) \sin\left(\pi - p(\phi) + \frac{\chi + \theta}{2}\right) d\phi d\chi dt \\
&= \left(\frac{v_n \lambda r(\phi)}{\pi}\right) \sin\left(\frac{\chi - \theta}{2}\right) \sin\left(\frac{\chi + \theta}{2} - p(\phi)\right) d\phi d\chi dt.
\end{aligned}$$

It follows that the rate with which nodes with direction of travel in $[\chi, \chi + d\chi]$ enter $\mathcal{F}$ at the portion of the boundary corresponding to the range $[\phi, \phi + d\phi]$, when $\mathcal{F}$ travels with direction θ is $\gamma(\chi, \phi; \theta) d\chi d\phi$ where

$$\gamma(\chi, \phi; \theta) = \left(\frac{v_n \lambda r(\phi)}{\pi}\right) \sin\left(\frac{\chi - \theta}{2}\right) \sin\left(\frac{\chi + \theta}{2} - p(\phi)\right), \tag{A.2}$$

where

$$\theta \in [-\pi, \pi], \quad \chi \in [\theta, \theta + 2\pi], \quad p^{-1}\left(\frac{\chi + \theta}{2} - \pi\right) \leq \phi \leq p^{-1}\left(\frac{\chi + \theta}{2}\right). \tag{A.3}$$

When ϕ is not in the set specified in the last of (A.3), $\gamma\left(\chi, \phi; \theta\right) = 0$.

A.2 Calculation of $\gamma(\chi; \theta)$

To calculate $\gamma(\chi; \theta)$, in principle we can substitute $\gamma(\chi, \phi; \theta)$ (given in (A.2)) in (5.10) and integrate. However, using the technique employed in Lemma 1 (Section 4.7), we arrive at

$$\gamma(\chi; \theta) = \left(\frac{v_n \lambda}{\pi}\right) P\left(\frac{\chi + \theta + \pi}{2}\right) \sin\left(\frac{\chi - \theta}{2}\right), \qquad -\pi \leq \theta \leq \pi, \qquad \theta \leq \chi \leq \theta + 2\pi. \tag{A.4}$$

The above can also be written as

$$\gamma(\chi; \theta) = \left(\frac{v_n \lambda}{\pi}\right) P\left(\frac{\chi + \theta + \pi}{2}\right) \left|\sin\left(\frac{\chi - \theta}{2}\right)\right|, \qquad -\pi \leq \theta, \chi \leq \pi. \tag{A.5}$$

A.3 Calculation of $\gamma(\phi; \theta)$

We will use (5.11). To this effect, let $\mathcal{R}$ be the intersection of $[-\chi_m, \chi_m]$ with the range of angles $\chi \in [\theta, \theta + 2\pi]$ for which condition (A.1) holds, such that the integrand

$\gamma(\chi, \phi; \theta)$ is not zero. We have

$$\begin{aligned}
\gamma(\phi;\theta) &= \int_{\mathcal{R}} \left(\frac{v_n \lambda r(\phi)}{2\pi}\right) 2\sin\left(\frac{\chi-\theta}{2}\right) \sin\left(\frac{\chi+\theta}{2} - p(\phi)\right) \, d\chi \\
&= \frac{v_n \lambda r(\phi)}{2\pi} \int_{\mathcal{R}} \left[\cos\left(\frac{\chi}{2} - \frac{\theta}{2} - \frac{\chi}{2} - \frac{\theta}{2} + p(\phi)\right) - \cos\left(\chi - p(\phi)\right)\right] d\chi \\
&= \frac{v_n \lambda r(\phi)}{2\pi} \int_{\mathcal{R}} \left[\chi \cos(p(\phi) - \theta) - \sin(\chi - p(\phi))\right]' \, d\chi.
\end{aligned}$$

In the previous, we used the identity

$$2 \sin a \sin b = \cos(a-b) - \cos(a+b).$$

To conclude,

$$\gamma(\phi;\theta) = \frac{v_n \lambda r(\phi)}{2\pi} \int_{\mathcal{R}} \left[\chi \cos(p(\phi) - \theta) - \sin(\chi - p(\phi))\right]' \, d\chi. \tag{A.6}$$

What is left, is to find explicit expressions for $\mathcal{R}$. We will have to take cases. The complexity arises from the fact that $\mathcal{R}$ is an intersection of two intervals of χ that must be taken modulo 2π. In the following, to simplify the notation, we will denote $p(\phi)$ simply by p. All cases appear graphically and in a tabular form in Fig. A.2.

1. **Case A:** $\boxed{\theta \leq p \leq \theta + \pi}$, from which it readily follows that $\theta \leq 2p - \theta \leq \theta + 2\pi$. In this case, as a preliminary, note that

$$\frac{\theta + \chi + \pi}{2} = p + \frac{\pi}{2} \Leftrightarrow \chi = 2p - \theta.$$

Let $\chi_1 = 2p - \theta$. χ_1 is the smallest value of χ in the range $[\theta, \theta + 2\pi]$ for which the nodes arrive at the location $[\phi, \phi + d\phi]$ from outside. So we must find the intersection of $[\chi_1, \theta + 2\pi] \cup [-\chi_m, \chi_m]$. We consider sub-cases, on which additional conditions hold on. Note that, for all of them, we also assume that $-\pi \leq p, \theta \leq \pi$.

(a) **Case A1:** $\boxed{-\chi_m \leq \theta \leq \chi_m,\ p \leq \dfrac{\theta + \chi_m}{2}}$. In this case, observe that $\chi_1 \geq \theta \Rightarrow \chi_1 \geq -\chi_m$ and $p \leq \frac{\theta + \chi_m}{2} \Rightarrow \chi_1 \leq \chi_m$. It follows that

$$\mathcal{R} = [2p - \theta, \chi_m] \cup [2\pi - \chi_m, \theta + 2\pi].$$

(b) **Case A2:** $\boxed{-\chi_m \leq \theta \leq \chi_m,\ \dfrac{\chi_m + \theta}{2} \leq p \leq \dfrac{\theta - \chi_m}{2} + \pi}$. In this case, $\frac{\chi_m + \theta}{2} \leq p \Rightarrow \chi_1 \geq \chi_m$, and $p \leq \frac{\theta - \chi_m}{2} + \pi \Rightarrow \chi_1 \leq 2\pi - \chi_m$, so it follows that

$$\mathcal{R} = [2\pi - \chi_m, \theta + 2\pi].$$

(c) **Case A3:** $\boxed{-\chi_m \leq \theta \leq \chi_m,\ \dfrac{\theta-\chi_m}{2}+\pi \leq p \leq \theta+\pi}$. As $\frac{\theta-\chi_m}{2}+\pi \leq p \Rightarrow \chi_1 \geq 2\pi-\chi_m$, it follows that

