Opportunistic Routing
in
Wireless Mesh Networks

Amir Darehshoorzadeh

Department of Computer Architecture
Universitat Politècnica de Catalunya

Advisor: Dr. Llorenç Cerdà i Alabern

A thesis submitted for the degree of
Doctor of Philosophy

Abstract

Advances in communication and networking technologies are rapidly making ubiquitous network connectivity a reality. In recent years, Wireless Mesh Networks (WMNs) have already become very popular and been receiving an increasing amount of attention by the research community. Due to the limited transmission range of the radio, many pairs of nodes in WMN may not be able to communicate directly, hence they need other intermediate nodes to forward packets for them. Routing in such networks is an important issue and it poses great challenges.

Opportunistic Routing (OR) has been investigated in recent years as a way to increase the performance of WMNs by exploiting its broadcast nature. In OR, in contrast to traditional routing, instead of pre-selecting a single specific node to be the *next-hop* as a forwarder for a packet, an ordered set of nodes (referred to as *candidates*) is selected as the potential *next-hop* forwarders. Thus, the source can use multiple potential paths to deliver the packets to the destination. More specifically, when the current node transmits a packet, all the candidates that successfully receive it will coordinate with each other to determine which one will actually forward it, while the others will simply discard the packet. This dissertation studies the properties, performance, maximum gain, candidate selection algorithms and multicast delivery issues about Opportunistic Routing in WMNs.

Firstly, we focus on the performance analysis of OR by proposing a Discrete Time Markov Chain (DTMC). It can be used to evaluate OR mechanisms in wireless networks in terms of expected number of transmissions from the source to the destination. The only ingredients needed to build the transition probability matrix are the candidates of each node, and the delivery probabilities to reach them. We compare different scenarios in terms of the expected value and variance of the number of transmissions needed to send a packet from the source to the destination.

Secondly, we apply our Markov model to compare four relevant candidate selection algorithms that have been proposed in the literature. They

range from non-optimum, but simple, to optimum, but with a high computational cost. However, increasing the number of candidates increases also the coordination overhead. Therefore, in practice, the maximum number of candidates that can be used is limited. This fact has been often neglected in candidate selection algorithms that have been proposed in the literature. Since we believe that this will be a practical constraint, we do consider it in our analysis. To do so, we propose some modifications to the candidate selection algorithms under study that allow limiting the maximum number of candidates.

Thirdly, the set of candidates which a node uses and priority order of them have a significant impact on the performance of OR. Therefore, using a good metric and algorithm to select and order the candidates are key factors in designing an OR protocol. As the next contribution we propose a new metric that measures the expected distance progress of sending a packet using a set of candidates. Based on this metric we propose a candidate selection algorithm which its performance is very close to the optimum algorithm although our algorithm runs much faster.

Fourthly, in an other contribution we investigate the maximum gain that can be obtained using OR. We obtain some equations that yield the distances of the candidates in OR such that the per transmission progress towards the destination is maximized. Based on these equations we propose a novel candidate selection algorithm. Our new algorithm only needs the geographical location of nodes. The performance of our proposal is very close to the optimum candidate selection algorithm although our algorithm runs much faster.

Finally, using OR to support multicast is an other topic that we investigate in this thesis. We do so by proposing a new multicast protocol which uses OR. Unlike traditional multicast protocols, there is no designated next-hop forwarder for each destination in our protocol, thus the delivery ratio is maximized by taking advantage of spatial diversity. We compare our protocol with two well known protocols: ODMRP multicast mesh protocol and ADMR multicast tree protocol. The simulations results show that our protocol outperforms traditional ODMRP and ADMR multicast protocols, reducing the number of data transmissions and increasing the data delivery ratio.

Contents

List of Figures

List of Tables

Acknowledgments

This dissertation could not have been realized without help from a wonderful group of people whom I have had the great fortune of meeting.

First of all, I would like to express my gratitude to my research advisor, Llorenç Cerdà-Alabern, for his support, for always keeping his door open to me whenever I needed to see him, willing to help in any way possible and for guiding me back on track whenever I felt lost. I am also very grateful for Llorenç's time and patience in giving detailed feedback on the many revisions of this dissertation, as well as for all of my prior papers and presentations.

I have been fortunate to work with Vicent Pla from Universitat Politécnica de Valéncia. He always gives me positive energy with his comments and encourages me to go ahead. I appreciate Vicent for his kind attention to my ideas and his effort to improve my work. It has been a pleasure collaborating with and learning from him.

I would also like to thank the members of my thesis committee, Prof. Vicente Casares Giner, Prof. Chris Blondia, Prof. Jorge García Vidal, Prof. José María Barceló, Prof. Felix Freitag and Prof. Davide Careglio for their time and insightful suggestions that help greatly improve the quality of this dissertation.

Prof. Mehdi Dehghan and Prof. Mohammad Reza Jahed-Motlagh, my advisors of the master project, helped me in a key moment and supported my decision to pursue a PhD degree.

My time spent at UPC would not have been nearly as memorable without my officemate in C6-221: Abhishek Deb, for sharing with me the good and the bad moments and accompanying me for having a light dinner every Friday after my weekly meeting. I would like to thank my friends from UPC- Manel, Ruben, Maziar, Pere - for being available for coffee breaks and being lunch buddies. Talking with Victor who was interested in my culture makes me happy to feel less far from home.

The companionship from all of my friends makes me feel like the luckiest person on the planet. Reza (my best-friend for more than 14 years) simply gives without expecting anything in return. I couldn't ask anything more from a friend like you. I appreciate how we remain close friends after all these years being apart. I thank Leila who is always smiling with Reza through the Skype and remind me good memories with them. I'm grateful for Hojjat who treated and supported me like his brother and putting a big smile on my face every afternoon at 16:00 whenever I phone him and hear his voice. I also thank him and his wife, Mahzad, for sharing their time on the weekends with me and Maryam. I thank Safa, Hélio and their cute girl Lunna for their many home-cooked meals and for inviting me over for the holidays. There are many, many, many other nice people whom I have left out but were a big part of enriching my life and keeping me sane and in good shape at UPC.

I am most grateful to my parents and my sister, Sareh, without whom I could not possibly have gone this far. Although we are thousands kilometers apart, my heart has never left you. Finally, I would like to thank my dear wife Maryam who was beside me from the first to the end. She has been my colleague sitting next to my table and the first one listening my ideas and questions patiently. She has given me relentless support, warmth, comfort, and love all along.

I am grateful for the financial support of AGAUR-Generalitat de Catalunya through the grant FI-DGR (2009FI_B1-00257). I want to also thank the Spanish government through projects CICYT TIN2010-21378-C01-02 (Cooperative and Opportunistic Wireless Heterogeneous Access Networks (COWHAN)), CICYT 2007-66869-C02-01 (Ad Hoc, Mesh and Cellular Networks Integration (IRAMCEL)) and the Generalitat de Catalunya through project 2009-SGR-1167 (Regional Project).

CHAPTER 1

Introduction

In recent years, Wireless Mesh Networks (WMNs) have already become very popular and been receiving an increasing amount of attention by the research community. In WMNs, nodes are comprised of mesh routers and mesh clients. Each node operates not only as a host but also as a router, forwarding packets on behalf of other nodes that may not be within direct wireless transmission range of their destinations [5, 6]. A WMN is dynamically self-organized and self-configured, with the nodes in the network automatically establishing and maintaining mesh connectivity among themselves. This feature brings many advantages to WMNs such as easy network maintenance, robustness and reliable service coverage [4].

Compared to wired networks, routing in WMNs is specially challenging because of two fundamental differences. The first one is the heterogeneous characteristics of the wireless links: due to the strong dependency of radio transmission impediments between the nodes with their distance and the environmental elements influencing the radio waves propagation. As a consequence, packet delivery probabilities may be significantly different for every link of a WMN. The second one is the broadcast nature of wireless transmissions: unlike wired networks, where links are typically point to point, when a node transmits a packet in a wireless network, the packet can be received by several neighboring nodes simultaneously.

Traditional routing protocols proposed for wireless networks perform best path routing, i.e., they pre-select one fixed route before transmissions starts. Each node in a route uses a fixed neighbor to forward toward the destination. Doing this way, in the routing table of every node participating in the routing between a source and a destination, there is a forwarding entry which points to a neighbor (referred to as *next-hop*), over which packets addressed to the destination will be sent. Note that, once all next-hops have been chosen, all packets between a source and destination follow the same path. This motivates the name of *uni-path* routing for such type of protocols. These approaches borrowed from the routing protocols for wire-line networks, and

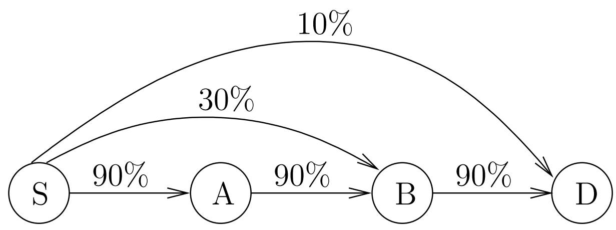


Figure 1.1: An example of Opportunistic Routing.

do not adapt well to the dynamic wireless environment where transmission failures occur frequently.

Opportunistic Routing (OR), also referred to as diversity forwarding [47] or any-path routing [30, 32], is being investigated to increase the performance of WMNs by taking the advantage of its broadcast nature. In OR, in contrast to traditional routing, instead of pre-selecting a single specific node to be the *next-hop* as a forwarder for a packet, an ordered set of nodes (referred to as *candidates*) is selected as the potential *next-hop* forwarders. We shall refer to the ordered set of candidates of a node as its Candidates Set (CS). Thus, the source can use multiple potential paths to deliver the packets to the destination. More specifically, when the current node transmits a packet, all the candidates that successfully receive it will coordinate with each other to determine which one will actually forward it, while the others will simply discard the packet.

For a better understanding of the inherent benefits associated to OR, consider the example shown in figure 1.1 (the example has been taken from [11]). It presents the possibility that one transmission may reach a node which is closer to the destination than the particular *next-hop* in traditional routing.

Assume that S is the source, D is the destination and the packet transmissions in each link are Bernoulli with the delivery probabilities specified over the links. The best path from the source to the destination using traditional routing is S-A-B-D which has $0.9 \times 0.9 \times 0.9 \approx 0.72$ end-to-end delivery probability. It minimizes the expected number of transmissions from node S to the destination D, $3 \times 1/0.9 \approx 3.33$. If a packet sent by S is correctly received by B but not node A, it has to be retransmitted by S until it reaches the designated *next-hop* A. Another situation that might happen is that a packet sent by S is correctly received by both node A and B. Although node B is closer to the destination than node A, it is not allowed to forward the packet. In contrast to the traditional routing protocols, OR takes the advantage of these situations to maximize the packet progress to the destination. An OR protocol can use $\{D, B, A\}$ as the candidates (D is the highest priority candidate, and A the least one) to forward the packet. If both nodes A and B receive the packet but not D, since node B has

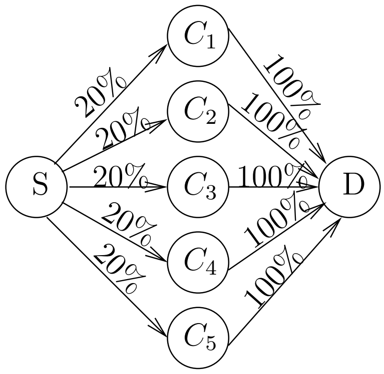
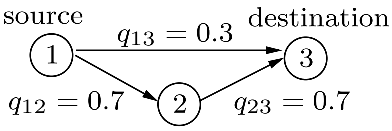


Figure 1.2: An illustration of virtual link in Opportunistic Routing.

more priority than A (it is closer to the destination), then it will forward the packet while node A will simply discard it.

Another benefit of OR is that it increases the reliability of transmissions by combining weak physical links into one strong *virtual link*. In other words, it acts like OR has additional backup links and the possibility of transmission failure is reduced [35]. As shown in figure 1.2 the sender has a low delivery probability to all its neighbors, while they have a perfect link to the destination. Under a traditional routing protocol, we have to pick one of the five intermediate nodes as the relay node. Thus, altogether we need 5 transmissions on average to send a packet from the source to the relay node and 1 transmission from the relay node to the destination. In comparison, under OR, we can select the five intermediate nodes as the candidates. The combined link has a success rate of $(1-(1-20\%)^5) \approx 67\%$. Therefore, on average only $1/0.67 = 1.48$ transmissions are required to deliver a packet to at least one of the five candidates, and another transmission is required for a candidate to forward the packet to the destination, so on average it takes only 2.48 transmissions to deliver a packet to the destination.

The following simple example illustrates how OR works, and its potential to improve the network performance. Consider the network topology and the Candidates Set (CS) of each node in figure 1.3. Node 1 is the source and node 3 is the destination. Assume that packet transmissions in each link are Bernoulli with the delivery probabilities from node i to node j, q_{ij}, indicated in the figure. Note that, in the Candidates Set (CS) of node 1, node 3 has higher priority than node 2. Note also that node 2 has only one candidate (the destination). Therefore, upon being the next forwarder, node 2 would behave as in traditional routing. We now compute the expected number of transmissions from node 1 to the destination using OR (E_1^{OR}). We can write: $E_1^{OR} = 1 + \sum_{i=1}^{3} p_i\, E_i$, where p_i is the probability of node i being the next forwarder (or the destination), and E_i is the expected



Network topology.

node	candidates to 3
1	3, 2
2	3

Candidates Set of each node.

Figure 1.3: An example of simple network topology.

number of transmissions from node i to the destination (note that $E_3 = 0$ and $E_1 = E_1^{OR}$). Grouping terms we have $E_1 = (1 + p_2 \times E_2)/(p_2 + p_3) = (1+(1-q_{13})\, q_{12} \times 1/q_{23})/((1-q_{13})\, q_{12}+q_{13})$. Substituting we get $E_1^{OR} \approx 2.15$. By comparing with E_1^{uni}, we obtain that using OR in this simple example could reduce the expected number of transmissions about 25 %.

1.1 Dissertation Overview

In the following we outline the contents of each of the remaining chapters.

Chapter 2 "Background and Related Work" reviews the material and the related work. This includes work related to main issues in OR like metrics which are used in OR, candidate selection algorithms, related works on using OR in multicast protocols and analytical works that have been done in OR.

Chapter 3 "Performance Modeling of Opportunistic Routing" proposes a Markov chain model that can be used to evaluate OR mechanisms in wireless networks in terms of the number of transmissions needed to reach the destination node. For each node, the ordered list of candidates and the delivery probability to each of them are inputs to our model. Hence, our model does not require any specific assumptions about the network topology nor the mechanism for selection and prioritization of candidates. This work was published in [17].

Chapter 4 "Candidate Selection Algorithms in Opportunistic Routing" compares four candidate selection algorithms that have been proposed in the literature. They range from non-optimum, but simple, to optimum, but with a high computational cost. We address the questions: Is there a big difference in performance between the simple and optimal algorithms? What is the computational cost as a function of maximum number of candidates? So, under which conditions it is worth using an optimal algorithm? We have published the results of this chapter in [22, 26].

Chapter 5 "Distance Progress Based Opportunistic Routing" proposes a new metric that we call *Expected Distance Progress (EDP)*. It measures the expected distance progress of sending a packet using a set of candidates. Based on EDP, we propose a hop-by-hop candidate selection and prioritization algorithm. This contribution is accepted in [24].

Chapter 6 "Maximum Performance of Opportunistic Routing" studies the maximum gain that can be obtained using OR. We address the question: What is the maximum gain that can be obtained using OR? More specifically, we are interested to answer that question in the case that, the maximum number of candidates per node is limited. We derive equations that yield the distances of the candidates in OR such that the per transmission progress towards the destination is maximized. Based on the equations that we have obtained in this chapter we have proposed a new candidate selection algorithm. We have published some of the results of this chapter in [16].

Chapter 7 "Multicast Delivery using Opportunistic Routing" investigates how OR can be used to improve multicast delivery. There are few works that have been made to adapt Opportunistic Routing (OR) in multicast. In this chapter we propose a new multicast routing protocol based on Opportunistic Routing for Wireless Mesh Networks (WMNs), named Multicast Opportunistic Routing Protocol (MORP). We compare our proposal with two well known On-Demand Multicast Routing Protocol (ODMRP) and Adaptive Demand-Driven Multicast Routing (ADMR) multicast protocols. Our results demonstrate that MORP outperforms ODMRP and ADMR, reducing the number of data transmissions and increasing the delivery ratio. A part of this work is published in [23].

Chapter 8 "Conclusions and Future Research Directions" closes this dissertation with an enumeration of some of our important results, some avenues of future work, and a discussion of the outlook for Opportunistic Routing.

1.2 Thesis Contributions

The main contributions of this dissertation are listed as follows:

- **Chapter 3 "Performance Modeling of Opportunistic Routing"**
  - We propose a Discrete Time Markov Chain to analyze the performance of OR in terms of total number of transmissions needed to reach the destination.
  - We use our model to investigate the performance of different scenarios in terms of the expected value and variance of the number

of transmissions needed to send a packet from the source to the destination.

  - We show that using a small number of candidates may be a sensible choice to have small expected number of transmissions.

- **Chapter 4 "Candidate Selection Algorithms in Opportunistic Routing"**
  - We compare a selected group of candidate selection algorithms that have been proposed in the literature.
  - We introduce some changes in the algorithm to adapt them with the constraint maximum number of candidates.
  - We investigate the performance of each candidate selection algorithm in terms of the expected value, variance and the probability of the number of transmissions needed to send a packet from a source to the destination. We also computed the execution time to construct the Candidates Sets.
  - We show that optimal algorithms have a high computational cost, even for a small maximum number of candidates.
  - We find that a fast and simple OR candidate selection algorithm is preferable in dynamic networks, where the Candidates Sets (CSs) are likely to be updated frequently.

- **Chapter 5 "Distance Progress Based Opportunistic Routing"**
  - We propose a new metric for candidate selection based on the Expected Distance Progress (EDP) of sending a packet under OR.
  - We propose a hop-by-hop candidate selection and prioritization algorithm based on the distance progress metric.
  - We show that, the performance of our algorithm is almost the same as the optimum candidate selection algorithm, while our algorithm requires less information and runs much faster.

- **Chapter 6 "Maximum Performance of Opportunistic Routing"**
  - We derived the equations that yield the distances of the candidates in OR such that the per transmission progress towards the destination is maximized.
  - We show that the Maximum Progress Distances (MPDs) for the already existing candidates do not change if we decide to add a new candidate to the Candidates Set.

  - We propose a lower bound to the expected number of transmissions needed to send a packet using OR.
  - We investigate the sensitivity of OR performance to the position of the candidates and we conclude that in order to minimize the expected number of transmissions, choosing the distance of the first two candidates near to their optimal positions, is the most critical.
  - We propose a rule of thumb in the design of the node positions in a static network using OR.
  - We also investigate the maximum performance using OR in the grid scenarios and conclude that using our rule of thumb in the design of a grid topology causes a good performance close to the optimum.
  - Based on the MPDs, we propose a new candidate selection algorithm which its performance is very close to the optimum algorithms while runs much faster than the others.

- **Chapter 7 "Multicast Delivery using Opportunistic Routing"**
  - We investigate the advantages of using OR to support multicast delivery.
  - We propose a new multicast routing protocol based on OR.
  - We compare our proposal with two well known multicast routing protocols.
  - We show that using OR can be a useful technique to implement reliable multicast protocols in WMNs.

CHAPTER 2

Background and Related Work

2.1 Introduction

This chapter gives an overview of related work with the emphasis of the contributions of this dissertation. We survey the main issues in OR which are candidate selection algorithms, routing metrics and candidate coordination methods. Then, we survey most important research contributions in each issue.

2.2 Issues in Opportunistic Routing

Three main issues arise in the design of OR protocols:

- **Candidate selection** All nodes in the network must run an algorithm for selecting and sorting the set of neighboring nodes (candidates) that can better help in the forwarding process to a given destination. We shall refer to this algorithm as *candidate selection*. The aim of candidate selection algorithms is to minimize the expected number of transmissions from the source to the destination. In section 2.5 we briefly describe some noteworthy candidate selection algorithms that have been proposed in the literature.

- **OR metric** In order to accurately select and prioritize the Candidates Sets (CSs), OR algorithms require a metric. First OR algorithms were based on simple metrics inherited from traditional unicast routing, as those used by Shortests Path First (SPF) algorithms. However, some researchers realized that more accurate metrics were required in OR. Different metrics in OR will be discussed in detail in section 2.3.

- **Candidate coordination** is the mechanism used by the candidates to discover which one has the highest priority that has received, and thus, must forward the packet. Coordination requires signaling among the nodes, and imperfect coordination may cause duplicate transmission of packets. In section 2.4 we describe three main categories of candidate coordination used in the literature.

2.2.1 Research Directions in Opportunistic Routing

In this section we give an overview of the main research contributions in OR. Table 2.1 shows some of the OR research contributions found in the literature that will be described in the following sections. The meaning of the columns is the following:

- **Protocol**: Here there is the name of the protocol coined by the authors, or NA if no name was given. The corresponding reference is also provided here.
- **Year**: Year of publication.
- **Type**: Method to obtain the numerical results presented in the paper. We use the keys: *S*, for Simulation; *A*, for Analytical; and *E*, for Experimental.
- **Topic**: Main topic of the paper.
- **Metric**: Metric used by the candidate selection algorithm.
- **Coord.**: Coordination method used in the paper. The table shows *NA* in those papers where a perfect coordination is assumed without relying in any specific type of coordination.
- **Cand. Sel.**: Information used by the candidate selection algorithm: *Topology* when it is related with the topological graph of the network, and *Location* when it uses the geographical position of the nodes.

Entries in table 2.1 are sorted in chronological order. The table shows the increasing interest that has emerged related with OR in the last decade.

Table 2.1: Classification of research works in Opportunistic Routing protocol.

Protocol	Year	Type	Topic	Metric	Coord.	Cand. Sel.
SDF [47]	2001	S	Candidate coordination	ETX	Ack	Topology
GeRaF [97]	2003	A/S	Candidate coordination	Geo.	RTS-CTS	Location
ExOR ver-1 [12]	2004	S	Candidate selection	ETX	Ack	Topology
ExOR ver-2 [11]	2005	E	Candidate coordination	ETX	Timer	Topology
NA [72]	2005	A/S	Sensor networks	Geo.	RTS-CTS	Location
COPE [43, 44]	2005	E	Network coding	ETX	Net. coding	Topology
OAPF [96]	2006	S	Candidate selection	ETX/EAX	Ack	Topology
LCOR [31]	2007	S	Candidate selection	EAX	NA	Topology
MORE [18]	2007	E	Network coding	ETX	Net. coding	Topology
GOR [88]	2007	S	Candidate selection	Geo.	Timer	Location
NA [63]	2008	A	Analytical	Geo.	NA	Location
NA [7]	2008	A/S	Analytical	Geo.	NA	Location
NA [33]	2008	E	Candidate selection	ETX	Ack	Topology
CORE [84, 83]	2008	S	Network coding	Geo.	Timer	Location
MTS [58]	2009	S	Candidate selection	EAX	Timer	Topology
POR [85]	2009	S	Candidate selection	Geo.	Timer	Location
SOAR [70]	2009	S/E	Candidate selection	ETX	Timer	Topology
Pacifier [45]	2009	S	Multicast	ETX	Net. coding	Topology
NA [16]	2010	A	Maximum performance	EAX	NA	Location
MSTOR [52]	2010	S	Multicast	EAX/ETX	Ack	Topology
MORP [23]	2011	S	Multicast	ETX	Ack	Topology
NA [27]	2011	A	Analytical/cand. selec.	ETX/EAX	NA	Topology

NA in the column **Coord.**: Perfect coordination.

Most of the research in OR is related with the issues described in section 2.2, but there are other areas of OR that have been investigated as well. We have identified the following as the main topics on research in OR:

- Metrics, section 2.3.
- Candidate coordination, section 2.4.
- Candidate selection algorithms, section 2.5.
- Network Coding (NC), section 2.6.
- Geographic OR, section 2.7.
- Multicast OR, section 2.8.
- Analytical models in OR, section 2.9.
- Sensor networks.

Each of these topics will be addressed in the next sections as indicated above.

Although many of the OR proposals can be adapted for sensors networks, there are some contributions that specifically study OR in this context. As an example, we have included [72] in table 2.1. In this paper the authors take into consideration how OR can be exploited when there are the characteristic power down periods that occur in sensor networks. Due to the limited number of works in this specific area, we do not analyze this topic further.

2.3 Routing Metrics

The general aim of OR is to minimize the expected number of transmission required to carry a packet from the source to the destination. The set of candidates which each node uses and and priority order of them have a significant impact on the performance that OR can achieve. Therefore, using a good metric to select and order the candidates is a key factor in designing an OR protocol.

Candidates in OR can be prioritized based on hop count [87, 78, 36], geographic-distance [97, 93] (Geo-Distance), Expected Transmission Count (ETX) [29], Expected Any-path Transmission (EAX) [96] and so on. Utilization of hop count, ETX or EAX needs an underlying routing protocol (either reactive or proactive) to gather such information. Geo-Distance requires the availability of location information of nodes. The accuracy of a metric depends on the proper measurement of link quality and timely dissemination of such information [61, 76]. Below, we describe the two usual metrics ETX and EAX that have been used in the literature.

Expected Transmission Count (ETX) [29]: is the average number of transmissions required to reliably send a packet across a link or route including retransmissions. The ETX of a single path route is the sum of the ETX for each link in the route. With the assumption of the packet transmission between nodes i and j as Bernoulli trials with delivery probability p_{ij}, the expected transmission count of the link is:

$$ETX(i,j) = \frac{1}{p_{ij}} \tag{2.1}$$

In OR, however, it is necessary to consider the fact that there are some candidates which can receive the packet, thus, a packet may travel along any of the potential paths. Authors in [31, 58] have shown that using ETX may give suboptimal selection of candidates and in [62] it was shown that OR in combination with ETX could degrade the performance of the network. Because of that Zhong et al. [96] proposed another metric which has been widely adopted in OR.

Expected Any-path Transmission (EAX) [96]: is an extension of ETX and can capture the expected number of transmissions taking into account the multiple paths that can be used under OR. Alternative methods to compute EAX have been proposed by different authors [31, 58, 17]. In chapter 3 we present a model for OR that can be used to calculate the expected number of transmissions from source to the destination.

2.4 Candidate Coordination Methods

One of the important issues of OR is the candidate coordination, i.e, the mechanism used by the candidates to discover which is the highest priority candidate that has received, and thus, must forward the packet. Coordination requires signaling between the nodes, and imperfect coordination may cause duplicate transmission of packets. A good coordination approach should select the best candidate without duplicate transmissions while using the smallest time/or control overhead.

Existing coordination approaches are divided into three main categories based on the mechanism used: acknowledgment-based (ACK-based), timer-based, Network Coding (NC) and Request-To-Send/Clear-To-Send (RTS-CTS) Coordination. In the following subsections we briefly describe these approaches.

2.4.1 Acknowledgment-Based Coordination

It is one of the first methods that was proposed for candidate coordination. Upon receiving a data packet, candidates send back a short acknowledgment (ACK) in decreasing order of candidate priority.

This method was first proposed in [47] as the coordination mechanism for the Selection Diversity Forwarding (SDF) protocol. In SDF, coordination is achieved by means of a four-way-handshaking: the candidates receiving the data packet send back an acknowledgment to the sender. Based on the acknowledgments, the sender sends a forwarding order to the best candidate, which is also acknowledged.

A similar approach is used in Extremely Opportunistic Routing (ExOR) [12], which uses a modified version of the 802.11 MAC which reserves multiple slots of time for the receiving nodes to return acknowledgments. Instead of only indicating that the packet was successfully received, each ACK contains the ID of the highest priority successful recipient known to the ACKs sender. All the candidates listen to all ACK slots before deciding whether to forward, in case a low-priority candidates ACK reports a high-priority candidate ID and whose ACK was not correctly received. Including the ID of the sender of the highest-priority ACK heard so far helps to suppress duplicate forwarding. This strategy requires that candidates be neighbors of each other such that the transmission of an ACK can be overheard by all of them.

As an example of the ACK-based coordination, consider a network with source S and destination D. Assume that the CS of S is $\{A, B, C\}$ (A has the highest priority and C has the lowest). Suppose that all candidates receive a transmission from source. Figure 2.1 shows ACK-based coordination method for this example. All candidates transmit acknowledgments in decreasing order of candidate priority: the first acknowledgment slot belongs to node A, the second slot belongs to node B and the third slot is dedicated to C. In figure 2.1 we suppose that the acknowledgment from A does not receive by B, but node C does hear the A's ACK (see figure 2.1). Suppose further that node B hears node C's ACK. If ACKs did not contain IDs, node B would forward the packet, since to its knowledge it is the highest priority recipient. The fact that node C's ACK contains node A's ID indirectly notifies B that node A did receive the packet. Once node A has successfully determined itself as the responsible node, it forwards the packet.

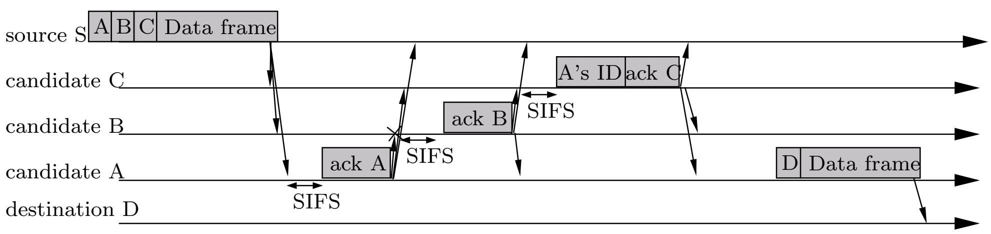


Figure 2.1: Acknowledgment-based coordination using a modified 802.11 MAC.

2.4.2 Timer-Based Coordination

In this method, all candidate which are included in the packet are ordered based on a metric. After a data packet is broadcasted, candidate will respond in order, i.e, ith candidate will respond at the ith time slot. A candidate forwards data packet at its turn only when it does not hear other candidates that forward the packet. Thus, when a candidate forwards a data packet, it means that all other higher priority candidates failed to receive the data packet. In another words, forwarding a data packet by a candidate will prevent the lower priority ones to forward it. In the example of figure 1.1, assume that $\{B, A\}$ is the CS of source S to reach destination D (B is the better candidate and given the higher priority). After receiving the packet sent by S, candidate B forwards the packet in the first time slot, while A schedules to transmit in the second time slot. If A is in the range of B, overhearing the data packet sent from B by A means that a higher priority candidate received the packet and has forwarded it, thus A simply discard the packet.

This approach is simple and easy to implement and no control packet is required. The overhead of the timer-based coordination is candidate waiting time. The main drawback of this solution is duplicate transmission because of not all candidates are guaranteed to overhear the forwarding from the selected candidate [35].

2.4.3 Network Coding Coordination

Another approaches to prevent duplicate transmission is combining Opportunistic Routing (OR) with Network Coding (NC) [3] which provides an elegant method for candidate coordination [18, 13, 84]. The basic principle behind combining NC with OR is that forwarders can combine the packets to be transmitted so as to deliver multiple data packets through a single transmission. When transmitting packets from source to a destination, a flow is divided into batches which contain several native packets (original packets without coding). The source broadcasts random linear combinations of native packets, and candidates forward the linear combinations of received coded packets. When the destination has enough linearly independent coded packet, then it can decode them to reconstruct the set of initial packets.

In order to better clarify the advantage of combining NC with OR, consider the example in figure 2.2. Assume that source S transmits two packets a and b using Candidates Set $\{C_1, C_2\}$. Assume that C_2 receives both packets but C_1 receives only one of them (see figure 2.2). Node C_1 transmits first because it is closer than C_2 to the destination. Node C_2 has the following three choices: forwarding a, b, or both a and b. In the Network Coding (NC),

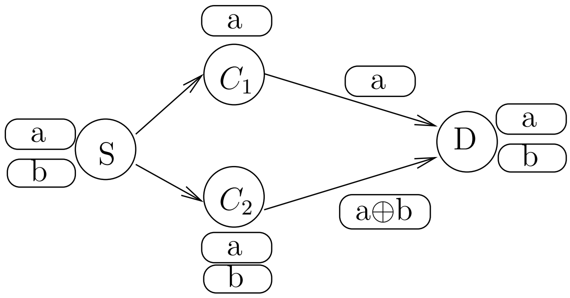


Figure 2.2: Network Coding coordination approach.

node C_2 can forward a coded packet $a \oplus b$. When D receives transmitted packets from C_1 and C_2, it can decode and restore the original packets. It performs an XOR operation on the two received packets: $a \oplus b \oplus a = b$. Thus, no duplicate transmission occurs at D.

However, using NC with OR may lead to a high number of potential forwarders sending coded packets, and thus, resulting in redundant transmissions. There exists a trade-off between transmitting a sufficient number of coded packets to guarantee that the destination has enough coded packets to reconstruct the native packets, and avoiding to inject in the network unnecessary packets [13].

2.4.4 RTS-CTS Coordination

Some other mechanisms like in [36, 97] use explicit control packet(s) exchanged immediately before sending a data packet. In this approach the sender multicasts the RTS to the its CS (it is actually a broadcast control packet). The RTS contains all the candidates addresses which are ordered according to a metric. When an intended candidate receives the RTS packet, it responds by a CTS. These CTS transmissions are sent in decreasing order of candidate priority: the first candidate in priority transmits the CTS after a SIFS, the second one after 2×SIFS, and so on. When the sender receives a CTS, it transmits the DATA packet to the sender of this CTS (which would be the highest priority candidate that responded) after a SIFS interval. This ensures that other lower priority candidates hear the DATA before they send CTS and suppress any further CTS transmission. All such receivers then set their Network Allocation Vector (NAV) until the end of ACK period. This mechanism is guaranteed to have a single winner and it can avoid duplicate transmissions.

Figure 2.3 shows an example of RTS-CTS coordination. Assume that there are three candidates A, B and C to reach the destination (A the highest priority candidate and C the least one). After receiving RTS by candidates

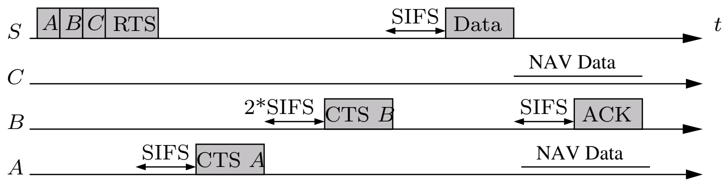


Figure 2.3: RTS-CTS coordination approach.

they send the CTS packet in order of their priorities. Here we assume that the first CTS which belong to A was not received, but the second one was received. When the sender S receives the first CTS from B, it sends the data packet to it, therefore the highest priority candidate whose its CTS is received by the source will forward the data packet.

2.5 Candidate Selection Algorithms

Another important component of OR is candidate selection, which is similar to building routing tables in traditional routing. Selection of good candidates can affect the performance of the network.

According to the amount of information is needed to select and prioritize the candidates, candidate selection algorithms can be divided into two categories; Location-based and Topology-based selection. In location-based selection [97], each node maintains a limited state information and independently determines its own CS along the path to the intended destination. Topology-based selections [12, 11] find the CSs according to the global topology information of the network. Therefore, a node requires to maintain global network state information, for example, the network topology, state information on each link, and flow-related information (e.g., path and data rate), what can run into a scalability problem. In general, topology-based strategy outperforms location-based strategy, since the former can optimize the selection of a CSs with more network state information gathered. However, the location-based strategy might be easier to implement, requires less signaling and scales better than topology-based [61].

Biswas and Morris proposed ExOR [12, 11], one of the firsts and most referenced OR protocols. In the first version of ExOR [12], all candidates must be one hop neighbors of each other, such that they are able to hear the acks sent by the others. The candidates are ordered by a priority according to their best position to forward the packets. When a packet is sent, the ordered list of candidates is included in the MAC header. All candidates that receive the packet correctly send in turn an ACK. Then, based on the ACKs received from the other candidates, decide locally whether to retransmit the packet.

Simple Opportunistic Adaptive Routing (SOAR) [70] has been proposed after ExOR. In order to leverage path diversity while avoiding duplicate transmissions, SOAR relaxes the actual route that data traverses to be along or near the default path but constrains the nodes involved in routing a packet to be near the default path. Moreover, this forwarding node selection also simplifies coordination since all the nodes involved are close to nodes on the default path and can hear each other with a reasonably high probability. It selects the shortest path between source and destination using ETX, and the nodes near to the shortest path can act as the Candidates Set. SOAR uses timer-based approach for candidate coordination.

In [96, 95] Zhong et al. proposed a new candidate selection and prioritization rule based on EAX. They analyzed the efficacy of OR by using this metric and did a comparison using the link-level measurement trace of MIT Roofnet [1]. Moreover, they claimed that the number of candidates should be kept moderate in order to reduce the number of transmissions. Consequently, only those nodes that would reduce significantly the forwarding cost are included in the candidate set. In [57] Li et al. introduced a new metric —*Successful Transmission Rate (STR)*– for choosing the forwarder list. They considered multi-links contribution, instead of one best link information used in [96, 95]. They proposed a fair OR (FORLC) protocol that used STR as a metric to select the Candidates Set.

In [30] a distributed algorithm for computing minimum cost opportunistic routes, which is a generalization of the well-known Bellman-Ford algorithm, is presented. The authors also alert about the risk of using too many relay candidates. An optimization based approach to OR when multiple transmission rates are possible has been considered in [89, 30, 49]. In [58] the key problem of how to optimally select the forwarder list is addressed, and an optimal algorithm (MTS) that minimizes the expected total number of transmissions is developed.