$$\mathcal{R} = [2p-\theta, \theta+2\pi].$$

(d) **Case A4:** $\boxed{\theta \leq -\chi_m,\ \dfrac{\theta-\chi_m}{2} \leq p \leq \dfrac{\chi_m+\theta}{2}}$. In this case, observe that $\frac{\theta-\chi_m}{2} \leq p \Rightarrow \chi_1 \geq -\chi_m$, therefore

$$\mathcal{R} = [2p-\theta, \chi_m].$$

(e) **Case A5:** $\boxed{\theta \leq -\chi_m,\ \dfrac{\chi_m+\theta}{2} \leq p \leq \theta+\pi}$. In this case, we have $\frac{\chi_m+\theta}{2} \leq p \Rightarrow \chi_1 \geq \chi_m$, therefore

$$\mathcal{R} = \emptyset.$$

(f) **Case A6:** $\boxed{\theta \leq -\chi_m,\ \theta \leq p \leq \dfrac{\theta-\chi_m}{2}}$. In this case, we have $p \leq \frac{\theta-\chi_m}{2} \Rightarrow \chi_1 \leq -\chi_m$, therefore

$$\mathcal{R} = [-\chi_m, \chi_m].$$

(g) **Case A7:** $\boxed{\theta \geq \chi_m,\ p \geq \theta}$. As $p \leq \frac{\theta-\chi_m}{2}+\pi$, it follows that also $\chi_1 \leq 2\pi-\chi_m$, and so

$$\mathcal{R} = [-\chi_m, \chi_m].$$

2. **Case B:** $\boxed{\theta-\pi \leq p \leq \theta}$. In this case, observe first that

$$\frac{\chi+\theta+\pi}{2} = p + \frac{3\pi}{2} \Leftrightarrow \chi = 2p+2\pi-\theta.$$

Let $\chi_1 = 2p+2\pi-\theta$. Note that $p \leq \theta \Rightarrow 2p \leq 2\theta \Rightarrow 2p+2\pi-\theta \leq \theta+2\pi$. Also, $p \geq \theta-\pi \Rightarrow 2p+2\pi-\theta \geq \theta$. Therefore, the value χ_1 is the maximum value of χ in the range $[\theta, \theta+2\pi]$ for which the nodes arrive at the location $[\phi, \phi+d\phi]$ from outside. So we must find the intersection of $[\theta, \chi_1] \cup [-\chi_m, \chi_m]$. We take cases:

(a) **Case B1:** $\boxed{-\chi_m \leq \theta \leq \chi_m,\ \dfrac{\theta-\chi_m}{2} \leq p \leq \theta}$. We have $\frac{\theta-\chi_m}{2} \leq p \Rightarrow \chi_1 \geq 2\pi-\chi_m$, so that

$$\mathcal{R} = [\theta, \chi_m] \cup [2\pi-\chi_m, 2p+2\pi-\theta].$$

(b) **Case B2:** $\boxed{-\chi_m \leq \theta \leq \chi_m,\ \dfrac{\chi_m+\theta}{2}-\pi \leq p \leq \dfrac{\theta-\chi_m}{2}}$. Observe that $p \leq \frac{\theta-\chi_m}{2} \Rightarrow \chi_1 \leq 2\pi-\chi_m$ and $\frac{\chi_m+\theta}{2}-\pi \leq p \Rightarrow \chi_1 \geq \chi_m$, therefore

$$\mathcal{R} = [\theta, \chi_m].$$

(c) **Case B3:** $\boxed{-\chi_m \leq \theta \leq \chi_m,\ \theta - \pi \leq p \leq \dfrac{\chi_m + \theta}{2} - \pi}$. In this case $p \leq \frac{\chi_m+\theta}{2} - \pi \Rightarrow \chi_1 \leq \chi_m$, and it follows that

$$\mathcal{R} = [\theta, 2p + 2\pi - \theta].$$

(d) **Case B4:** $\boxed{\theta \geq \chi_m,\ \dfrac{\theta - \chi_m}{2} \leq p \leq \dfrac{\theta + \chi_m}{2}}$. Observe that $p \leq \frac{\theta+\chi_m}{2} \Rightarrow \chi_1 \leq 2\pi + \chi_m$ and $\frac{\theta-\chi_m}{2} \leq p \Rightarrow \chi_1 \geq 2\pi - \chi_m$, so

$$\mathcal{R} = [2\pi - \chi_m,\ 2p + 2\pi - \theta].$$

(e) **Case B5:** $\boxed{\theta \geq \chi_m,\ \theta - \pi \leq p \leq \dfrac{\theta - \chi_m}{2}}$. Observe that $p \leq \frac{\theta-\chi_m}{2} \Rightarrow \chi_1 \leq 2\pi - \chi_m$, so

$$\mathcal{R} = \emptyset.$$

(f) **Case B6:** $\boxed{\theta \geq \chi_m,\ \dfrac{\chi_m + \theta}{2} \leq p \leq \theta}$. As $\frac{\chi_m+\theta}{2} \leq p \Rightarrow \chi_1 \geq 2\pi + \chi_m$, it follows that

$$\mathcal{R} = [-\chi_m, \chi_m].$$

(g) **Case B7:** $\boxed{\theta \leq -\chi_m,\ p \leq \theta}$. Since $p \geq \frac{\theta+\chi_m}{2} - \pi$, we have $\chi_1 \geq \chi_m$, and so

$$\mathcal{R} = [-\chi_m, \chi_m].$$

3. **Case C:** $\boxed{p \leq \theta - \pi}$. In this case, observe first that

$$\frac{\chi + \theta + \pi}{2} = p + 2\pi + \frac{\pi}{2} \Leftrightarrow \chi = 2p + 4\pi - \theta.$$

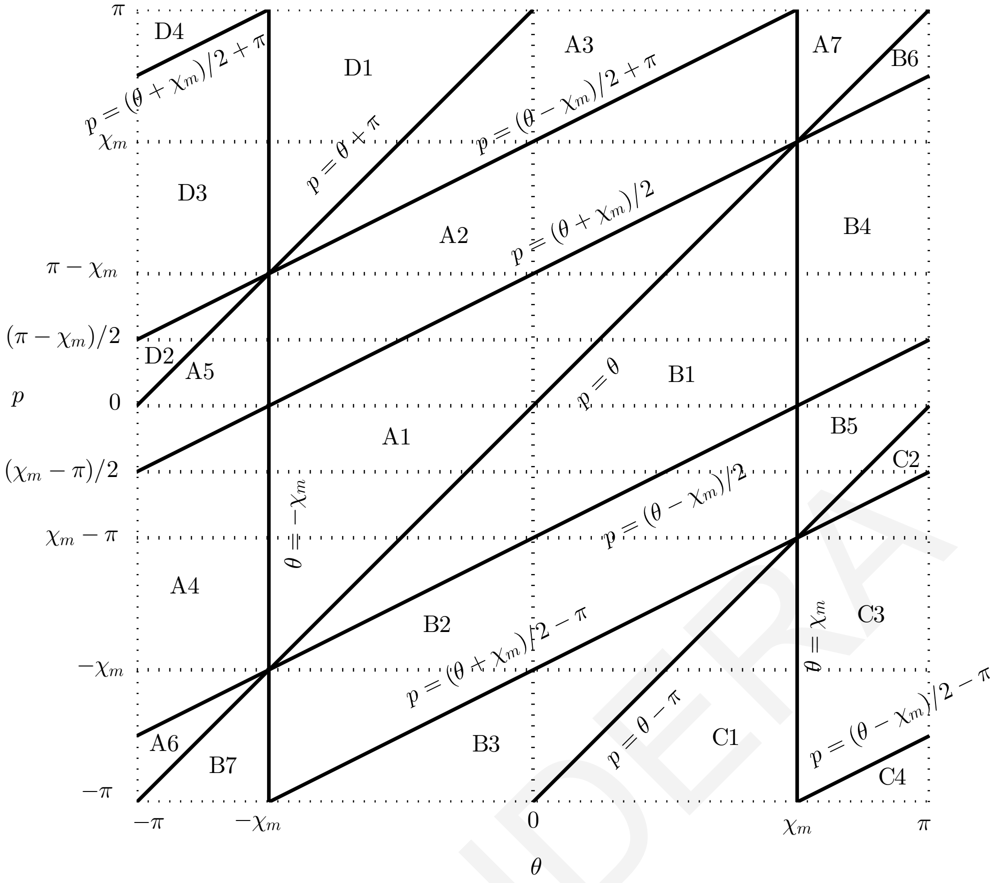
Let $\chi_1 = 2p + 4\pi - \theta$. Observe that $p \leq \theta - \pi \Rightarrow 2p - \theta \leq \theta - 2\pi \Rightarrow \chi_1 \leq \theta + 2\pi$. Also, $p \geq \theta - 2\pi \Rightarrow 2p - \theta \geq \theta - 4\pi \Rightarrow \chi_1 \geq \theta$. Therefore, χ_1 is the minimum value of χ in the range $[\theta, \theta + 2\pi]$ for which the nodes arrive at the location $[\phi, \phi + d\phi]$ from outside. So we must find the intersection of $[\chi_1, \theta + 2\pi] \cup [-\chi_m, \chi_m]$. We take cases:

(a) **Case C1:** $\boxed{\theta \leq \chi_m,\ p \leq \theta - \pi}$. Because $p \geq \frac{\theta-\chi_m}{2} - \pi$, it follows that $\chi_1 \geq 2\pi - \chi_m$. Therefore,

$$\mathcal{R} = [2p + 4\pi - \theta, \theta + 2\pi].$$

(b) **Case C2:** $\boxed{\theta \geq \chi_m,\ \dfrac{\theta + \chi_m}{2} - \pi \leq p \leq \theta - \pi}$. Because $p \geq \frac{\theta+\chi_m}{2} - \pi$, it follows that $2p + 4\pi - \theta \geq 2\pi + \chi_m \Rightarrow \chi_1 \geq 2\pi + \chi_m$, therefore

$$\mathcal{R} = \emptyset.$$



Case	Description	$\mathcal{R}$
A1	$-\chi_m \le \theta \le \chi_m,\ \theta \le p \le \frac{\theta+\chi_m}{2}$	$[2p-\theta,\chi_m] \cup [2\pi-\chi_m,\theta+2\pi]$
A2	$-\chi_m \le \theta \le \chi_m,\ \frac{\chi_m+\theta}{2} \le p \le \frac{\theta-\chi_m}{2}+\pi$	$[2\pi-\chi_m,\theta+2\pi]$
A3	$-\chi_m \le \theta \le \chi_m,\ \frac{\theta-\chi_m}{2}+\pi \le p \le \theta+\pi$	$[2p-\theta,\theta+2\pi]$
A4	$\theta \le -\chi_m,\ \frac{\theta-\chi_m}{2} \le p \le \frac{\chi_m+\theta}{2}$	$[2p-\theta,\chi_m]$
$\mathrm{A5, B5,}$ $\mathrm{C2, D2}$	$\theta \le -\chi_m,\ \frac{\chi_m+\theta}{2} \le p \le \frac{\theta-\chi_m}{2}+\pi$ $\theta \ge \chi_m,\ \frac{\theta+\chi_m}{2}-\pi \le p \le \frac{\theta-\chi_m}{2}$	$\emptyset$
$\mathrm{A6, A7,}$ $\mathrm{B6, B7,}$ $\mathrm{C4,}$ D4	$\theta \le -\chi_m,\ p \le \frac{\theta-\chi_m}{2}$ $\theta \le -\chi_m,\ p \ge \frac{\theta+\chi_m}{2}+\pi$ $\theta \ge \chi_m,\ p \le \frac{\theta-\chi_m}{2}-\pi$ $\theta \ge \chi_m,\ p \ge \frac{\theta+\chi_m}{2}$	$[-\chi_m,\chi_m]$
B1	$-\chi_m \le \theta \le \chi_m,\ \frac{\theta-\chi_m}{2} \le p \le \theta$	$[\theta,\chi_m] \cup [2\pi-\chi_m,2p+2\pi-\theta]$
B2	$-\chi_m \le \theta \le \chi_m,\ \frac{\chi_m+\theta}{2}-\pi \le p \le \frac{\theta-\chi_m}{2}$	$[\theta,\chi_m]$
B3	$-\chi_m \le \theta \le \chi_m,\ \theta-\pi \le p \le \frac{\chi_m+\theta}{2}-\pi$	$[\theta,2p+2\pi-\theta]$
B4	$\theta \ge \chi_m,\ \frac{\theta-\chi_m}{2} \le p \le \frac{\theta+\chi_m}{2}$	$[2\pi-\chi_m,\ 2p+2\pi-\theta]$
C1	$\theta \le \chi_m,\ p \le \theta-\pi$	$[2p+4\pi-\theta,\theta+2\pi]$
C3	$\theta \ge \chi_m,\ \frac{\theta-\chi_m}{2}-\pi \le p \le \frac{\theta+\chi_m}{2}-\pi$	$[2p+4\pi-\theta,\chi_m+2\pi]$
D1	$\theta \ge -\chi_m,\ p \ge \theta+\pi$	$[\theta,2p-2\pi-\theta]$
D3	$\theta \le -\chi_m,\ \frac{\theta-\chi_m}{2}+\pi \le p \le \frac{\chi_m+\theta}{2}+\pi$	$[-\chi_m,2p-2\pi-\theta]$

Figure A.2: The cases we consider for calculating $\mathcal{R}$.