2.6 Network Coding Opportunistic Routing

MAC-independent Opportunistic Routing & Encoding (MORE) [18] is an OR protocol that uses both the idea of OR and Network Coding (NC). It deploys the advantages of NC to improve performance of OR in wireless multicast networks. Duplicate transmissions are avoided by randomly mixing packets before forwarding. The sender creates a linear combinations of packets and broadcasts the resulting packet after adding a MORE header containing the CS. Each receiving node discards the packet if it is not linearly independent from the other packets received before, or if its ID does not appear in the candidates list. Otherwise, it linearly combines the received coded packets and rebroadcasts the new packet.

COPE [43, 44] is a practical NC mechanism for supporting efficient unicast communication in a Wireless Mesh Network. It employs opportunistic listening to enable each node to learn local state information and encoded packet broadcasting to improve the network throughput. It exploits the shared nature of the wireless medium which broadcasts each packet in a small neighborhood around its path. Each node stores the overheard packets for a short time [43]. It also tells its neighbors which packets it has heard by annotating the packets it sends. When a node transmits a packet, it uses its knowledge of what its neighbors have heard to perform opportunistic coding; the node XORs multiple packets and transmits them as a single packet if each intended next-hop has enough information to decode the encoded packet. Motivated by COPE, several other coding-aware routing mechanisms have been proposed [81, 50], which are aimed at improving the network throughput by combining routing with inter-flow NC.

Coding-aware Opportunistic Routing mechanism & Encoding (CORE) [84, 83] is a coding-aware OR mechanism that combines OR and localized inter-flow NC for improving the throughput performance of a WMN. Through OR, CORE allows the next-hop node with the most coding gain to continue the packet forwarding. Through localized NC, CORE attempts to maximize the number of packets that can be carried in a single transmission. When a node has a packet to send, it simply broadcasts the packet, possibly encoded with other packet(s), which may be received by some of the candidates in its CS. The candidates receiving the packet collaborate to select the best candidate among them in a localized manner, which is the one with the most coding opportunities. This forwarding process is repeated until the packet reaches its intended destination. In CORE, geo-distance metric and timer-based coordination have been used to select and coordinate the candidates, respectively.

2.7 Geographic Opportunistic Routing

Geographic Random Forwarding (GeRaF) [97] is a forwarding protocol which selects set of candidates and prioritizes them using geographical location information. Only those neighboring nodes closer to the destination than the sender can be candidates. The priority of selected candidates is based on their geo-distances to the destination. The candidate coordination can easily be implemented via an RTS-CTS dialog at the MAC layer, which also ensures that a single forwarder can be chosen.

Geographic Opportunistic Routing (GOR) [88] is used in geographic routing scenarios and adopts timer-based coordination with local candidates order. Authors showed that giving the nodes closer to the destination higher priority is not always the optimal way to achieve the best throughput. They proposed a local metric named Expected One-hop Throughput (EOT) to characterize the local behavior of GOR in terms of bit-meter advancement per second. Based on EOT, which considers the coordination overhead, they proposed a candidate selection scheme.

S.Yang et. al. [85] proposed a protocol called Position based Opportunistic Routing (POR). In POR, when a source wants to send data packet to the destination, it finds its CS according to the distance between its neighbors and the destination. The neighbor which is the nearest to the destination will have the highest priority. They fixed the maximum number of candidates in each node to 5. When a candidate receives a packet, it checks its position in the CS and waits for some time slots to forward the packet. If it hears the same packet being sent by the other nodes, it will simply discard the packet.

2.8 Multicast Routing Protocols

Multicast is an important communications paradigm in wireless networks. It comes into play when a host needs to send the same message or data stream to multiple destinations. Due to the unique characteristics of the wireless networks such as limited resources and unreliable channels, traditional multicast protocols in the wired networks do not perform well in wireless, and new protocols have been proposed.

2.8.1 Traditional Multicast Routing Protocols

One of the most popular methods to classify multicast routing protocols is based on how distribution paths among group members are constructed. According to this method, existing multicast routing approaches can be classified into tree-based, mesh-based and hybrid protocols [39, 40].

In the tree-based protocols only a single shortest path must be established between source-receiver pair, therefore the multicast tree is composed of a unique path from the multicast source to each of the multicast receivers.

Tree-based proposals are also divided into two sub-categories: source-based tree and shared-based tree approaches. A source-based tree maintains an individual route towards all the multicast receivers for each multicast group. Some source-based multicast protocols are Differential Destination Multicast (DDM) [38], Preferred Link Based Multicast (PLBM) [74], Adaptive Demand-driven Multicast Routing [37] and probabilistically reliable on-demand (PROD) [94].

Since the construction of a separate tree for each source is costly, some tree-based multicast protocols use a shared-based (core-based) tree to distribute the multicast messages. In shared-based tree a single tree is constructed to support the whole groups. Since the shared-based multicast tree only permits the multicast traffic to be sent out from the root to the multicast receivers, each multicast source must forward its multicast traffic to the root initially. Multicast traffic of each source is then forwarded along the shared tree. Ad-hoc Multicast Routing utilizing Increasing ID numbers (AMRIS) [79], Multicast Ad-hoc On-demand Distance Vector routing (MAODV) [69], Multicast Zone Routing (MZRP) [92] and Adaptive Core based Multicast routing (ACMP) [41] are some popular shared-based tree multicast routing protocols.

The main advantage of a tree as the underlying forwarding structure is that the number of forwarding nodes tends to be reduced. However, they generally suffer from fragile tree structure [40]. Besides the previous problem, source-based tree proposals also suffer from large memory space requirements and wasteful usage of limited bandwidth because each source constructs its own tree. But, it performs better than shared-based tree proposals at heavy loads due to efficient distribution of trees. Although shared-based tree proposals are more scalable, they have the vulnerability of the single core problem [8].

In a mesh-based multicast routing protocol, multiple routes may exist between any pair of source and destination, which is intended to enrich the connectivity among group members. The major difference between the tree-based and mesh-based protocols lies in the manner in which a multicast message is relayed. In tree-based protocols, each intermediate node on the tree has a well-defined list of the next-hop nodes for a specific multicast session. It will send a copy of the received multicast message to only the neighboring nodes on its next-hop list. In mesh-based protocols, each node on the mesh will broadcast the message upon its first reception of the message. Mesh-based multicast routing protocols generally are robust due to the penalty of multiple paths between different nodes. But many of these proposals suffer from excessive control overhead which will affect on scalability and utilization of limited bandwidth. Examples of mesh based multicast routing protocols include On-Demand Multicast Routing (ODMRP) [55, 56] and its variations (PatchODMRP [53], PoolODMRP [15], PDAODMRP [71], EnhancedODMRP [65] and Resilient ODMRP [66]), Forwarding Group Multicast (FGMP) [19], Core-Assisted Mesh (CAMP) [34], Clustered Group Multicast (CGM) [60], Neighbor-Supporting Multicast (NSMP) [54], Dynamic Core based Multicast routing (DCMP) [28] and link stability based multicast routing in MANETs (LSMRM) [9].

Hybrid multicast routing protocols combine the advantages of both tree-based and mesh-based multicast approaches, i.e., the robustness of the mesh-based multicast routing protocols and low overhead of tree-based protocols.

Therefore, the hybrid multicast routing protocols are able to address both efficiency and robustness issues. Multicast Core-Extraction Distributed Ad Hoc Routing (MCEDAR) [73], Ad-hoc Multicast Routing (AMRoute) [82] and Efficient Hybrid Multicast Routing (EHMRP) [10] are some well-known hybrid multicast routing protocols.

2.8.2 Multicast Opportunistic Routing Protocol

There are few works that have been made to adapt OR in multicast. The availability of multiple destinations can make the selection of CSs and the coordination between candidates complicated. There are few works that have tried made to adapt OR for multicast scenarios.

MORE [18] which is explained in section 2.6 is a MAC independent protocol that uses both the idea of OR and NC. It can be used in both unicast and multicast scenarios.

In [45] the source first creates the shortest path tree to reach all destinations based on the ETX of each link. Then the nodes not only receive packets from their father in the tree, but also can overhear packets from its sibling nodes. It uses random linear NC to improve multicast efficiency and simplify node coordination.

The authors in [77] used a Steiner tree based on ETX and data packets were forwarded through the links using OR. Their protocol constrains the nodes involved in routing a packet to be near the default multicast tree. The average EAX of each candidate to reach a sub-group of destinations is used as the cost of reaching to multiple destinations.

The authors in [51] proposed a Multicast Opportunistic Routing (MOR) algorithm. It opportunistically employs a set of forwarders to push a packet closer to all receivers round-by-round. They proposed a new metric – *Expected Transmission Advancement (ETA)* – which is the expected number of OR transmissions achieved after one transmission from a source node toward the destination using the Candidates Set of source. Based on packet receptions at the end of each round, a new forwarder set is constructed to maximize the ETA towards all destinations. They developed an event-driven simulator to measure the performance of their proposal. For the propagation model they used a simple packet loss which is only related to the geographic distance between two nodes. They believe that implementing of MOR using packet-level simulators is not straightforward.

In a recent work, Le and Liu [52] propose Minimum Steiner Tree with Opportunistic Routing (MSTOR) which is an overlay multicast to adapt OR in wireless network. They construct a minimum overlay Steiner tree, and map it into unicast Opportunistic Routing relay path connecting the source with all destinations. They employed unicast OR on each link of the tree. Their protocol does not exploit opportunistic receptions across different links in that tree.

2.9 Opportunistic Routing Analytical Models

There are some papers which propose analytical models to study the performance of OR. Baccelli et al. [7] used simulations to show that OR protocols significantly improve the performance of multihop wireless networks compared to the shortest path routing algorithms, and elaborated a mathematical framework to prove some of the observations obtained by the simulations.

In [63] an analytical approach for studying OR in wireless multi-hop networks have been proposed. They used lognormal shadowing and Rayleigh fading models for packet reception. In their model they assume that the nodes are uniformly distributed over the plane. The authors did not consider any specific candidate selection algorithm, but simply compute the expected progress of the packet transmissions based on the probability of any node in the progressing region successfully receives the packet. In [20] they extend their work by using directional antennas and different radio propagation models and spatial node distributions.

Zubow et al. in [46] claimed that shadow fading losses for spatially close candidates are not independent from each other, unlike commonly assumed. They presented measurements obtained from an indoor testbed and concluded that correlations can not be neglected if nodes are separated by less than 2 m. The authors of [80] proposed an utility-based model for OR and claimed that for the optimal solution it is necessary to search all loop-free routes from the source to the destination. They proposed both optimal and heuristic solutions for selecting the candidates according to their utility function. In [86, 62] an algebraic approach is applied to study the interaction of OR routing algorithms and routing metrics. They showed that OR in combination with ETX could degrade the performance of network.

In [14], the issue of optimal CS selection in the OR has been addressed. They provide an analytical framework to model the problem of selecting the optimal CS for both the constrained and unconstrained CS selection. They proposed two algorithms for optimal CS selection, one for the constrained and one for the unconstrained case.

CHAPTER 3

Performance Modeling of Opportunistic Routing

3.1 Introduction

This chapter presents a Discrete Time Markov Chain (DTMC) that can be used to evaluate OR mechanisms in terms of the number of transmissions needed to reach the destination. Using this model we have obtained a formula, which is straightforward from the DTMC, to calculate the expected number of transmissions from source to the destination under Opportunistic Routing approach. Unlike previous studies, from the proposed model closed-form expressions not only for the mean but also for higher moments and the full probability distribution are easily derived. For each node, the ordered list of candidates and the delivery probability to each of them are inputs to our model. Hence, our model does not require any specific assumptions about the network topology nor the mechanism for selection and prioritization of candidates.

3.2 Markov Model

We have considered one *tagged* connection. Each node has a set of candidates that can opportunistically route the packets towards the destination. In order to simplify the explanation of our model, we will first describe a simple scenario, and then we will generalize it.

Consider a linear network topology of N nodes equally spaced a distance $x = \frac{D}{N-1}$, being D the distance between the source s and the destination d of the tagged connection (see figure 3.1).

Let $p(x)$ be the probability of successfully delivering a packet to node located at a distance x. The nodes retransmit the packets until successful delivery. With the assumption of independent delivery probabilities, and the nodes always routing the packets to their closest neighbor, the average number of

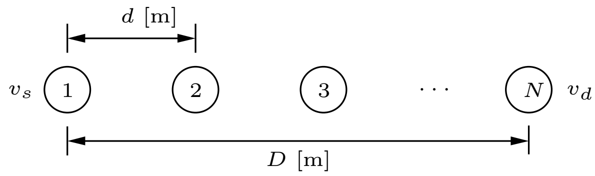


Figure 3.1: Linear network topology

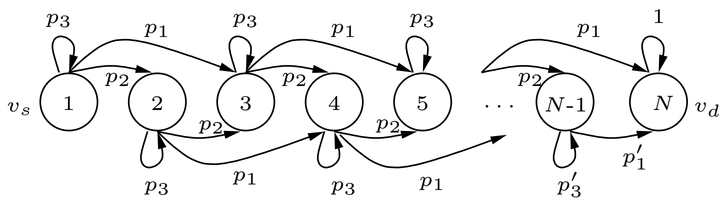


Figure 3.2: Opportunistic Routing with 2 candidates.

transmissions N_t in uni-path routing is given by:

$$N_t = \frac{N-1}{p(x)}. \tag{3.1}$$

Assume now that Opportunistic Routing (OR) is used with a list of 2 candidates. That is, we assume that upon transmission, if any of the next 2 neighbors toward the destination receive the packet, the closest node to the destination opportunistically becomes the next-hop forwarder towards the destination. We can model this routing by means of the absorbing DTMC depicted in figure 3.2. The transition probabilities are given by:

$$\begin{aligned} p_1 &= p(2\,x) \\ p_2 &= p(x)\,(1-p_1) \\ p_3 &= 1-(p_1+p_2) \\ p_1' &= p(x) \\ p_3' &= 1-p_1'. \end{aligned} \tag{3.2}$$

A similar DTMC can be easily derived for 3 candidates and so on, until all possible nodes are chosen as the candidates (we shall refer to this case as *infinite candidates*). Furthermore, the model is readily extended to an arbitrary network. The only ingredients needed to build the transition probability matrix are the Candidates Sets (CSs) involved in the routing from s to d, and the delivery probabilities to reach them. Notice that these Candidates Sets are: the candidates of node s towards d, the candidates of these candidates towards d, and so on until d (whose candidates is the empty set). This is explained in the following.

We will utilize graph theory notation for the sake of being concise. Let $G = (V, E)$ be the graph of the network. The vertex s is the source and d is the destination of the tagged connection ($s, d \in V$). Note that we will use node/vertex and link/edge interchangeably. Let $p(i, j) > 0$ the delivery probability of the edge between the pair of vertices i, j. Let $C_{i,d} = \{c_i(1), c_i(2), \cdots c_i(n_i)\}$, $C_{i,d} \subseteq V$ the ordered set of candidates of vertex i ($c_i(1)$ is the best candidate to reach d and $c_i(n_i)$ is the worst). As before, each vertex of the graph is a state of the DTMC, being d the absorbing state. The transition probabilities $p_{ij} \neq 0$ are given by:

$$p_{ij} = p(i, j), \qquad i \neq d,\ j = c_i(1) \tag{3.3}$$

$$p_{ij} = p(i, j) \prod_{l=1}^{k-1} (1 - p(i, c_i(l))), \qquad i \neq d,\ j = c_i(k),\ k = 2, \cdots, n_i \tag{3.4}$$

$$p_{ii} = 1 - \sum_{l \in C_{i,d}} p_{il} = \prod_{l=1}^{n_i} (1 - p(i, c_i(l))),\ i \neq d \tag{3.5}$$

$$p_{ii} = 1, \qquad i = d\,. \tag{3.6}$$

Note that the two expressions given for p_{ii} in equation (3.5) follow from the stochastic nature of the transition matrix (the first one), and because p_{ii} is the probability that none of the candidates ($C_{i,d}$) receives the packet (the second one).

Without loss of generality, we can number the nodes such that the source and the destination are respectively 1 and N, and for any node i, its candidates satisfy: $c_i(l \in C_{i,d}) > i$. Note that, neglecting self-transitions, the former condition implies that the graph is loop free. This condition holds assuming that the candidate selection algorithm uses some kind of strict order, i.e., for a node j to be included into the set of candidates of i, it must be strictly *closer* to the destination than i. Hence, a loop $i = j_0 \rightarrow j_1 \rightarrow j_2 \rightarrow \cdots \rightarrow j_n = i$ (by transitivity) would imply that i is strictly closer to the destination than i, which is a contradiction. This is an obvious assumption for a well designed candidate selection algorithm. Otherwise, a node i would choose as candidate a node having a larger cost to reach the destination than the node i. With these assumptions, the transition matrix of the resulting chain has the triangular form:

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & \cdots & p_{1N} \\ 0 & p_{22} & p_{23} & \cdots & p_{2N} \\ 0 & 0 & p_{33} & \cdots & p_{3N} \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{T} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \quad . \tag{3.7}$$

where $\mathbf{T}$ governs the transmissions before reaching the destination, and $\mathbf{t} = [p_{1N}\ p_{2N}\ \cdots\ p_{N-1N}]^{\mathrm{T}}$ are the probabilities to reach the destination in one transmission from the nodes $1, \cdots, N-1$.

Let X_1 be the random variable equal to the number of transitions from the source (node 1) until absorption. Note that, in our model this is the number of transmissions since the source first transmits the packet, until it is received by the destination. The DTMC obtained in our model represents a discrete phase-type distribution [48]. Thus, the point probabilities and factorial moments of X_1 are given by:

$$P\{X_1 = n\} = \boldsymbol{\tau}\, \mathbf{T}^{n-1}\, \mathbf{t},\ n \geq 1 \tag{3.8}$$

$$\mathrm{E}[X_1(X_1-1)\cdots(X_1-k+1)] = n!\, \boldsymbol{\tau}\, (\mathbf{I}-\mathbf{T})^{-n}\, \mathbf{T}^{n-1}\, \mathbf{1} \tag{3.9}$$

where we define $\boldsymbol{\tau} = \begin{bmatrix}1 & 0 & \cdots & 0\end{bmatrix}^{\mathrm{T}}$ and $\mathbf{1}$ a column vector of 1's. Note that $\mathbf{T}$, and thus also $\mathbf{I}-\mathbf{T}$, are triangular matrices, which simplifies the computation of their inverses.

3.2.1 Expected Number of Transmissions

If we are only interested on the expected number of transmissions, we can derive a recursive equation as follows. Let X_i $(i \neq d)$ be the random variable equal to the number of transitions from the state i until absorption. Clearly:

$$\mathrm{E}[X_i] = 1 + \sum_{j \in C_{i,d}} p_{ij}\, \mathrm{E}[X_j] = 1 + p_{ii}\, \mathrm{E}[X_i] + \sum_{l=1}^{n_i} p_{il}\, \mathrm{E}[X_l]$$

grouping $\mathrm{E}[X_i]$ we get:

$$\mathrm{E}[X_i] = \frac{1 + \sum_{l=1}^{n_i} p_{il}\, \mathrm{E}[X_l]}{1 - p_{ii}}, \quad i \neq d\,. \tag{3.10}$$

Taking $\mathrm{E}[X_d] = 0$, the equation (3.10) can be used to compute the expected number of transmissions needed to send a packet from the source s to the destination d by using $C_{s,d}$ as the Candidates Set (CS) of s to reach node d. Note that the loop free property of the chain guarantees that the recursive equation (3.10) is finite.

At the time we proposed this model, Li and Zhang published an analytical framework to estimate the transmission costs of packet forwarding in wireless networks [59]. Both approaches are similar in their formulation, although differ in the way the model is solved: our model leads to a discrete phase-type distribution, while in [59] transmission costs are computed using spectral graph theory.

Equation (3.10) has been obtained by other methods in [30] and [96], where it is referred to as *least cost any-path* and *Expected Any-path Transmission (EAX)* respectively (see appendix C for more details). As explained in section 2.5, some candidate selection algorithms for OR use the Expected Transmission Count (ETX) metric. Although ETX is much simple to compute than EAX, it does not accurately compute the expected number of transmissions under OR.

3.3 Selection of Candidates

In this section we elaborate on the model of section 3.2, describing the algorithm for the selection of candidates that we will use in the evaluation.

For every pair of vertices i, j from the graph at a distance d_{ij}, we use the shadowing propagation model (See appendix B for more details). The delivery probability $p(i, j) = p(d_{ij})$ between two nodes i and j is computed using equation (B.5). We have set the shadowing parameters to the default values used by the network simulator (Ns-2) [2], given in table B.2. We have assumed that an edge exists between the vertices i, j only if $p(i, j) \geq min.dp$. In practice, this minimum delivery probability threshold ($min.dp$) would be the error probability at which the routing protocol suppose the link is broken.

When no OR is used, we employ the Shortests Path First (SPF) algorithm to select the vertices that form the path from the source vertex v_s to the destination v_d. We have run the SPF algorithm using the following weights:

- *hops:* $w_{ij} = 1$, i.e. classical SPF minimizing the number of hops.
- c_1: $w_{ij} = \frac{1}{p_{ij}}$, thus, minimizing the mean number of transmissions. The reason of referring to this case as c_1 will become clear in the following.
- *log:* $w_{ij} = log(\frac{1}{p_{ij}})$. The idea behind this weight is choosing the closest neighbor among those that approach to the destination.

When OR is used, in order to construct the ordered list of the candidates of each vertex $v \in V \setminus v_d$, we have used Algorithm 1. This algorithm is similar to the one proposed in [12]. First, it runs SPF with weights $w_{ij} = \frac{1}{p_{ij}}$. The first node after the source in this path is selected as the candidate c. Then the link between the source and the candidate c is removed, and the loop is repeated until no more paths to the destination are available, or the maximum number of candidates is reached. The ETX of each node to the destination is used to sort the Candidates Set. In chapter 4 we will study this algorithm in more details. Note that there is not a simple algorithm for the

Algorithm 3.1: Compute the candidate list of vertex v.

G_{tmp} = temporal copy of the graph G
$v_{cost} \leftarrow$ "SPF cost from v to v_d in G"
while *"the number of candidates $\leq$ Maximum number of candidates"* **and** *there is connectivity from v to v_d in G_{tmp}* **do**
    $c \leftarrow$ first vertex after v in the SPF path from v to v_d in G_{tmp}
    if $c == v_d$ **then**
        add candidate v_d
        cost=0
    else
        $c_{cost} \leftarrow$ SPF cost from c to v_d in G_{tmp}
        if $c_{cost} < v_{cost}$ **then**
            add candidate c
            cost=c_{cost}
        end
    end
    delete the edge from v to c in $Gtmp$
end

optimum candidates selection. In fact, instead of using a single-path metric, as we do in the Algorithm1, an optimum algorithm would need a metric that captures the multiple paths that can be used to reach the destination (see [30]).

We have run the Algorithm 1 with a maximum number of candidates equal to 2, 3, 5 and ∞. We shall referrer to this cases as c_2, c_3, c_5 and c_∞, respectively in the rest of this chapter. Recall that we refer as *infinite candidates*, c_∞, the case where all possible nodes are candidates. Therefore, this case will be equivalent of running the algorithm 1 with a maximum number of candidates = ∞. Note also that running the algorithm 1 with a maximum number of candidates equal to 1 is equivalent to using SPF with weights $w_{ij} = \frac{1}{p_{ij}}$. This motivates, as we previously said, that we shall refer to this case as c_1.

3.4 Numerical Results

We have considered three different network topologies:

1. *Linear Topology:* The nodes are equally spaced over a line of length D. The source and the destination are placed at the line end points (see figure 3.1).
2. *Random:* The nodes are randomly placed in a square of diagonal D, except the source and the destination which are placed at the diagonal end points (see figure 3.3 (a)).

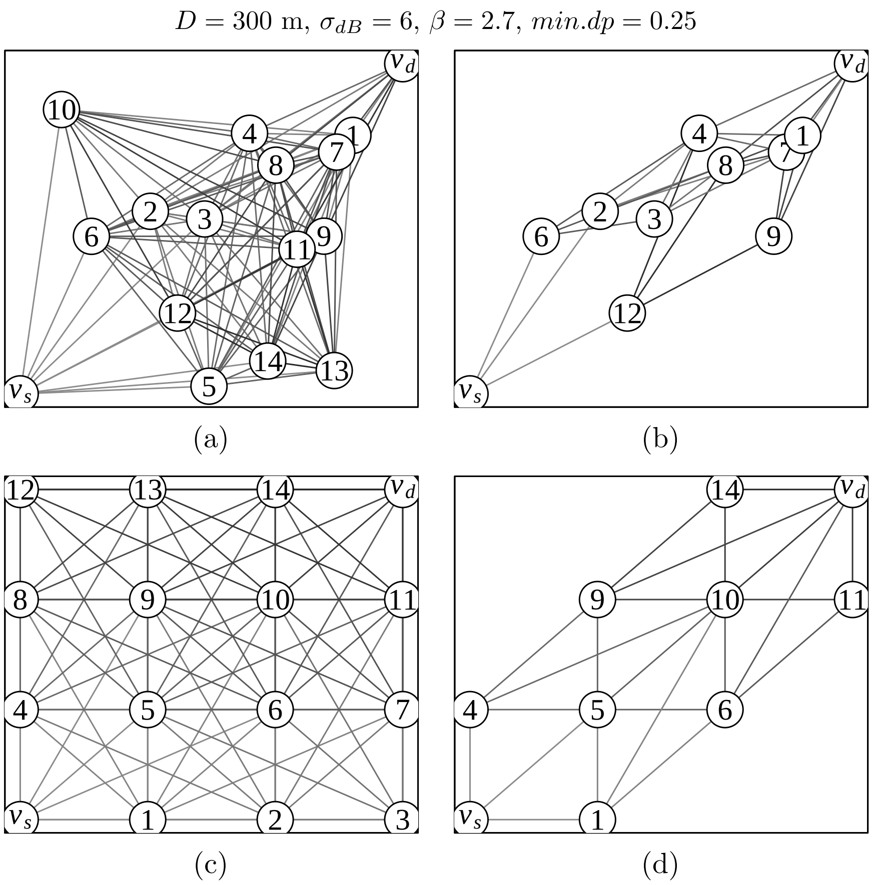


Figure 3.3: (a) Random network topology with 16 vertices. (b) The same graph with 3 candidates per vertex. The vertices that are no candidates have been pruned. (c), (d) idem for the grid topology.

3. *Grid:* The nodes are placed in a grid of diagonal D. The source and the destination are placed at the diagonal end points (see figure 3.3 (c)).

We will first do a detailed study using the linear topology in section 3.4.1. Then, we will extend our results comparing the three topologies in section 3.4.2.

3.4.1 Linear Topology

Figure 3.4 shows the results for the linear topology. It depicts the expected number of transmissions for the SPF routing using the weights *log*, *hops* and c_1 described in section 3.3, and OR using 2, 3, 5 and infinite candidates (indicated in the legend as c_2, c_3, c_5 and c_∞, respectively). The curves have been obtained varying the number of nodes (N), but maintaining the distance $D = 300$ m between the source and the destination, thus, increas-

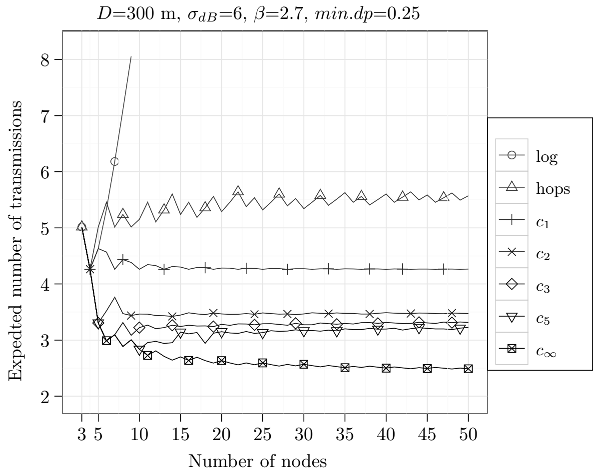


Figure 3.4: Expected number of transmissions for the linear network topology.

ing the density of the network. Unless otherwise specified, the shadowing propagation model parameters used throughout this section are $\sigma_{dB} = 6$ and $\beta = 2.7$, and the threshold for considering a link is $min.dp = 0.25$ (see appendix B).

Initially (for a number of nodes equal to 3 and 5) all curves are coincident, since there is only one possible route to reach the destination. Since $p(D/2) > min.dp > p(D)$, in the *hops* case a path with 2 hops will be selected to reach the destination. Among all paths with 2 hops it can be easily shown that minimum number of transmissions is achieved when the relay node is placed exactly at $D/2$, and then the average number of transmissions would be $2/p(150) \approx 5$. However, when N is even there will not be a node placed at $D/2$. Additionally, in the *hops* case the same weight for all reachable nodes is used (equal to 1), therefore, the algorithm will select any of the nodes that allows reaching the destination in two hops with equal probability. Thus, in the *hops* case the number of transmissions is equal or higher than 5 as it is shown in figure 3.4.

Regarding the *log* case, since it always chooses the closest neighbor, the expected number of transmissions asymptotically tends to be equal to the number of hops between the source and the destination $(N - 1)$. Clearly, the *hops* and *log* cases will yield in most cases a routing path far from optimal. Therefore, we will focus only in the c_1 case, where SPF minimizes the expected number of transmissions, and also the OR strategies with a varying number of candidates. For the scenario depicted in figure 3.4, it can be easily derived that the optimal number of hops is 3 $(N = 4)$. Since $p(100) \approx 0.70$, with 3 hops we have an expected number of transmissions of $3/0.70 \approx 4.28$, as shown in figure 3.4 for the c_1 case.

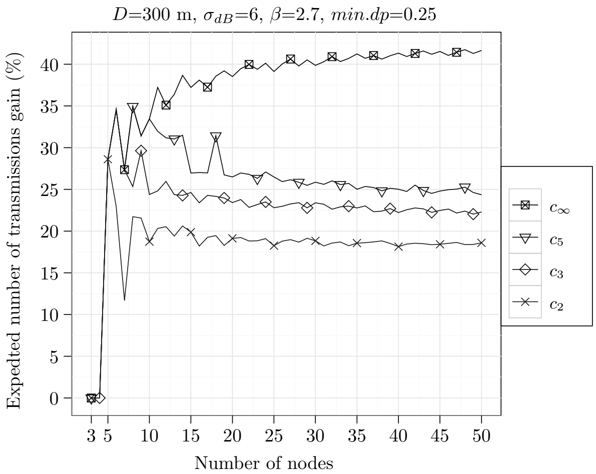


Figure 3.5: Gain for the linear network topology.

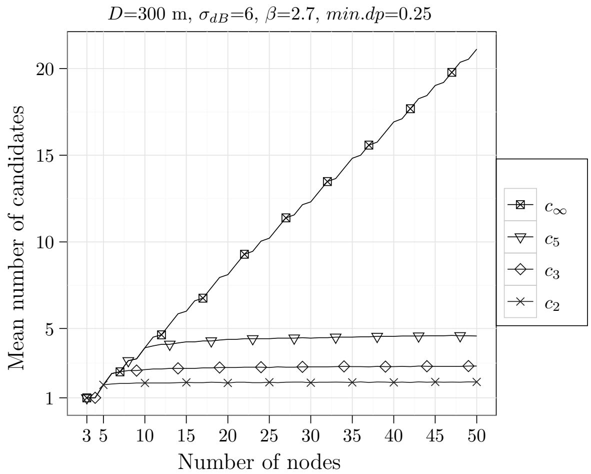


Figure 3.6: Mean number of candidates for the linear network topology.

In order to measure the improvement that can be reached using OR we define the *gain* (G_i) as the relative difference of the expected number of transmissions required with the OR with i candidates ($\mathrm{E}[T_i]$), with respect to the c_1 case ($\mathrm{E}[T_1]$) i.e.:

$$G_i = \frac{\mathrm{E}[T_1] - \mathrm{E}[T_i]}{\mathrm{E}[T_1]} \tag{3.11}$$

Figure 3.5 shows the gain obtained for the corresponding values depicted in figure 3.4. Additionally, the average number of candidates measured over all nodes except the destination, and the variance of number of transmissions, are shown in figures 3.6 and 3.7, respectively.

Figures 3.4, 3.5 and 3.6 show some interesting results: From figure 3.5 we see that the OR may achieve a significant reduction in the expected number

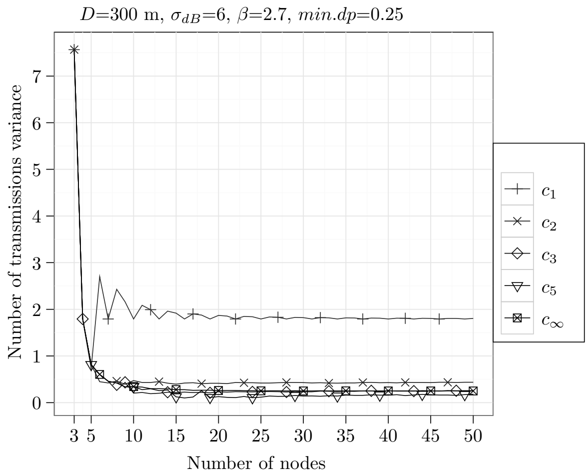


Figure 3.7: Variance of the number of transmissions for the linear network topology.

of transmissions. Furthermore, the gain increases fast with the number of candidates. For instance, the gain is close to 30% with only 5 nodes. In the c_∞ case, the gain is more than 40% with 30 nodes, but, from figure 3.6 we see that this gain is obtained at the cost of using a large number of candidates (the mean number of candidates is higher than 10). If the maximum number of candidates is fixed to only 2, the maximum gain is 28%. However, figure 3.5 shows that after the maximum point, the gain slightly decreases to an asymptotic value (approximately 18% for the c_2 case). In fact, figure 3.4 shows that the expected number of transmissions using OR slightly increases to an asymptotic value with increasing the number of nodes, for all cases except for c_∞. This counterintuitive result could be explained by the fact that the candidates selection algorithm we use is not optimum.

Figure 3.7 shows that using OR, the variance of the number of transmissions is significantly reduced. Additionally, the reduction is approximately the same regardless of the maximum number of candidates. This reduction has two important benefits. Firstly, the variability of the transmission delays may be significantly reduced using OR. Secondly, this fact indicates that the number of retransmissions of a packet by the same node may be also reduced using OR. This may also contribute on the reduction of the transmission delay variability, due to the back-off algorithm used at the MAC layer.

Finally, for a more detailed comparison we have included the probability of the number of transmissions for the scenario previously discussed, for a small number of nodes ($N = 9$) and a large one ($N = 49$), in figures 3.8 and 3.9 respectively. These values of N represent a network with a low density and high density of nodes, and will also be used in the comparison carried out in section 3.4.2.

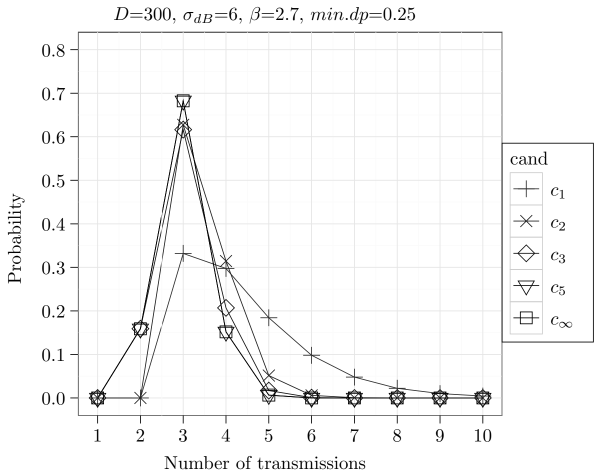


Figure 3.8: Probability of the number of transmissions for the linear network topology with $N = 9$ nodes.

Comparing the c_1 case in figures 3.8 and 3.9 we observe that the probability curves are almost the same. This comes from the fact, as explained above, that in the c_1 case, SPF chooses 2 intermediate nodes to reach the destination (3 hops), and the position of these nodes chosen as relays, is similar with $N = 9$ and $N = 49$. These figures show that in the c_1 case only 35% of packets reach the destination in 3 transmissions, while 10% of packets need 6 or more transmissions.

Regarding the OR cases, figure 3.8 shows that the probabilities are similar for all of them. This is logical since with $N = 9$ there are few choices for the selection of candidates. However, we can see that the number of transmissions needed to reach the destination is significantly reduced with respect to the c_1 case: With OR more than 60% of packets reach the destination with only 3 transmissions, and more than 10% with only 2 transmissions.

Comparing the figures 3.8 and 3.9 we can see that the probabilities are very similar for the c_2 case. This is consequence of the fact that increasing the density will not change very much the position of the candidates chosen by the c_2 case. On the other hand, the probabilities change significantly for the c_∞ case. For instance, in figure 3.9 we see that 50% of the packets reach the destination with only 2 transmissions. Looking at figure 3.6 we can see that with $N = 49$ nodes, the c_∞ case uses more than 20 candidates. With this large number of candidates it is likely that some candidate close to the destination will receive the packet, thus, allowing the delivery to the destination with only two transmissions. However, implementing an OR protocol with a high number of candidates is difficult, and possibly will introduce large signaling overhead and duplicated transmissions. Therefore, a maximum number of candidates between 2 and 5 will possibly be more

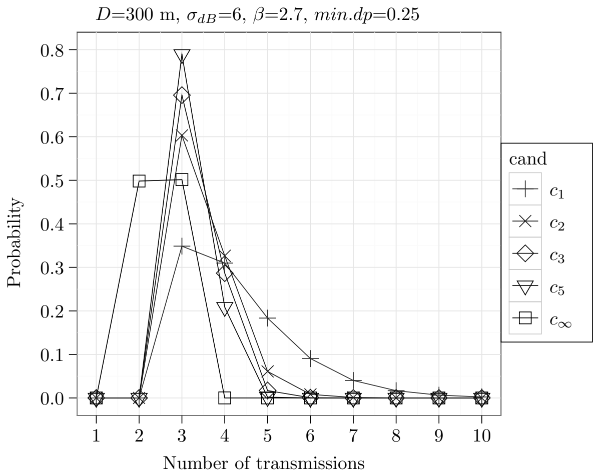


Figure 3.9: Probability of the number of transmissions for the linear network topology with $N = 49$ nodes.

convenient. Figure 3.9 shows that, even on a dense network, there are not significant differences on the number of transmissions required for such values of the maximum number of candidates.