(c) **Case C3:** $\boxed{\theta \geq \chi_m,\ \dfrac{\theta-\chi_m}{2}-\pi \leq p \leq \dfrac{\theta+\chi_m}{2}-\pi}$. Again, because $p \geq \frac{\theta-\chi_m}{2}-\pi$, it follows that $\chi_1 \geq 2\pi-\chi_m$. In addition, $p \leq \frac{\theta+\chi_m}{2}-\pi \Rightarrow 2p+4\pi-\theta \leq 2\pi+\chi_m \Rightarrow \chi_1 \leq 2\pi+\chi_m$. Therefore,

$$\mathcal{R} = [2p+4\pi-\theta, \chi_m+2\pi].$$

(d) **Case C4:** $\boxed{\theta \geq \chi_m,\ p \leq \dfrac{\theta-\chi_m}{2}-\pi}$. From $p \leq \frac{\theta-\chi_m}{2}-\pi$ it follows that $\chi_1 \leq 2\pi-\chi_m$, therefore

$$\mathcal{R} = [-\chi_m, \chi_m].$$

4. **Case D:** $\boxed{p \geq \theta+\pi}$. In this case, observe first that

$$\frac{\chi+\theta+\pi}{2} = p-\frac{\pi}{2} \Leftrightarrow \chi = 2p-2\pi-\theta.$$

Let $\chi_1 = 2p-2\pi-\theta$. Note that $p \geq \theta+\pi \Rightarrow \chi_1 \geq \theta$, and $p \leq \theta+2\pi \Rightarrow \chi_1 \leq \theta+2\pi$. Therefore, χ_1 is the maximum value of χ in the range $[\theta, \theta+2\pi]$ for which the nodes arrive at the location $[\phi, \phi+d\phi]$ from outside. So we must find the intersection of $[\theta, \chi_1] \cup [-\chi_m, \chi_m]$. We take cases:

(a) **Case D1:** $\boxed{\theta \geq -\chi_m,\ p \geq \theta+\pi}$. Observe that $p \leq \frac{\chi_m+\theta}{2}+\pi$, therefore $\chi_1 \leq \chi_m$, and it follows that

$$\mathcal{R} = [\theta, 2p-2\pi-\theta].$$

(b) **Case D2:** $\boxed{\theta \leq -\chi_m,\ \theta+\pi \leq p \leq \dfrac{\theta-\chi_m}{2}+\pi}$. In this case, $p \leq \frac{\theta-\chi}{2}+\pi \Rightarrow \chi_1 \leq -\chi_m$, therefore

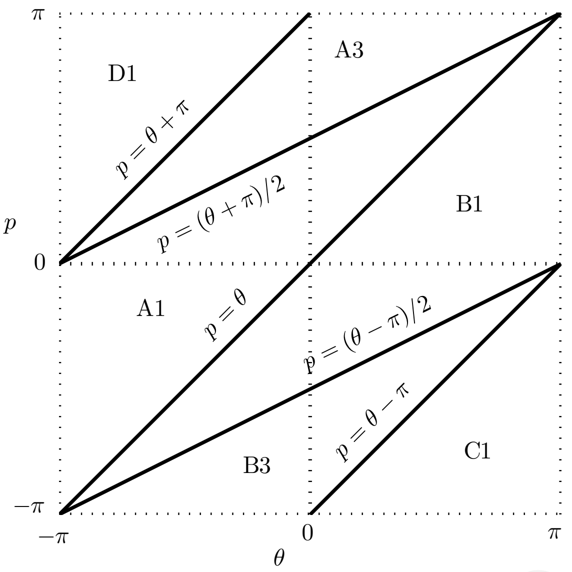
$$\mathcal{R} = \emptyset.$$

(c) **Case D3:** $\boxed{\theta \leq -\chi_m,\ \dfrac{\theta-\chi_m}{2}+\pi \leq p \leq \dfrac{\chi_m+\theta}{2}+\pi}$. In this case, $p \leq \frac{\chi_m+\theta}{2}+\pi \Rightarrow \chi_1 \leq \chi_m$. Also, $p \geq \frac{\theta-\chi_m}{2}+\pi \Rightarrow 2p-2\pi-\theta \geq -\chi_m \Rightarrow \chi_1 \geq -\chi_m$, therefore

$$\mathcal{R} = [-\chi_m, 2p-2\pi-\theta].$$

(d) **Case D4:** $\boxed{\theta \leq -\chi_m,\ p \geq \dfrac{\chi_m+\theta}{2}+\pi}$. From the second inequality it follows that $\chi_1 > \chi_m$. Therefore,

$$\mathcal{R} = [-\chi_m, \chi_m].$$



Case	Description	$\mathcal{R}$
A (A1,A3)	$\theta \le p \le \theta + \pi$	$[2p - \theta, \theta + 2\pi]$
B (B1,B3)	$\theta - \pi \le p \le \theta$	$[\theta, 2p + 2\pi - \theta]$
C (C1)	$p \le \theta - \pi$	$[2p + 4\pi - \theta, \theta + 2\pi]$
D (D1)	$p \ge \theta + \pi$	$[\theta, 2p - 2\pi - \theta]$

Figure A.3: The cases we consider for calculating $\mathcal{R}$, in the special case $\chi_m = \pi$.

It follows from (A.6) that

$$\gamma(\phi;\theta) = \frac{v_n \lambda r(\phi)}{2\pi} \times \begin{cases} G(\chi_B) - G(\chi_A), & \mathcal{R} = [\chi_A,\ \chi_B], \\ G(\chi_B) - G(\chi_A) + G(\chi_D) - G(\chi_C), & \mathcal{R} = [\chi_A,\ \chi_B] \cup [\chi_C,\ \chi_D], \end{cases} \tag{A.7}$$

where the function $G(\cdot)$ is defined as

$$G(\chi) = \chi \cos(p(\phi) - \theta) - \sin(\chi - p(\phi)),$$

and the values χ_A, χ_B, χ_C, and χ_D are collected in Fig. A.2.

Finally, let us also assume that $\chi_m = \pi$. Of the 20 cases of Fig. A.2, only 6 remain, as shown in Fig. A.3.

A.4 Calculation of $\gamma(\theta)$

Finally, we calculate the incidence rate $\gamma(\theta)$ with which nodes with any direction $\chi \in [-\chi_m, \chi_m]$ hit the forwarding region $\mathcal{F}$ at any location ϕ. This incidence rate can be calculated using either of the two integrals of (5.12), however we will use the first one.

To this end, we use (A.5), and therefore

$$\gamma(\theta) = \int_{-\chi_m}^{\chi_m} \left(\frac{\upsilon_n \lambda}{\pi}\right) P\left(\frac{\chi+\theta+\pi}{2}\right) \left|\sin\left(\frac{\chi-\theta}{2}\right)\right| \, d\chi, \qquad -\pi \leq \theta \leq \pi. \tag{A.8}$$

Appendix B

Special Case: Disk

In this appendix we show in detail the calculation involved in deriving delay-cost pairs for the special case of the disk defined in Section 5.5.