3.4.2 Linear, Random and Grid Topologies

For the other network topologies considered in this chapter we have obtained similar conclusions as those previously described for the linear topology. We have summarized our results for the linear, random and grid topologies in tables 3.1 (a), (b) and (c), respectively. In the random topology we have taken averages after solving the model for 10 different random position of the nodes. These tables show the expected value and variance of the number of transmissions (indicated as $E[T]$ and $Var[T]$ in the tables) for two number of nodes: $N = 9$ and $N = 49$. As a measure of the density of the nodes, we have added the mean number of candidates obtained in the c_∞ case (column M_∞). Additionally, we have solved the model for two values of the loss exponent of the propagation model: $\beta = 2.7$ and $\beta = 3$. For instance, the results obtained in figures 3.4- 3.9 are summarized with the rows corresponding to $\beta = 2.7$ in table 3.1 (a).

These tables give an idea about how much the expected number of transmissions and variance can be reduced using OR under different conditions, as we explain in the following.

Comparing the three topologies, we observe that when the network density is the highest ($N = 49$, $\beta = 2.7$), the values of $E[T]$ and $Var[T]$ obtained in each topology are approximately the same for all the corresponding cases. For instance, $E[T_1] = 4.3$ and $E[T_2] = 3.5$ for all topologies (18% of gain),

Table 3.1: Expected and variance number of transmissions (E[T] and Var[T]) obtained for the Linear, Random and Grid network topologies.

(a) Linear topology

			E[T]					Var[T]				
N	β	M_∞	c_1	c_2	c_3	c_5	c_∞	c_1	c_2	c_3	c_5	c_∞
9	2.7	3.2	4.4	3.4	3.1	3.0	3.0	2.2	0.4	0.4	0.3	0.3
	3.0	1.9	7.1	5.4	5.4	5.4	5.4	5.5	0.7	0.7	0.7	0.7
49	2.7	20.5	4.3	3.5	3.3	3.2	2.5	1.8	0.4	0.3	0.2	0.3
	3.0	14.2	6.8	5.5	5.2	5.1	3.8	2.4	0.5	0.2	0.1	0.2

(b) Random topology

			E[T]					Var[T]				
N	β	M_∞	c_1	c_2	c_3	c_5	c_∞	c_1	c_2	c_3	c_5	c_∞
9	2.7	3.7	4.8	3.7	3.4	3.2	3.2	3.3	1.2	1.1	1.1	1.1
	3.0	1.9	9.2	8.0	7.8	7.8	7.8	12.6	9.5	9.4	9.4	9.4
49	2.7	21.7	4.3	3.5	3.3	3.2	2.6	2.0	0.5	0.3	0.2	0.3
	3.0	11.8	7.2	5.7	5.2	4.8	4.3	3.4	0.9	0.6	0.5	0.4

(c) Grid topology

			E[T]					Var[T]				
N	β	M_∞	c_1	c_2	c_3	c_5	c_∞	c_1	c_2	c_3	c_5	c_∞
9	2.7	2.2	5.0	3.8	3.6	3.6	3.6	7.6	1.9	1.5	1.5	1.5
	3.0	1.5	14.4	10.5	10.5	10.5	10.5	37.7	19.9	19.9	19.9	19.9
49	2.7	19.6	4.3	3.5	3.4	3.1	2.6	1.8	0.4	0.3	0.1	0.3
	3.0	9.3	7.1	5.5	5.1	4.9	4.4	1.2	0.7	0.5	0.3	0.4

Legend: N is the number of nodes, β is the path loss exponent of the propagation model, and M_∞ is the average number of candidates obtained in the c_∞ case. The other parameters are $D = 300$ m, $\sigma_{dB} = 6$, $min.dp = 0.25$

which goes down approximately to E[T_∞] = 2.6 (41% of gain) for the c_∞ case. Additionally, we can observe that E[T] only decreases from 3.5 to 3.2 when we move from a maximum number of candidates of 2 to 5 (c_2 to c_5). In practice, implementing an OR protocol with a high number of candidates is difficult. This is because the coordination between them increases the signaling overhead and also the possibility of having duplicated transmissions, which reduces the efficiency of the protocol. Therefore, these results suggest that using a small number of candidates (even 2) may be a sensible choice.

For $\beta = 3$, the delivery probabilities between the nodes is significantly reduced (see figure B.1). This has two effects: First, the SPF algorithm will choose a larger number of hops to reach the destination, thus, increasing E[T] and Var[T]. Second, the number of available links between the nodes will decrease, and thus the average number of available candidates (M_∞)

will reduce. For instance, in the linear topology with $N = 49$ nodes, when β increases from 2.7 to 3, $\mathrm{E}[T_1]$ increases from 4.3 to 6.8, and M_∞ is reduced from 20.5 to 14.2. However, in this case ($N = 49$), the density of the network is still sufficiently high such that we obtain approximately the same gains and conclusions as those for $\beta = 2.7$ described previously.

If we consider the scenarios with low density of nodes ($N = 9$), the average number of available candidates (M_∞) is very small (around 2). Therefore, using OR with a maximum number of candidates larger than 2 has a small effect. E.g. in the Linear topology with $\beta = 2.7$ we obtain $\mathrm{E}[T_2]$ to $E[T_\infty]$ ranging respectively from 3.4 to 3.

The scenarios with the lowest density ($N = 9$, $\beta = 3$) exhibit a higher sensitivity to the topology. For instance, $\mathrm{E}[T_2] = 5.4$ (linear), 8.0 (random), 10.5 (grid) when $N = 9$, $\beta = 3$, whereas $\mathrm{E}[T_2] = 3.4$, 3.7, 3.8 when $N = 9$, $\beta = 2.7$ or $\mathrm{E}[T_2] = 5.5$, 5.7, 5.5 when $N = 49$, $\beta = 3$. This is because the short transmission range and the reduced number of nodes, offer few choices for the selection of good candidates to OR, thus, resulting a performance more sensitive to their position.

3.5 Conclusions

In this chapter we have proposed a Discrete Time Markov Chain (DTMC) to analyze the performance that may be achieved using Opportunistic Routing. In our model the nodes are the states of the chain, and the state transitions model how the packet progresses through the network. This model leads to a discrete phase-type distribution for which there exist simple expressions for their distribution and moments. We have shown how our model can be used to analyze OR in a diversity of scenarios that take into account the radio propagation model, the network topology and the maximum number of candidates.

We have used our model to obtain numerical results for a shadowing propagation model and three different network topologies: Linear, random and grid. We have compared different scenarios in terms of the expected value and variance of the number of transmissions needed to send a packet from the source to the destination.

Our results show that using Opportunistic Routing the expected number of transmissions can be reduced by 20% or more in a typical scenario. Furthermore, this gain can be reached with moderate to low density of nodes. We have also observed that the variance of the number of transmissions may be reduced very much using Opportunistic Routing. This result is specially important in networks requiring QoS. Finally, our results suggest that using a small number of candidates (even 2) may be a sensible choice.

CHAPTER 4

Candidate Selection Algorithms in Opportunistic Routing

4.1 Introduction

The works in the literature usually compare Opportunistic Routing (OR) using the proposed candidate selection algorithm against a traditional unipath routing algorithm. Furthermore, most of the algorithms have been designed to select all possible candidates to reduce the expected number of transmissions. However, increasing the number of candidates increases also the coordination overhead. Therefore, as we mentioned before, in practice the maximum number of candidates that can be used is limited.

In this chapter we compare four candidate selection algorithms that have been proposed in the literature. They range from non-optimum, but simple, to optimum, but with a high computational cost. We have modified the algorithms under study to adopt them with the constraint number of candidates. In this chapter we address the questions: Is there a big difference in performance between the simple and optimal algorithms? What is the computational cost as a function of maximum number of candidates? So, under which conditions it is worth using an optimal algorithm?

4.2 Candidate Selection Algorithms

In this section, we describe the candidate selection algorithms under study. These algorithms are: Extremely Opportunistic Routing (ExOR) [12]; Opportunistic Any-Path Forwarding (OAPF) [96]; Least-Cost Opportunistic Routing (LCOR) [31]; and Minimum Transmission Selection (MTS) [58].

ExOR is one of the first and most referenced OR protocols, it is based on ETX and is simple to implement. OAPF has an intermediate complexity: It uses the EAX metric but it does not guarantee to yield the optimum set of candidates. Finally, we have chosen LCOR and MTS because, to the best of our knowledge, they are the only two algorithms in the literature that select the optimum set of candidates (i.e. the Candidates Sets (CSs) that minimize the expected number of transmissions).

Here we introduce some notations that we use throughout this chapter:

- $ncand$ is the maximum number of candidates in each node. Like in chapter 3 we shall refer as $ncand = \infty$ to the case when the maximum number of candidates is not limited. We shall also use the notation ExOR(n) to refer to ExOR with $ncand = n$, and similarly for the other algorithms under study (see the legends of figures 4.2-4.5).
- $C^{v,d}$ is the Candidates Set (CS) of node v to reach node d.
- $ETX(v, d)$ is the uni-path ETX between two nodes v and d.
- $EAX(C^{v,d}, v, d)$ is the Expected Any-path Transmission (EAX) between two nodes v and d by using $C^{v,d}$ as the CS.
- $N(v)$ is the set of all neighbors of node v.
- $|S|$ is the cardinality of the set S.

In the following subsections we describe the implementation that we have done for each of the candidate selection algorithms under study. For the sake of being precise, we shall give a pseudocode summarizing our implementations.

4.2.1 Extremely Opportunistic Routing (ExOR)

Extremely Opportunistic Routing (ExOR) [12] uses ETX as the metric for selecting the candidates. Algorithm 4.1 shows our implementation of ExOR. Node s runs this algorithm to find its Candidates Set ($C^{s,d}$) to reach the destination d. The basic idea of ExOR is running the Shortests Path First (SPF) with $weight = \frac{1}{p_{ij}}$ where p_{ij} is the delivery probability between two nodes i and j (see section 2.3). The first node after the source in this path is selected as candidate ($cand$). Then the link between s and $cand$ is removed, and the loop is repeated until no more paths to d are available, or the maximum number of candidates is reached. $ETX(cand, d)$ (or 0 if the $cand$ is the destination) is used to sort the Candidates Set (CS).

Assume that node S in figure 4.1 want to find its CS using ExOR. According to ExOR's algorithm (see algorithm 4.1), node S finds the Shortest Path First (SPF) to D which is S-A-D with ETX=3.99 (see table 4.1). Therefore, node A is selected as the candidate for node S. Then, edge S-A is removed from the topology and SPF is run again. The new shortest path from S to D

Algorithm 4.1: Candidate selection ExOR(s, d, $ncand$).

$G_{tmp} \leftarrow$ temporal copy of the network topology
$cost(s) \leftarrow ETX(s, d)$ in G_{tmp}
$C^{s,d} \leftarrow \emptyset$
while $|C^{s,d}| < ncand$ *& (s, d) connected in* G_{tmp} **do**
    $cand \leftarrow$ first node after s in the $SPF(s, d)$ in G_{tmp}
    if $cand == d$ **then**
        $C^{s,d} \leftarrow C^{s,d} \cup d$
        $cost(cand) \leftarrow 0$
    else
        $cost(cand) \leftarrow ETX(cand, d)$ in G_{tmp}
        if $cost(cand) < cost(s)$ **then**
            $C^{s,d} \leftarrow C^{s,d} \cup cand$
        end
    end
    $G_{tmp} \leftarrow$ delete $edge(s, cand)$ in G_{tmp}
end
$C^{s,d} \leftarrow C^{s,d}$ ordered by $cost$

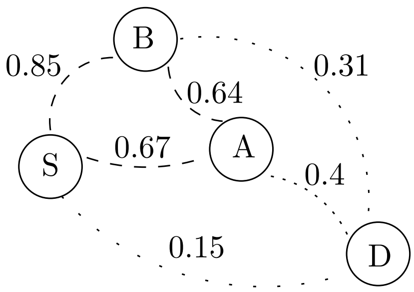


Figure 4.1: An example of Candidate selection.

is S-B-D with ETX=4.40 and B is selected as the next candidate. Finally, the ETX of each candidate to the destination d is used to sort the CS (see table 4.1). The final CS of node S is $C^{S,D} = \{A, B\}$. ExOR uses ETX to estimate the closeness to the destination but, this metric does not account the fact that packets are delivered by the candidates under opportunistic forwarding.

There is a second version of ExOR [11] proposed in 2005. To cope with the acknowledgment and the coordination issue, ExOR adopts batch transmission; 10-100 packets are collected in a batch to transfer and the next batch starts only when the current batch has completed. For each packet of the same batch, the source node selects a CS, which are prioritized by closeness to the destination. The closeness property of a node is evaluated employing the ETX metric.

Table 4.1: Expected Transmission Count of each node to D in figure 4.1

Node	ETX(Node,D)
S	3.99
A	2.5
B	3.22
D	0

Each packet has a batch map. For each packet in the batch, this map indicates the highest-priority node known to have received a copy of that packet. Then, as the packet progresses towards the destination the batch map contained in the packet is used to update the local batch maps stored in the receiving nodes. A forwarder is allowed to broadcast only received packets that its local batch map indicates have not been forwarded by any other higher priority node. Therefore, the coordination is done using timer-based coordination (see section 2.4.2). The evaluation was performed on Roofnet [1], an outdoor roof-top 802.11b network. This version of ExOR guarantees to transmit 90% of a batch using opportunistic forwarding, while the remaining packets are sent with traditional unicast routing.

4.2.2 Opportunistic Any-Path Forwarding (OAPF)

This algorithm [96] is a hop-by-hop Opportunistic Routing which is based on ETX and EAX. The pseudocode of OAPF is shown in Algorithm 4.2.

Assume that node s wants to select its Candidates Set to reach the destination d. It creates an initial Candidates Set ($\hat{C}^{s,d}$). A neighbor v of s will be included in the initial Candidates Set ($\hat{C}^{s,d}$) only if $ETX(v,d) < ETX(s,d)$. Note that, all nodes in the initial Candidates Set must select their Candidates Set (CS) before s. The actual Candidates Set (CS) of s will be a subset of the initial CS. After initiating the Candidates Set, s selects the best candidate among the nodes in the initial CS. Here, the best candidate is the one that mostly reduces the expected number of transmission from s to the destination. Node s adds the best candidate to its actual Candidates Set ($C^{s,d}$) and removes it from its initial set. Node s tries again to find the best node from its new initial Candidates Set ($\hat{C}^{s,d}$). This process is repeated until there is not any other suitable node to be included in the CS of s, or the number of candidates in the $C^{s,d}$ reaches the maximum number of candidates ($ncand$). Finally, the Candidates Set is ordered by the EAX of each selected candidate.

Now assume that node S in figure 4.1 wants to find its CS using OAPF. First, it creates its initial Candidates Set $\hat{C}^{S,D}$. Since the ETX of all its neighbors (A, B and D) to the destination D is less than $ETX(S,D)$ (see

Algorithm 4.2: Candidate selection OAPF(s, d, $ncand$).

1 $C^{s,d} \leftarrow \emptyset$; $\hat{C}^{s,d} \leftarrow \emptyset$
2 $m_p \leftarrow \infty$
3 **forall the** $v \in N(s)$ **do** `/* Initialization */`
4   **if** $ETX(v,d) < ETX(s,d)$ **then**
5         $\hat{C}^{s,d} \leftarrow \hat{C}^{s,d} \cup v$
6   **end**
7 **end**
8 **while** $|C^{s,d}| < ncand$ **do** `/* search for the best candidate */`
9     $cand \leftarrow \arg\min_{c \in \hat{C}^{s,d}} EAX(C^{s,d} \cup c, s, d)$
10     $m_c \leftarrow EAX(C^{s,d} \cup cand, s, d)$
11   **if** $m_c < m_p$ **then**
12         $C^{s,d} \leftarrow C^{s,d} \cup cand$
13         $\hat{C}^{s,d} \leftarrow \hat{C}^{s,d} \setminus cand$
14         $m_p \leftarrow m_c$
15   **else**
16         $cost(s) \leftarrow m_p$
17   *break*
18   **end**
19 **end**
20 $C^{s,d} \leftarrow C^{s,d}$ ordered by *cost*

Table 4.2: Candidates Set of A and B in figure 4.1 using OAPF

Node	Candidates Set	EAX
A	$\{D\}$	2.5
B	$\{D, A\}$	2.79

table 4.1) then, the initial CS of S is $\hat{C}^{S,D} = \{A, B, D\}$. Note that, all nodes in the initial CS must select their CSs before S. In table 4.2 we summarize the CS and related expected number of transmissions for node A and B.

Table 4.3 shows the process of selecting candidates for the source S using OAPF. In the first iteration source selects B as its candidate. Because B is the one that reduces the expected number of transmissions from S to D the most. Then, node B is removed from initial CS. The CS of S in the first iteration would be $C^{S,D} = \{B\}$. In the iteration of while-loop in algorithm 4.2 (line 8-19), source looks for the second candidate from the remaining potential candidates in $\hat{C}^{S,D} = \{A, D\}$. As we can see in table 4.3, the second candidates that reduces the expected number of transmissions from S to D the most is D. Therefore, the final CS for source using OAPF is $C^{S,D} = \{D, B\}$ with EAX equal to 3.46.

Table 4.3: OAPF Operation

Iteration	Selection
1	$EAX(\{A\}, S, D) = 3.99$, $EAX(\{B\}, S, D) = 3.97$, $EAX(\{D\}, S, D) = 6.66$
2	$EAX(\{A, B\}, S, D) = 3.64$, $EAX(\{D, B\}, S, D) = 3.46$

Algorithm 4.3: Candidate selection LCOR(s, d, $ncand$).

1 **forall the** v *in the network* $\setminus d$ **do**
2     $cost_{curr}(v) \leftarrow \infty$; $cost_{prev}(v) \leftarrow \infty$
3 **end**
4 $cost_{curr}(d) \leftarrow 0$
5 **repeat**
6   flag $\leftarrow$ TRUE
7   **forall the** v *in the network* $\setminus d$ **do** `/* search for the best Candidates Set */`
8         $C^{v,d} \leftarrow \underset{S \in 2^{|N(v)|}, |S| \leq ncand}{\arg\min} EAX(S, v, d)$
9         $cost_{curr}(v) \leftarrow EAX(C^{v,d}, v, d)$
10   **end**
11   **forall the** v *in the network* $\setminus d$ **do**
12   **if** $cost_{curr}(v) \neq cost_{prev}(v)$ **then**
13             $cost_{prev}(v) \leftarrow cost_{curr}(v)$
14   flag $\leftarrow$ FALSE
15   **end**
16   **end**
17 **until** *flag* $==$ *TRUE*
18 $C^{s,d} \leftarrow C^{s,d}$ ordered by $cost_{curr}$

4.2.3 Least-Cost Opportunistic Routing (LCOR)

The goal of this algorithm is to find the optimal Candidates Sets (CSs). Least-Cost Opportunistic Routing (LCOR) [31] uses EAX as the metric to select candidates as shown in Algorithm 4.3.

The algorithm starts by initializing the cost (EAX) of each node v to reach the destination d (lines 1-3). Since in the initializing phase the Candidates Sets (CSs) for all nodes are empty, the cost to reach the destination for all nodes is equal to ∞ ($cost_{curr}(v) \leftarrow \infty$). Note that, the cost for the destination d is always equal to 0 ($cost_{curr}(d) \leftarrow 0$).

To find the optimal Candidates Sets (CSs) in each iteration, and for every node v except the destination, the algorithm runs an exhaustive search over all possible subsets of $N(v)$ with cardinality no exceeding $ncand$ (line 8). The algorithm terminates when the cost to reach the destination does not change for all nodes in two consecutive iterations (lines 11-16).

In each iteration the algorithm checks for all the nodes but the destination, all subsets of their neighbors with length equal to $1, 2, \cdots, ncand$. The number of such subsets for a node v is $\sum_{i=1}^{ncand} \binom{|N(v)|}{i}$. Therefore, for dense networks the computational cost of the algorithm increases extremely fast due to the combinatorial explosion of the exhaustive search of line 8. In section 4.3.4, we will carry out an experimental of the computational time of the algorithms under study.

Applying LCOR on the topology in figure 4.1 yields as a result $C^{S,D} = \{D, A\}$ with the expected number of transmissions equal to 3.36.

4.2.4 Minimum Transmission Selection (MTS)

Minimum Transmission Selection (MTS) [58] selects the optimal Candidates Set (CS) for any node to a given destination d. It uses EAX as a metric for selecting the Candidates Sets (CSs). The general idea of MTS consists of moving from the nodes closest to the destination d (in terms of the EAX) backwards to the source, and using the following principle: If u and v are neighbors and $EAX(C^{u,d}, u, d) < EAX(C^{v,d}, v, d)$, then adding u and its candidates to the CS of node v will reduce the expected number of transmissions from v to d, i.e. $EAX(C^{v,d} \cup u \cup C^{u,d}, v, d) < EAX(C^{v,d}, v, d)$.

Given a general wireless topology, for a given destination d initially let $\mathbb{S}$ be the set of all nodes except d. The MTS algorithm for computing the optimal CS from any source node $v \in \mathbb{S}$ to d is described in pseudo-code in algorithm 4.4. The algorithm starts by initializing the cost (EAX) of each node v to reach the destination d (lines 2–8 in Algorithm 4.4). If d is one of the neighbors of v, then v adds the destination to its CS and the cost to reach the destination ($cost(v)$) is set to $\frac{1}{q_{vd}}$, where q_{vd} is the delivery probability of link between the two nodes v and d (note that $EAX(C^{v,d}, v, d) = \frac{1}{q_{vd}}$ when $C^{v,d} = \{d\}$).

At each subsequent iteration while $\mathbb{S}$ is not empty the algorithm looks for the node *minnode* with the minimum cost in terms of the expected number of transmissions to the destination (line 10). The neighbors of *minnode*, $N(minnode)$, add *minnode* and its candidates to their CS. Then, *minnode* is removed from $\mathbb{S}$. This process is done by means of the function *merge*, which combines both CSs and order them in increasing order of their cost (EAX). Note that proceeding this way, MTS finishes in $N - 1$ iterations, where N is the number of nodes in the network.

In the description of MTS given above, the optimal CSs for all the nodes in the network are computed assuming there is not any limitation in the

Algorithm 4.4: Candidate.selection.MTS($\mathbb{S}$, d,$ncand$).

Data: $\mathbb{S}$ is the set of all nodes except d.

1 $cost(d) \leftarrow 0$
2 **forall the** $v \in \mathbb{S}$ **do**
3   **if** $v \in N(d)$ **then**
4     $cost(v) \leftarrow \frac{1}{q_{vd}}$; $C^{v,d} \leftarrow d$
5   **else**
6     $C^{v,d} \leftarrow \emptyset$; $cost(v) \leftarrow \infty$
7   **end**
8 **end**
9 **while** $\mathbb{S}$ *is not empty* **do**
10   $minnode \leftarrow \arg\min_{v \in \mathbb{S}} cost(v)$
11   $\mathbb{S} \leftarrow \mathbb{S} \setminus \{minnode\}$
12   **forall the** $v \in N(minnode)$ **do**
13     $C^{v,d} \leftarrow merge(C^{v,d}, minnode, C^{minnode,d})$
14     $cost(v) \leftarrow EAX(C^{v,d}, v, d)$
15   **end**
16 **end**
17 $\mathbb{S} \leftarrow$ all nodes in the network $\setminus\{d\}$ ordered by $cost$
18 **forall the** $v \in \mathbb{S}$ **do**
19   $C^{v,d} \leftarrow \underset{T \in C^{v,d}, |T| \leq ncand}{\arg\min} EAX(T, v, d)$
20 **end**

number of candidates, as proposed in the original version of this algorithm.

In order to limit the maximum number of candidates, maintaining the optimality of the algorithm, we have added the lines 18–20. Here the nodes are visited in increasing order of their cost, and an exhaustive search is done over all subsets of the CSs with cardinality $\leq ncand$. Since we first find the optimal CSs in the case of infinite number of candidates, and then we look for the best subset of Candidates Sets (CSs) with at most $ncand$ elements, the final Candidates Sets (CSs) will be the optimal Candidates Sets (CSs).

Like the previous algorithms, we apply MTS in the example of figure 4.1 to find the CSs. According to MTS algorithm $\mathbb{S} = \{S, A, B\}$. The cost of nodes S, A and B to the destination D is 1/0.15, 1/0.4 and 1/0.31, respectively. The result of each iteration of the MTS algorithm is shown in table 4.4, where the first item in each cell is the current best CS for the corresponding node, and the second item is the current smallest expected number transmissions using that set. In the first iteration since the EAX of A is the minimum ($EAX(\{D\}, A, D) = 2.5$), it is removed from $\mathbb{S}$. Then, MTS adds A and its candidates ($C^{A,D} = \{D\}$) to the CSs of all neighbors of A, i.e, nodes S and B (see iteration 1 in table 4.4). Note that, in each iteration the EAX of each node is updated according to the new CS. In the second iteration

Table 4.4: Candidates Set selection for the figure 4.1 using MTS

Iteration	S	A	B
1	$\{D, A\}$, 3.36	-	$\{D, A\}$, 2.79
2	$\{D, A, B\}$, 3.22	-	-

Table 4.5: Candidates Set and EAX of each node in figure 4.1 using different algorithms.

Node	ExOR	OAPF	LCOR or MTS
S	{A, B}, 3.64	{D, B}, 3.46	{D, A}, 3.36
A	{D}, 2.50	{D}, 2.50	{D}, 2.50
B	{D, A}, 2.79	{D, A}, 2.79	{D, A}, 2.79

the node with the minimum EAX is B ($EAX(\{D, A\}, B, D) = 2.79$); it is removed from $\mathbb{S}$ and its candidates $C^{B,D} = \{D, A\}$, and B are added to the CSs of all neighbors of B which are still in $\mathbb{S}$, i.e, node S. Now, each node has a set of candidates to reach D. Note that, until this step there is not any limitation on the number of candidates. Doing an excursive search with constraint $ncand = 2$ over the sets which are found by the original version of MTS results in the optimum CS with length at most equal to 2.

We summarize the CSs and EAX of each node in figure 4.1 using different algorithms under study in table 4.5. The first item in each cell is the CS for the corresponding node, and the second item is the Expected Any-path Transmission (EAX) of the corresponding node using the said set. As we can see in table 4.5 the algorithms that use EAX to select CSs have better expected number of transmissions than ExOR which uses ETX for selection of candidates.

4.3 Numerical Results

In this section we use the proposed model in chapter 3 to study the performance of the candidate selection algorithms described in this chapter.

We have proceed as follows:

- First the network topology is set up randomly placing the nodes in a square field with diagonal $D = 300$ m, except the source and the destination which are placed at the end points of one of the diagonals. We consider scenarios with different number of nodes ($10 \leq N \leq 50$).
- To assess the delivery probabilities of the links we have used the shadowing propagation model. In our simulation we have used $\beta = 2.7$ and $\sigma_{dB} = 6$ dBs. We have set the model parameters to the default values

used by the network simulator (Ns-2) [2] (see table B.2). We have assumed that a link between any two nodes exists only if the delivery probability between them is greater (or equal) than $min.dp = 0.1$.

- We assume that the topology and delivery probabilities are known by all nodes, and for each of them it is used one of the algorithms described in sections 4.2.1– 4.2.4 to compute the Candidates Set (CS) for the given destination.

- Finally, we use the DTMC model described in chapter 3 to compute the following performance measures: expected number of transmissions, variance of the expected number transmissions and probability of the number of transmissions.

In order to compare the algorithms under study, and since we want to focus on the effect of candidate selection, we have considered the following scenario: (i) in the network there is only one active connection; (ii) perfect coordination between the candidates, i.e. the most priority candidate successfully receiving the packet will be the next forwarder; (iii) the nodes retransmit the packets until successful delivery.

We have done this evaluation using the R numerical tool [67]. Each point in the plots is an average over 100 runs with different random node positions. We have used this methodology for each of the algorithms described in sections 4.2.1– 4.2.4, and for a different maximum number of candidates: $ncand = 2, 3, 4, 5, \infty$. Recall that we refer as $ncand = \infty$ to the case when there is no limit on the maximum number of candidates and all possible nodes can be selected as the candidates.

As an estimation of the computational cost of the algorithms, we have measured the execution time it takes to compute the CSs in each scenario. These times have been obtained running the algorithms on a computer with an Intel Xeon Dual-Core-2 2.33 GHz, FSB 1333 MHz, with 4 MB cache and 12 GB of memory.

4.3.1 Expected number of transmission

First, we examine in details the case with at most 3 candidates for each node ($ncand = 3$), as shown in figure 4.2. For the sake of comparison, we have included the scenarios using uni-path routing and also the optimal candidate selection algorithm in the case $ncand = \infty$ (we shall refer to it as Opt(∞)). Note that, uni-path routing is equivalent to use $ncand = 1$ in any of the OR algorithms under study. The curves have been obtained varying the number of nodes, but maintaining the distance $D = 300$ m between the source and the destination, thus, increasing the density of the network.

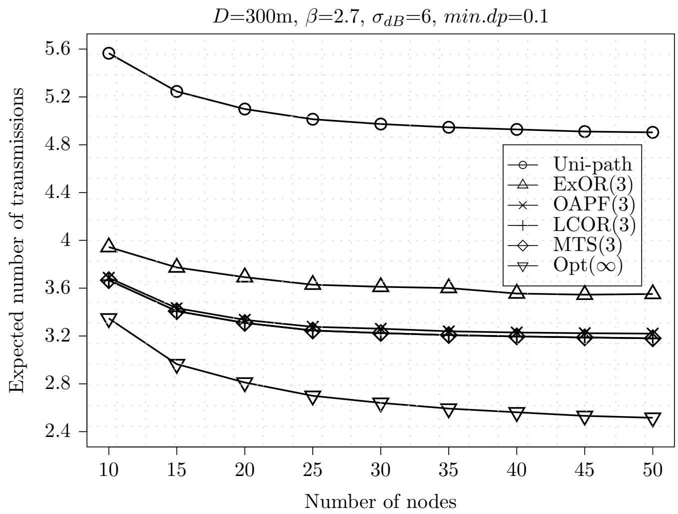


Figure 4.2: Expected number of transmissions in the case $ncand = 3$.

As a first observation in figure 4.2, we can see that using any OR algorithm outperforms the traditional uni-path routing. Regarding the optimal algorithms, LCOR and MTS, we have validated that they choose exactly the same CSs, and thus, the curves are the same. Additionally, for $ncand = \infty$ the expected number of transmissions for LCOR and MTS are the same, so we show only one of the curves obtained with LCOR(∞) and MTS(∞) (indicated as Opt(∞)).

We can see that the expected number of transmissions obtained with OAPF is only slightly larger than those obtained with the optimal algorithms. Finally, we observe that the expected number of transmissions required by ExOR is significantly larger than any other OR algorithms. The reasons that motivate this inferior performance of ExOR are the following: recall that ExOR is a simple algorithm that uses ETX as the metric for selecting candidates. It looks for the candidates running SPF after removing the links to the nodes that have already been selected as candidates. By doing this, the candidates tend to be chosen close to each other. In chapter 6 we have investigated the optimal position of the candidates and we have shown that they are not clustered, but distributed over distances that approximate to the destination. Therefore, we conclude that ExOR does a coarse selection of the CS. On the other hand, recall that OAPF incrementally adds the nodes to the CS that are most effective at reducing the Expected Any-path Transmission (EAX). Although this does not guarantee choosing the optimal CSs, we can see from the figure 4.2 that the results are very close to the optimal algorithm.

Regarding the scenario with $ncand = \infty$, figure 4.2 shows that it achieves a noticeable reduction of the expected number of transmissions compared to

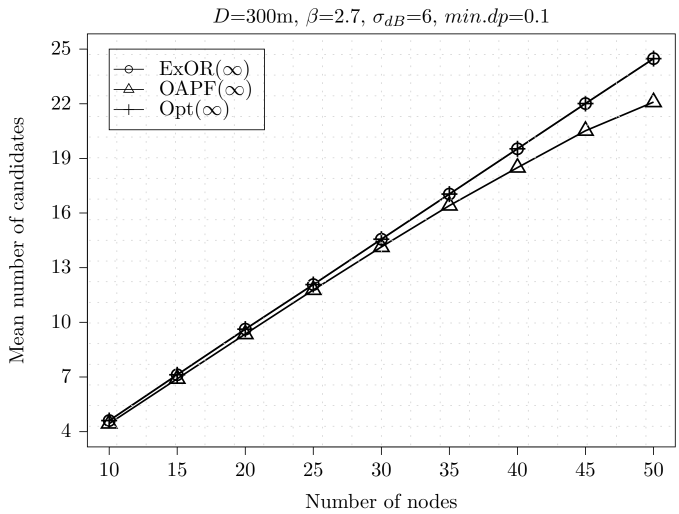


Figure 4.3: Mean number of candidates in the case $ncand = \infty$.

the scenario with $ncand = 3$. However, as shown in figure 4.3, this is at cost of using a large number of candidates. As we mentioned in chapter 3, implementing an OR protocol with a high number of candidates is difficult and possibly will introduce large signaling overhead and duplicate transmissions. Therefore, the differences obtained with $ncand = 3$ and $ncand = \infty$ in a real scenario, are likely to be much smaller than those shown in figure 4.2.

For other scenarios we have obtained similar results. For instance, figures 4.4 and 4.5 have been obtained, respectively, maintaining the total number of nodes equal to $N = 10$ and $N = 50$ (thus, representing a low and high density network), and varying the maximum number of candidates to: $ncand = 1, 2, \cdots, 5$ and ∞. Note that $ncand = 1$ is equivalent to uni-path routing, thus, the expected number of transmissions obtained for $ncand = 1$ is the same for all algorithms.

In the case of $ncand = \infty$ all algorithms have almost the same expected number of transmissions. This comes from the fact that in this case there is not any limitation on the maximum number of candidates. Therefore, all nodes which are closer to the destination than the source can be selected as candidates, and all of the algorithms have almost the same CSs. Note that since ExOR uses ETX as the metric to select candidates, the order of candidates may be different compared with the CSs in the other algorithms. Because of that the expected number of transmissions in the case of ExOR with $ncand = \infty$ has a very small difference compared with the other algorithms (not noticeable in the graphs).

By comparing figures 4.4 and 4.5 we can see that the difference between ExOR and the other algorithms is higher in a dense network ($N = 50$).

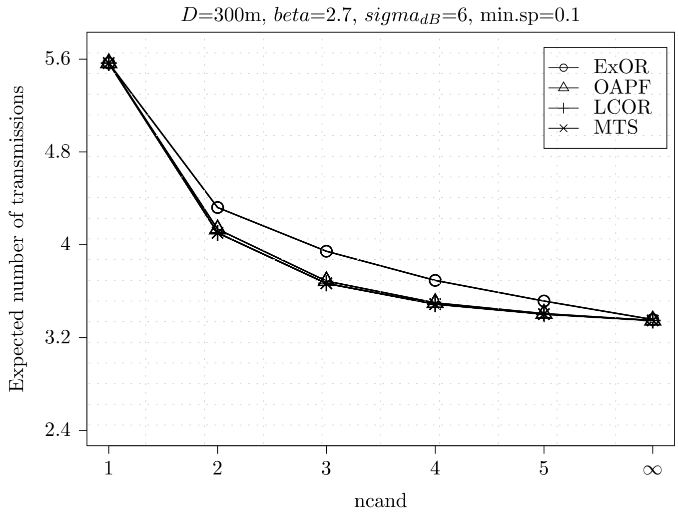


Figure 4.4: Expected number of transmissions for the random topology with $N = 10$ nodes varying the maximum number of candidates.

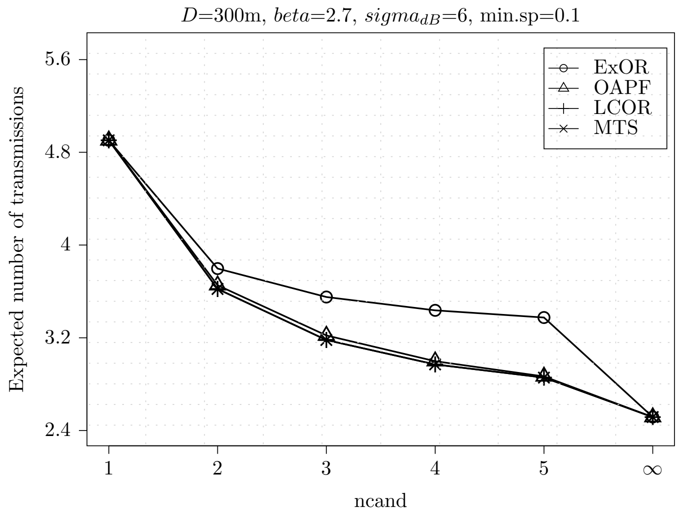


Figure 4.5: Expected number of transmissions for the random topology with $N = 50$ nodes varying maximum the number of candidates.

This comes from the fact that in a dense network there is a larger number of possible choices of the CSs. Thus, limiting the maximum number of candidates makes the selection of the candidates sets more critical. However, we can see that the difference between OAPF and the optimal algorithms is kept small even in a dense network. We can see that increasing the maximum number of candidates (*ncand*) from 1 to 2 results in an important gain in all cases and increasing *ncand* from 5 to ∞ is more important in the dense topology.