In this case, $x_0 = x_1 = -R$ and $x_2 = R$. Also, $r(\phi) = R$ and $p(\phi) = \phi$ for all $\phi \in [-\pi, \pi]$, and $y_1(x) = -\sqrt{R^2 - x^2}$, $y_2(x) = \sqrt{R^2 - x^2}$ for $x \in [-R, R]$.

B.1 Calculations when $\mathcal{F}$ is Not Empty

Observe that the shape of the forwarding region is such, that the points satisfying

$$\frac{C}{x} = q \Leftrightarrow \frac{x^2 + y^2}{x} = q \Leftrightarrow \left(x - \frac{q}{2}\right)^2 + y^2 = \left(\frac{q}{2}\right)^2$$

are either completely within $\mathcal{F}$, when $q \leq 2R$, or completely outside $\mathcal{F}$, when $q > 2R$.

We also have that

$$A(q) = \begin{cases} \pi \left(\frac{q}{2}\right)^2, & q \leq 2R, \\ \pi R^2, & q > R, \end{cases} \qquad A'(q) = \begin{cases} \frac{\pi q}{2}, & q \leq 2R, \\ 0, & q > 2R. \end{cases}$$

It follows that

$$f_Q(q) = \begin{cases} \frac{\lambda' \pi q}{2(1 - \exp(-\lambda' |\mathcal{F}|))} \exp\left(-\frac{\lambda' \pi q^2}{4}\right), & q \leq 2R, \\ 0, & q > 2R. \end{cases}$$

Using symmetry, it is straightforward to show that

$$E(X_{T,1} | \mathcal{X}_1 = \theta, M', Q = q) = \frac{q}{2}.$$

We also have

$$\begin{aligned}
E(C_{T,1}|\mathcal{X}_1=\theta, M', Q=q) &= \int_0^{2\pi} \frac{1}{2\pi}\left[\left(\frac{q}{2}+\frac{q}{2}\cos\phi\right)^2+\left(\frac{q}{2}\sin\phi\right)^2\right] d\phi \\
&= \int_0^{2\pi} \frac{q^2}{8\pi}\left[1+\cos^2\phi+2\cos\phi+\sin^2\phi\right] d\phi \\
&= \int_0^{2\pi} \frac{q^2}{8\pi}\left(2+2\cos\phi\right) d\phi = \frac{q^2}{8\pi}\times 4\pi = \frac{q^2}{2}.
\end{aligned}$$

B.2 Calculations when $\mathcal{F}$ is Empty

Since $p(\phi)=\phi$ and $r(\phi)=R$, Eqns. (A.2) and (A.3) readily become

$$\gamma\left(\chi,\phi;\theta\right)=\frac{v_n\lambda R}{\pi}\sin\left(\frac{\chi-\theta}{2}\right)\sin\left(\frac{\chi+\theta}{2}-\phi\right),$$

where

$$\theta\in[0,2\pi],\quad \chi\in[\theta,\theta+2\pi],\quad \phi\in\left[\frac{\chi+\theta}{2}-\pi,\frac{\chi+\theta}{2}\right].$$

When ϕ is not in the above set, $\gamma\left(\chi,\phi;\theta\right)=0$.

Also, (A.5) gives

$$\gamma(\chi;\theta)=\left(\frac{2v_n\lambda R}{\pi}\right)\left|\sin\left(\frac{\chi-\theta}{2}\right)\right|,\qquad -\pi\leq\theta,\chi\leq\pi.$$

We have used the fact that $P(\omega)=2R$ for any angle ω.

Let us now also assume that $\chi_m=\pi$. Since $p(\phi)=\phi$ and $r(\phi)=R$, (A.6) becomes

$$\gamma(\phi;\theta)=\frac{v_n\lambda R}{2\pi}\left[\chi\cos(\phi-\theta)-\sin(\chi-\phi)\right]_{\chi_A}^{\chi_B}, \tag{B.1}$$

where χ_A and χ_B are the two boundaries of $\mathcal{R}$ as specified in Fig. A.3. We consider separately the cases of this figure:

1. **Case A:**

$$\begin{aligned}
\gamma(\phi;\theta) &= \frac{v_n\lambda R}{2\pi}\left[(2\pi+2\theta-2\phi)\cos(\phi-\theta)-\sin(\theta-\phi)+\sin(\phi-\theta)\right] \\
&= \frac{v_n\lambda R}{\pi}\left[(\pi+\theta-\phi)\cos(\phi-\theta)+\sin(\phi-\theta)\right].
\end{aligned}$$

2. **Case B:**

$$\begin{aligned}
\gamma(\phi;\theta) &= \frac{v_n\lambda R}{2\pi}\left[(2\pi+2\phi-2\theta)\cos(\phi-\theta)-\sin(\phi-\theta)+\sin(\theta-\phi)\right] \\
&= \frac{v_n\lambda R}{\pi}\left[(\pi+\phi-\theta)\cos(\phi-\theta)+\sin(\theta-\phi)\right].
\end{aligned}$$

3. **Case C:**

$$\begin{aligned}\gamma(\phi;\theta) &= \frac{v_n\lambda R}{2\pi}\left[(2\theta-2\phi-2\pi)\cos(\phi-\theta)-\sin(\theta-\phi)+\sin(\phi-\theta)\right]\\ &= \frac{v_n\lambda R}{\pi}\left[(\theta-\phi-\pi)\cos(\phi-\theta)+\sin(\phi-\theta)\right].\end{aligned}$$

4. **Case D:**

$$\begin{aligned}\gamma(\phi;\theta) &= \frac{v_n\lambda R}{2\pi}\left[(2\phi-2\pi-2\theta)\cos(\phi-\theta)-\sin(\phi-\theta)+\sin(\theta-\phi)\right]\\ &= \frac{v_n\lambda R}{\pi}\left[(\phi-\pi-\theta)\cos(\phi-\theta)+\sin(\theta-\phi)\right].\end{aligned}$$

Finally, using (B.1), $\gamma(\theta)$ becomes

$$\begin{aligned}\gamma(\theta) &= \left(\frac{2v_n\lambda R}{\pi}\right)\int_\theta^{\theta+2\pi}\sin\left(\frac{\chi-\theta}{2}\right)\,d\chi = \left(\frac{2v_n\lambda R}{\pi}\right)\int_0^{2\pi}\sin\left(\frac{\chi}{2}\right)\,d\chi\\ &= \frac{4v_n\lambda R}{\pi}\left[-\cos\left(\frac{\chi}{2}\right)\right]_0^{2\pi} = \frac{8v_n\lambda R}{\pi}.\end{aligned}$$

As expected, there is no dependence of $\gamma(\theta)$ on θ.

Bibliography

[1] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *ACM SIGCOMM*, Portland, OR, USA, Aug.-Sep. 2004, pp. 145–157.

[2] T. Small and Z. J. Haas, "The shared wireless infostation model – a new ad hoc networking paradigm (or where there is a whale, there is a way)," in *ACM MOBIHOC*, Annapolis, MD, June 2003.

[3] P. Juang, H. Oki, Y. Wang, and M. Martonosi, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiments with Zebranet," in *Proc. ASPLOS-X*, San Jose, CA, Oct. 2002.

[4] P.-C. Cheng, K.-C. Lee, M. Gerla, and J. Harri, "GeoDTN+Nav: Geographic DTN routing with navigator prediction for urban vehicular environments," *Mobile Networks and Applications*, vol. 15, no. 1, pp. 61–82, 2010.

[5] A. Sidera and S. Toumpis, "Delay tolerant firework routing: A geographic routing protocol for wireless delay tolerant networks," *EURASIP Journal on Wireless Communications and Networking*, 2013.

[6] E. Kuiper and S. Nadjm-Tehrani, "Geographical routing with location service in intermittently connected MANETs," *IEEE Trans. on Vehicular Technology*, vol. 60, no. 2, pp. 592–604, 2011.

[7] L. Xue, D. Kim, Y. Zhu, D. Li, W. Wang, and A. Tokuta, "Multiple heterogeneous data ferry trajectory planning in wireless sensor networks," in *IEEE INFOCOM*, Toronto, Canada, Apr. 2014, pp. 2274–2282.

[8] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments," in *SIGCOMM*, Philadelphia, PA, USA, Aug. 2005, pp. 244–251.

[9] B. Kotnyek, “An annotated overview of dynamic network flows,” INRIA, Tech. Rep. RR-4936, 2003.

[10] R. C. Shah, S. Roy, S. Jain, and W. Brunette, “Data MULEs: modeling and analysis of a three-tier architecture for sparse sensor networks,” *Ad Hoc Networks*, vol. 1, no. 6, pp. 215–233, 2003.

[11] N. Laoutaris, G. Smaragdakis, P. Rodriguez, and R. Sundaram, “Delay tolerant bulk data transfers on the internet,” in *Proc. ACM Sigmetrics 2009*, Seattle, WA, June 2009, pp. 229–238.

[12] M. Grossglauser and D. Tse, “Mobility increases the capacity of ad-hoc wireless networks,” in *IEEE INFOCOM*, vol. 3, Anchorage, AL, Apr. 2001, pp. 1360–1369.

[13] S. Toumpis and A. J. Goldsmith, “Large wireless networks under fading, mobility, and delay constraints,” in *IEEE INFOCOM*, Hong Kong, China, Mar.-Apr. 2004.

[14] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Efficient routing in intermittently connected mobile networks: the single-copy case,” *IEEE Trans. Networking*, vol. 16, no. 1, pp. 63–76, Feb. 2008.

[15] ——, “Efficient routing in intermittently connected mobile networks: the multiple-copy case,” *IEEE Trans. Networking*, vol. 16, no. 1, pp. 77–90, Feb. 2008.

[16] J. LeBrun, C.-N. Chiah, D. Ghosal, and M. Zhang, “Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks,” in *Proc. IEEE VTC Spring*, vol. 4, Florence, Italy, May-June 2005, pp. 2289–2293.

[17] S. Ali, J. Qadir, and A. Baig, “Routing protocols in delay tolerant networks,” in *International Conference on Emerging Technologies (ICET)*, 2010, pp. 70–75.

[18] M. Liu, Y. Yang, and Z. Qin, “A survey of routing protocols and simulations in delay-tolerant networks,” *Lecture Notes in Computer Science*, pp. 243–253, 2011.

[19] A. Silva, S. Burleigh, C. Hirata, and K. ObraczkaSi, “A survey on congestion control for delay and disruption tolerant networks,” *Ad Hoc Networks*, pp. 480–494, Feb. 2015.

[20] J. Park, U. Lee, S. Oh, M. Gerla, and D. Lun, “Emergency related video streaming in VANETs using network coding,” in *VANET*, 2006, pp. 102–103.

[21] http://en.wikipedia.org/wiki/MontBlancTunnel.

[22] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Computer Networks*, pp. 2292–2330, 2008.

[23] U. Lee, B. Zhou, M. Gerla, E. Magistretti, P. Bellavista, and A. Corradi, “MobEyes: smart mobs for urban monitoring with a vehicular sensor network,” *IEEE Trans. on Wireless Communications*, vol. 13, no. 5, pp. 52–57, Oct. 2006.

[24] Y. Yun and Y. Xia, “Maximizing the lifetime of wireless sensor networks with mobile sink in delay-tolerant applications,” *IEEE Trans. on Mobile Computing*, pp. 1308–1318, 2010.

[25] J. Segui, E. Jennings, and S. Burleigh, “Enhancing contact graph routing for delay tolerant space networking,” in *GLOBECOM*, Dec. 2011, pp. 1–6.

[26] A. Vasilakos, Y. Zhang, and T. Spyropoulos, *Delay Tolerant Networks: Protocols and Applications.* CRC Press, 2011.

[27] H. Ntareme, M. Zennaro, and B. Pehrson, “Delay tolerant network on smartphones: Applications for communication challenged areas,” in *ExtremeCom*, 2011.

[28] A. Lindgren and P. Hui, “The quest for a killer app for opportunistic and delay tolerant networks,” in *CHANTS*, 2009, pp. 59–66.

[29] C. Lochert, M. Mauve, H. Fubler, and H. Hartenstein, “Geographic routing in city scenarios,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 1, Jan. 2005.

[30] K.-C. Lee, P.-C. Cheng, and M. Gerla, “GeoCross: A geographic routing protocol in the presence of loops in urban scenarios,” *Ad Hoc Networks*, vol. 8, pp. 474–488, Jul. 2010.

[31] http://en.wikipedia.org/wiki/Fireworks\#Palm.

[32] http://pages.cs.aueb.gr/~toumpis/dtns.html.

[33] A. Sidera and S. Toumpis, “DTFR: A geographic routing protocol for wireless delay tolerant networks,” in *Proc. IFIP MedHocNet 2011*, Favignana Island, Sicily, Italy, June 2011.

[34] ——, “On the delay/cost tradeoff in wireless mobile delay-tolerant networks,” in *WiOpt*, Hammamet, Tunisia, May. 2014.

[35] M. Bayir and M. Demirbas, “On the fly learning of mobility profiles for routing in pocket switched networks,” *Ad Hoc Networks, Elsevier*, 2013.