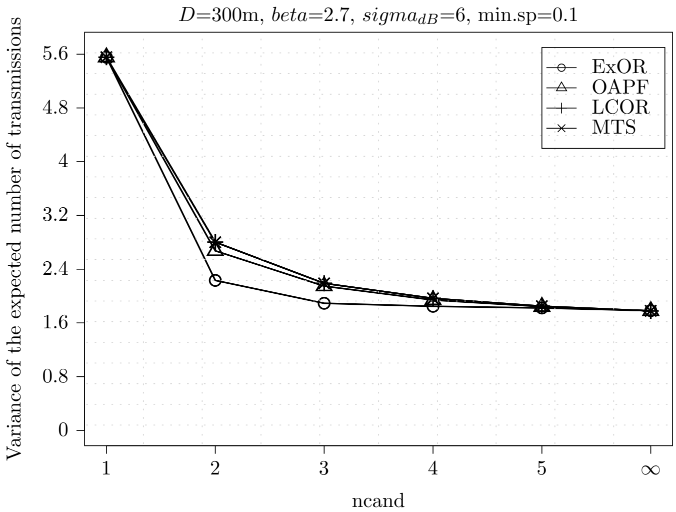


Figure 4.6: Variance of the expected number of transmissions for the random topology with $N = 10$ nodes varying the maximum number of candidates.

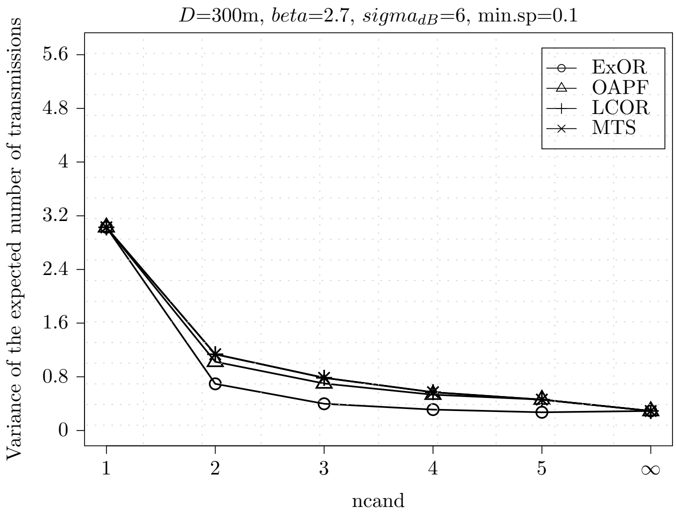


Figure 4.7: Variance of the expected number of transmissions for the random topology with $N = 50$ nodes varying the maximum number of candidates.

4.3.2 Variance of Expected Number of Transmissions

One of the metrics which can also be calculated with the markov model proposed in chapter 3 is the variance of expected number of transmissions from source to the destination. Figures 4.6 and 4.7 show the variance of expected number of transmissions for a low ($N = 10$) and high ($N = 50$) density network, respectively. Since with one candidate all algorithms have the same result as uni-path routing the variances of the expected number of transmissions for $ncand = 1$ are the same.

Figures 4.6 and 4.7 show that using OR the variance of the expected number of transmissions is significantly reduced compared with uni-path routing. It is also observed that while the variance decreases with the value of *ncand*, just a small number of candidates (typically 2 or 3 candidates per node) is enough to attain a significant part of the potential reduction. This effect is even more noticeable when the candidate selection algorithm employed is ExOR. Furthermore, while ExOR is the algorithm that yields the highest mean number of transmissions, as it was shown above, it achieves the lowest variance.

The reduction of variance of the expected number of transmissions, compared with uni-path routing, has two important benefits. Firstly, the variability of the transmission delays may be significantly reduced using OR. Secondly, this fact indicates that the number of retransmissions of a packet by the same node may be also reduced using OR. This may also contribute on the reduction of the transmission delay variability, due to the back-off algorithm used at the MAC layer.

4.3.3 Probability Distribution of the Number of Transmissions

For having a more detailed comparison, we have included the probability distribution of the number of transmissions for $ncand = 1, 3$ and ∞ for a small number of nodes $N = 10$ and a large one $N = 50$, in figures 4.8 and 4.9, respectively.

The probability curves for the $ncand = 1$ case (uni-path routing) in both figures 4.8 and 4.9 are almost the same. These figures show that for $N = 10$ in the uni-path routing, about 14% of packets reach the destination with 3 transmissions, while about 40% of packets need 6 or more transmissions. In figures 4.8 and 4.9 we can see that, by using OR algorithms, the number of transmissions needed to reach the destination is significantly reduced with respect to the uni-path routing approach. The curves for all algorithms except ExOR are almost the same. In a low density network ($N = 10$), using the optimal candidate selection algorithms (LCOR or MTS) in the $ncand = 3$ case, 18% and 37% of packets reach the destination with 2 and 3 transmissions, respectively, while using ExOR only about 5% of packets reach the destination with 2 transmissions. In the network with more nodes ($N = 50$), LCOR, MTS and even OAPF can select the candidates which are close to the destination. Therefore, as we can see in figure 4.9 by using these algorithms with $ncand = 3$ about 20% and 50% of packets reach the destination with 2 and 3 transmissions, respectively.

By comparing the figures 4.8 and 4.9 we can see that the probabilities change significantly for the $ncand = \infty$ case. For instance, in figure 4.9 about 50%

of packets reach the destination only with 2 transmissions, while in the low dense network ($N = 10$) only 25% of packets reach the destination with 2 transmissions. Looking at figure 4.3 we can see that, the $ncand = \infty$ case uses 25 candidates in a dense network ($N = 50$). With such a large number of candidates it is likely that some candidate close to the destination will receive the packet, thus, allowing the delivery to the destination with only two transmissions.

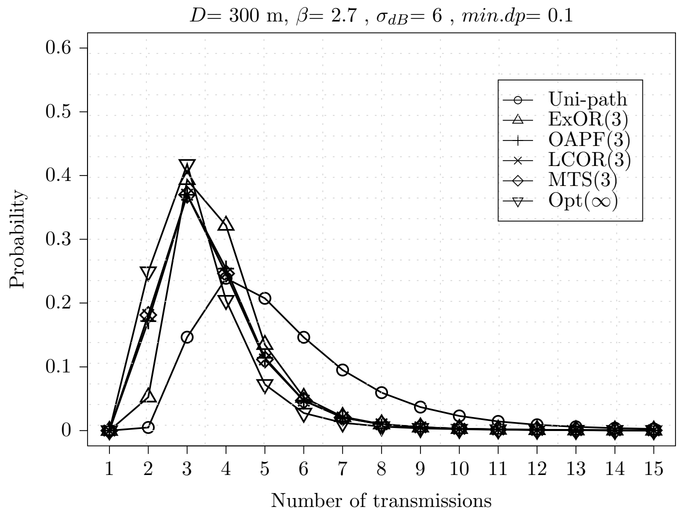


Figure 4.8: Probability of the number of transmissions for the random topology with $N = 10$ nodes

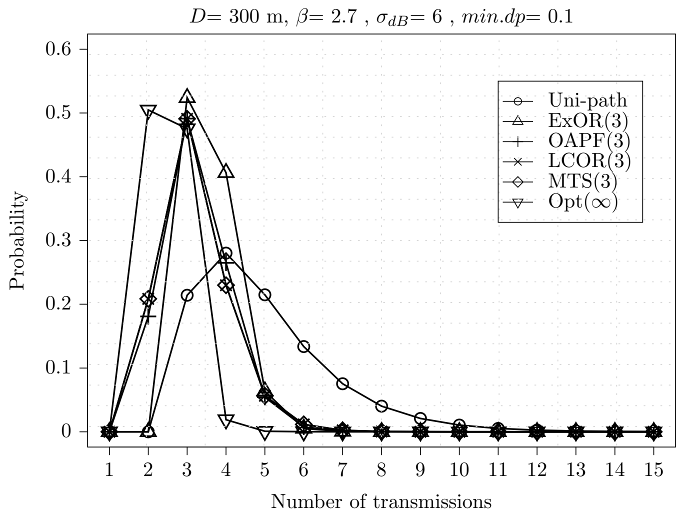


Figure 4.9: Probability of the number of transmissions for the random topology with $N = 50$ nodes

4.3.4 Execution Time

In this section we estimate the computational cost of the algorithms under study by measuring the *execution time* it takes to compute the CSs towards the destination. Recall that these CSs are: the candidates of the source s towards the destination d, the candidates of these candidates towards d, and so on until d (whose candidates is the empty set). Notice that for ExOR this requires calling Algorithm 4.1 for the source s, for its candidates, the candidates of these candidates, and so on until d. For the other algorithms, computing the CSs of the source requires the computation of all the necessary CSs. This comes from the fact that the other algorithms are based on the EAX metric, which requires the CSs. Therefore, for the algorithms OAPF, LCOR and MTS, the execution time is the time it takes calling only once the algorithms 4.2, 4.3 and 4.4, respectively.

Figure 4.10 shows the expected number of transmissions versus the execution time in logarithmic scale. We have selected $ncand = 3$ as a sample case for our study. So, the points in figure 4.10 have been obtained by averaging over the 100 runs of the corresponding points in figure 4.2. The values next to the points represent the number of nodes in the network (N).

We can see that for all the algorithms, the larger is the number of nodes the lower is the expected number of transmissions and the higher is the execution time. As expected, the fastest algorithm is ExOR whereas LCOR is the slowest. For instance, when the number of nodes in the network is 50, LCOR needs about 3.3 hours to finish. Obviously, with a maximum number of candidates larger than 3 the execution time will be much longer. OAPF lies between the exhaustive search of the optimal algorithms and the simplicity of ExOR, and thus, has an execution time that falls in between these algorithms, e.g. 0.6 to 47 seconds for the low and high density networks, respectively.

MTS and LCOR have the same expected number of transmissions while the execution time of MTS is much lower than LCOR. For instance in the high density network ($N = 50$), MTS needs about 40 minutes to finish while LCOR needs about 3.3 hours. Recall that MTS(3) first looks for the optimal CSs without limiting the maximum number of candidates, and then the CSs are pruned to at most 3 elements. Therefore, the searching space for finding the optimal sets in MTS(3) is less than LCOR(3), which examines all the subsets of the neighbors of the nodes.

By comparing the two optimal algorithms that have been proposed in the literature, we can conclude that MTS outperforms LCOR in terms of the execution time. Additionally, it is possible to obtain candidate selection algorithms, as OAPF, that have a performance close to the optimal algorithms

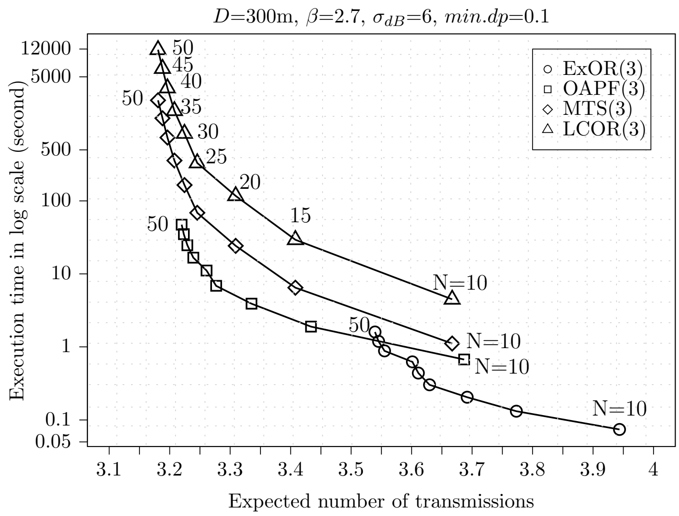


Figure 4.10: Expected number of transmissions and execution time of all algorithms.

with a much lower execution time. With simple algorithms as ExOR, the performance may be significantly less than the optimal.

4.4 Conclusions

In this chapter we have compared four relevant algorithms that have been proposed in the literature for the candidate selection in Opportunistic Routing: Extremely Opportunistic Routing (ExOR); Opportunistic Any-Path Forwarding (OAPF); Least-Cost Opportunistic Routing (LCOR); and Minimum Transmission Selection (MTS). They range from the simplicity of ExOR, the intermediate computational complexity of OAPF, to the optimal but high computational cost of LCOR and MTS. We have modified the algorithms such that the maximum number of candidates can be limited.

We have used the model proposed in chapter 3 to obtain numerical results for a shadowing propagation model and different candidate selection algorithms. We have compared different scenarios in terms of the expected number of transmissions needed to send a packet from source to the destination, variance of expected number of transmissions and the probability of the number of transmissions needed to reach the destination. We also computed the execution time to construct the Candidates Sets (CSs).

Our results show that using any OR algorithm outperforms the traditional uni-path routing. Furthermore, if the maximum number of candidates is not limited, all of the algorithms have almost the same expected number of transmissions. Such assumption is not realistic since the algorithms may choose a large number of candidates, which will introduce large signaling overhead and duplicate transmissions. When the maximum number of candidates is limited, our results show that the expected number of transmissions required by ExOR is larger than the other OR algorithms. This is because of the coarse selection of the CSs of ExOR. However, the performance obtained with OAPF has always been very close to the optimal algorithms. We have also observed that the variance of the number of transmissions may be reduced very much using Opportunistic Routing. This result is specially important in networks requiring QoS.

Regarding the execution times, that fact that ExOR is based on ETX makes this algorithm much faster than the others. For the optimum algorithms, we have obtained that MTS outperforms LCOR. However, both algorithms require extremely large times to compute the CSs in a dense network (on the order of hours in a modern PC). On the other hand, OAPF is able to run the candidate selection with execution time orders of magnitude lower than the optimum algorithms (on the order of minutes). Therefore, we conclude that a fast and simple Opportunistic Routing candidate selection algorithm (like as OAPF) may be preferable in dynamic networks, where the Candidates Sets are likely to be updated frequently.

CHAPTER 5

Distance Progress Based Opportunistic Routing

5.1 Introduction

In candidate selection algorithms, selecting nodes which are closer to the destination intuitively improves the performance of OR and reduces the expected number of transmissions. However, this might only be true if there is no constraint on the number of candidates. Since increasing the number of candidates increases also the coordination overhead, in practice the maximum number of candidates that can be used is limited. This fact has often been neglected in candidate selection algorithms proposed in the literature. That is, the algorithms have been designed to select all possible candidates to reduce the expected number of transmissions, while assuming perfect coordination and no signaling overhead.

When there is no constraint over the number of candidates, the delivery ratio between the node and its candidates is not important. That is, all candidates that can help reaching the destination are chosen, regardless of the probability of reaching them. However, if the maximum number of candidates is limited, considering the links delivery probability may be essential to choose an appropriate set of candidates. For instance, choosing only candidates close to the destination, but having low links delivery probability, would be a bad selection.

An example is given in figure 5.1. Suppose s and d as the source and destination, respectively. Each edge is labeled with the associated delivery probability. Furthermore, assume that the nodes are allowed to select only 2 candidates. If node s selects nodes c_1 and c_2 which are the closest nodes to the destination, then the expected number of transmissions from s to d using these candidates is about 6.37. However, if s selects $\{c_1, c_3\}$ or $\{c_2, c_3\}$ as the candidates, the expected number of transmission would be 2.75. To address the above issues, in this chapter, we define a new metric that we

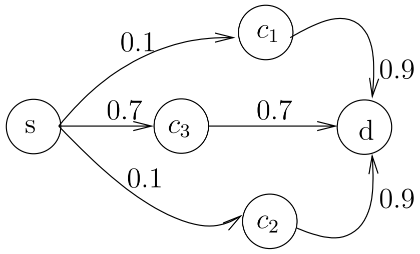


Figure 5.1: Example of selecting the closest nodes to the destination.

call *EDP*. EDP measures the expected distance progress of sending a packet using a set of candidates. We also propose a hop-by-hop candidate selection and prioritization algorithm based on EDP. We shall refer to our proposal as *Distance Progress based Opportunistic Routing (DPOR)*. In DPOR, each node selects its Candidates Set independently without considering the other nodes' Candidates Set (CS). In fact, the selection of candidates in each node i is based on the knowledge of its neighbors' geographic position and the link delivery probability between i and its neighbors.

5.2 Distance Progress based Opportunistic Routing

In this section, we define a new metric to estimate the Expected Distance Progress of sending a packet using a set of candidates. Then, based on this metric we propose a candidate selection and prioritization algorithm to maximize the Expected Distance Progress of a sending data packet.

Assume that there are $\mathbb{N}$ nodes in the network with a source s and destination d. We assumed that: (i) all nodes $v \in \mathbb{N}$ know the position coordinates of their neighbors, $N(v)$, (ii) each node v knows the link delivery probability between v and its neighbors ($p_{v,i}, i \in N(v)$), and (iii) all nodes know the position of the destination. This assumptions could be easily implemented, e.g. by using a location registration and look up service which maps node addresses to locations as in [85, 42].

5.2.1 Expected Distance Progress

Let $D_{i,d}$ be the geographic distance between node i and destination d. The *Distance Progress* of a data packet sent by source s towards destination d using next-hop c_i is given by: $DP_{c_i}^{s,d} = D_{s,d} - D_{c_i,d}$. We define the *Expected Distance Progress* (EDP) from node s to the destination d using candidates set $C^{s,d} = \{c_1, c_2, ..., c_{ncand}\}$ (with c_1 being the highest priority, and c_{ncand} the least one) as:

$$EDP(s, d, C^{s,d}) = \sum_{i=1}^{ncand} (D_{s,d} - D_{c_i,d}) \times p_{s,c_i} \prod_{j=1}^{i-1} (1 - p_{s,c_j})$$
$$= \sum_{i=1}^{ncand} DP_{c_i}^{s,d} \times p_{s,c_i} \prod_{j=1}^{i-1} (1 - p_{s,c_j}) \quad (5.1)$$

Where $p_{i,j}$ is the delivery probability of link between node i and j. Note that upon a packet transmission, the higher is EDP, the higher is the approach of the packet to the destination.

Intuitively, increasing the number of candidates would result in a larger EDP. Additionally, the maximum EDP for a given candidates set of $C^{s,d}$ can only be achieved by assigning the priority to each node based on their distances to the destination. That is, the furthest node receiving the packet should try to forward it first; if it did not receive the packet, the second furthest node should try, and so on.

5.2.2 DPOR

In this section we propose a candidate selection algorithm that we call Distance Progress based Opportunistic Routing (DPOR), which tries to maximize the EDP. Algorithm 6 shows the pseudo-code of DPOR. In fact, our algorithm does not only consider the closeness of candidates to the destination, but it also takes into account the links delivery probability between the forwarder and the candidates. Basically, DPOR tries to find the candidates which are close to the destination, but having a link delivery probability between the forwarder and the candidates not too small.

Algorithm 6 works as follows: Assume that a generic node s wants to choose its Candidates Set (CS) for a specific destination d. First, node s finds its neighbors which are closer to the destination than itself. We shall refer this set as $N(s)$. A neighbor j of s is included in $N(s)$ only if $D_{j,d} < D_{s,d}$. Then, node s selects the best candidate among its neighbors that increases the most EDP from s to the destination (lines 5). It adds the best candidate to its Candidates Set $C^{s,d}$ and removes it from its neighbors set (lines 8 and 9). Then it tries again to find the best node from its new neighbors set. This process is repeated until there is not any other suitable node to be included in the CS of s, or the number of candidates in $C^{s,d}$ reaches the maximum number of candidates ($ncand$). Note that, in each iteration $EDP(s, d, C^{s,d})$ is calculated ordering the candidates $C^{s,d} = \{c_1, c_2, ..., c_{ncand}\}$ by their distance to the destination, i.e, $D_{c_1,d} < D_{c_2,d} < ... < D_{c_{ncand},d}$. We remark that in DPOR each node i selects its CS independently from other nodes' Candidates Set, and only by knowing the position of its neighbors and the

Algorithm 5.1: Candidate.selection.DPOR(s, d, $ncand$).

$m_p \leftarrow -1$
$N(s) = \{n = neighbor(s) | D_{n,d} < D_{s,d}\}$
Sort $N(s)$ according to $D_{n,d}, n \in N(s)$
while $|C^{s,d}| < ncand$ **do**
    $cand \leftarrow \arg\max_{c \in N(s)} EDP(s, d, C^{s,d} \cup c)$
    $m_c \leftarrow EDP(C^{s,d} \cup cand, s, d)$
    if $m_c < m_p$ **then**
        $C^{s,d} \leftarrow C^{s,d} \cup cand$
        $N(s) \leftarrow N(s) \setminus cand$
        $m_p \leftarrow m_c$
    else
        break
    end
end

link delivery probability toward them.

5.3 Methodology

We have compared our proposal (DPOR) with the candidate selection algorithms that have been explained in Chapter 4. They range from non-optimum, but simple: Extremely Opportunistic Routing (ExOR) [12], Position based Opportunistic Routing (POR) [85] and Opportunistic Any-Path Forwarding (OAPF) [96], to optimum, but having a high computational cost: Minimum Transmission Selection (MTS) [58].

POR is a position based OR protocol. When a source wants to send data packet to the destination, it finds its CS according to the distance between its neighbors and the destination. The neighbor which is nearer to the destination will have higher priority.

In order to compare different algorithms, and since we want to focus on the effect of candidate selection, like in chapter 4 we have assumed that there is a perfect coordination between the candidates, i.e. the most priority candidate successfully receiving the packet will be the next forwarder. The nodes retransmit the packets until successful delivery. Furthermore, we have assumed that there is only one active connection in the network.

We compare the performance of each algorithm in terms of the expected number of transmissions needed to send a packet from the source to the destination and the execution time of each algorithm to create the Candidates Sets (CSs).

The numerical results have been obtained using R [67]. After setting up the network topology, the delivery ratio of the links is assessed using a shadowing propagation model. In our simulation we have used $\beta = 2.7$ and $\sigma_{dB} = 6$ dBs. We have set the model parameters to the default values used by the network simulator (Ns-2) [2] (see appendix B). We shall use these values in the performance evaluation presented in section 5.4.

5.4 Performance Evaluation

In this section we study the performance of our proposal, DPOR, compared to the other candidate selection algorithms under study. We consider scenarios with different number of nodes ($20 \le N \le 80$) randomly placed in a square field with diagonal $D = 500$ m, except the source and the destination which are placed at the diagonal end points. Each point in the plots is an average of 100 runs with different random node positions. The delivery probabilities have been assigned with the shadowing model.

In the candidate selection algorithm, we have assumed that a link between any two nodes exists only if the delivery probability between them is greater (or equal) than $min.dp = 0.1$. We have compared the algorithms for different maximum number of candidates: $ncand = 2, 3, 4, 5$. We shall use the notation ExOR(n) to refer to ExOR with $ncand = n$, and similarly for the other algorithms under study (see the legend of figures 5.2–5.6).

5.4.1 Expected Number of Transmissions

First, we examine the case with at most 2 candidates for each node ($ncand = 2$), as shown in figure 5.2. For the sake of comparison, we have included the scenario using uni-path routing (computing the routes using SPF with weights equal to the delivery probabilities). The curves have been obtained varying the number of nodes, but maintaining the distance $D = 500$ m between the source and the destination, thus, increasing the density of the network. In all figures we use the notation Opt(n) to refer to the optimum candidate selection algorithm (MTS).

Figure 5.2 shows that all candidate selection algorithms, except POR, outperforms the traditional uni-path routing. Furthermore, we can see that increasing the number of nodes causes decreasing the expected number of transmissions in all algorithms except POR. This comes from the fact that the candidate selection in POR is only based on the closeness of neighbors to the destination. In fact, POR does not consider the links delivery probability between forwarder and its candidates. Therefore, increasing the density of the network causes having more candidates closer to the destination that will be chosen by POR, and thus, decreasing its performance.

ExOR looks for the candidates running SPF after removing the links to the

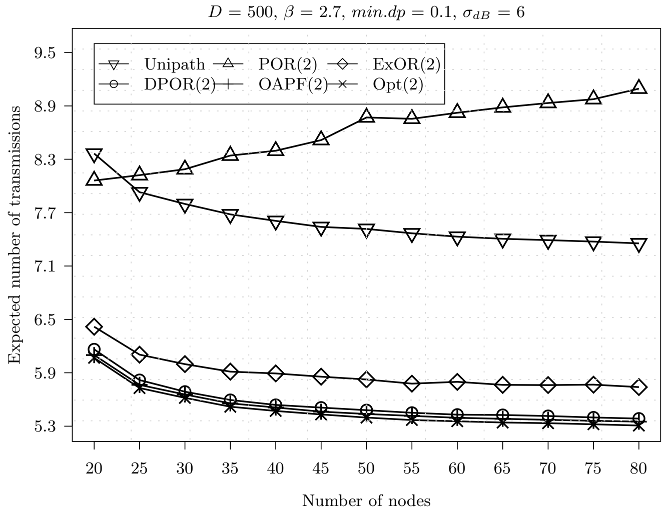


Figure 5.2: Expected number of transmissions for the case $ncand = 2$

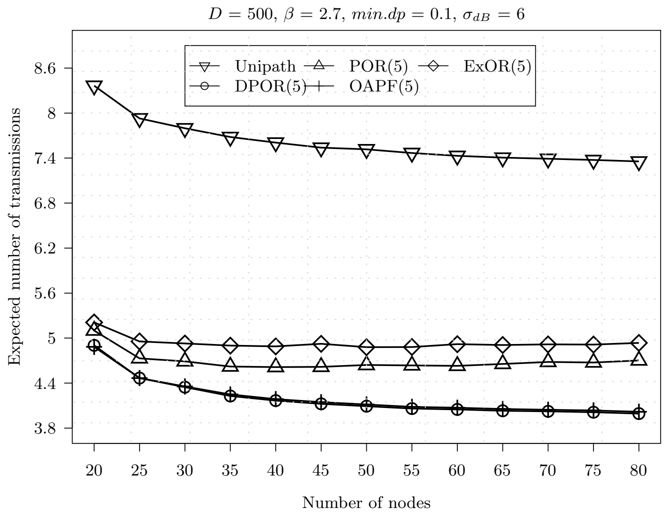


Figure 5.3: Expected number of transmissions for the case $ncand = 5$.

nodes that have already been selected as candidates. Although it is simple, this candidate selection algorithm is far from the optimum one. In fact, figure 5.2 shows that ExOR has larger expected number of transmissions than DPOR, OAPF and the Opt.

In figure 5.2 we can see that the results for DPOR are very close to the optimum algorithm and OAPF. Recall that DPOR chooses the most effective candidates at increasing the Expected Distance Progress (EDP). By doing this, DPOR does not only consider the closeness of candidates to the destination (like POR), but it also considers the links delivery probability between the forwarder and the candidates as well.

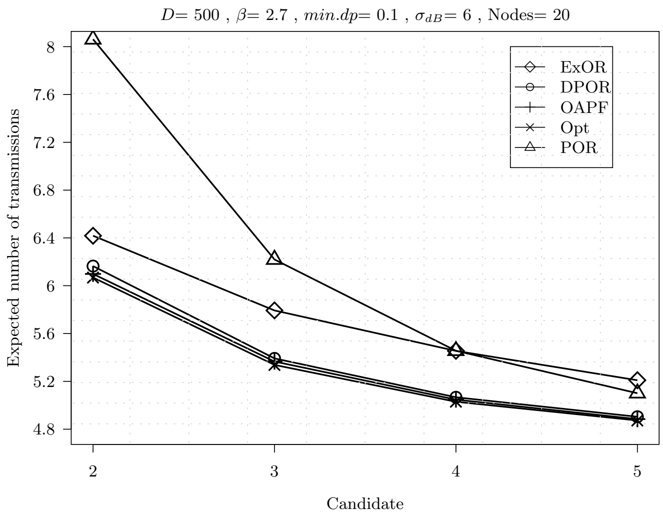


Figure 5.4: Expected number of transmissions for the case $N = 20$ varying the number of candidates.

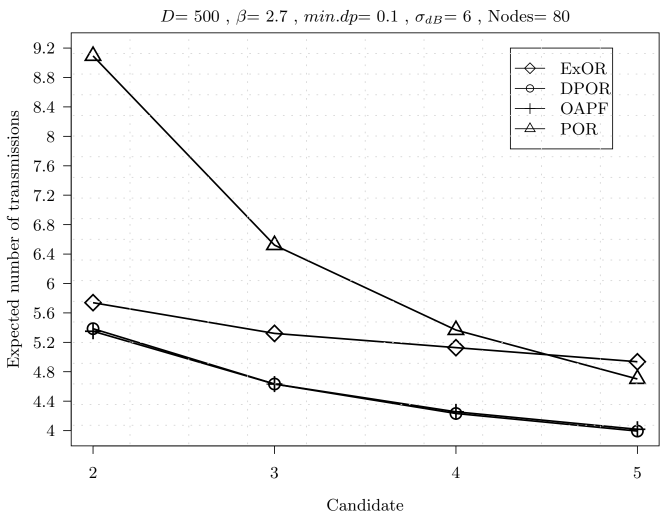


Figure 5.5: Expected number of transmissions for the case $N = 80$ varying the number of candidates.

Obviously, increasing the number of candidates in each node decreases the expected number of transmissions. Figure 5.3 shows the expected number of transmissions of each algorithm with the maximum number of candidate set to 5 ($ncand = 5$). Note that, since the computational cost of the optimum algorithm with 5 candidates is extremely high, we could not obtain the results for the optimum algorithm in the case of $ncand = 5$, in a reasonable amount of time. However it was shown in chapter 4 that the results of OAPF is almost close to the optimum algorithm.

Figure 5.3 shows that, with 5 candidates the expected number of transmis-

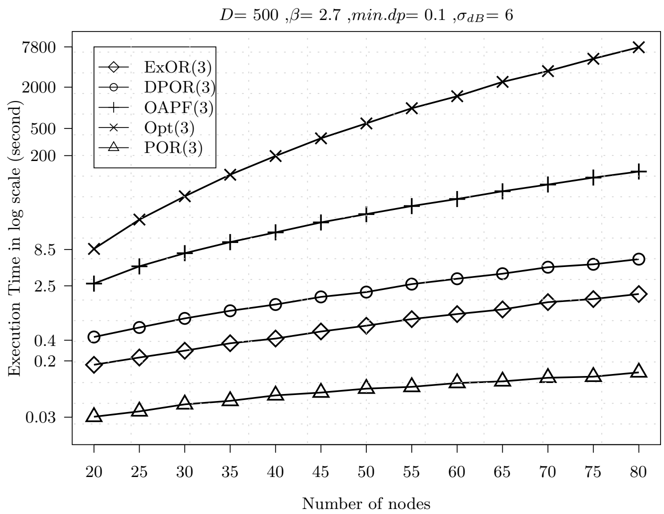


Figure 5.6: Execution time for the case $ncand = 5$.

sions of POR is improved about 50%, compared to the case of 2 candidates (see figure 5.2), and outperforms ExOR. However, implementing an OR protocol with a high number of candidates is difficult, and possibly will introduce large signaling overhead and duplicate transmissions. Regarding DPOR, figure 5.3 depicts that, it achieves almost the same performance as OAPF.

For other scenarios we have obtained similar results. For instance, figures 5.4 and 5.5 have been obtained, respectively, maintaining the total number of nodes equal to $N = 20$ and $N = 80$ (thus, representing a low and high density network), and varying the maximum number of candidates to: $ncand = 2, 3, \cdots, 5$. The expected number of transmissions of POR in both scenarios is only better than ExOR when the maximum number of candidates is set to 5. On the other hand, DPOR, OAPF and Opt give always almost the same results.

5.4.2 Execution Time

In this section we estimate the computational cost of the algorithms under study by measuring the *execution time* that it takes to compute the Candidates Sets (CSs) of all nodes in the network towards the destination. The algorithms were run on a PC with 2 processors Intel Xeon Quad-Core 2.13 GHz and 24 GB of memory.

The execution time is shown in figure 5.6, in logarithmic scale. We have selected $ncand = 3$ as a sample case for our study. As we expected, the two fastest algorithms are POR and ExOR, and the optimum algorithm has the longest execution time. For instance, when the number of nodes in the network is 80, the optimum algorithm needs about 2 hours and 10 minutes to finish. Obviously, with a maximum number of candidates larger than 3 the execution time will be much longer. Although the execution time of POR or ExOR is lower than the other algorithms, recall from section 5.4.1 that POR or ExOR perform worse in terms of expected number of transmissions.

On the other hand, even if DPOR achieves almost the same expected number of transmissions than OAPF and Opt, figure 5.6 shows that DPOR has an execution time significantly lower than these algorithms. E.g. only 37 seconds were needed to compute all Candidates Sets with DPOR in a network with 80 nodes, while more than 100 seconds and 2 hours were needed, respectively, to compute the candidates sets using OAPF and Opt. We conclude that DPOR achieves a performance close to the optimum, but with a low computational cost.

5.5 Conclusions

In this chapter we have proposed a new metric that we call *Expected Distance Progress* (EDP). EDP measures the expected distance progress towards the destination when a packet is sent using Opportunistic Routing. We propose a hop-by-hop candidate selection and prioritization algorithm based on EDP called *Distance Progress based Opportunistic Routing* (DPOR). DPOR tries to maximize EDP, and thus, minimize the expected number of transmissions using OR. In DPOR, each node selects its Candidates Set (CS) independently without considering the other nodes' Candidates Sets. Our proposal does not need the whole topology information of the network to find the CSs. Only the neighbors' position and the links delivery probability to reach them are required.

We have compared DPOR with four other relevant candidate selection algorithms proposed in the literature: ExOR, OAPF, MTS and POR. Numerical results show that the expected number of transmissions of ExOR and POR are far from the optimum algorithm. On the other hand, DPOR and OAPF have almost the same expected number of transmissions as the optimum algorithm. We also have evaluated the computational cost of the algorithms. Our results show that obtaining the Candidates Sets with DPOR is much faster than OAPF and MTS.

We conclude that DPOR is a fast and efficient OR candidate selection algorithm. Additionally, each node requires only neighbors' and destination geographic positions and the delivery probability to the neighbors nodes. Indeed DPOR needs much less information compared to the most of candidate selection algorithms proposed in the literature, which require the whole topology information.

CHAPTER 6

Maximum Performance of Opportunistic Routing

6.1 Introduction

In this chapter we study the maximum gain that can be obtained using Opportunistic Routing (OR). Previous works have studied OR selecting the OR candidates on a given network topology, and comparing the efficiency with the traditional uni-path routing. The efficiency is measured in terms of the expected number of transmissions from the source to the destination. Therefore, we shall refer to gain as the relative difference of the expected number of transmissions required with OR with respect to the traditional uni-path routing.

6.2 Optimal Positions of Candidates

We study the position of the candidates in order to maximize the progress towards the destination. The ingredients of our model are: The maximum number of candidates per node n, and the formula for the delivery probability at a distance d, $p(d)$, which we suppose to be the same for all the nodes. Assume that the destination is far from a generic test node for which we are looking the candidates. Clearly, the optimum candidates will be located over the segment between the test node and the destination (see figure 6.1).

Let $\{c_1, c_2, \cdots c_n\}$ be the ordered set of candidates of the generic test node (c_n the highest priority, and c_1 the least one), and d_i the distance from the test node to the candidate c_i (see figure 6.1). We assume that a coordination protocol exist among the candidates, such that the highest priority candidate receiving the packet will forward the packet (if it is not the destination), while the other nodes will simply discard it. Assume that $p(d_i)$ is the delivery probability from the test node to the candidate c_i, and let Δ_n be the random

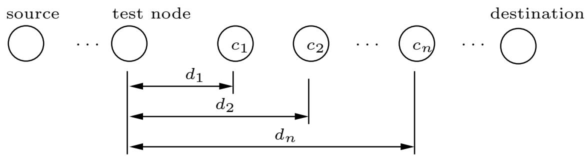


Figure 6.1: Test node and its candidates.

variable equal to the distance reached after one transmission shot. Clearly,

$$\begin{aligned} \mathrm{E}[\Delta_n] = \; & d_n \, p(d_n) + \\ & d_{n-1} \, p(d_{n-1}) \, (1 - p(d_n)) + \\ & \cdots + d_1 \, p(d_1) \prod_{i=2}^{n} (1 - p(d_i)) \\ = \; & d_n \, p(d_n) + (1 - p(d_n)) \, \mathrm{E}[\Delta_{n-1}]. \end{aligned} \tag{6.1}$$

We are interested in looking for the value $d_n \in (d_{n-1}, \infty)$ that maximizes equation (6.1). Note that, this value maximizes also the function

$$f(x) = (x - a) \, p(x) \tag{6.2}$$

where $a = \mathrm{E}[\Delta_{n-1}]$. Notice that $f(a) = 0$ and $f(x)$ is increasing in the neighborhood of a. We shall assume that the delivery probability $p(x)$ is differentiable and $\lim_{x \to \infty} x \, p(x) = 0$, which make plausible to further assume that the function $f(x)$ is quasi-concave in $x \in (a, \infty)$, having a unique critical point equal to its global maximum in this interval. This condition holds e.g. for the shadowing model we use to assess $p(d)$. Additionally, since $\mathrm{E}[\Delta_{n-1}] < d_{n-1} < d_n$, we can reduce the optimization domain to $d_n \in (d_{n-1}, \infty)$. Under these conditions we can compute the distances d_i, $i = 1, \cdots, n$ that maximize (6.1) by solving:

$$\frac{\partial \mathrm{E}[\Delta_i]}{\partial d_i} = 0, \quad d_i \in (d_{i-1}, \infty), \; i = 1, \cdots, n$$

which gives the set of equations:

$$p(d_i) + (d_i - \mathrm{E}[\Delta_{i-1}]) \, p'(d_i) = 0, \qquad d_i \in (d_{i-1}, \infty), \; i = 1, \cdots, n \tag{6.3}$$

where $\mathrm{E}[\Delta_0] = 0$ and $d_0 = 0$. Note that using equation (6.3) we can compute d_1 by solving $p(d_1) + p'(d_1) \, d_1 = 0$, after which we can compute d_2 and so on until d_n. We shall refer to these distances as the *MPD*. In the sequel we shall refer to them as $d_1, \cdots, d_n$, and denote the expected number of transmissions given by equation (6.1) using these distances as $\mathrm{E}[\Delta_n^*]$. Note also that a consequence of equation (6.3) is that the Maximum Progress Distances (MPDs) for the already existing candidates do not change if we decide to add a new candidate to the candidate set.

6.3 Maximum Performance of OR

In this section we investigate the performance of OR in terms of the expected number of transmissions to send a packet from the source to the destination. To do so, we define τ_n to be the random variable equal to the number of transmissions required to send a packet from the source to the destination using n candidates per node. We are thus interested in obtaining bounds to $\mathrm{E}[\tau_n]$.

6.3.1 A Lower Bound with Infinite Candidates

We first derive a result that will be useful in the bounds derived afterwards. Assume an infinitely dense network where the nodes can choose an infinite number of candidates. Assume further that there is not limitation on the minimum delivery probability that live links can have. Let τ_∞ be the random variable equal to the number of transmissions required to send a packet from the source to the destination in such network. With these assumptions, some node as close to the destination as we want can receive the packet with probability 1 (we can choose a region arbitrarily close to the destination that contains an infinite number of candidates). Therefore, if the destination does not receive the packet after it is firstly transmitted by the source, some candidate arbitrarily close to it will receive it an relay it to the destination with just one more transmission, and thus, $\tau_\infty = 2$. Let D be the distance between the source and the destination. From the previous discussion we conclude that:

$$\mathrm{E}[\tau_\infty] = p(D) + 2(1 - p(D)) = 2 - p(D). \tag{6.4}$$

6.3.2 A Lower Bound for the Expected Number of Transmissions

Assume a network with n candidates per node. Since $\mathrm{E}[\Delta_n^*]$ computed in section 6.2 using the MPDs given by equations (6.3) is the maximum progress towards the destination after every transmission shot, we have that the expected number of transmissions to send a packet from the source to the destination ($\mathrm{E}[\tau_n]$) is lower bounded as follows

$$\mathrm{E}[\tau_n] \geq \frac{D}{\mathrm{E}[\Delta_n^*]} \tag{6.5}$$

where D is the distance between the source and the destination.

The bound given by equation (6.5) will be tight as long as the distance D is large compared with d_n, and the nodes are located at the Maximum Progress Distances (MPDs). Clearly, when the nodes become closer than d_n to the destination, the optimal positions cannot be given by the MPDs. In this case the highest priority candidate will be the destination. Thus, the distance of the most priority candidate will be the distance to the destination, and the optimal position of the other candidates should be computed taking the distances that minimize the EAX formula. In fact, this "boundary effect" will propagate to the position of the other nodes between the source and the destination, and their optimal positions may be slightly different than those obtained using the Maximum Progress Distances (MPDs) (we shall investigate this in section 6.6). Nevertheless, the expected distance progress after each transmission could not be as high as the one obtained using the MPDs, which guarantees that (6.5) is a lower bound.

We can use the result obtained for an infinite number of candidates to improve the bound given by (6.5). First, the expected number of transmissions cannot be less that the value given by equation (6.4). Therefore, we have that:

$$\mathrm{E}[\tau_n] \geq \max\left(2 - p(D),\ \frac{D}{\mathrm{E}[\Delta_n^*]}\right). \tag{6.6}$$

The bound given by (6.6) can still be improved when $n > 1$ as we explain next. As we said before, when the nodes are closer than d_n to the destination, the position of the nodes cannot be the MPDs. Therefore, using $\mathrm{E}[\Delta_n^*]$ as the progress in this region may be a coarse approximation. To estimate the progress in this region we note that before the packet reaches the destination, at least one node in the interval $[D - d_n, D)$ will receive it, because the furthest candidate of any node is at a distance d_n. We shall refer to the first node in this interval that receives the packet as $v(x)$, where x is the distance from this node to the destination (we assume that the source is located at 0, and the destination at D). Now, the number of transmissions from the source to $v(x)$ can be lower bounded by $(D - x)/\mathrm{E}[\Delta_n^*]$ (i.e. assuming the maximum progress), and the number of transmissions from $v(x)$ to the destination can be lower bounded assuming an infinite number of candidates between $v(x)$ and the destination (equation (6.4)). Adding both terms we have $\mathrm{E}[\tau_n|v(x)] \geq (D-x)/\mathrm{E}[\Delta_n^*]+2-p(x) = D/\mathrm{E}[\Delta_n^*]+2-p(x)-x/\mathrm{E}[\Delta_n^*]$. Thus, if we want a lower bound we must take x that minimizes $\mathrm{E}[\tau_n|v(x)]$ in the interval $x \in (0, d_n]$.

Summing up, we have that:

$$E[\tau_n] \geq \begin{cases} \max\left(2 - p(D),\ \frac{D}{E[\Delta_1^*]}\right),\ n = 1 \\ \max\left(2 - p(D),\ \frac{D}{E[\Delta_n^*]} + \right. \\ \left. \quad \inf_{x \in (0,\, d_n]} \left\{ 2 - p(x) - \frac{x}{E[\Delta_n^*]} \right\} \right),\ n > 1 \end{cases} \tag{6.7}$$

6.3.3 An Upper Bound for the Gain

Let us denote by τ_n the number of transmissions when the candidates are optimally placed. In order to measure the improvement that can be reached using OR we define the *gain* (G_n) as the relative difference of the expected number of transmissions required with the OR with n candidates ($E[\tau_n]$), with respect to the uni-path routing case. Note that OR with only 1 candidate per node is equivalent to uni-path routing. Therefore, we shall refer to the expected number of transmissions with uni-path routing as $E[\tau_1]$, and thus:

$$G_n = \frac{E[\tau_1] - E[\tau_n]}{E[\tau_1]} = 1 - \frac{E[\tau_n]}{E[\tau_1]}. \tag{6.8}$$

Using the same intuition as in (6.5) we can write

$$\frac{D}{E[\Delta_1^*]} \leq E[\tau_1] \leq \frac{\lceil D/d_1 \rceil d_1}{E[\Delta_1^*]} \tag{6.9}$$

and using the lower bound for $E[\tau_n]$ and the upper bound for $E[\tau_1]$ it follows from (6.8) that

$$G_n \leq 1 - \frac{D/d_1}{\lceil D/d_1 \rceil} \frac{E[\Delta_1^*]}{E[\Delta_n^*]}. \tag{6.10}$$

6.4 Numerical Results

We now give some numerical examples of the formulas derived in the previous sections. We shall assume that the delivery probability $p(d)$ is given by equation (B.5). Substituting $p(d)$ in the equations (6.3) and solving them numerically we obtain the maximum progress distances for the candidates shown in figure 6.2.

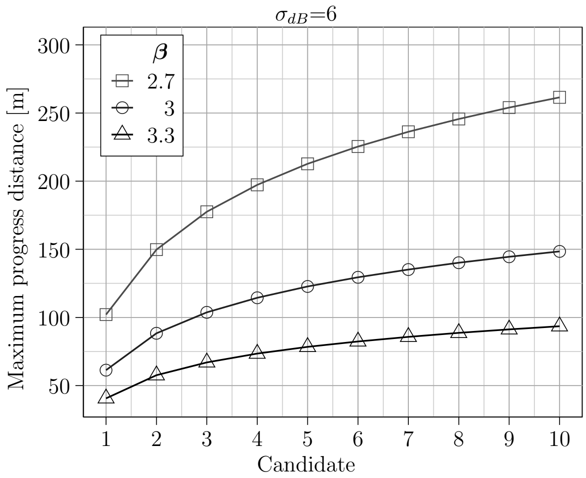


Figure 6.2: Maximum Progress Distances for the candidates.

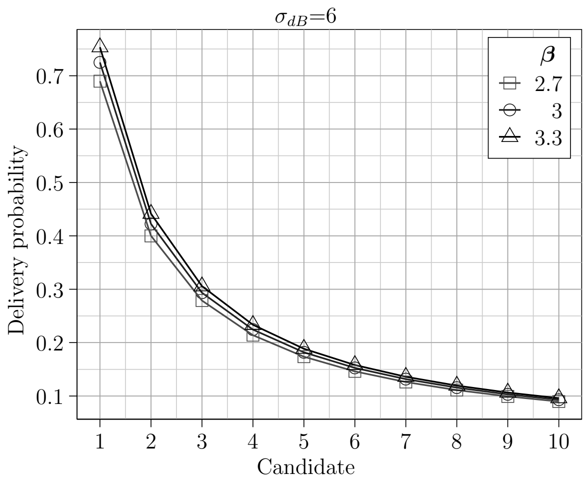


Figure 6.3: Delivery probability to each candidate located at the maximum progress distances.

In our numerical experiments we have set the model parameters to the default values used by the network simulator (ns-2) [2], given in table B.2. There are three curves, that correspond to three values of the loss exponent of the propagation model: $\beta = 2.7$, $\beta = 3$ and $\beta = 3.3$. Note that the larger is β, the lower is the transmission range of the nodes, and thus, the shorter are the distance of the candidates.

Figure 6.3 shows the delivery probabilities obtained for the corresponding points shown in figure 6.2. It is interesting that the probabilities are very similar for all values of β. This fact could be use as a rule of thumb in the selection of candidates.

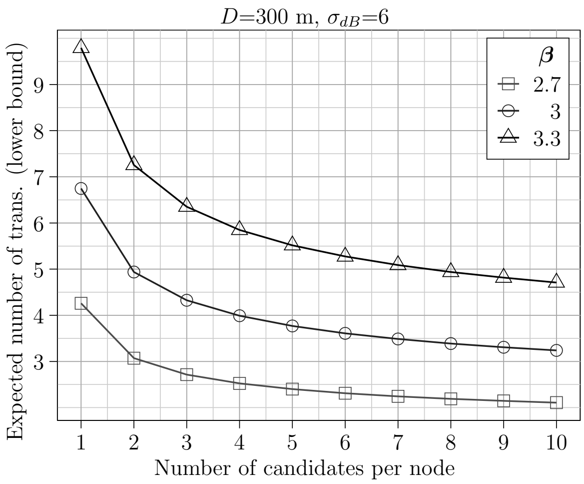


Figure 6.4: Expected number of transmissions (lower bound).

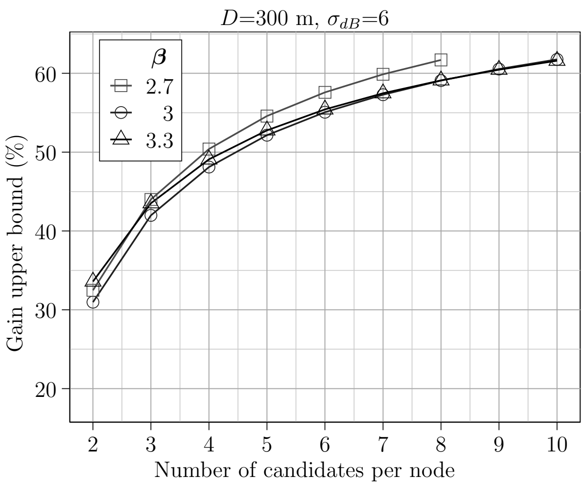


Figure 6.5: Gain (upper bound).

Finally, figure 6.4 depicts the lower bound of the expected number of transmissions (equation (6.7)) for a distance $D = 300$ m between the source and the destination, and figure 6.5 shows the corresponding upper bound to the gain (equation (6.10)). As we shall see in section 6.5, the lower bounds given by equation (6.7) are very tight. Consequently, the gains that can be obtained using OR are close to the upper bounds depicted in figure 6.5. These figures show that the highest gain increase occurs when we move from 1 to 2 candidates (approximately 30% of gain). After which the gain increases approximately up to 60% with 10 candidates. However, implementing an OR protocol with a high number of candidates is difficult, and possibly will introduce large signaling overhead and duplicated transmissions that would prevent to reach such large gains. This motivates that selecting a maximum

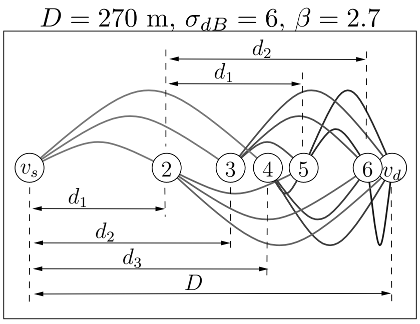


Figure 6.6: Quasi Optimal Opportunistic Routing network with a maximum of 3 candidates per node.

number of candidates per node equal to 2 or maybe 3 is possibly a sensible choice.

6.5 Quasi Optimal Opportunistic Routing Network

In this section we compute an upper bound for the expected number of transmissions by computing EAX in a network where the candidates are positioned using the Maximum Progress Distances (MPDs) computed as in section 6.2. Note that not all the candidates can be located using these distances, since for some nodes the distance to the destination can be shorter than the distance to the candidate. For these nodes we will use the destination and its closest neighbors located between the node and the destination as candidates. Since these candidates, at least, are not located at the optimum positions, the expected number of transmissions computed for such network will be an upper bound to the minimum expected number of transmissions that can be achieved using OR. We shall refer to such network as *Quasi Optimal Opportunistic Routing Network (QOO)*.

Figure 6.6 depicts an example of a network with 3 candidates per node build using these rules. The source is v_s and the destination is v_d. Nodes 2, 3 and 4 are located at the MPDs from v_s: d_1, d_2 and d_3 respectively. Nodes 5 and 6 are located at the maximum progress distances from node 2: d_1 and d_2 respectively. Since v_d is closer from node 2 than d_3, v_d is taken as the third candidate of node 2. Since node 6 is at a distance d_1 from node 3, and v_d is closer from this node than d_2 and d_3, the candidates of node 3 are nodes 5, 6 and v_d. Likewise it is done for the other nodes.

Figure 6.7 shows the expected number of transmissions varying the distance D between the source and the destination for a QOO network build as explained before. The curves shown in the figure have been obtained using

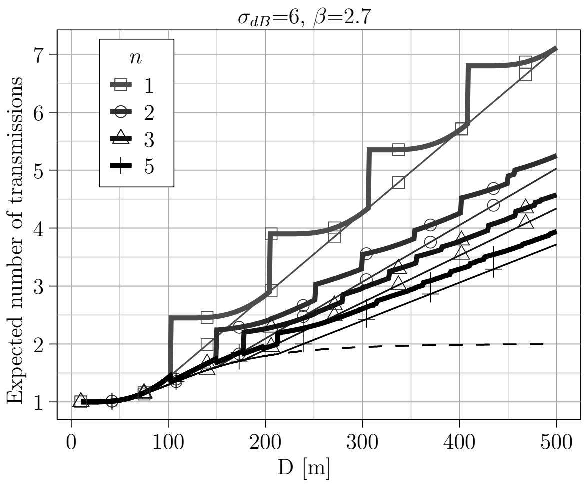


Figure 6.7: Lower and upper bounds (thin and thick lines respectively) of the minimum expected number of transmissions achievable with Opportunistic Routing for n = 1, 2, 3 and 5 maximum number of candidates per node. The dashed line corresponds to infinite number of candidates.

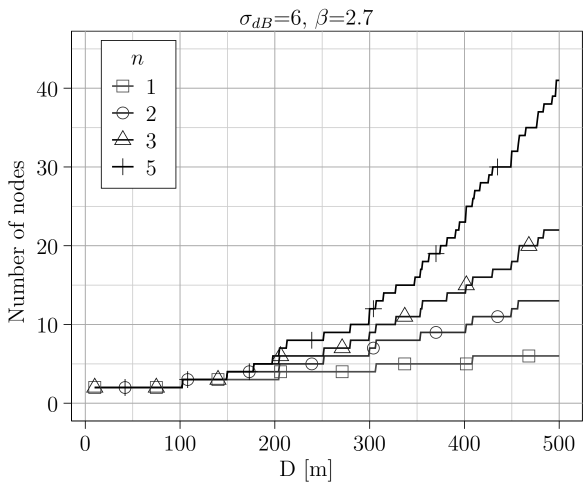


Figure 6.8: Number of nodes of the quasi optimal Opportunistic Routing networks used in figure 6.7

a maximum number of candidates per node equal to 1, 2, 3 and 5 (cfr. the numbers in the legend). Figure 6.8 shows the number of nodes that resulted in the QOO networks used to obtain the corresponding values of figure 6.7. In figure 6.7 we have also added the lower bounds of equation (6.7) (thin lines), and the lower bound for an infinite number of candidates given by equation (6.4) (dashed line).

The delivery probability of the links ($p(d)$) has been obtained using the shadowing model (equation (B.5)) with a path loss exponent $\beta = 2.7$. The

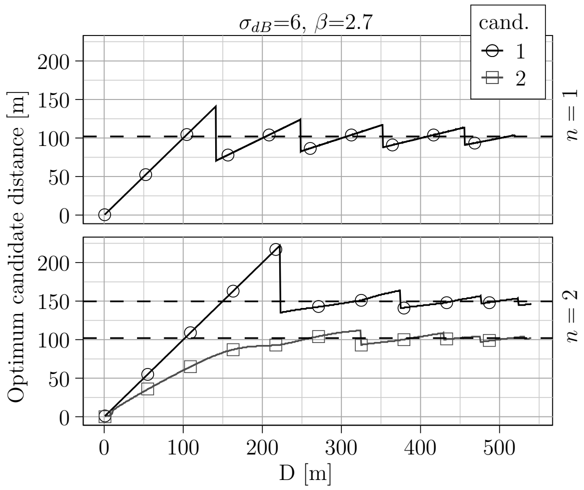


Figure 6.9: Optimum distances of the candidates (d_i^*) with a maximum of $n = 1$ and $n = 2$ candidates per node (top and bottom respectively). The dashed lines are the MPDs (d_i).

expected number of transmissions has been obtained using the Markov chain that we have proposed in [17]. These values could have been obtained also using the recursive formula of the expected number of transmissions that has been proposed by several authors (see e.g. [96, 30]). However, we have noticed that solving the Markov chain was faster than using the recursive formula.

Figure 6.7 confirms that the lower bounds of the expected number of transmissions obtained with equation (6.7) are very tight, since they are very close to the upper bound obtained with the QOO network. Furthermore, this result seems to indicate that the MPDs are very close to the optimum distances. We shall investigate this in the next section. Note that the discontinuities of the upper bound occur at the distances where a new node is added to the QOO network. E.g. in the scenario with 1 candidate, which occurs when the distance between the source and the destination (D) is a multiple of d_1.

6.6 Validation

In the previous sections the *Maximum Progress Distances* have been obtained and used to derive bounds, which are rather accurate approximations as well, of the performance of OR measured by the mean number of transmission required to reach the destination.

For a network of finite length ($D < \infty$), the optimal distances of the candidates —in the sense of minimizing the mean number of transmissions re-

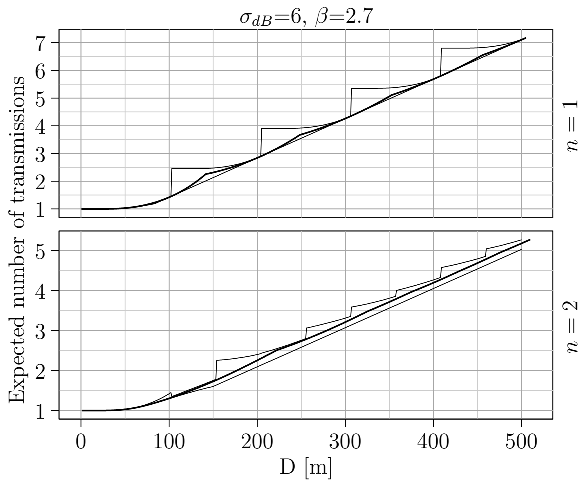


Figure 6.10: Expected number of transmissions obtained with the optimum distances of figure 6.9, and its upper and lower bounds.

quired to reach the destination— are more complex to obtain and in general may not coincide with the *Maximum Progress Distances*.

In this section, we use a numerical approximation to estimate the optimal distances in a finite length network with the aim of empirically confirming some of the intuitions that have been applied previously, and provide a further insight into the optimal distances problem.

Let $V_n(x)$ be the minimum mean number of transmissions required to reach the destination that is at distance x from the source node, when a maximum of n candidates per node is used. We can write

$$V_n(x) = \min_{x_1 < \cdots < x_n} \left\{ \frac{1}{1 - \prod_{i=1}^{n} q(x_i)} \times \left(1 + p(x_n)V_n(x - x_n) + \cdots + \prod_{i=2}^{n} q(x_i)p(x_1)V_n(x - x_1)\right)\right\} \quad (6.11)$$

where $q(d) = 1 - p(d)$ and $V_n(0) = 0$. If the number of nodes between the source and the destination is less than n, then the destination and the intermediate nodes are taken as candidates. We shall refer as d_i^* to the optimal distances x_i that minimize equation (6.11).

We have solved the optimization problem of (6.11) in an approximate fashion by considering a discrete network (a finite number of nodes are evenly distributed between source and destination) and then performing an exhaustive optimum search. The network density, i.e., the number of nodes, have

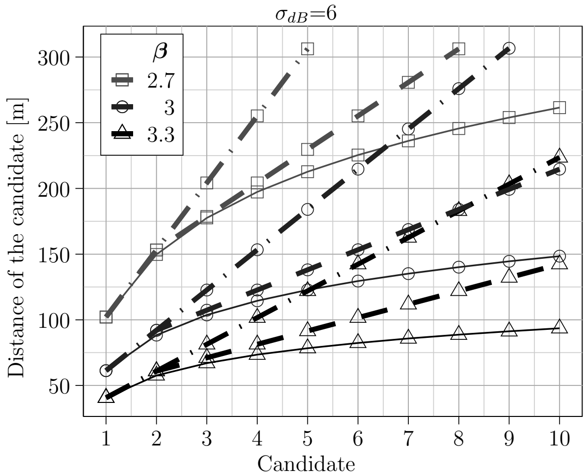


Figure 6.11: Maximum Progress Distances of the candidates (thin lines) and its $d_1/4$ (dashed line) and $d_1/2$ (dot-dashed line) approximations.

been increased until the minimum obtained does not vary significantly. Obviously, the exhaustive search becomes un feasible as the maximum number of candidates, n, or the network size, D, grow. For this reason, we have limited this method to a maximum number of candidates equal to $n = 1$ and $n = 2$. Nevertheless, as we will see in the following, these two scenarios are enough for validation purposes.

Figure 6.9 compares the optimal distances (d_i^*) and the Maximum Progress Distances (d_i) as functions of D. The optimum mean number of transmission obtained for the optimal distances are shown in Figure 6.10 along with their corresponding lower and upper bounds.

We observe that the optimal distances converge to the Maximum Progress Distances ($d_1 \approx 102\ m$ and $d_2 \approx 150\ m$) when D grows. It is also observed that while the MPDs of the first candidate (d_1) are the same for different values of the maximum number of candidates, the optimum distances d_1^* are different for $n = 1$ and for $n = 2$, although they converge to the same value (that of the Maximum Progress Distance, d_1, as we said before).

Notice that, as expected, when $n = 1$ and D is a multiple of d_1, the optimal distance equals the Maximum Progress Distance ($d_1^* = d_1$) and the lower bound of $V_1(D)$ turns out to yield an exact value. Also, as it has been predicted, the lower bound for $V_1(D)$ is tighter than that for $V_2(D)$. On the other hand, in both cases ($n = 1, 2$) when D grows the shape of $V_n(D)$ tends to be a straight line whose slope is matched by that of the lower bound, i.e., by $1/\mathrm{E}[\Delta_n^*]$. Moreover, the shape of $V_2(D)$ gets smooth more rapidly than $V_1(D)$ does. A similar observation can be made about the rate of convergence of the optimal distances to the MPDs.

6.7 Sensitivity to Node Positions

The Maximum Progress Distances computed in section 6.2 can be of practical interest in the design of a static network using OR. E.g the back-haul of a mesh network, or the position of the nodes in a sensor network. A first approach could be the Quasi Optimal OR Network described in the previous section. However, for such network the number of nodes increases nearly exponentially with the distance between the source and the destination, D, as shown in figure 6.8. In this section we look for positions of the nodes that, being close to their optimal values, allow reducing the number of nodes of the network.

Looking at the MPDs obtained for different parameters of the propagation model (figure 6.2), we can observe that $d_2 \approx d_1 + d_1/2$ and $d_3 \approx d_1 + d_1/2 + d_1/4$. This suggest that a good compromise is positioning the nodes equally spaced at a distance $d_1/4$, choosing d_1 for the first candidate, $\hat{d_2} = d_1 + d_1/2$ for the second, and $\hat{d_i} = d_1 + d_1/2 + (i-2) \times d_1/4$ for the candidates $i > 2$. Doing this way, a distance D would require a number of nodes $N \leq 4 \cdot \lceil D/d_1 \rceil$. If only 2 candidates are going to be used, or if we wish to reduce further the number of nodes, a coarser approach would be positioning the nodes equally spaced at a distance $d_1/2$, choosing d_1 for the first candidate and $\hat{d_i} = d_1 + (i-1) \times d_1/2$ for the candidates $i > 2$. Doing this way, the required number of nodes would be $N \leq 2 \cdot \lceil D/d_1 \rceil$. We shall refer to these approximations as $d_1/4$ and $d_1/2$ respectively. Figure 6.11 shows the Maximum Progress Distances computed as in section 6.2 and its $d_1/4$ and $d_1/2$ approximations.

Figures 6.12 and 6.13 show the sensitivity of the expected number of transmissions to the $d_1/4$ and $d_1/2$ approximations. As in section 6.4, we have used a distance between the source and the destination $D = 300\ m$, and three values of the loss exponent of the propagation model: $\beta = 2.7$, $\beta = 3$ and $\beta = 3.3$. For each value of β figure 6.12 shows four curves of the expected number of transmissions: (i) the lower bound computed as in section 6.3 (note that these curves are the same than those shown in figure 6.4); (ii) using the QOO network of section 6.5 (solid lines); and (iii, iv) using its $d_1/4$ (dashed line) and $d_1/2$ (dot-dashed line) approximations. Figure 6.13 shows the number of nodes of the networks that where used to compute the expected number of transmissions for the corresponding cases (ii, iii, iv) of figure 6.12.

Figure 6.12 shows that the expected number of transmissions obtained for the QOO network is very close to the lower bound. Nevertheless, figure 6.13 shows that building the QOO network requires a high number of nodes. The maximum value is 665 nodes, obtained for $\beta = 3.3$ (where the nodes' coverage is the shortest) and 10 candidates per node. Figure 6.12 shows too

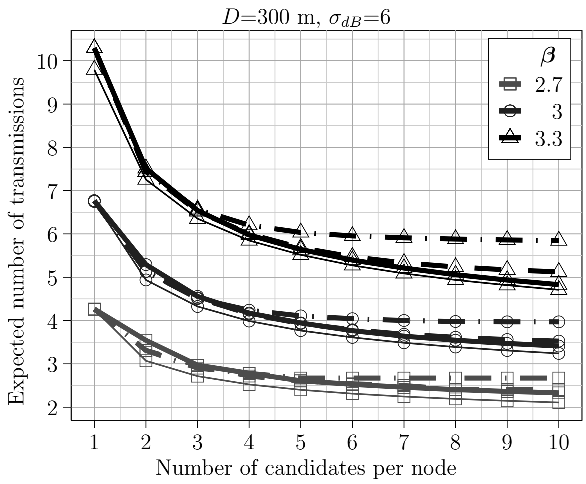


Figure 6.12: Expected number of transmissions: (i) Lower bound (thin lines), (ii) using the QOO network of section 6.5 (solid lines), (iii, iv) using, respectively, its $d_1/4$ (dashed line) and $d_1/2$ (dot-dashed line) approximations.

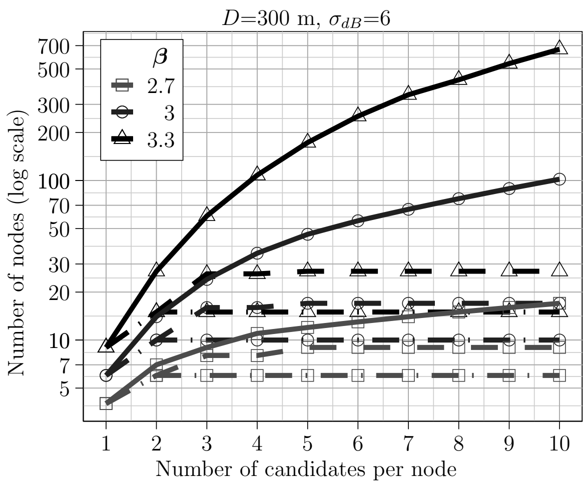


Figure 6.13: Number of nodes of the networks used in figure 6.11 (i) using the QOO network of section 6.5 (solid lines), (ii, iii) using its $d_1/4$ (dashed line) and $d_1/2$ (dot-dashed line) approximations.

that the expected number of transmissions obtained for the $d_1/4$ and $d_1/2$ approximations it is also very close to the lower bound. Only for $\beta = 3.3$ and more than 5 candidates per node the difference is noticeable. However, in figure 6.13 we can see that the number of nodes using the $d_1/4$ and $d_1/2$ approximations is enormously reduced (e.g. it is 27 and 15 nodes respectively for the $d_1/4$ and $d_1/2$ approximations in the same scenario for which 665 nodes are used with the QOO network).

We conclude that choosing the position of the 2 candidates closest to the sender near to their optimal positions, is the most critical in order to minimize the expected number of transmissions. Consequently, what we have called $d_1/2$ approximation may be a sensible rule of thumb in the design of the node positions in a static network using OR.

6.8 Candidate Selection Algorithm Based on MPD

In this section we proposed a new candidate selection algorithm based on MPDs that we call *Candidate selection based on Maximum Progress Distance (CMPD)*. It tries to select the candidates that are located at the Maximum Progress Distance.

Assume that there are $\mathbb{N}$ nodes in the network with a source s and destination d. We assumed that: all nodes $v \in \mathbb{N}$ know the position coordinates of their neighbors, $N(v)$ and the destination d. We have used $D_{x,y}$ to refer the geographic distance between two nodes x and y.

Algorithm 7 shows the pseudo-code of CMPD for a node v to select its Candidates Set to reach the destination d. The parameter $ncand$ in algorithm 7 is the maximum number of candidates in each node. Let $\hat{c}_i$ be a virtual candidate of v that lies on the straight line between v and the destination d at distance d_i. The value of d_i is given by the previous results obtained in section 6.2 (see figure 6.2). Note that, the obtained value for d_i, $i \in \{1, 2, \dots, ncand\}$, in section 6.2 is valid when the destination is far away from the forwarder i.e., $D_{v,d} > d_{ncand}$. Therefore, when the distance between source and the destination is shorter than d_{ncand} we shrink the MDP distances (d_i, $i \in \{1, 2, \dots, ncand\}$) such that $d_{ncand} = D_{v,d}$ (see lines 1– 5). The corresponding candidate c_i is chosen from the nodes in $N(v)$ which is the one closest to $\hat{c}_i$, i.e, $c_i = \arg\min_{c \in N(v)} D_{c,\hat{c}_i}$. Note that c_i should be closer than v to the destination ($D_{c_i,d} < D_{v,d}$). Finally, the candidates set is order according to the closeness of each candidate to the destination. The candidates which is nearer to the destination will have higher priority.

6.8.1 Performance Evaluation of CMPD

In this section we study the performance of CMPD, compared to the other candidate selection algorithms. We have compared our proposal (CMPD) with four candidate selection algorithms that have been studies in chapters 5. The algorithms for our comparisons are: ExOR [12], OAPF [96], MTS [58] and DPOR [25].

Algorithm 6.1: Candidate.selection.MPD(v, d, $ncand$).

Data:
$D_{x,y}$: Geographic distance between nodes x and y
$\hat{c}_i$: ith candidate which is located at the optimum position.
d_i: Geographic distance between v and $\hat{c}_i$.

if $D_{v,d} < d_{ncand}$ **then**
  for i=1 **to** $ncand$ **do**
    $d_i \leftarrow d_i * D_{v,d}/d_{ncand}$;
  end
end
$N(v) = \{n = neighbor(v) | D_{n,d} < D_{v,d}\}$
$i \leftarrow 1$
$C^{v,d} \leftarrow \emptyset$
while $|C^{v,d}| < ncand$ & $N(v) \neq \emptyset$ **do**
  $cand \leftarrow \arg\min_{c \in N(v)} D_{c,\hat{c}_i}$
  $C^{v,d} \leftarrow C^{v,d} \cup cand$
  $N(v) \leftarrow N(v) \setminus cand$
  $i \leftarrow i + 1$
end
Order $C^{v,d}$ according to $D_{c_i,d}, c_i \in C^{v,d}$

6.8.2 Methodology

In order to compare different algorithms, and since we want to focus on the effect of candidate selection, we have assumed that there is a perfect coordination between the candidates, i.e. the most priority candidate successfully receiving the packet will be the next forwarder. The nodes retransmit the packets until successful delivery. Furthermore, we have assumed that there is only one active connection in the network.

We compare the performance of each algorithm in terms of the expected number of transmissions needed to send a packet from the source to the destination and the execution time of each algorithm.

We consider scenarios with different number of nodes ($20 \leq N \leq 80$) randomly placed in a square field with diagonal $D = 500$ m, except the source and the destination which are placed at the diagonal end points. Each point in the plots is an average of 100 runs with different random node positions. The delivery probabilities have been assigned with the shadowing model with $\beta = 2.7$ and $\sigma_{db} = 6.0$.

In the candidate selection algorithm, we have assumed that a link between any two nodes exists only if the delivery probability between them is greater (or equal) than $min.dp = 0.1$. We have compared the algorithms for different maximum number of candidates: $ncand = 2$ and 3. We shall use the

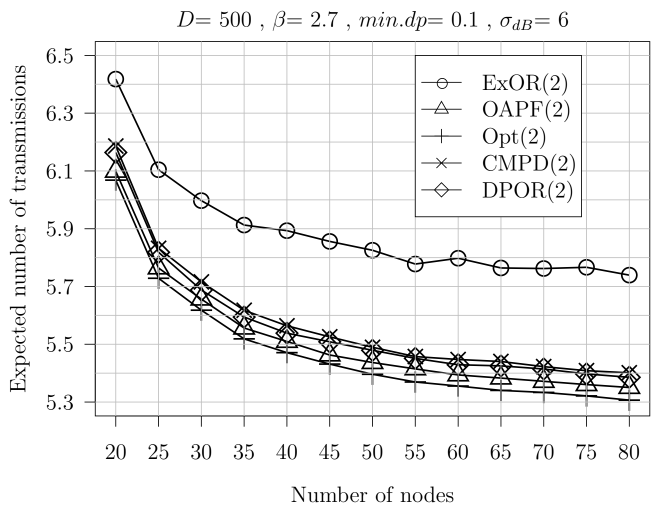


Figure 6.14: Expected number of transmissions for the case $ncand = 2$

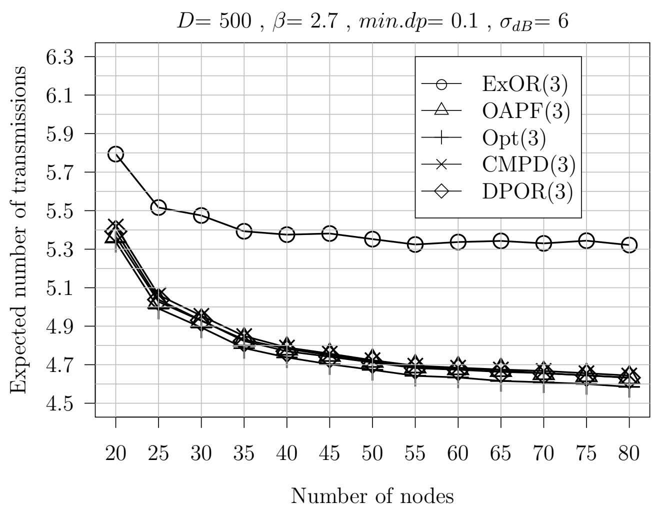


Figure 6.15: Expected number of transmissions for the case $ncand = 3$.

notation ExOR(n) to refer to ExOR with $ncand = n$, and similarly for the other algorithms under study.

6.8.3 Expected Number of Transmissions

Figures 6.14 and 6.15 show the expected number of transmissions of different algorithm in the case of $ncand = 2$ and 3 candidates, respectively. The curves have been obtained varying the number of nodes, but maintaining the distance $D = 500$ m between the source and the destination, thus, increasing the density of the network. In all figures we use the notation Opt(n) to refer to the optimum candidate selection algorithm. Note that, we have used MTS [58] for the optimum candidate selection algorithm.

As a first observation in figure 6.14, we can see that increasing the number of nodes causes decreasing the expected number of transmissions in all algorithms. The results for CMPD in figure 6.14 are very close to OAPF, DPOR and the optimum algorithm. Recall that CMPD chooses the most closest node to the virtual node which is located at the optimum position.

Obviously, increasing the number of candidates in each node decreases the expected number of transmissions. Figure 6.15 shows the expected number of transmissions of each algorithm with the maximum number of candidate set to 3 ($ncand = 3$). The expected number of transmissions for CMPD is very close to DPOR, OAPF and the optimum algorithm while ExOR has much more expected number of transmissions than the others.

6.8.4 Execution Time

In this section we evaluate the computational cost of the algorithms under study by measuring the *execution time* that it takes to compute the candidates sets of all nodes in the network towards the destination. Like in chapter 5, the algorithms were run on a PC with 2 processors Intel Xeon Quad-Core 2.13 GHz and 24 GB of memory.

The execution time is shown in figure 6.16, in logarithmic scale. We have selected $ncand = 3$ as a sample case for our study. As expected, the fastest algorithms are CMPD and ExOR, and the optimum one has the longest execution time. For instance, when the number of nodes in the network is 80, the optimum algorithm needs about 2 hours to finish. Obviously, with a maximum number of candidates larger than 3 the execution time will be much larger. OAPF and DPOR are in the middle of the exhaustive search of the optimal algorithms and the simplicity of CMPD and ExOR, and thus, has an execution time that falls in between these algorithms. E.g. for OAPF the execution time is 2.6 to 117 seconds for $N = 20$ and $N = 80$ nodes, respectively.

Although the execution time of ExOR is close to CMPD, its expected number of transmissions is much higher than CMPD (see figure 6.15). Furthermore, ExOR needs to know the whole network topology while CMPD just need the position of the destination and the neighbors.