[36] E. Jones, L. Li, J. Schmidtke, and P. Ward, “Practical routing in delay-tolerant networks,” *IEEE Trans. on Mobile Computing*, vol. 6, no. 8, pp. 943–959, 2007.

[37] P. Hui, J. Crowcroft, and E. Yoneki, “BUBBLE rap: Social-based forwarding in delay-tolerant networks,” *IEEE Trans. on Mobile Computing*, pp. 1576–1589, Nov. 2011.

[38] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, “SUMO - simulation of urban mobility - an overview,” in *SIMUL*, Barcelona, Spain, Oct. 2011.

[39] A. Sidera and S. Toumpis, “Routing using partition-wide information in wireless delay tolerant networks,” in *Proc. IFIP MedHocNet 2013*, Ajaccio, Corsica, France, June 2013.

[40] ——, “Wireless mobile DTN routing with the extended minimum estimated expected delay protocol,” *submitted to journal publication*, 2015.

[41] H. Takagi and L. Kleinrock, “Optimal transmission ranges for randomly distributed packet radio terminals,” *IEEE Trans. on Communications*, pp. 246–57, Mar. 1984.

[42] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, “A distance routing effect algorithm for mobility,” in *ACM MOBICOM*, Dallas, TX, Oct. 1998, pp. 76–84.

[43] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, “Routing with guaranteed delivery in ad hoc wireless networks,” *Wireless networks*, pp. 609–616, 2001.

[44] S. Giordano, I. Stojmenovic, and L. Blazevic, “Position based routing algorithms for ad hoc networks: A taxonomy,” *Ad Hoc Wireless Networking*, pp. 103–136, 2004.

[45] B. Karp and H.-T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *ACM MOBICOM*, Boston, MA, Aug. 2000.

[46] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter, "MDDV: a mobility-centric data dissemination algorithm for vehicular networks," in *ACM MOBICOM*, Philadelphia, Pennsylvania, USA, Sep. 2004, pp. 47–56.

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

[48] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke, Tech. Rep. CS-2000-06, 2000.

[49] D. Goodman, J. Borras, N. Mandayam, and R. Yates, "Infostations: a new system model for data and messaging services," in *IEEE 47th Vehicular Technology Conference*, Phoenix, AZ, USA, May 1997, pp. 969–973.

[50] A. Jindal and K. Psounis, "Optimizing multi-copy routing schemes for resource constrained intermittently connected mobile networks," in *Fortieth Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, California, USA, Oct. 2006, pp. 2142–2146.

[51] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and Focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility," in *IEEE PerCom Workshops*, White Plains, NY, USA, Mar. 2007, pp. 79–85.

[52] A. Jindal and K. Psounis, "Performance analysis of epidemic routing under contention," in *ACM IWCMC*, Vancouver, Canada, July 2006, pp. 539–544.

[53] P. Tournoux, J. Leguay, F. Benbadis, V. Conan, M. Dias, and J. Whitbeck, "The accordion phenomenon: Analysis, characterization, and impact on DTN routing," in *IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp. 1116–1124.

[54] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," in *ACM SIGMOBILE*, Jul. 2003, pp. 19–20.

[55] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, "MaxProp: routing for vehicle-based dirsuption-tolerant networks," in *IEEE INFOCOM*, Barcelona, Spain, Apr. 2006, pp. 1–11.

[56] C. Chen and Z. Chen, "Evaluating contacts for routing in highly partitioned mobile networks," in *MobiOpp*, 2007, pp. 17–24.

[57] K. Tan, Q. Zhang, and W. Zhu, "Shortest path routing in partially connected ad hoc networks," in *GLOBECOM*, San Francisco, CA, USA, Dec. 2003, pp. 1038–1042.

[58] B. Burns, O. Brock, and B. Levine, "MV routing and capacity building in disruption tolerant networks," in *IEEE INFOCOM*, Miami, FL, USA, Mar. 2005, pp. 398–408.

[59] R. Ramanathan, R. Hansen, P. Basu, R. Rosales-Hain, and R. Krishnan, "Prioritized epidemic routing for opportunistic networks," in *ACM MobiOpp*, San Juan, Puerto Rico, June 2007, pp. 62–66.

[60] V. Conan, J. Leguay, and T. Friedman, "Fixed point opportunistic routing in delay tolerant networks," *IEEE Journal on Selected Areas in Communications*, pp. 773–782, June 2008.

[61] M. Uddin, H. Ahmad, T. Abdelzaher, and R. Kravets, "Intercontact routing for energy constrained disaster response networks," *IEEE Trans. on Mobile Computing*, 2013.

[62] E. Bulut, S. Geyik, and B. Szymanski, "Utilizing correlated node mobility for efficient DTN routing," *Pervasive and Mobile Computing, Eslevier*, 2013.

[63] Z. Guo, B. Wang, and J. Cui, "Generic prediction assisted single-copy routing in underwater delay tolerant sensor networks," *Ad Hoc Networks, Elsevier*, 2013.

[64] A. Elwhishi, P. Ho, K. Naik, and B. Shihada, "Self-adaptive contention aware routing protocol for intermittently connected mobile networks," *IEEE Trans. on Parallel and Distributed Systems*, 2013.

[65] Q. Fu, B. Krishnamachari, and L. Zhang, "DAWN: A density adaptive routing for deadline-based data collection in vehicular delay tolerant networks," *Tsinghua Science and Technology*, 2013.

[66] J. Jeong, K. Lee, Y. Yi, I. Rhee, and S. Chong, "ExMin: A routing metric for novel opportunity gain in delay tolerant networks," *Computer Networks, Elsevier*, 2013.

[67] Y. Wang, J. Wu, and W. Yang, "Cloud-based multicasting with feedback in mobile social networks," *IEEE Trans. on Wireless Communications*, 2013.

[68] S. Wang, M. Liu, X. Cheng, Z. Li, J. Huang, and B. Chen, "Opportunistic routing in intermittently connected mobile P2P networks," *IEEE Journal on Selected Areas in Communications*, 2013.

[69] H. Chen and W. Lou, "GAR: Group aware cooperative routing protocol for resource-constraint opportunistic networks," *Computer Communications, Elsevier*, 2014.

[70] E. Bulut and B. Szymanski, "Exploiting friendship relations for efficient routing in mobile social networks," *IEEE Trans. on Parallel and Distributed Systems*, 2012.

[71] D. Zeng, S. Guo, A. Barnawi, S. Yu, and I. Stojmenovic, "An improved stochastic modeling of opportunistic routing in vehicular CPS," *IEEE Trans. on Computers*, 2014.

[72] K. Akkarajitsakul, E. Hossain, and D. Niyato, "Cooperative packet delivery in hybrid wireless mobile networks: A coalitional game approach," *IEEE Trans. on Mobile Computing*, 2013.