6.9 Node Position in Grid OR Networks

In this section we investigate whether the $d_1/2$ approximation proposed in previous section can be also used in an OR network with a grid topology. We consider a square grid with N nodes, and denote by d the distance

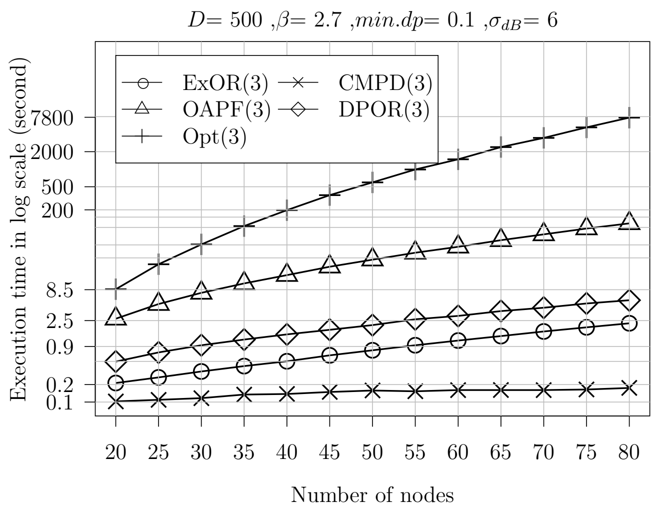


Figure 6.16: Execution time in the case $ncand = 3$.

between two adjacent nodes (see figure 6.17). Clearly, the best positions for those candidates to reach destinations located in verticals or horizontals lines with respect the source will be the MPD. Thus, setting $d = d_1/2$ would be a good rule of thumb for these destinations. However, most destinations are not located in these lines, thus, it is not clear whether a shorter value for d may be a better choice.

In order to investigate an appropriate value for d we have varied its value, while maintaining the number of nodes of the grid, N. We have assumed that the source V_s and the destination V_d are located at positions (1,1) and (i,n) where $i \in \{1, ..., n\}$, respectively (see figure 6.17). Furthermore, we refer to D as the distance between V_s and V_d. We have used CMPD which is proposed in section 6.8, to select the candidates of the node towards each destination. We shall use the notation CMPD(n) to refer to CMPD with $ncand = n$. The link delivery probabilities between two nodes have been assigned with the shadowing model with $\beta = 2.7$ and $\sigma_{db} = 6.0$. We have assumed that a link between any two nodes exists only if the delivery probability between them is greater (or equal) than $min.dp = 0.1$.

Note that varying d, the value of D, and possibly the nodes chosen by the candidate selection algorithm also change. Thus, in order to compare the goodness of the different values of d, we have defined the metric that we shall call *Average Distance Progress (ADP)* as:

$$ADP(d, V_s, V_d) = \frac{D(d, V_s, V_d)}{EAX(d, V_s, V_d)} \tag{6.12}$$

where $D(d, V_s, V_d)$ is the distance between the source V_s and the destination V_d, and $EAX(d, V_s, V_d)$ is expected number of transmissions (EAX) from the

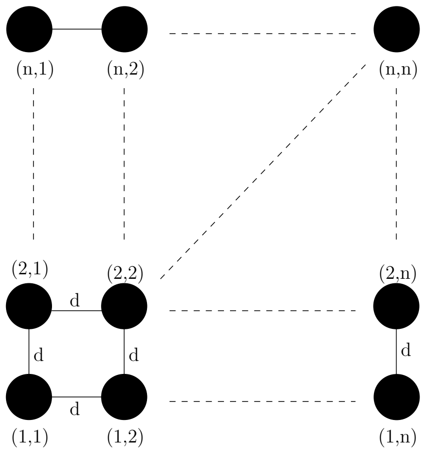


Figure 6.17: Grid topology.

source V_s to the destination V_d using the candidates chosen for a given value of d. Note that $ADP(d, V_s, V_d)$ represents the average number of meters that a packet progresses towards V_d at each transmission shot. Therefore, the optimum value of d would maximize equation (6.12).

Figures 6.18 shows the value of $ADP(d, V_s, V_d)$ for $N = 400$ nodes varying the distance between adjacent nodes on the square side (d) in the range $1 \leq d \leq 200$. The curves have been obtained for different maximum number of candidates: $ncand = \{2, 3, 5\}$. In figure 6.18, we have fixed the position of the source and the destination at the diagonal end points, i.e. $V_s = (1, 1)$ and $V_d = (20, 20)$. For the sake of comparison, we have included the value of $d_1/2$ (see legend $Op-Position$ in figure 6.18). For a shadowing propagation model with parameters $\beta = 2.7$ and $\sigma_{db} = 6.0$, it is $d_1/2 \approx 51$ m (see figure 6.2).

Figure 6.18 shows that the larger is the maximum number of candidates ($ncand$), the higher is the ADP. For each $ncand$, the vertical lines in figure 6.18 show the values of d that maximize ADP. Figure 6.18 shows that the curves of ADP are rather flat around their maximum point. For instance, with $ncand = 2$ ADP reaches its maximum (102 m) at $d = 36$ m. However, using $d = d_1/2 = 51$ m it is obtained ADP= 97 m, only 5% smaller that the maximum.

Figure 6.18 shows that the optimum d is smaller than $d = d_1/2 = 51$ m for $ncand = 2$, while it is higher for $ncand = 3, 5$. The reason is the following. Recall that the optimum position of the first candidate is d_1, and the second is very close to $d_1/2$. Therefore, choosing $d = (d_1/2)/\sqrt{2} \approx 36$ m we obtain the best position for the candidates in the diagonal. If more than 2 candidates are used, the best position of the other candidates is not well fitted by nodes in the diagonal. Thus, the candidate selection algorithm

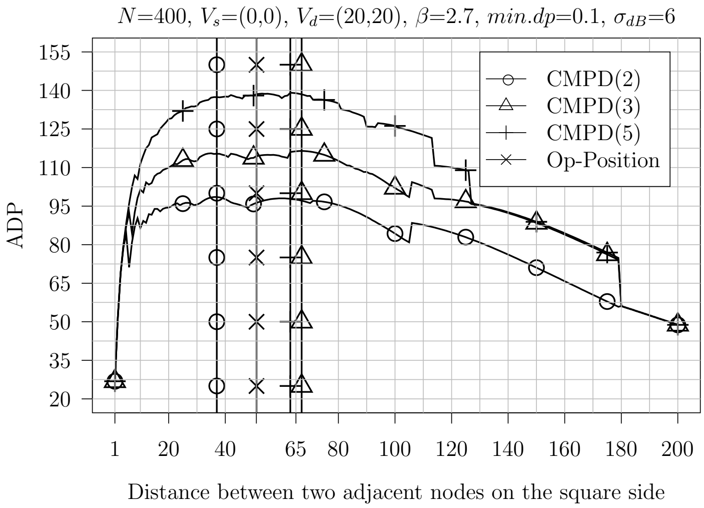


Figure 6.18: ADP for $V_s = (1, 1)$ and $V_d = (20, 20)$, varying the distance between two adjacent nodes on the square side.

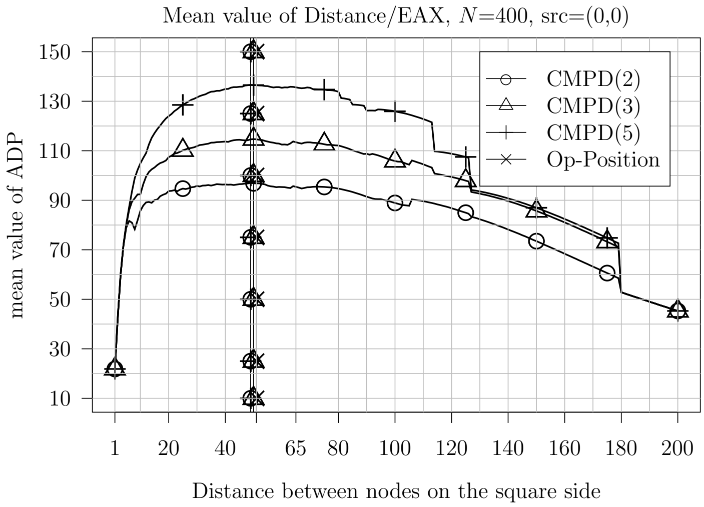


Figure 6.19: Mean value of ADP for $V_s = (1, 1)$ varying the distance between nodes on the square side.

chooses also nodes out of the diagonal, and closer to the sending node. This motivates that a slightly higher performance can be achieved using a higher value for d, as shown in figure 6.18.

Figure 6.19 shows the average value of ADP over all destinations in one side of the square opposite to the source, i.e. $V_d = (i, 20), i \in \{1, 2, ..., 20\}$. As before, the vertical lines are located at $d = d_1/2 = 51$ m and the values of d that maximize ADP. Figure 6.19 shows that all vertical lines almost overlap.

We can conclude that in a grid topology ADP is not very sensitive to d in a rather large interval. Additionally, setting $d = d_1/2$ is a good rule of thumb, yielding an ADP close to the optimum for all possible destinations.

In an other experiment we have investigated the position of candidates in a grid scenario. We have created a grid topology fixing the diagonal of square to 500 m and varying the position of destination. The number of nodes in this experiment is set to $N = 400$ nodes. Like the previous scenario we have fixed the position of source to $V_s = (1, 1)$ and varying the position of destination by choosing the nodes on the diagonal of the square ($V_d = (i, i), 2 \leq i \leq 20$). For the candidate selection algorithm we have run CMPD with the maximum number of candidates equal to $ncand = 5$.

Figure 6.20 shows the distance of selected candidates from the source node V_s varying the distance between source and the destination. To be more precise, we have also shown the node coordinates of each selected candidate in figure 6.20. As we can see in figure 6.20 when the distance between V_s and V_d is increasing the distance between the selected candidate and the source node does not change. The distance of each candidate to the source is almost the same as we have obtained in section 6.2 (see figure 6.2). For instance the distance between the most priority candidate (cand-5) and V_s is about 212 m which is almost the same with $d_5 = 212.74$.

6.10 Conclusions

In this chapter we have derived the equations that yield the distances of the candidates in Opportunistic Routing (OR) such that the per transmission progress towards the destination is maximized. We have called them as the *Maximum Progress Distances (MPDs)*. The only ingredient to obtain these distances is the law for the delivery probability between nodes as a function of distance. An important consequence of our derivation is that the MPDs for the already existing candidates do not change if we decide to add a new candidate to the candidate set.

Based on these MPDs, we have proposed a lower bound to the expected number of transmissions needed to send a packet using OR. The lower bound has proven to be very tight. By modeling the delivery probabilities with a shadowing propagation model, we obtained numerical results showing that the expected number of transmissions can be reduced up to a 30% with only 2 candidates, whereas in order to reduce it another 30% the number of candidates has to be increased up to 10.

We have constructed a quasi optimum OR network locating the nodes and their candidates at the Maximum Progress Distances whenever possible. Solving the expected number of transmissions in these networks we have

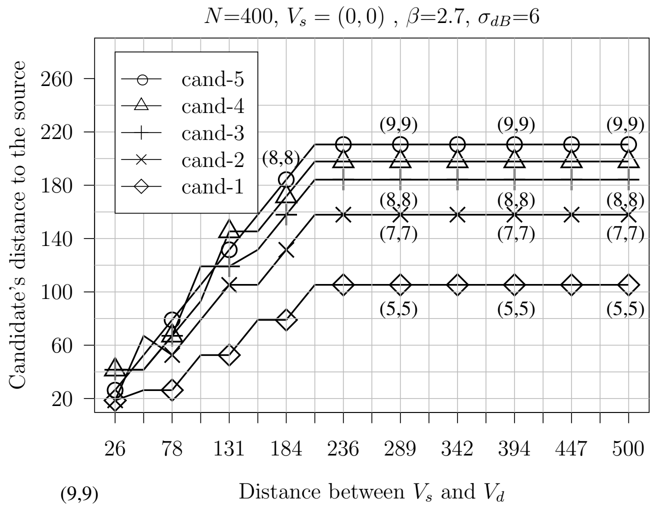


Figure 6.20: Position of candidates to the source in the grid scenario for $V_s = (1,1)$ varying the distance between V_s and V_d.

confirmed that our lower bound is very tight. We have further validated these results by building a dense network and computing the optimal distances of the candidates by an exhaustive optimum search. We have seen that the optimal distances of the candidates converge rapidly to the Maximum Progress Distances as the length of the network increases.

We have investigated the sensitivity to the position of the candidates. We have concluded that choosing the distance of the first two candidates near to their optimal positions, is the most critical in order to minimize the expected number of transmissions. Based on this result we have used the Maximum Progress Distances to provide a rule of thumb for placing the nodes in a static network using OR. Compared to the optimal layout, this method will slightly increase the average number of transmissions while the total number of nodes required is reduced enormously. This can be of practical interest in the design of the back-haul of a mesh network, or in the positioning of the nodes in a sensor network.

Based on MPD, we have proposed a new candidate selection algorithm that we call *Candidate selection based on Maximum Progress Distance* (CMPD). It select the candidates that are close to the optimum position of candidates. We have shown that, the performance of CMPD is very close to the optimum candidate selection algorithm, while CMPD requires much less information and runs much faster.

Finally, we have investigated the maximum performance using OR in the grid scenarios by defining a new metric that we call *Average Distance Progress* (ADP). It represents the average number of meters that a packet progresses towards the destination at each transmission shot. We have concluded that using our rule of thumb for placing the nodes in a grid network using OR, yielding an ADP close to the optimum for all possible destinations.

CHAPTER 7

Multicast Delivery using Opportunistic Routing

7.1 Introduction

Previous researches have shown that OR can significantly reduce the expected number of transmissions to deliver a packet to a particular destination. It is therefore tempting to adapt OR to improve the efficiency of wireless multicast. The main challenge in adapting of OR with multicast is how to share the opportunistic forwarders paths between multiple destinations. In OR algorithms for unicast protocols, since a packet is addressed to only one destination, upon transmitting a packet, only one of the candidates receiving it would actually forward the packet. On the other hand, since there are more than one destination in the multicast protocols, using OR might cause that more than one candidate has to forward the packet to reach all the destinations. Another challenge of using OR in multicast, in contrast to unicast, is that the selected candidates might have to forward the packets toward more than one destination.

This chapter presents a new multicast routing protocol that we call *Multicast Opportunistic Routing Protocol* (MORP). Unlike traditional multicast protocols, there is no designated next-hop forwarder for each destination in our protocol, thus the delivery ratio is maximized by taking advantage of spacial diversity. MORP uses three-way-handshaking approach to transmit the data packet. The basic idea of MORP is as follow: when a source node wants to transmit a data packet, it creates its Candidates Set and include it into the packet. The candidates which successfully receive the packet send an acknowledgment. Then, the sender selects some candidates, and towards which destinations they have to forward the packet. This information is sent to the candidates, which repeat the algorithm until reaching all destinations of the multicast group. Compared with the traditional multicast protocols, our protocol does not build a complete tree or mesh before the transmission starts. Instead, MORP builds a tree on the fly, depending on the candidates that successfully receive the packet in each transmission.

7.2 Multicast Opportunistic Routing Protocol (MORP)

In this section we propose a new multicast routing protocol that we call *Multicast Opportunistic Routing Protocol* (MORP). In the following we first introduce the network model and notation used in the description of MORP, then we describe the protocol and its components.

7.2.1 Network Model

We consider a network of N static wireless nodes, including 1 source node s and a destinations set $\mathcal{D}$ with $k < N$ destinations $\mathcal{D} = \{d_1, d_2, ..., d_k\}$.

Denote $\mathcal{C}^{i,d_j}_{ncand} = \{c_1,\, c_2, \cdots, c_{ncand}\}$ as the Candidates Set (CS) of node i with at most $ncand$ candidates to reach the destination d_j using unicast OR (c_1 the highest priority candidate, and c_{ncand} the least one). In this chapter we have used $ncand = 2$ and 10. From this point forward we shall call $\mathcal{C}^{i,d_j}_{2}$ and $\mathcal{C}^{i,d_j}_{10}$ the "small candidates set" and "large candidates set" of node i to reach the destination d_j, respectively. Each node in the network must compute these CSs using one of the candidates selection algorithms that have been proposed in the literature for unicast OR, like ExOR [12]. All this information (small and large candidates sets) is stored in a *Candidate-Table*.

We define the *Multicast Candidates Set* of the source node s, denoted by $\mathcal{C}^{s,\mathcal{D}}$, as the set of candidates that allows reaching all destinations in $\mathcal{D}$. MORP computes this set as the union of the small Candidates Sets (CSs) of all destinations in $\mathcal{D}$:

$$\mathcal{C}^{s,\mathcal{D}} = \bigcup_{d_j \in \mathcal{D}} \mathcal{C}^{s,d_j}_{2} \tag{7.1}$$

Equation (7.1) uses the small CSs instead of the large Candidates Sets (CSs) in order to maintain the cardinality of $C^{s,\mathcal{D}}$ as small as possible. The reason is that the lower is the cardinality of $C^{s,\mathcal{D}}$, the less nodes are involved in the packet delivery, and thus, the lower is the signaling overhead.

MORP also uses a sequence number to distinguish each data packet created by the multicast source. We shall refer as ID the node identifier used by MORP.

7.2.2 Description of MORP

Each time the source s wants to transmit a packet, the following three-way-handshaking is carried out: First the source inserts its *Multicast Candidates Set* in the data packet and transmits it. The node also stores the packet in a *Message-Cache* table to retransmit it later, if it is necessary.

Each node which successfully receives the data packet checks if its ID is included in the packet's header. If so, it stores the data packet in its buffer and sends back an acknowledgment (ACK), otherwise it simply discards the packet. Note that, a node may receive a packet with the same sequence number from different neighbor nodes. In this case the node does not consider the packet as duplicated, and will process it.

Upon receiving the ACKs from the candidates, the source stores candidates' IDs in an *Ack-Table*. After a period of time (T_{ACK}), the source checks if it received ACKs from enough candidates to reach all destinations in $\mathcal{D}$. If there are not enough ACKs, it retransmits the packet which is stored in its *Message-Cache*. This is done up to a maximum number of retransmissions (MAX_{ReTx}). Then, according to the candidates which successfully received the packet, the sender selects the candidates responsible to forward the packet, and to which destinations. We shall refer these nodes and their destinations as the *Forwarding-Set* and *Bind-Destinations*, respectively, and denote them as $\mathcal{F}$ and $\mathcal{D}_i, i \in \mathcal{F}$. If none of the destinations are reached, the sets $\mathcal{D}_i, i \in \mathcal{F}$ are disjoint and their union is $\mathcal{D}$. Otherwise their union is $D \setminus d_i$, $d_i \in \{\text{Destinations receiving the packet}\}$. Note that, we can consider the source node s as the initial *Forwarding-Set*, with *Bind-Destinations* equal to the multicast destinations set, i.e. $\mathcal{D}_s = \mathcal{D}$. The algorithm to compute the *Forwarding-Set* and *Bind-Destinations* is explained in section 7.2.3.

Then the source s builds a control packet with the *Forwarding-Set* and its *Bind-Destinations*, and broadcast it. We shall refer to this packet as the *Forwarding-Packet*. Each node i that receives the *Forwarding-Packet* and its ID is included in it, must forward the packet following the same rules as the source, except that its *Bind-Destinations*, $\mathcal{D}_i$, indicated in the *Forwarding-Packet* will be used instead of $\mathcal{D}$. This process will be continued until the forwarding nodes directly deliver the packet to their *Bind-Destinations*.

7.2.3 Forwarding Set

As explained in the previous section, upon receiving the candidates' ACKs, the node must select the *Forwarding-Set* and its *Bind-Destinations*. In this section we describe the algorithm used by MORP to select these sets (*Forwarding-Set* and *Bind-Destinations*). We classify the candidates which sent back the acknowledgment and the destinations in the four following sets:

Definition 1 Non-Redundant-Destinations-Set (NRDestSet)*: is the set of destinations reachable by only one candidate. I.e. for each destination* $d_j \in$ NRDestSet *there is only one candidate* c_i *in the* Ack-Table *which is able to reach* d_j*. Additionally, we shall refer to the set of such candidates as the* Non-Redundant-Candidates-Set (NRCandSet)*.*

Definition 2 Redundant-Destinations-Set (RDestSet)*: is the set of destinations* d_k *reachable by at least two candidates, e.g.,* c_i *and* c_j*. We shall refer to the set of such candidates as the* Redundant-Candidates-Set (RCandSet)*. So, if a candidate, e.g.,* c_i*, is removed from the* RCandSet*, then there is, at least, another candidate in* RCandSet *which is able to reach any destination* $d_k \in$ RDestSet*.*

Note that, the destination sets *NRDestSet* and *RDestSet* are disjoint. However, this might not be true for the Candidates Sets *NRCandSet* and *RCandSet*.

To create the non-redundant and redundant sets of candidates and destinations, node s uses its large candidate set, $\mathcal{C}_{10}^{s,d_j}$, $d_j \in \mathcal{D}$, defined in section 7.2.1. Here, the large CS is used instead of the small one in order to increase the chance of reaching all destinations with the minimum number of candidates. For example, it may happen that a candidate c_i does not appear in the small candidates set to reach destination d_j, $c_i \notin C_2^{s,d_j}$, but it is in the small candidate set of another destination d_k, $c_i \in C_2^{s,d_k}$. If c_i receives the packet and appears in the large Candidates Set (CS) of d_j ($c_i \in \mathcal{C}_{10}^{s,d_j}$), then node s can also use c_i to reach destination d_j.

Algorithm 7.1 shows the pseudocode used by a node to compute the *Forwarding-Set* and its *Bind-Destinations*. The general aim of algorithm 7.1 is to select few and good candidates to reach all destinations such that the expected number of transmissions is minimized. The algorithm works as follows: First node s creates the Non-Redundant-Set and Redundant-Set for both candidates and destinations (*NRCandSet*, *NRDestSet*, *RCandSet* and *RDestSet*). For each destination $d_j \in$ *NRDestSet* the algorithm assigns the only possible candidate $c_i \in$ *NRCandSet* (lines 2-6). Recall that *NRDestSet* is the set of destinations d_j reachable by only one candidate. Therefore, for each destination in the *Non-Redundant-Destinations-Set* there is only one possible choice from *Non-Redundant-Candidates-Set* to add to the *Forwarding-Set*.

Then the algorithm chooses the candidates from *RCandSet* to reach the destinations in the *RDestSet*. For these destinations there are multiple choices

Algorithm 7.1: Computation of the *Forwarding-Set* and its *Bind-Destinations* by node s.

Data: $\mathcal{D}_s$, *Bind-Destinations* of node s.
Find *RCandSet, RDestSet, NRCandSet* and *NRDestSet*
forall the $d_j \in$ NRDestSet **do**
  $c \leftarrow c_i \in NRCandSet$ and $c_i \in \mathcal{C}_{10}^{s,d_j}$
  Add c to the *Forwarding-Set*
  Add d_j as the *Bind-Destinations* of c
end
$\mathcal{S} \leftarrow RCandSet$
while *TRUE* **do**
  $C \leftarrow CostFunc(\mathcal{S})$
  $\mathcal{R} \leftarrow \arg\min_{\mathcal{T}=\mathcal{S}\setminus c_i} CostFunc(\mathcal{T})$
  $C' \leftarrow CostFunc(\mathcal{R})$
  if $(C' - C)/C >$ *Threshold* **then**
    break
  else
    $\mathcal{S} \leftarrow \mathcal{R}$
  end
end
forall the $d_j \in$ RDestSet **do**
  $c \leftarrow \arg\min_{c_i \in \mathcal{S} \,\&\, c_i \in \mathcal{C}_{10}^{s,d_j}} ETX(c_i, d_j)$
  Add c to the *Forwarding-Set*
  Add d_j as the *Bind-Destinations* of c
end

of candidates. The optimum choice would minimize the expected number of transmissions to reach all destinations. However, even for a single destination, computing the expected number of transmissions is an equation with a high computational cost (see e.g. [32]). For multiple destinations there has not been proposed any exact equation to compute the expected number of transmissions, and in any case, the computational cost would be extremely high. Additionally, in [16] we have observed that the performance results are not very sensitive to the selection of best candidates. Therefore, MORP builds the *Forwarding-Set* using the following simple cost function as an estimation of the expected number of transmissions to reach all destinations in *RDestSet*, using the candidates in the set $\mathcal{S}$:

$$CostFunc(\mathcal{S}) = \sum_{d_j \in RDestSet} \min_{c_i \in \mathcal{S}} ETX(c_i, d_j) \tag{7.2}$$

where $ETX(c_i, d_j)$ is the expected transmissions count [29] from candidate c_i to the destination d_j. Note that, equation 7.2 gives the expected number of transmissions that would be obtained using unicast delivery to each destination, choosing the candidate in $\mathcal{S}$ that is closest to each destination in *RDestSet*. Therefore, this will be an upper-bound to the expected number of transmissions obtained using OR.

Lines 8-22 of algorithm 7.1 show the selection of the candidates for the destinations in *RDestSet*. In each iteration of the while-loop, the algorithm runs an exhaustive search over all possible subsets of the set $\mathcal{S}$ by removing one candidate. The algorithm uses equation (7.2) to choose the subset having the minimum cost (line 10). If the difference between the cost of new set (C') and the previous one (C) to reach the *Redundant-Destinations-Set* is not very large (e.g., Threshold=1), the algorithm will continue with the new set to eliminate more candidates.

The output of the while-loop of lines 8-17 is a reduced set of candidates able to reach all destinations in *RDestSet*. In order to assign the *Bind-Destinations* to these candidates, it is used the minimum ETX (lines 18-22).

7.2.4 Candidate Coordination and Data forwarding

After running algorithm 7.1, the source puts the *Forwarding-Set* and its *Bind-Destinations* in the *Forwarding-Packet* and broadcasts it. Each node i receiving the *Forwarding-Packet* having its ID in the *Forwarding-Set* will forward the data packet stored in its buffer to its *Bind-Destinations*. The candidates with IDs not included in the *Forwarding-Packet* will simply discard the packet. This process will be continued until the forwarding nodes directly deliver the data packet to their *Bind-Destinations*.

7.2.5 Data Structures

This section summarizes the data structures that nodes running MORP are required to maintain:

- *Candidate-Table*: It is created before the transmission starts and stores the CSs to reach each destination. Each entry in the *Candidate-Table* is the destination ID, the multicast group address and the list of candidates to reach the destination. Recall that we have used two different maximum number of candidates to form the small and large candidates sets. Therefore, in each node there are two *Candidate-Tables*.

- *Ack-Table*: It stores the ID of the candidates from which ACK packets have been received. Each entry of this table consists of the ID of the candidate, the sequence number of the packet which has been received and acknowledged and the multicast group address of the packet.

- *Bind-Destinations-Table*: When a node forwards the data packet, it stores its *Bind-Destinations*. This information will be used when the ACKs are received and the node wants to decide to which destination each candidate should forward the packet. Indeed, *Bind-Destinations-Table* of node i stores its *Bind-Destinations*, $\mathcal{D}_i$, for each packet, until the corresponding *Forwarding-Packet* is sent.

- *Message-Cache*: The *Message-Cache* is maintained by each node to prevent duplicated packets. It is also used to retransmit a packet which is not acknowledged by enough candidates. When a node forwards a data packet, it stores the source ID, the multicast group address and the sequence number of the packet. An age timer is used to remove old entries.

7.2.6 An Example of MORP

We finish the description of MORP by means of a simple example. Consider the network topology shown in figure 7.1. Assume that the delivery probability is a function of the distance between the nodes shown in the figure. The source node is s and the destinations set is $\mathcal{D} = \{d_1, d_2, d_3, d_4\}$. An unicast OR candidates selection algorithm (e.g. ExOR) is used by all nodes to compute the small and large Candidates Sets. Table 7.1 shows these sets for node s. In each row, candidates are ordered in descending priority from left to right.

When s wants to send a packet, it puts its multicast candidates set (see equation (7.1)), which is $\mathcal{C}^{s,\mathcal{D}} = \{a, b, c, d_3, d_4, f\}$ in the data packet and sends it. The source sets the timer T_{ACK} and waits for the ACKs from the candidates that have received the packet successfully. Assume that only the candidates a, b and d_3 receive the data and send back an ACK to the source.

When s receives ACK from a, b and d_3, it stores their ID in its *Ack-Table*. After T_{ACK} expires in node s, it runs the algorithm 7.1 to find the candidates which should forward the packet. Since one destination, d_3, has received the packet, node s looks for the candidates to reach destinations d_1, d_2 and d_4. First, it finds the non-redundant and redundant sets of candidates and destinations. As we mentioned in section 7.2.3, the algorithm 7.1 uses the large candidates set to create the non-redundant and redundant sets. The only candidate which has received the packet and can reach the destination d_4 is d_3 (see large Candidates Set in table 7.1). Therefore the *Non-Redundant-Destinations-Set* (*NRDestSet*) is $\{d_4\}$, and the candidate d_3 will be added to the *Forwarding-Set* with destination d_4 as its *Bind-Destination*.

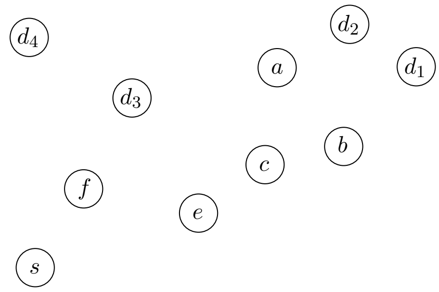


Figure 7.1: Example of MORP.

Table 7.1: Small and large candidates sets of s

(a) Small and large candidates sets

dest.	small		large				
d_1	b	c	a	b	c	e	f
d_2	b	a	b	a	c	e	f
d_3	d_3	f	d_3	c	e	f	
d_4	d_4	f	d_4	f	d_3	e	

(b) ETX Table

node	d_1	d_2
a	4.3	4.1
b	4.8	3.8

The benefit of considering the large Candidates Set instead of small candidates set becomes apparent for destination d_4. If the algorithm would have just considered the small Candidates Sets, since none of the candidates d_4 and f received the packet, the destination d_4 would be considered unreachable, and s would retransmit the data packet.

To reach destinations d_1 and d_2 there are two candidates a and b which received the data packet. Therefore, the *Redundant-Destinations-Set* (*RDest-Set*) and *Redundant-Candidates-Set* (*RCandSet*) are $\{d_1, d_2\}$ and $\{a, b\}$, respectively.

In the first iteration of the while-loop of algorithm 7.1, the cost of reaching $RDestSet = \{d_1, d_2\}$ using $\mathcal{S} = \{a, b\}$ is estimated as: $C = ETX(a, d_1) + ETX(b, d_2) = 8.1$ (see equation 7.2). Then it reduces the number of candidates in the *RCandSet* and uses formula 7.2 again to find the set with the minimum cost (line 10 in algorithm 7.1). This is given by the set $\mathcal{R} = \{a\}$ with cost $C' = 8.4$. Since the relative difference between new cost and the previous one ($C = 8.1$) is small, the algorithm takes the new set $\mathcal{S} = \{a\}$. Then the while-loop finishes.

Thus, the final *Forwarding-Set* is $\mathcal{F} = \{a, d_3\}$ with *Bind-Destinations* $\mathcal{D}_a = \{d_1, d_2\}$ and $\mathcal{D}_{d_3} = \{d_4\}$. Node s will put these sets in the *Forwarding-Packet* and send it. Upon receiving the *Forwarding-Packet*, a and d_3 will know that they must forward the packet to $\{d_1, d_2\}$ and $\{d_4\}$, respectively, and will repeat the forwarding process for these destinations.

Note that, as the data packets approach the destinations, the size of the *Bind-Destinations* sets will be decreased or remain unchanged. Thus, it is like MORP builds a tree on the fly, depending on the candidates that successfully receive the data packet in each transmission.

7.3 Implementation of MORP

As explained in section 7.2.1, MORP computes the Candidates Sets using one of the candidates selection algorithms that have been proposed in the literature for unicast OR. To do so, the nodes need to be aware of the network topology and the delivery probability of the wireless links. This information can be gathered in different ways. One possible implementation could be the method described in ExOR [11], where nodes collects measurements and send them to a central server which distributes the required information to all nodes. Distributed algorithms similar to the topology discovery mechanism used by OLSR [21] would also be possible.

MORP could be implemented at link or network layer. A link layer implementation would permit the design of an efficient signaling protocol. For instance, the three-way-handshaking used by MORP (see section 7.2.2) could be implemented using a modified 802.11 MAC as shown in figure 7.2. In this figure the *Multicast Candidates Set* consists of the nodes $\{a, b, c\}$. The candidates send back an ACK which is immediately followed by the *Forwarding-Packet*. A similar proposal to send the ACKs was proposed in [12].

A network layer implementation would allow using current off-the-shelf 802.11 network cards. In this case ACKs and *Forwarding-Packets* would be sent using unicast 802.11 data frames, thus, increasing the overhead and delays of the three-way-handshaking used by MORP. Nevertheless, for the sake of investigating the feasibility to implement MORP with current hardware, in the numerical results presented in section 7.7 we have assumed a network layer implementation using standard 802.11 cards.

7.4 Summary of the ODMRP Protocol

The On-Demand Multicast Routing Protocol is a mesh based multicast protocol where group membership and multicast routes are established and updated by the source on demand [56, 8, 64]. It introduces the concept of *forwarding groups*. A multicast source will transmit packets to the destinations via the *forwarding group*. The *forwarding group* is a set of nodes in charge of forwarding multicast packets. When a multicast source has data packets to send, but there is no route to the multicast group, it broadcasts a *Join-Query* control packet to the entire network. This control packet is periodically sent every *REFRESH INTERVAL*, e.g., every 3 seconds to refresh

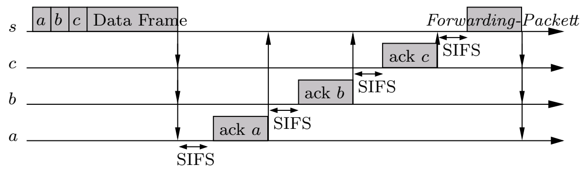


Figure 7.2: Three-way-handshaking of MORP using a modified 802.11 MAC.

the membership information and update routes. When a node receives a non-duplicate *Join-Query*, it stores the upstream node ID and rebroadcasts the packet.

When the *Join-Query* packet reaches a multicast destination, it creates and broadcasts a *Join-Table* to its neighbors. This packet is forwarded along the shortest path back to the multicast source that originated the *Join-Query*. When a node receives a *Join-Table*, it checks if its ID matches with the ID of the next node of one of the entries in the *Join-Table*. If it matches, the node realizes that it is on the path to the source, and thus, is part of *forwarding group*. Then it sets the forwarding flag *FG-Flag* and broadcasts its own *Join-Table*. The *Join-Table* is propagated by each *forwarding group* member until it reaches the multicast source. The *FG-Flag* of forwarding nodes expires after a multiple of the interval between successive *Join-Query* floods.

When a node receives a data packet, it forwards the packet only when it is non-duplicated, and the *FG-Flag* for the multicast group of this node has not expired. Note that a multicast destination can also be a *forwarding group* node if it is on the path between a multicast source and another destination.

These procedures allow for redundant forwarding to each receiver, increasing the packet delivery ratio of the protocol: if a packet is dropped on one path as a result of collision or a link break, the receiver can receive it along another path. The benefit of this redundancy comes at the cost of additional overhead and additional load on the network.

7.5 Summary of the ADMR Protocol

Adaptive Demand-Driven Multicast Routing (ADMR) [37, 75] protocol is an on demand protocol like ODMRP. It creates a source-based forwarding tree connecting the source with the destinations of the multicast group. Each multicast packet is dynamically forwarded from the source along the shortest delay path through the tree to the destinations of the multicast group. In ADMR, packet forwarding is based on two types of flooding: *tree flood* and *network flood*. In the tree flooding the packets are constrained to the nodes in the multicast tree, while network flooding is the flooding among all nodes in the network. Note that the tree flooding in ADMR is similar to the *forwarding group* concept in ODMRP.

When a source has packet to send, but no routing state yet exists for this sender and group, it floods a packet called *Source Information* to all nodes in the network using *network flood*. Each node in the network that receives this packet, forwards it unless it has already forwarded a copy of it. In addition, the node records in its *Node-Table* the ID of the node from which it received the packet. When this packet reaches a multicast destination, it creates a reply packet called *Receiver Join* packet back toward the source. The *Receiver Join* packet is sent automatically along the shortest path traversed by the flood back towards the source. Each node that forwards the *Receiver Join* creates a forwarding entry in its *Membership-Table*, indicating that it is a forwarder for this sender and group.

When a destination wants to join a group, the node checks its *Membership-Table* to determine if it is already connected to the group. If it is not, it sends a *Multicast Solicitation* packet as a network flood. Each node in the network forwards the *Multicast Solicitation*. In this case, if a node receiving the *Multicast Solicitation* already belongs to the group, it will unicast the *Multicast Solicitation* only to the previous hop address. Therefore, the packet follows the multicast tree towards the source, speeding up and decreasing the overhead of the receiver join. When the source receives the *Multicast Solicitation* packet, the source replies to the *Multicast Solicitation* to advertise to the destination its existence as a sender for the group.

ADMR sends *Keep-Alive* messages to maintain the existing forwarding state for the multicast tree. The absence of data packets and *Keep-Alive* messages within a certain period of time is an indication of forwarding tree disconnection. Firstly, a local repair procedure is performed to reconnect the tree; if it fails a global reconnect procedure is used.

7.6 Evaluation Methodology

To evaluate the performance of MORP we compare it with ODMRP and ADMR, which have been shown to perform well in previous studies. The simulation code has been implemented within the Global Mobile Simulation (GloMoSim) library [91]. The number of multicast groups and sources is set to one in all scenarios. Members join the multicast group at the start of the simulation and remain throughout the simulation. The simulation field consists of a square with diagonal equal to 500 m. We have run simulations varying the number of nodes in the range $20 \leq N \leq 100$. One node is the source, and it is located in a square corner, the others are placed randomly inside the square. The destinations of the multicast group are chosen randomly among the nodes inside the square. Each simulation runs for 300 seconds of simulation time. Each point in our performance graphs represents the average of 20 simulation runs. For this number of runs we obtained reasonably small confidence intervals. The IEEE 802.11 Distributed Coordination Function was used as the medium access control protocol.

The multicast application-layer source in our simulations generates Constant Bit Rate (CBR) traffic with 4 packet per second and 64 bytes of payload. This sending rate was chosen to challenge the routing protocols' abilities to successfully deliver data packets in a wireless network. It was not chosen to represent any particular or class of applications, although it could be considered to abstractly model a very simple broadcast audio distribution application [37].

For a more realistic simulation of an 802.11 network, we have considered that packets can be transmitted at two different transmission rates: a data rate of 11 Mbps, and a basic rate of 2 Mbps. Packets transmitted at the data rate are subject to a shadowing propagation model, which introduces random transmission losses. Packets transmitted at the basic rate does not suffer transmission losses. We have assumed that data packets are always transmitted at data rate. However, the protocols can transmit signaling packets using the basic rate to prevent losses due to impairments of the radio channel. More specifically, we have assumed that in MORP, all signaling packets (i.e. ACKs and *Forwarding-Packets*) are transmitted at the basic rate. In ODMRP, *Join-Query* packets are sent at the data rate. This is because these packets are used to build the routing tables, and thus, they need to have the same transmission properties over the wireless links as those of data packets. For the same reason, *Source Information* and *Multicast Solicitation* packets are sent at the data rate in ADMR, although *Receiver Join* packets are sent at the basic rate.

We have assumed that in MORP nodes are aware of the network topology and the delivery probability of the wireless links, due to the shadowing propagation model of the radio channel. MORP uses this information and applies ExOR [12] to compute the Candidates Sets.

We have set the shadowing parameters to the default values used by the GloMoSim, given in table B.3. The value of path loss exponent and deviation value are set to $\beta = 2.7$ and $\sigma_{dB} = 6$ dBs, respectively. We have assumed that a link between any two nodes exists only if the delivery probability between them is greater (or equal) than $min.dp = 0.1$

7.6.1 Protocols Parameters

We have evaluated two different variations of the ODMRP parameters. The "ODMRP-3-9" variation represents ODMRP using the parameter values chosen by ODMRP's designers: 3 seconds for the *Join-Query* flooding interval (*REFRESH INTERVAL*=3 seconds) and a forwarding state lifetime of 3 times of this interval (a total of 9 seconds). The "ODMRP-3-3.3" variation reduces the forwarding state lifetime to 1.1 times of the *Join-Query* flooding interval; it shows the effect of reducing the forwarding redundancy of ODMRP (see section 7.4). For ADMR parameters we have used the default values which are used in [37]: 30 seconds for the periodic data flood interval and 2 missing packets to trigger disconnection detection procedure.

In MORP we have used ExOR [12] as the candidate selection algorithm, fixing the maximum number of candidates for the small and large candidates sets to $ncand = 2$ and 10, respectively. In our protocol, we have used 12 milliseconds to receive ACKs from the Candidates Set ($T_{ACK} = 12$ ms). The legend MORP-ExOR(n) in the following sections refers to MORP with $MAX_{ReTx} = n$.

7.6.2 Performance Metrics

We have evaluated all protocols as a function of number of nodes in the network, and number of destinations of the multicast group. The measures of interest are:

- Packet delivery ratio: The ratio of the number of data packets delivered to the destinations versus the number of data packets supposed to be received.
- Multicast group reachability: Let X be a random variable equal to the number of destinations of the multicast group receiving a given data packet. We have computed the empirical complementary cumulative distribution function (EC-CDF) of X. This gives a measure of the number of destinations of the multicast group receiving data packets.
- Forwarding cost: Total number of data packets transmitted by all nodes in the network over the total number of data packets sent by the source. This metric represents the delivery cost in terms of transmissions of each multicast packet. Note that, to make the comparisons more clear, in this metric we take as the reference originated instead of delivered packets.
- Normalized packet overhead: The total number of all data and control packets transmitted by any node in the network (either originated or forwarded), divided by the total number of all data packets received across all multicast receivers.
- End-to-End delay: Average end-to-end delay of all data packets received by the destinations.

7.7 Numerical Results

7.7.1 Packet Delivery Ratio

One important parameter of MORP is the maximum number of retransmissions (MAX_{ReTx}). Recall that if the forwarder does not receive enough ACKs from its candidates, it retransmits the data packet up to MAX_{ReTx} times before it is forwarded. To see the effect of this parameter, figures 7.3 and 7.4 depict the delivery ratio varying MAX_{ReTx} from 1 to 5. The two curves correspond to a number of destinations of the multicast group equal to 2 and 10. The legend MORP-ExOR-Dest(n) in these two figures refers to MORP with number of destinations equal to n. These figures have been obtained with a total number of nodes equal to $N = 20$ (figure 7.3) and $N = 100$ (figure 7.4). In the rest of this chapter we shall refer to the scenarios having these number of nodes as *sparse* and *dense* networks, respectively. The 95% confidence intervals have been added in figure 7.3. It can be observed that the intervals are relatively small, and the same was obtained for the other figures. So, for the sake of clarity, confidence intervals are not depicted in the rest of the figures.

As expected, figures 7.3 and 7.4 show that the higher is MAX_{ReTx}, the higher is the delivery ratio. Additionally, we observe that the maximum delivery ratio improvement is obtained when MAX_{ReTx} is increased from 1 to 2. For instance, in the sparse network (figure 7.3), we can see that the delivery ratio of MORP for 2 destinations with $MAX_{ReTx} = 1$ is about 74%, while it improves to 94% with $MAX_{ReTx} = 2$ (improvement around 27%). Increasing from $MAX_{ReTx} = 2$ to 3 yields a delivery ratio of 98% (improvement around 4%).

Comparing figures 7.3 and 7.4 we can see that packet delivery ratio is always higher in a dense than in a sparse network. This comes from the fact that in the dense network, MORP uses better candidates than in the sparse network. For instance, the packet delivery ratio of MORP in a sparse network with 2 destinations and $MAX_{ReTx} = 1$ is about 74%, while it increases to 90% in a dense network.

Figure 7.5 shows the packet delivery ratio of MORP in comparison with ODMRP and ADMR. The curves are obtained varying the number of nodes from 20 to 100. In this figure the number of destinations has been set to 5 (NumDest = 5). The results of MORP are shown for MAX_{ReTx} is set to 1 and 2 (MORP-ExOR(1) and MORP-ExOR(2), respectively).

As we can see in figure 7.5, MORP with any MAX_{ReTx} outperforms both ODMRP (ODMRP-3-9 and ODMRP-3-3.3) and ADMR. For instance, even with $MAX_{ReTx} = 1$, MORP has about 92% packet delivery ratio, while

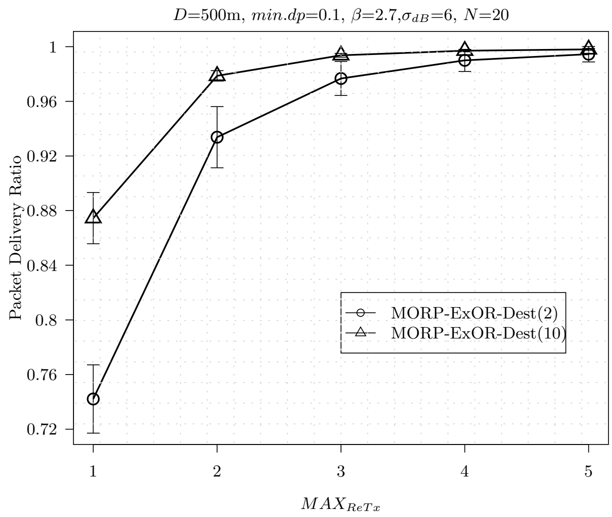


Figure 7.3: Packet delivery ratio of MORP in a sparse network as a function of MAX_{ReTx}.

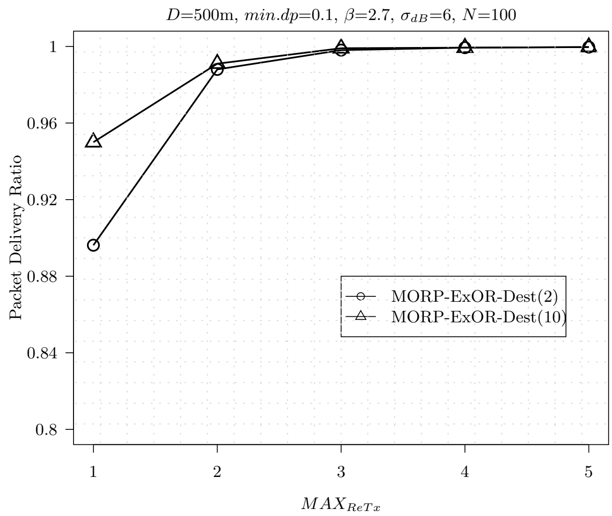


Figure 7.4: Packet delivery ratio of MORP in a dense network as a function of MAX_{ReTx}.

ODMRP-3-9, ODMRP3-3.3 and ADMR have about 83%, 48% and 89%, respectively. This comes from the fact that the construction of the routes in ODMRP and ADMR are subject to the random losses that may have the *Join-Query* packets in ODMRP and the *Source Information* and *Multicast Solicitation* packets in ADMR. On the other hand, MORP takes routing decisions "on the fly" (when the forwarding nodes are chosen), and thus, adapts faster to random losses.

Figure 7.5 shows that the packet delivery ratio of ODMRP-3-3.3 is signifi-

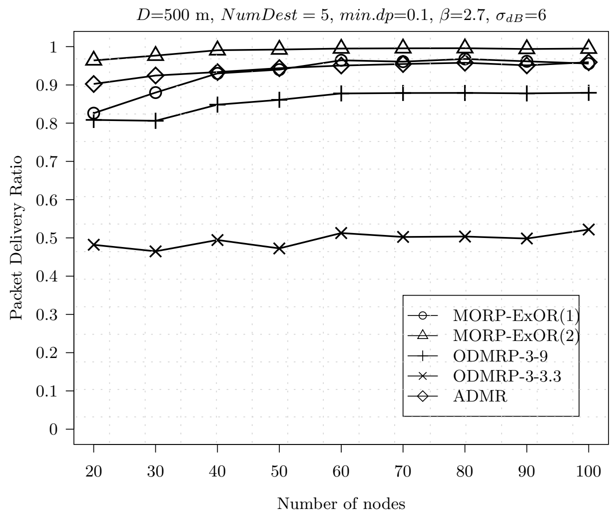


Figure 7.5: Packet delivery ratio for 5 destinations as a function of number of nodes in the network.

cantly lower than ODMRP-3.9 (about 35%). As we described in section 7.4, ODMRP creates forwarding groups within nodes in the network that expires after a fixed timeout. In ODMRP-3-3.3 the forwarding state timeout (3.3 seconds) is shorter than in ODMRP-3-9 (9 seconds). Therefore, ODMRP-3-3.3 has less number of nodes in the forwarding group than in ODMRP-3-9, resulting in a lower delivery ratio.

Figures 7.6 and 7.7 show the delivery ratio of the protocols under study, varying the number of destinations. In these figures we can see that all protocols achieve a higher delivery ratio in the dense scenario than in the sparse network. In the sparse network (figure 7.6) the packet delivery ratio of MORP with any maximum number of retransmissions (MAX_{ReTx}) outperforms both variations of ODMRP. The same figure shows that ADMR has a delivery ratio about 6% better than MORP-ExOR(1). However, in section 7.7.3 we will see that this small improvement is at cost of having much more data transmissions than MORP. Nevertheless, figure 7.6 shows that MORP outperforms ADMR when MAX_{ReTx} is increased to 2 or 3.

For a dense network, we can see in figure 7.7 that the packet delivery ratio of MORP with any MAX_{ReTx} is higher than ODMRP and ADMR. This comes from the fact that in a dense network, MORP can choose better candidates to forward the packets. For instance, the packet delivery ratio of MORP-ExOR(1) and MORP-ExOR(2) in the dense network is about 94% and 98%, respectively. Although the delivery ratio of ADMR in the case of a dense network is close to MORP-ExOR(1), we will see in section 7.7.3 that it is at cost of a large amount of data transmissions.

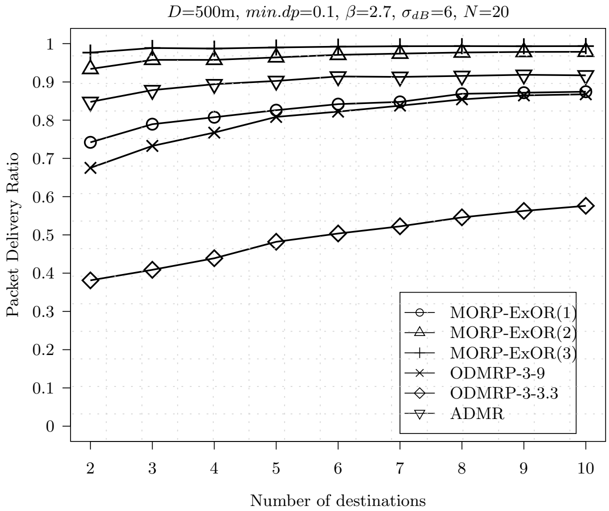


Figure 7.6: Packet delivery ratio in a sparse network as a function of number of destinations.

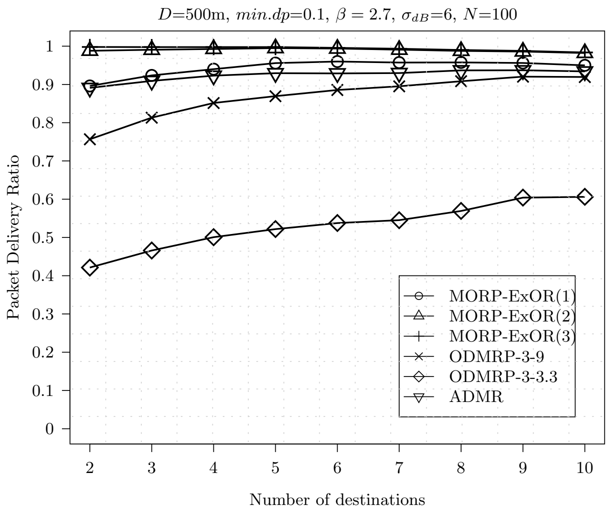


Figure 7.7: Packet delivery ratio in a dense network as a function of number of destinations.

7.7.2 Multicast Group Reachability

In this section we investigate the number of destinations of the multicast group receiving data packets. To do so, we have calculated its empirical complementary cumulative distribution function (EC-CDF). This is shown in figures 7.8 and 7.9 in a scenario with 10 destinations for the sparse and dense networks (with 20 and 100 nodes), respectively.

Figure 7.8 shows that ODMRP-3-3.3 performs much worst than the others: the probability of reaching the multicast group decreases sharply with in-

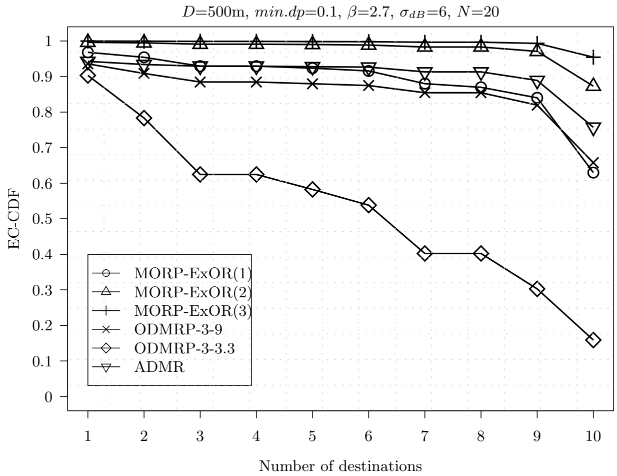


Figure 7.8: Distribution of received packets for 10 destinations and 20 nodes.

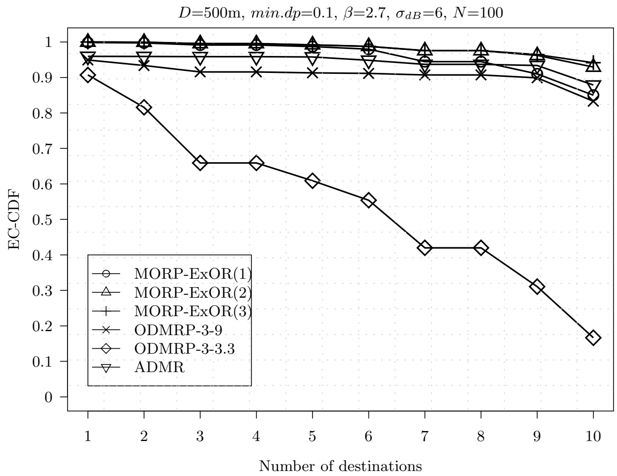


Figure 7.9: Distribution of received packets for 10 destinations and 100 nodes.

creasing the number of destinations, and less than 20% of the packets reach all destinations.

Regarding the other protocols, figure 7.8 shows that in the sparse scenario around 5% of packets do not reach any destination in ODMRP-3-9, ADMR and MORP-ExOR(1). Then, the curves are approximately flat and decrease at the end (specially for 10 destinations). This behavior is explained by the simulation topology. Recall that the simulation field is a square with the source in one corner and the other nodes distributed randomly. This distribution of nodes favors that when the source reach the first next hop, it reaches also most of the destinations with high probability.

Figure 7.8 also shows that the difference between MORP-ExOR(2) and

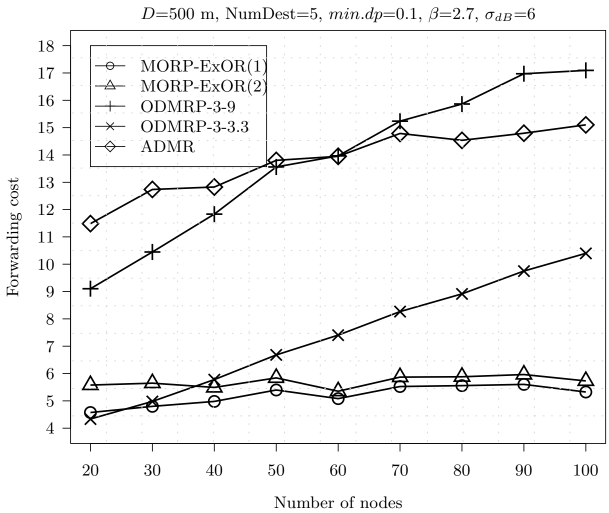


Figure 7.10: Forwarding cost for 5 destinations as a function of number of nodes in the network.

MORP-ExOR(3) is small. In both cases almost 100% of packets reach up to 9 destinations, and about 90% of packets are delivered to 10 destinations in MORP-ExOR(2), and 95% in MORP-ExOR(3).

Finally, the same conclusions can be derived from the dense network scenario shown in figure 7.9. Of course, the group reachability increases, due to the higher proximity between the nodes.

7.7.3 Forwarding Cost

In this section we compare the forwarding cost of MORP, ODMRP and ADMR. Recall that we have defined forwarding cost as the number of data packets transmitted by all nodes in the network over the total number of data packets sent by the source.

Figure 7.10 shows the forwarding cost of the protocols varying the number of nodes from 20 to 100 in the case of 5 destinations. The results of MORP, like in figure 7.5, are obtained for $MAX_{ReTx} = 1$ and 2.

Figure 7.10 shows that the forwarding cost of MORP outperforms ODMRP-3-9 and ADMR. In fact, the forwarding cost of both variations of ODMRP and ADMR is rather sensitive to the number of nodes, while in MORP is not. This is because using Opportunistic Routing, as in MORP, only some useful nodes are selected as candidates to forward the packets. Figure 7.10 also shows that there is only a slight increase of the forwarding cost when MORP increases MAX_{ReTx} from 1 to 2.

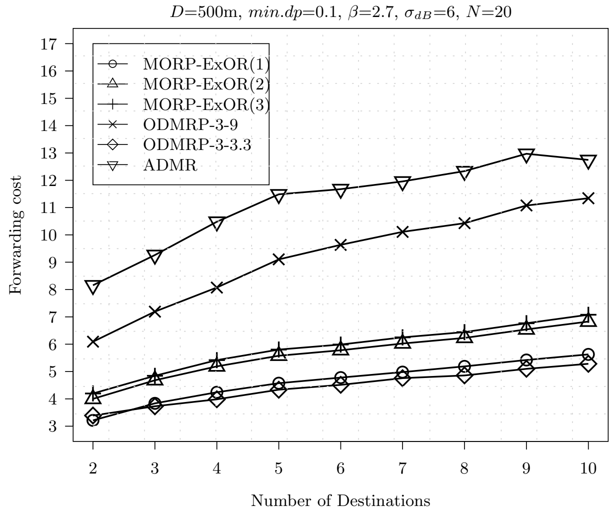


Figure 7.11: Forwarding cost in a sparse network as a function of number of destinations.

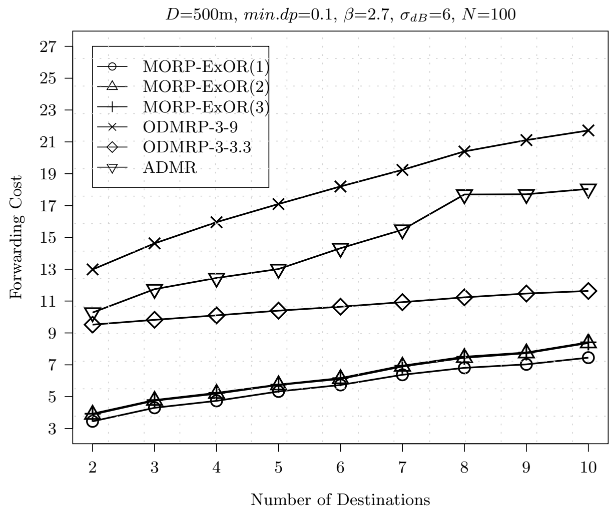


Figure 7.12: Forwarding cost in a dense network as a function of number of destinations.

As described in section 7.4, ODMRP periodically floods a data packet together with a *Join-Query* packet. I.e., it piggybacks the *Join-Query* information on the data packet periodically to update the routes. Because of this, new nodes may become forwarders, while forwarders created during a previous periodic flood still have a set forwarding flag. Consequently, redundant routes are created, and the number of data transmissions increases with increasing the network density. In fact, the forwarding cost of both variations of ODMRP is dominated by the flooding packets and forwarding state timeout. Since in ODMRP-3-9 the forwarding state timeout (9 seconds) is longer than in ODMRP-3-3.3, there are more nodes with the forwarding flag set in ODMRP-3-9 than in ODMRP-3-3.3. Therefore, the forwarding cost of ODMRP-3-9 is much higher than in ODMRP-3-3.3.

The construction of the routes in ADMR is subject to the random losses that may have the *Source Information* and *Multicast Solicitation* packets. Recall that the absence of some data packets in ADMR is an indication of forwarding tree disconnection and the local or global repair procedure is triggered to repair the path. As we mentioned in section 7.6.1, we used 2 missing data packets to trigger disconnection detection. When a node on the tree does not receive 2 consecutive data packets, it starts repairing algorithm, which may add new nodes to the tree. This is exacerbated with increasing the density of the network, thus, increasing the forwarding cost.

Figure 7.10 shows that for $N = 20$ ODMRP-3-3.3 is slightly better than MORP. However, recall from figure 7.5 that in this scenario the delivery ratio of ODMRP-3-3.3 is much lower than in MORP.

In section 7.7.1 we have shown that MORP outperforms ODMRP and ADMR in terms of packet delivery ratio for different number of destinations. Figures 7.11 and 7.12 give the forwarding cost for the same scenarios. Figure 7.11 shows that the forwarding cost of MORP in the sparse network, and with any MAX_{ReTx}, is much lower than ADMR and ODMRP-3-9. The figure shows that only ODMRP-3-3.3 is slightly better than MORP. However, as we showed in figure 7.6, the delivery ratio of ODMRP-3-3.3 is much lower than MORP.

Regarding ADMR, figure 7.6 in section 7.7.1 showed that the delivery ratio in a sparse network is slightly better than MORP-ExOR(1). However, we observe in figure 7.11 that, this is at cost of ADMR having a forwarding cost of about 3 times higher than MORP-ExOR(1).

Figures 7.11 and 7.12 confirm, as in figure 7.10, that the forwarding cost of ODMRP and ADMR is much higher in a dense than in a sparse network, while MORP is rather insensitive to network density. Nevertheless, it can be observed that the forwarding cost increment in MORP is slightly higher in a dense than in a sparse network. This may be counterintuitive, since MORP can choose better candidates in a dense network. However, since MORP looks for candidates to reach all destinations, in a dense network MORP chooses more candidates, thus, increasing the forwarding cost. However, recall from figures 7.6 and 7.7, this slightly higher forwarding cost in a dense network is rewarded with a significantly higher packet delivery ratio.

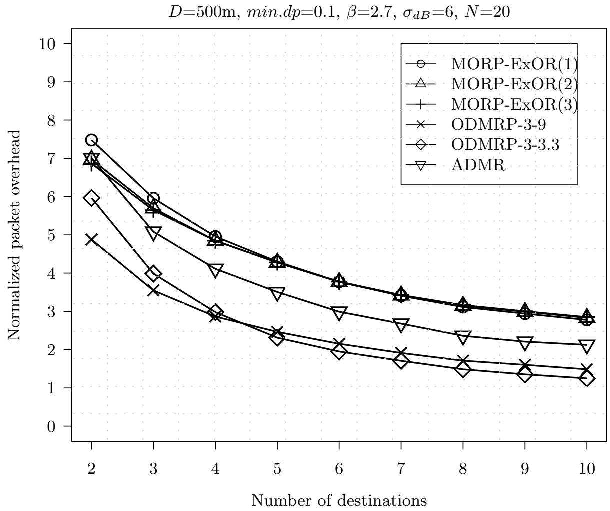


Figure 7.13: Packet overhead in a sparse network as a function of number of destinations.

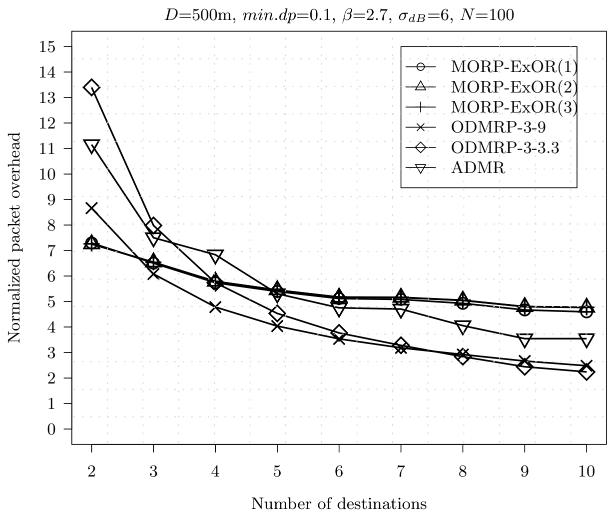


Figure 7.14: Packet overhead in a dense network as a function of number of destinations.

7.7.4 Packet Overhead

In this section we compare the packet overhead of the protocols under study. Recall that we compute the packet overhead as the ratio of data and control packets transmitted by any node to deliver data packets. We count as the control packets for ODMRP the *Join-Query* and *Join-Table*, for ADMR the *Source Information*, *Receiver Join*, *Multicast Solicitation* and packets related to the repair process. The *ForwardingPacket* and ACK packets in

MORP are counted as the control packets.

Figures 7.13 and 7.14 show the packet overhead of all protocols varying the number of destinations for a sparse and dense network, respectively. Figure 7.13 shows that in a sparse network MORP has a higher packet overhead than ADMR and ODMRP. This is due to the ACKs sent by the candidates in MORP. Note that, in this comparison we are giving the same weight to data and control packets. Recall from figures 7.11 and 7.12 that MORP performs better than the other protocols in terms of forwarding cost, where only data packets are considered. Thus, in case of sending data packets with large payload, the overhead considered in this section would be a pessimistic comparison with respect to MORP. Furthermore, as explained in section 7.3, the overhead of control packets in MORP could be reduced implementing the three-way handshaking used by MORP at MAC layer.

7.7.5 End-To-End Delay

Figures 7.15 and 7.16 show the average end-to-end delay for different number of destinations for a sparse and dense network, respectively.

These figures show that the end-to-end delay in MORP is higher than in ODMRP and ADMR. This is because each time a node transmits a data packet in MORP, it waits for the ACKs sent by the candidates during T_{ACK} =12 ms. However, recall from section 7.3 that implementing the three-way handshaking used by MORP at MAC layer could reduce this delay, significantly.

Comparing figures 7.15 and 7.16 we can see that the decrease of the end-to-end delay in a dense network for MORP is much more noticeable than in the other protocols. This is because using OR in a dense network, the probability of reaching a candidate close to the destination increases, thus, reducing the average number of end-to-end hops. It can also be observed that the difference between the average delay using MORP with $MAX_{ReTx} = 1$ and $MAX_{ReTx} > 1$ is higher in a sparse network than in a dense network. This is because in the dense network the probability that the sending node receives enough ACKs to reach all destinations is higher than in the sparse network. Therefore, MORP requires less retransmissions of data packets in a dense network.

7.8 Conclusion

In this chapter we have investigated how Opportunistic Routing (OR) can be exploited to implement a multicast protocol for wireless mesh networks. This has been done by proposing a new protocol called *Multicast Opportunistic Routing Protocol*, MORP. MORP uses a three-way-handshaking approach

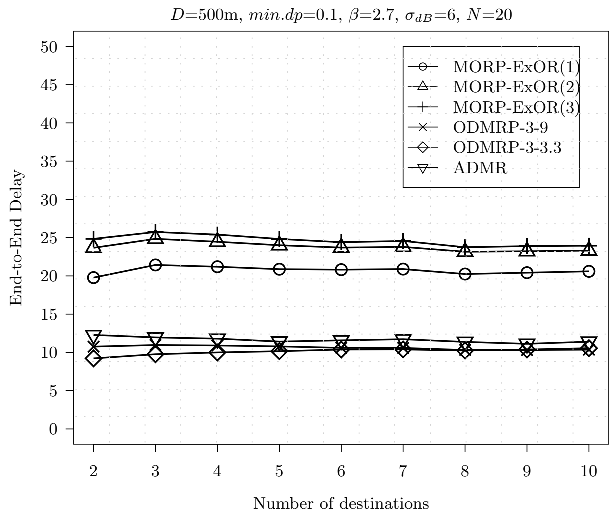


Figure 7.15: End-to-end delay in a sparse network as a function of number of destinations.

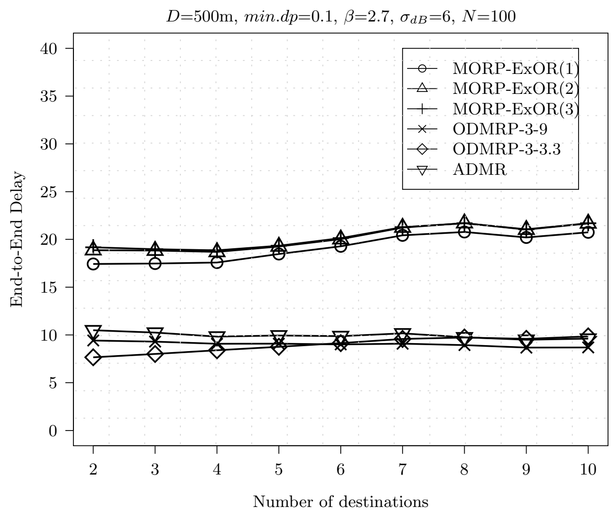


Figure 7.16: End-to-end delay in a dense network as a function of number of destinations.

where candidates send ACKs to the sender node upon successfully receiving data packets. Then, the sender node partitions the set of destinations and assigns each subset to the most appropriate candidate. This information is sent to the candidates which repeat the same approach for each subset of destinations. Compared with traditional multicast protocols, MORP does not build a complete tree or mesh before the transmission starts. Instead, MORP builds a tree on the fly, depending on the candidates that successfully receive the packet in each transmission.

We have compared MORP with other relevant multicast protocols that have been proposed in the literature: On Demand Multicast Routing Protocol (ODMRP) and Adaptive Demand-Driven Multicast Routing (ADMR). The comparison is done taking into consideration realistic simulations using 802.11 standard MAC layer. A lossy shadowing propagation model has been used for the radio channel.

Simulation results show that in most cases ODMRP and ADMR have a number of data transmissions much higher than in MORP, while the achieved delivery ratio is not as good as in MORP. Although, the signaling overhead and end-to-end delay in MORP is a bit higher than in ODMRP and ADMR, the overhead of control packets could be reduced significantly by implementing the three-way-handshaking used by MORP at MAC layer.

We conclude that MORP outperforms traditional ODMRP and ADMR multicast protocols, reducing the number of data transmissions and increasing the data delivery ratio. Hence, using OR can be a useful technique to implement reliable multicast protocols in wireless mesh networks.

CHAPTER 8

Conclusions and Future Research Directions

We close this dissertation with an enumeration of some of our important results, some avenues of future work, and a discussion of the outlook for Opportunistic Routing.

8.1 Achieved Results

The essential idea of Opportunistic Routing is to exploit the broadcast nature and space diversity provided by the wireless medium. The source can use multiple potential paths to deliver the packets to the destination. By having multiple forwarding candidates, the successful rate of each transmission can be much improved.

In this dissertation we described the meaning of Opportunistic Routing (OR) in Chapter 1. Then, we surveyed the main research contributions of this topic in Chapter 2 by classifying different research areas: routing metrics, candidate selection, candidate coordination, geographic OR and multicast OR. We summarize the results of other chapters below.

- Chapter 3. In this chapter, we proposed a Discrete Time Markov Chain (DTMC) model that can be used to evaluate OR mechanisms in wireless networks in terms of expected, variance and probability of number of transmissions from the source to the destination. The only ingredients needs to build the transition probability matrix are the candidates of each node, and the delivery probabilities to reach them. As a consequence, the proposed model can be applied independently of the candidate selection algorithm that is employed. The model leads to a discrete phase-type representation for the distribution of the number of transmissions that are needed to reach the destination node. An important advantage of the phase-type representation is that, there exist simple and closed-form expressions for its distribution and moments. We used our model to investigate the performance of different scenarios in terms of the expected number of transmissions needed to send a packet from the source to the destination. Our results show that using Opportunistic Routing (OR) the expected number of transmissions can be reduced by 20% or more in a typical scenario. We showed that using a small number of candidates (like 2) may be a sensible choice to have small expected number of transmissions.

- Chapter 4. We applied our model proposed in Chapter 3 to compare four relevant algorithms that have been proposed in the literature. They range from non-optimum, but simple, to optimum, but with a high computational cost. The algorithms under study were: Extremely Opportunistic Routing (ExOR); Opportunistic Any-Path Forwarding (OAPF); Least-Cost Opportunistic Routing (LCOR); and Minimum Transmission Selection (MTS). We modified the algorithms under study in order to limit the maximum number of candidates per node. We compared different scenarios in terms of the expected, the variance and the probability of the number transmissions needed to send a packet from the source to the destination. The algorithms had also been compared from the perspective of the execution time which is needed to construct the Candidates Sets (CSs). Our numerical results show that if the maximum number of candidates is not limited, all of the algorithms obtain almost the same expected number of transmissions. When the maximum number of candidates is limited, our results show that the expected number of transmissions required by ExOR is larger than that of the other OR algorithms. On the other hand, the performance obtained with OAPF has shown to be very close to the optimal algorithm. We also observed that the variance of the number of transmissions can be substantially reduced by using OR.

  Regarding the execution times of each algorithm under study, the fact that ExOR is based on ETX makes this algorithm much faster than the others. For the optimum algorithms, we observed that MTS outperforms LCOR. However, both algorithms require extremely large execution times to compute the CSs in a network with 50 nodes (on the order of hours in a modern PC). On the other hand, OAPF is able to run the candidate selection with execution times orders of magnitude lower than the optimum algorithms (on the order of minutes) while its performance in terms of the expected number of transmissions is very close to the optimum algorithms.

- Chapter 5. We proposed a new metric that we call *Expected Distance Progress* (EDP). It measures the expected distance progress of sending a packet using a set of candidates. Then, we proposed a hop-by-hop candidate selection and prioritization algorithm based on EDP called *Distance Progress based Opportunistic Routing* (DPOR). Our algorithm does not need the whole topology information of the network to find the CSs. Only the neighbors' position and the links delivery probability to reach them are required. We compared DPOR with four other relevant candidate selection algorithms studied in Chapter 4: ExOR, OAPF, MTS and Position based Opportunistic Routing (POR). Simulation results show that the expected number of transmissions of ExOR and POR are far from the optimum algorithm. On the other hand, DPOR has almost the same expected number of transmissions as the optimum algorithm MTS. We also evaluated the computational cost of the algorithms. Our results show that finding the CSs with DPOR is much faster than OAPF and MTS. We concluded that DPOR is a fast and efficient OR candidate selection algorithm. Additionally, each node requires only neighbors' and destination geographic positions and the delivery probability to neighbors nodes. Indeed DPOR needs much less information compared to the most of OR proposals in the literature, which require the whole network topology.