[73] X. Sun, Q. Yu, R. Wang, Q. Zhang, Z. Wei, J. Hu, and A. Vasilakos, "Performance of dtn protocols in space communications," *Wireless Networks, Springer*, 2013.

[74] G. Lee, S. Rallapalli, W. Dong, Y. Chen, L. Qiu, and Y. Zhang, "Mobile video delivery via human movement," in *IEEE International Conference on Sensing Communications and Networking (SECON)*, 2013.

[75] R. Lu, X. Lin, H. Zhu, and X. Shen, "Spark: A new VANET-based smart parking scheme for large parking lots," in *IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp. 1413–1421.

[76] B. Chen and M. Chan, "MobTorrent: A framework for mobile internet access from vehicles," in *IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp. 1404–1412.

[77] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Performance analysis of mobility-assisted routing," in *ACM MOBIHOC*, May 2006.

[78] W. J. Hsu, T. Spyropoulos, K. Psounis, and A. Helmy, "Modeling time-variant user mobility in wireless mobile networks," in *IEEE INFOCOM*, May 2007.

[79] A. Mei and J. Stefa, "SWIM: A simple model to generate small mobile worlds," in *IEEE INFOCOM*, Apr. 2009.

[80] S. Kosta, A. Mei, and J. Stefa, "Small world in motion (SWIM): Modeling communities in ad-hoc mobile networking," in *SECON*, June 2010.

[81] K. Lee, S. Hong, S. Kim, I. Rhee, and S. Chong, "SLAW: A new mobility model for human walks," in *IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp. 855–863.

[82] V. Conan, J. Leguay, and T. Friedman, "Characterizing pairwise intercontact patterns in delay tolerant networks," in *ACM Autonomics*, Rome, Italy, Oct. 2007.

[83] V. Borrel, M. Ammar, and E. Zegura, "Understanding the wireless and mobile network space: A routing-centered classification," in *CHANTS*, Montreal, Canada, Sep. 2007.

[84] S. Heimlicher, M. Karaliopoulos, H. Levy, and T. Spyropoulos, "On leveraging partial paths in partially-connected networks," in *IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp. 55–63.

[85] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *ACM MOBIHOC*, Tokyo, Japan, May 2004.

[86] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Applications, Technologies, Architectures, and Protocols for Computer Communication*, Karlsruhe, Germany, Aug. 2003, pp. 27–34.

[87] M. Zhang and R. Wolff, "Routing protocols for vehicular ad hoc networks in rural areas," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 126–131, Nov. 2008.

[88] C. Sommer and F. Dressler, "Progressing toward realistic mobility models in vanet simulations," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 132–137, Nov. 2008.

[89] Dartmouth College. The Dartmouth Wireless Trace Archive. http://cmc.cs.dartmouth.edu.

[90] http://www.haggleproject.org.

[91] N. Eagle and A. Pentland, “Reality mining: sensing complex social systems,” *Personal and Ubiquitous Computing*, vol. 10, no. 4, pp. 255–268, May 2006.

[92] https://dome.cs.umass.edu/UMassDieselNet.

[93] K. Lee, S.-H. Lee, R. Cheung, U. Lee, and M. Gerla, “First experience with CarTorrent in a real vehicular ad hoc network testbed,” in *Mobile Networking for Vehicular Environments*, Anchorage, AK, USA, May 2007, pp. 109–114.

[94] C. Perkins and E. Royer, “Ad-hoc on-demand distance vector routing,” in *IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, 1999.

[95] M. Heissenbuettel, T. Braun, M. Waelchli, and T. Bernoulli, “Evaluating the limitations of and alternatives in beaconing,” *Ad Hoc Networks*, vol. 5, pp. 558–578, July 2007.

[96] T. Clausen, P. Jacquet, C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot, “Optimized link state routing protocol (OLSR),” *Network Working Group, inria-00471712*, 2003.

[97] A. Sidera and S. Toumpis, “Delay tolerant firework routing,” AUEB, Tech. Rep., 2011, available at `pages.cs.aueb.gr/∼toumpis/dtns/technical_report.pdf`.

[98] T. Rappaport, *Wireless Communications: Principles and Practice.* Upper Saddle River, NJ: Prentice Hall, 2002.

[99] P. Sommer and R. Wattenhofer, “Gradient clock synchronization in wireless sensor networks,” in *Proc. ACM ISPN*, San Francisco, CA, Apr. 2009.

[100] http://www.nsnam.org.

[101] http://www.omnetpp.org/.

[102] A. Karänen, J. Ott, and T. Kärkkäinen, “The ONE simulator for DTN protocol evaluation,” in *Proc. Simutools 2009*, Rome, Italy, Mar. 2009.

[103] http://watwire.uwaterloo.ca/DTN/sim/.

[104] O. R. Helgason and K. V. Jónsson, "Opportunistic networking in OMNetT++," in *Proc. OMNeT++*, Marseille, France, Mar. 2008.

[105] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inform. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[106] R. Nelson and L. Kleinrock, "The spatial capacity of a slotted ALOHA multihop packet radio network with capture," *IEEE Trans. Commun.*, vol. COM-32, no. 6, pp. 684–694, June 1984.

[107] P. Jacquet, B. Mans, and G. Rodolakis, "Information propagation speed in mobile and delay tolerant networks," in *IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009.

[108] E. Baccelli, P. Jacquet, B. Mans, and G. Rodolakis, "Information propagation speed in bidirectional vehicular delay tolerant networks," in *IEEE INFOCOM*, Shanghai, China, Apr. 2011.

[109] S. M. Ross, *Introduction to Probability Models*, 8th ed. Academic Press, 2003.

[110] F. Baccelli and B. Blaszczyszyn, "Stochastic geometry and wireless networks," *Foundations and Trends in Networking*, 2009.

[111] P. Jacquet, B. Mans, and G. Rodolakis, "On space-time capacity limits in mobile and delay tolerant networks," in *IEEE INFOCOM*, San Diego, CA, Mar. 2010.

[112] R. Catanuto, S. Toumpis, and G. Morabito, "On asymptotically optimal routing in large wireless networks and geometrical optics analogy," *Computer Networks*, July 2009.

[113] A. Tasiopoulos, C. Tsiaras, and S. Toumpis, "Optimal and achievable cost/delay tradeoffs in delay-tolerant networks," *Computer Networks, Elsevier*, 2014.

[114] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - simulation of urban mobility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3,4, pp. 128–138, Dec. 2012.

[115] V. Lenders, G. Karlsson, and M. May, "Wireless ad hoc podcasting," in *IEEE SECON*, San Diego, CA, June 2007, pp. 273–283.

[116] J. Vermorel and M. Mohri, "Multi-armed bandit algorithms and empirical evaluation," *Lecture Notes in Computer Science*, pp. 437–448, 2005.