- Chapter 6. We investigated the maximum gain that can be obtained using OR. We derived the equations that yield the distances of the candidates in Opportunistic Routing (OR) such that the per transmission progress towards the destination is maximized. We called them as the *Maximum Progress Distances (MPDs)*. The only ingredient to obtain these distances is the law for the delivery probability between nodes as a function of distance. An important consequence of our derivation is that the MPDs for the already existing candidates do not change if we decide to add a new candidate to the CS. Based on MPDs, we proposed a lower bound to the expected number of transmissions needed to send a packet using OR. We investigated the sensitivity of our results to the position of the candidates and concluded that choosing the distance of the first two candidates near to their optimal positions, is the most critical in order to minimize the expected number of transmissions. We provided a rule of thumb for placing the nodes in a static network using OR. We also proposed a new candidate selection algorithm based on Maximum Progress Distances (MPDs) that we call *Candidate selection based on Maximum Progress Distance* (CMPD). It tries to select the candidates that are located at the MPDs. The results show that, the performance of CMPD is almost the same as the optimum candidate selection algorithm, while our algorithm requires less information and runs much faster. Finally, we investigated the maximum performance using OR in the grid scenarios.We concluded that using our rule of thumb for placing the nodes in a grid network using OR, yielding a good performance in terms of the expected number of transmissions from the source to the destination.

- Chapter 7. Previous researches have shown that OR can significantly reduce the expected number of transmissions to deliver a packet to a particular destination. It is therefore tempting to adapt OR to improve the efficiency of wireless multicast. In this Chapter, we investigated how OR can be used to improve multicast delivery. We proposed a new multicast routing protocol based on Opportunistic Routing for WMNs, named *Multicast Opportunistic Routing Protocol* (MORP). Compared with traditional multicast protocols, MORP does not build a complete tree or mesh before the transmission starts. Instead, MORP builds a tree on the fly, depending on the candidates that successfully receive the packet in each transmission. It uses a three-way-handshaking approach where candidates send ACKs to the sender node upon successfully receiving data packets. Then, the sender node partitions the set of destinations and assigns each subset to the most appropriate candidate. This information is sent to the candidates which repeat the same approach for each subset of destinations. We compared our proposal with two well known ODMRP and ADMR multicast protocols. The comparison had done taking into consideration realistic simulations using 802.11 standard MAC layer. Our results show that MORP outperforms ODMRP and ADMR, reducing the number of data transmissions and increasing the delivery ratio.

8.2 Outlook and Future Research Directions

Wireless Mesh Networks have been envisioned as an important solution for the next generation of wireless networking. Opportunistic Routing has been proposed as a way to exploit unique features of wireless multi-hop networks by selecting multiple nodes as the candidates for forwarding the traffic. Within this work we showed that Opportunistic Routing can reduce the expected number of transmissions of sending packets from a source to the destination. We conclude that Opportunistic Routing is a promising technique to improve the performance of WMNs.

To design an OR protocol two main issues should be considered: Candidate selection; and Candidate coordination. Candidate selection is an algorithm which selects and orders the set of neighboring nodes that can help in the forwarding process toward a given destination. An efficient candidate selection algorithm selects a set of appropriate forwarders as the Candidates Set and gives the priority to each of them within a reasonable amount of time. There are many candidate selection algorithms with different complexity and performance which have been proposed in the literature. We have investigated the performance of different candidate selection algorithms in detail within this dissertation. We found that depending on the scenario and the availability of the topology information, one algorithm may be more preferable than the others. We also found that considering the position information of nodes for selecting the candidates is an efficient and fast approach for selecting the Candidates Set. However, to improve the efficiency of OR in WMNs, more research in the candidate selection issue is needed.

Candidate coordination is another important key issue in the design of OR protocol which receives less attention than the candidate selection in the literature. Candidate coordination is the mechanism used by the candidates to discover and coordinate the priority of each candidate that has received the packet, such that the candidate with higher priority must forward the packet. Imperfect candidate coordination may cause duplicate transmissions which increases the control overhead and reduces the efficiency of the OR protocol. Candidate coordination requires reliable signaling among the nodes. A practical implementation of the candidate coordination algorithm needs to consider existing technologies. It can be implemented at the link or network layer. However a link layer implementation permits the design of an efficient signaling protocol. To implement an OR protocol in the link layer, the MAC layer IEEE 802.11 standard should be modified. Therefore, it needs more investigation to design a practical as well as an efficient candidate coordination mechanism.

Opportunistic Routing is usually investigated for wireless mesh networks where nodes do not have mobility. When dealing with mobile networks such as Ad-Hoc, Sensor networks and VANET, OR is leading to include the mobility properties of these kind of networks. In the presence of mobility using an efficient and fast candidate selection algorithm is unavoidable.

Multicast is an important communications paradigm in wireless networks. Using OR for multicast delivery can improve the performance of multicast protocols. As we mentioned in Chapters 2, little work has been done in this topic. In Chapter 7 we have presented a new multicast routing protocol based on OR and we showed that it can improve the performance of the network. Using different coordination approaches like implementing it in the MAC layer can reduce the control overhead and end-to-end delay.

Using OR approach for broadcasting a packet to all nodes in the networks can be another research direction. The multicast protocol proposed in Chapter 7 can be improved to be adapted in the broadcast scenarios.

The choice of metric has great impact on the performance of an OR protocol. Most of the works in the literature mainly focused on using ETX and EAX as primary metrics. Investigating the error of link delivery probability and its impact on the OR performance in different types of networks can be another research direction. Another potential direction for research is the routing metrics with various performance objectives, such as minimizing delay or maximizing energy efficiency.

Opportunistic Routing in multi-channel multi-radio networks can be an interesting research direction. It is interesting to investigate distributed algorithms to solve the channel assignment and candidate selection issues. In this issue, a sender may have different Candidates Sets on different channels. Authors in [90] computed an end-to-end throughput bound of OR in multi-radio multi-channel multi-hop wireless networks. Designing an efficient algorithm that leads to semi-optimal channel assignment and candidate selection is desirable.

Combining OR with network coding is a promising research direction.

Finally, security is another major concern in multihop wireless networks. It will be valuable to design secure OR protocols and integrate them into existing security framework to provide more robust and more secure information delivery service.

Acronym

WMN Wireless Mesh Network

OR Opportunistic Routing

CS Candidates Set

SPF Shortests Path First

ETX Expected Transmission Count

EAX Expected Any-path Transmission

NC Network Coding

ExOR Extremely Opportunistic Routing

SDF Selection Diversity Forwarding

RTS Request-To-Send

CTS Clear-To-Send

NAV Network Allocation Vector

OAPF Opportunistic Any-Path Forwarding

LCOR Least-Cost Opportunistic Routing

MTS Minimum Transmission Selection

SOAR Simple Opportunistic Adaptive Routing

GeRaF Geographic Random Forwarding

CORE Coding-aware Opportunistic Routing mechanism & Encoding

MORE MAC-independent Opportunistic Routing & Encoding

GOR Geographic Opportunistic Routing

EOT Expected One-hop Throughput

POR Position based Opportunistic Routing

MSTOR Minimum Steiner Tree with Opportunistic Routing

MORP Multicast Opportunistic Routing Protocol

DTMC Discrete Time Markov Chain

EDP Expected Distance Progress

DPOR Distance Progress based Opportunistic Routing

CMPD Candidate selection based on Maximum Progress Distance

MPD Maximum Progress Distance

QOO Quasi Optimal Opportunistic Routing Network

ODMRP On-Demand Multicast Routing Protocol

ADMR Adaptive Demand-Driven Multicast Routing

MPD Maximum Progress Distance

SIFS Short Interframe Space

MOR Multicast Opportunistic Routing

ETA Expected Transmission Advancement

ADP Average Distance Progress

Appendices

Appendix A List of Publications

A.1 Journals

1. Amir Darehshoorzadeh and Llorenç Cerdà-Alabern. **Multicast Delivery Using Opportunistic Routing in Wireless Mesh Networks.** Submitted to *Wireless Networks, 2012.*

2. Amir Darehshoorzadeh, Llorenç Cerdà-Alabern and Vicent Pla. **Modeling and Comparison of Candidate Selection Algorithms in Opportunistic Routing.** *Computer Networks, 2011.*

3. Llorenç Cerdà-Alabern, Amir Darehshoorzadeh and Vicent Pla. **Optimum Node Placement in Wireless Opportunistic Routing Networks.** (Under preparation)

4. Llorenç Cerdà-Alabern, Amir Darehshoorzadeh and Vicent Pla. **Candidate Selection in Opportunistic Routing Using Maximum Progress Distances.** (Under preparation)

A.2 Conferences

1. Amir Darehshoorzadeh and Llorenç Cerdà-Alabern. **Distance Progress Based Opportunistic Routing for Wireless Mesh Networks.** In *The* 8^{th} *International Wireless Communications & Mobile Computing Conference, (IWCMC'12). Limassol, CYPRUS, 2012.*

2. Amir Darehshoorzadeh and Llorenç Cerdà-Alabern. **A New Multicast Opportunistic Routing Protocol for Wireless Mesh Networks.** In *Performance Evaluation of Cognitive Radio Networks Workshop (PE-CRN co-located with Networking), 2011, Valencia, Spain, 2011.*

3. Amir Darehshoorzadeh and Llorenç Cerdà-Alabern. **Candidate Selection Algorithms in Opportunistic Routing.** In *Proceedings of the* 5^{th} *ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks (PM2HW2N co-located with MSWIM), Bodrum, Turkey, 2010.*

4. Llorenç Cerdà-Alabern, Amir Darehshoorzadeh and Vicent Pla. **On the Maximum Performance in Opportunistic Routing.** In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (IEEE WoWMoM 2010), Montreal, Canada, 2010.*

5. Llorenç Cerdà-Alabern, Vicent Pla and Amir Darehshoorzadeh. **On the Performance Modeling of Opportunistic Routing.** In *MobiOpp '10: Proceedings of the Second International Workshop on Mobile Opportunistic Networking, Pisa, Italy, 2010.*

A.3 Book Chapter

- Amir Darehshoorzadeh, Llorenç Cerdà-Alabern and Vicent Pla. **Opportunistic Routing in Wireless Mesh Networks**, in *Routing in Opportunistic Networks. Dr. Isaac Woungang (Editor), to be published by Springer Verlag, USA, in early 2013.* (**Under Review**)

Appendix B Propagation Model

The free-space and the two-ray model predict the received power as a deterministic function of distance. They both represent the communication range as an ideal circle. In reality, the received power at certain distance is a random variable due to multi-path propagation effects, which is also known as fading effects. In fact, free-space and two-ray models predict the mean received power at distance d. A more general model is called the shadowing model [68]. In this appendix we describe the shadowing propagation model that have been used in the numerical results of this dissertation.

The shadowing model consists of two parts. The first one is known as path loss model, which predicts the mean received power at distance d, denoted by $P_r(d)$. It uses a close-in distance d_0 as a reference. $\overline{P_r(d)}$ is computed relative to $P_r(d_0)$ as follows.

$$\frac{P_r(d_0)}{\overline{P_r(d)}} = (\frac{d}{d_0})^{\beta} \tag{B.1}$$

Where β is called the path loss exponent, and is usually empirically determined by field measurement. Larger values correspond to more obstructions and hence faster decrease in average received power as distance become larger. $P_r(d_0)$ can be computed using the free-space equation which is:

$$P_r(d) = \frac{P_t\, G_t\, G_r\, \lambda^2}{L\,(4\,\pi)^2\, d^2} \tag{B.2}$$

.

In equation B.2, G_t and G_r are the transmission and reception antenna gains respectively, L is a system loss, λ is the signal wavelength (c/f, with $c = 3 \times 10^8$ m/s).

The path loss is usually measured in dB. So from equation B.1 we have:

$$\left[\frac{\overline{P_r(d)}}{P_r(d_0)}\right]_{dB} = -10\beta log(\frac{d}{d_0}) \tag{B.3}$$

The second part of the shadowing model reflect the variation of the received power at certain distance. It is a log-normal random variable, I.e., it is of Gaussian distribution in dB. The overall shadowing model is represented by equation B.4.

$$P_r(d)|_{dB} = 10\,\log_{10}\left(\frac{P_t\,G_t\,G_r\,\lambda^2}{L\,(4\,\pi)^2\,d^\beta}\right) + X_{dB} \qquad \text{(B.4)}$$

Where $X_{dB}$is a Gaussian random variable with zero mean and standard deviation σ_{dB}. σ_{dB} is called the shadowing deviation, and like β is also obtained by measurement. Table B.1 shows some typical values for β and σ_{dB}.

The shadowing model extended the ideal circle to a richer statistic model: nodes can only probabilistically communicate when neat the edge of the communication range. Packets are correctly delivered if the received power is greater than or equal to `RXThresh`. Note that, this model shall no consider collisions. Thus, the delivery probability at a distance d ($p(d)$) is given by:

$$p(d) = Prob(P_r(d)|_{dB} \geq 10\,\log_{10}(\texttt{RXThresh})) = Q\left(\frac{1}{\sigma_{dB}} 10\,\log_{10}\left(\frac{\texttt{RXThresh}\,L\,(4\,\pi)^2\,d^\beta}{P_t\,G_t\,G_r\,\lambda^2}\right)\right) \qquad \text{(B.5)}$$

Where $Q(z) = \frac{1}{\sqrt{2\,\pi}} \int_z^\infty \mathrm{e}^{-y^2/2} dy$.

The default default values of shadowing propagation model in the network simulator (Ns-2) [2] and GloMoSim [91] are given in tables B.2 and B.3.

Figure B.1 depicts the delivery probability at a varying the distance, for two values of the path loss exponent (β) and a standard deviation $\sigma_{dB} = 6$ dBs with Ns-2 default values. With these parameters and $\beta = 2.7$ the link delivery probability is approximately 40% at the distance of 150 m.

Table B.1: Typical values for β and σ_{dB}.

Environment		β	σ_{dB}
Outdoor	Free space	2	4 ∼12
	urban	2.7 ∼ 5	
Office	Line-of-sight	1.6 ∼ 1.8	7 ∼ 9.6
	Obstructed	4 ∼ 6	

Table B.2: Default Ns-2 values for the shadowing propagation model.

Parameter	Value
P_t	0.28183815 Watt
RXThresh	3.652×10^{-10} Watt
G_t, G_r, L	1
f	914 MHz

Table B.3: Default GloMoSim values for the shadowing propagation model.

Parameter	Value
P_t	0.03162278 Watt
RXThresh	7.943282×10^{-12} Watt
G_t, G_r, L	1
f	2400 MHz

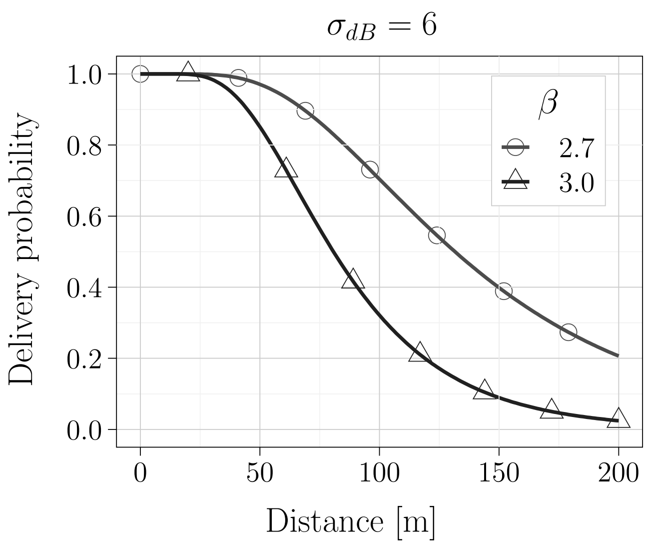


Figure B.1: Delivery probability versus distance for a standard deviation $\sigma_{dB} = 6$ dBs.

Appendix C Expected Number of Transmissions Recursive Formula

Expected Any-path Transmission (EAX): This metric was defined by Zhong et al. in [96] to capture the expected number of transmissions taking into account the multiple paths that can be used under OR. Let s be a source node, with c_i the candidate with priority i (with $i = 1$ being the highest priority) and p_i the delivery probability between s and c_i. Using the same assumptions as in equation (2.1), the expected number of transmissions to reach a destination node d is given by the recursive formula:

$$EAX(C^{s,d}, s, d) = S(C^{s,d}, s, d) + Z(C^{s,d}, s, d) \tag{C.1}$$

$$S(C^{s,d}, s, d) = \frac{1}{1 - \prod_{i=1}^{|C^{s,d}|}(1 - p_i)} \tag{C.2}$$

$$Z(C^{s,d}, s, d) = \frac{\sum_{i=1}^{|C^{s,d}|} EAX(C^{c_i,d}, c_i, d)\, p_i \prod_{j=1}^{i-1}(1 - p_j)}{1 - \prod_{i=1}^{|C^{s,d}|}(1 - p_i)} \tag{C.3}$$

where $C^{s,d}$ is the candidate set of node s to reach the destination d, and $|C^{s,d}|$ its cardinality; $S(C^{s,d}, s, d)$ is the expected number of transmissions from s until at least one of the nodes in $C^{s,d}$ receives the packet; and $Z(C^{s,d}, s, d)$ is the expected number of transmissions to reach the destination d from one of the nodes in $C^{s,d}$ which is responsible to forward the packet. Note that in equation (C.3) we take the product $\prod_{j=1}^{i-1}$ equal to 1 for $i = 1$.

Index

Bibliography

[1] MIT roofnet. http://pdos.csail.mit.edu/roofnet. 18, 42

[2] The Network Simulator ns-2. 29, 48, 63, 74, 132

[3] R. Ahlswede, Ning Cai, S.-Y.R. Li, and R.W. Yeung. Network information flow. *Information Theory, IEEE Transactions on*, 46(4):1204–1216, July 2000. 15

[4] Ian F. Akyildiz and Xudong Wang. *Wireless Mesh Networks*. John Wiley & Sons Ltd, 2009. 1

[5] Ian F. Akyildiz, Xudong Wang, and Weilin Wang. Wireless mesh networks: a survey. *Comput. Netw. ISDN Syst.*, 47(4):445–487, March 2005. 1

[6] I.F. Akyildiz and Xudong Wang. A survey on wireless mesh networks. *Communications Magazine, IEEE*, 43(9):S23 – S30, sept. 2005. 1

[7] F. Baccelli, B. Blaszczyszyn, and P. Muhlethaler. On the performance of time-space opportunistic routing in multihop mobile ad hoc networks. In *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops, 2008. WiOPT 2008. 6th International Symposium on*, pages 307–316, April 2008. 11, 23

[8] Osamah S. Badarneh and Michel Kadoch. Multicast routing protocols in mobile ad hoc networks: a comparative survey and taxonomy. *EURASIP J. Wirel. Commun. Netw.*, pages 26:1–26:42, 2009. 21, 101

[9] Rajashekhar Biradar, Sunilkumar Manvi, and Mylara Reddy. Link stability based multicast routing scheme in manet. *Comput. Netw.*, 54:1183–1196, May 2010. 21

[10] J. Biswas, M. Barai, and S.K. Nandy. Efficient hybrid multicast routing protocol for ad-hoc wireless networks. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pages 180 – 187, nov. 2004. 22

[11] S. Biswas and R. Morris. ExOR: opportunistic multi-hop routing for wireless networks. *ACM SIGCOMM Computer Communication Review*, 35(4):133–144, 2005. 2, 11, 17, 41, 101

[12] Sanjit Biswas and Robert Morris. Opportunistic routing in multi-hop wireless networks. *ACM SIGCOMM Computer Communication Review*, 34(1):69–74, 2004. 11, 14, 17, 29, 39, 40, 62, 83, 94, 101, 104, 105

[13] R. Bruno and M. Nurchis. Survey on Diversity Based Routing in Wireless Mesh Networks: Challenges and Solutions. *Computer Communications*, 33(3):269–282, 2010. 15, 16

[14] A.S. Cacciapuoti, M. Caleffi, and L. Paura. Optimal constrained candidate selection for opportunistic routing. In *IEEE GLOBECOM*, pages 1–5, December 2010. 23

[15] Shao-Bin Cai and Xiao-Zong Yang. The performance of poolodmrp protocol. In *MMNS'03*, pages 90–101, 2003. 21

[16] Llorenç Cerdà-Alabern, Amir Darehshoorzadeh, and Vicent Pla. On the maximum performance in opportunistic routing. In *IEEE WoWMoM 2010*, Montreal, Canada, 2010. 5, 11, 97

[17] Llorenç Cerdà-Alabern, Vicent Pla, and Amir Darehshoorzadeh. On the performance modeling of opportunistic routing. In *MobiOpp '10: Proceedings of the Second International Workshop on Mobile Opportunistic Networking*, pages 15–21, New York, NY, USA, 2010. ACM. 4, 13, 78

[18] Szymon Chachulski, Michael Jennings, Sachin Katti, and Dina Katabi. Trading structure for randomness in wireless opportunistic routing. In *SIGCOMM '07: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 169–180, New York, NY, USA, 2007. ACM. 11, 15, 18, 22

[19] Ching-Chuan Chiang, Mario Gerla, and Lixia Zhang. Forwarding group multicast protocol (fgmp) for multihop, mobile wireless networks. *Cluster Computing*, 1:187–196, April 1998. 21

[20] Wing-Cheong Lau Chun-Pong Luk and On-Ching Yue. Opportunistic routing with directional antennas in wireless mesh networks. In *IEEE INFOCOM*, 2009. 23

[21] Thomas Clausen, Philippe Jacquet (editors), Cédric Adjih, Anis Laouiti, Pascale Minet, Paul Muhlethaler, Amir Qayyum, and Laurent Viennot. Optimized link state routing protocol (OLSR). RFC 3626, pages 1-75, October 2003. Network Working Group. 101

[22] Amir Darehshoorzadeh and Llorenç Cerdà-Alabern. Candidate selection algorithms in opportunistic routing. In *PM2HW2N '10: Proceed-*

ings of the 5th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks, pages 48–54, New York, NY, USA, 2010. ACM. 4

[23] Amir Darehshoorzadeh and Llorenç Cerdà-Alabern. A new multicast opportunistic routing protocol for wireless mesh networks. In *Networking Workshops*, pages 62–72, 2011. 5, 11

[24] Amir Darehshoorzadeh and Llorenç Cerdà-Alabern. Distance progress based opportunistic routing for wireless mesh networks. In *International Wireless Communications & Mobile Computing Conference*. IEEE, 2012. 5

[25] Amir Darehshoorzadeh and Llorenç Cerdà-Alabern. Distance Progress Based Opportunistic Routing for Wireless Mesh Networks. (Accepted). In *Wireless Communications and Mobile Computing Conference, IWCMC '12. International*, 2012. 83

[26] Amir Darehshoorzadeh, Llorenç Cerdà-Alabern, and Vicent Pla. Modeling and comparison of candidate selection algorithms in opportunistic routing. *Computer Networks*, 55(13):2886 – 2898, 2011. 4

[27] Amir Darehshoorzadeh, Llorenç Cerdà-Alabern, and Vicent Pla. Modeling and comparison of candidate selection algorithms in opportunistic routing. *Computer Networks*, 55(13):2886– 2898, 2011. 11

[28] Subir Kumar Das, B. S. B. S. Manoj, and C. Siva Ram Murthy. A dynamic core based multicast routing protocol for ad hoc wireless networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '02, pages 24–35, New York, NY, USA, 2002. ACM. 21

[29] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. *Wireless Networks*, 11(4):419–434, 2005. 12, 13, 98

[30] H. Dubois-Ferriéandre, M. Grossglauser, and M. Vetterli. Least-cost opportunistic routing. In *Allerton Conference on Communication, Control, and Computing*, 2007. 2, 18, 29, 30, 78

[31] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. Least-cost opportunistic routing. In *Proceedings of 2007 Allerton Conference on Communication, Control, and Computing*, 2007. 11, 13, 39, 44

[32] H. Dubois-Ferriére, M. Grossglauser, and M. Vetterli. Valuable detours: Least-cost anypath routing. *Networking, IEEE/ACM Transactions on*, 19(2):333 –346, april 2011. 2, 97

[33] D. Fuste-Vilella, J. Garcia-Vidal, and J.D. Morillo-Pozo. Cooperative forwarding in ieee 802.11-based manets. In *Wireless Days, 2008. WD '08. 1st IFIP*, pages 1 –5, nov. 2008. 11

[34] J.J. Garcia-Luna-Aceves and E.L. Madruga. The core-assisted mesh protocol. *Selected Areas in Communications, IEEE Journal on*, 17(8):1380 –1394, aug 1999. 21

[35] Che-Jung Hsu, Huey-Ing Liu, and Winston Khoon Guan Seah. Opportunistic routing - a review and the challenges ahead. *Computer Networks*, 55(15):3592–3603, 2011. 3, 15

[36] Shweta Jain and Samir R. Das. Exploiting path diversity in the link layer in wireless ad hoc networks. *Ad Hoc Netw.*, 6(5):805–825, 2008. 12, 16

[37] Jorjeta G. Jetcheva and David B. Johnson. Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '01, pages 33–44, New York, NY, USA, 2001. ACM. 20, 102, 104, 105

[38] Lusheng Ji and M.S. Corson. Differential destination multicast-a manet multicast routing protocol for small groups. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 1192 –1201 vol.2, 2001. 20

[39] Luo Junhai, Ye Danxia, Xue Liu, and Fan Mingyu. A survey of multicast routing protocols for mobile ad-hoc networks. *Communications Surveys Tutorials, IEEE*, 11(1):78 –91, quarter 2009. 20

[40] Luo Junhai, Xue Liu, and Ye Danxia. Research on multicast routing protocols for mobile ad-hoc networks. *Computer Networks*, 52(5):988 – 997, 2008. 20, 21

[41] B. Kaliaperumal, A. Ebenezer, and Jeyakumar. Adaptive core based scalable multicasting networks. In *INDICON, 2005 Annual IEEE*, pages 198 – 202, dec. 2005. 21

[42] Brad Karp and H. T. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, MobiCom '00, pages 243–254, New York, NY, USA, 2000. ACM. 60

[43] Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel Médard, and Jon Crowcroft. Xors in the air: practical wireless network coding. In *Proceedings of the 2006 conference on Applications,*

technologies, architectures, and protocols for computer communications, SIGCOMM '06, pages 243–254, New York, NY, USA, 2006. ACM. 11, 19

[44] Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel Médard, and Jon Crowcroft. XORs in the air: practical wireless network coding. *IEEE/ACM Trans. Netw.*, 16(3):497–510, 2008. 11, 19

[45] D. Koutsonikolas, Y.C. Hu, and Chih-Chun Wang. Pacifier: High-throughput, reliable multicast without "crying babies" in wireless mesh networks. In *INFOCOM 2009, IEEE*, pages 2473 –2481, 2009. 11, 22

[46] M. Kurth, A. Zubow, and J.-P. Redlich. Cooperative opportunistic routing using transmit diversity in wireless mesh networks. In *IEEE INFOCOM*, pages 1310–1318, 2008. 23

[47] Peter Larsson. Selection diversity forwarding in a multihop packet radio network with fading channel and capture. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(4):47–54, 2001. 2, 11, 14

[48] G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. ASA-SIAM, 1999. 28

[49] R. Laufer, H. Dubois-Ferriere, and L. Kleinrock. Multirate anypath routing in wireless mesh networks. In *IEEE INFOCOM*, pages 37–45, 2009. 18

[50] Jilin Le, J. Lui, and Dah Ming Chiu. Dcar: Distributed coding-aware routing in wireless networks. In *Distributed Computing Systems, 2008. ICDCS '08. The 28th International Conference on*, pages 462–469, June 2008. 19

[51] Tan Le and Yong Liu. Exploring the gain of opportunistic routing in wireless multicast. Technical-report, Polytechnic Institute of New York University, 5 Metrotech Center, Brooklyn, NY 11201, US, 2009. 22

[52] Tan Le and Yong Liu. Opportunistic overlay multicast in wireless networks. In *GLOBECOM*, pages 1–5, 2010. 11, 22

[53] Meejeong Lee and Ye Kyung Kim. Patchodmrp: an ad-hoc multicast routing protocol. In *Information Networking, 2001. Proceedings. 15th International Conference on*, pages 537 –543, 2001. 21

[54] Seungjoon Lee and Chongkwon Kim. Neighbor supporting ad hoc multicast routing protocol. In *Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '00, pages 37–44, Piscataway, NJ, USA, 2000. IEEE Press. 21

[55] Sung-Ju Lee, M. Gerla, and Ching-Chuan Chiang. On-demand multicast routing protocol. In *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, pages 1298 –1302 vol.3, 1999. 21

[56] Sung Ju Lee, William Su, and Mario Gerla. On-demand multicast routing protocol in multihop wireless mobile networks. *Mob. Netw. Appl.*, 7:441–453, December 2002. 21, 101

[57] Yanhua Li, Yuan an Liu, and Pengkui Luo. Link probability based opportunistic routing metric in wireless network. In *Communications and Mobile Computing, 2009. CMC '09. WRI International Conference on*, volume 2, pages 308–312, January 2009. 18

[58] Yanhua Li, Wei Chen, and Zhi-Li Zhang. Optimal forwarder list selection in opportunistic routing. In *Mobile Adhoc and Sensor Systems. MASS '09. IEEE 6th International Conference on*, pages 670 –675, oct. 2009. 11, 13, 18, 39, 45, 62, 83, 86

[59] Yanhua Li and Zhi-Li Zhang. Random walks on digraphs: a theoretical framework for estimating transmission costs in wireless routing. In *29th conference on Information communications*, INFOCOM'10, pages 2775–2783. IEEE Press, 2010. 28, 29

[60] C.R. Lin and Shiang-Wei Chao. A multicast routing protocol for multihop wireless networks. In *Global Telecommunications Conference, 1999. GLOBECOM '99*, volume 1A, pages 235 –239 vol.1a, 1999. 21

[61] Haitao Liu, Baoxian Zhang, H. Mouftah, Xiaojun Shen, and Jian Ma. Opportunistic routing for wireless ad hoc and sensor networks: Present and future directions. *Communications Magazine, IEEE*, 47(12):103 –109, dec. 2009. 12, 17

[62] Mingming Lu and Jie Wu. Opportunistic routing algebra and its applications. In *IEEE INFOCOM*, pages 2374–2382, 2009. 13, 23

[63] Chun-Pong Luk, Wing-Cheong Lau, and On-Ching Yue. An analysis of opportunistic routing in wireless mesh network. In *Communications, 2008. ICC '08. IEEE International Conference on*, pages 2877–2883, May 2008. 11, 23

[64] M. Naderan-Tahan, A. Darehshoorzadeh, and M. Dehghan. Odmrp-lr: Odmrp with link failure detection and local recovery mechanism. In *Computer and Information Science, 2009. ICIS 2009. Eighth IEEE/ACIS International Conference on*, pages 818 –823, jun. 2009. 101

[65] S.Y. Oh, Joon-Sang Park, and M. Gerla. E-odmrp: enhanced odmrp with motion adaptive refresh. In *Wireless Communication Systems, 2005. 2nd International Symposium on*, pages 130 – 134, sept. 2005. 21

[66] Dhammika Pathirana and Minseok Kwon. Rodmrp: Resilient on-demand multicast routing protocol. *Advanced Information Networking and Applications Workshops, International Conference on*, 2:85–92, 2007. 21

[67] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0. 48, 62

[68] Theodore Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001. 131

[69] Elizabeth M. Royer and Charles E. Perkins. Multicast Ad hoc On-Demand Distance Vector (MAODV) Routing - draft-ietf-manet-maodv-00. Technical report, Mobile Ad Hoc Networking Working Group, 2000. 21

[70] E. Rozner, J. Seshadri, Y. Mehta, and Lili Qiu. Soar: Simple opportunistic adaptive routing protocol for wireless mesh networks. *Mobile Computing, IEEE Transactions on*, 8(12):1622–1635, December 2009. 11, 18

[71] L. Wang S. Cai, X. Yang. Pdaodmrp: An extended poolodmrp based on passive data acknowledgement. *Journal of Communications and Networks*, 6(4):362–375, 2004. cited By (since 1996) 3. 21

[72] R.C. Shah, S. Wietholter, and A. Wolisz. Modeling and analysis of opportunistic routing in low traffic scenarios. In *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 2005. WIOPT 2005. Third International Symposium on*, pages 294–304. IEEE, 2005. 11, 12

[73] P. Sinha, R. Sivakumar, and V. Bharghavan. Mcedar: multicast core-extraction distributed ad hoc routing. In *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, pages 1313 –1317 vol.3, 1999. 22

[74] R.S. Sisodia, I. Karthigeyan, B.S. Manoj, and C.S.R. Murthy. A preferred link based multicast protocol for wireless mobile ad hoc networks. In *Communications, 2003. ICC '03. IEEE International Conference on*, volume 3, pages 2213 – 2217 vol.3, may 2003. 20

[75] Radosveta Sokullu and Ozlem Karaca. Comparative performance study of admr and odmrp in the context of wireless lans and wireless sensor networks. In *Proceedings of the 7th WSEAS International Conference on Telecommunications and Informatics*, pages 183–187, Stevens Point, Wisconsin, USA, 2008. World Scientific and Engineering Academy and Society (WSEAS). 102

[76] S. Cañadas-Hurtado Triviño-Cabrera, Alicia. Survey on opportunistic routing in multihop wireless networks. *IJCNIS*, 3(2), 2011. 12

[77] Yang WenZhong, Zhang ZhenYu, Wang Bo, and Wu XiaoHong. A reliable multicast for manets based on opportunistic routing. In *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on*, pages 1–4, sept. 2010. 22

[78] Cedric Westphal. Opportunistic routing in dynamic ad hoc networks: the oprah protocol. In *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, pages 570–573, October 2006. 12

[79] C.W. Wu and Y.C. Tay. Amris: a multicast protocol for ad hoc wireless networks. In *Military Communications Conference Proceedings, 1999. MILCOM 1999. IEEE*, volume 1, pages 25 –29 vol.1, 1999. 21

[80] J. Wu, M. Lu, and F. Li. Utility-based opportunistic routing in multi-hop wireless networks. In *Distributed Computing Systems, 2008. ICDCS'08. The 28th International Conference on*, pages 470–477, 2008. 23

[81] Yunnan Wu, S.M. Das, and R. Chandra. Routing with a markovian metric to promote local mixing. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 2381–2385, May 2007. 19

[82] Jason Xie, Rajesh R. Talpade, Anthony Mcauley, and Mingyan Liu. Amroute: ad hoc multicast routing protocol. *Mob. Netw. Appl.*, 7:429–439, December 2002. 22

[83] Y. Yan, B. Zhang, J. Zheng, and J. Ma. Core: a coding-aware opportunistic routing mechanism for wireless mesh networks [accepted from open call]. *Wireless Communications, IEEE*, 17(3):96 –103, june 2010. 11, 19

[84] Yan Yan, Baoxian Zhang, H.T. Mouftah, and Jian Ma. Practical coding-aware mechanism for opportunistic routing in wireless mesh networks. In *Communications, 2008. ICC '08. IEEE International Conference on*, pages 2871–2876, May 2008. 11, 15, 19

[85] Shengbo Yang, Feng Zhong, Chai Kiat Yeo, Bu Sung Lee, and Jeff Boleng. Position based opportunistic routing for robust data delivery in manets. In *Proceedings of the 28th IEEE conference on Global telecommunications*, GLOBECOM'09, pages 1325–1330, Piscataway, NJ, USA, 2009. IEEE Press. 11, 20, 60, 62

[86] Yaling Yang and Jun Wang. Design guidelines for routing metrics in multihop wireless networks. In *IEEE INFOCOM 2008*, pages 1615–1623, April 2008. 23

[87] Y. Yuan, H. Yang, S.H.Y. Wong, S. Lu, and W. Arbaugh. ROMER: Resilient opportunistic mesh routing for wireless mesh networks. In *IEEE Workshop on Wireless Mesh Networks (WiMesh)*, 2005. 12

[88] Kai Zeng, Wenjing Lou, Jie Yang, and Donald R. Brown, III. On throughput efficiency of geographic opportunistic routing in multihop wireless networks. *Mob. Netw. Appl.*, 12:347–357, December 2007. 11, 19

[89] Kai Zeng, Wenjing Lou, and Hongqiang Zhai. On end-to-end throughput of opportunistic routing in multirate and multihop wireless networks. In *IEEE INFOCOM*, pages 816–824, 2008. 18

[90] Kai Zeng, Zhenyu Yang, and Wenjing Lou. Opportunistic routing in multi-radio multi-channel multi-hop wireless networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1 –5, march 2010. 124

[91] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. GloMoSim: A Library for Parallel Simulation of Large-Scale Wireless Networks. In *Proceedings of the 12th Workshop on Parallel and Distributed Simulation (PADS'98)*, pages 154–161. IEEE Computer Society, May 1998. 103, 132

[92] Xiaofeng Zhang and Lillykutty Jacob. Mzrp: An extension of the zone routing protocol for multicasting in manets. *J. Inf. Sci. Eng.*, pages 535–551, 2004. 21

[93] Bin Zhao, Rohit Iyer Seshadri, Matthew C. Valenti, Harbinger). Like Geraf, and Harbinger Assumes That. Geographic random forwarding with hybrid-arq for ad hoc networks with rapid sleep cycles. In *in Proc. IEEE Military Commun. Conf. (MILCOM*, 2004. 12

[94] Xin Zhao, Chun Tung Chou, Jun Guo, Sanjay Jha, and A. Misra. Probabilistically reliable on-demand multicast in wireless mesh networks. In *World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008. 2008 International Symposium on a*, pages 1 –9, june 2008. 20

[95] Zifei Zhong and S. Nelakuditi. On the efficacy of opportunistic routing. In *4th Annual IEEE Communications Society Conference onS ensor, Mesh and Ad Hoc Communications and Networks, SECON '07*, pages 441–450, June 2007. 18

[96] Zifei Zhong, Junling Wang, Srihari Nelakuditi, and Guor-Huar Lu. On selection of candidates for opportunistic anypath forwarding. *SIGMOBILE Mob. Comput. Commun. Rev.*, 10(4):1–2, 2006. 11, 12, 13, 18, 29, 39, 42, 62, 78, 83, 134

[97] M. Zorzi and R.R. Rao. Geographic random forwarding (geraf) for ad hoc and sensor networks: multihop performance. *Mobile Computing, IEEE Transactions on*, 2(4):337–348, Oct.-Dec. 2003. 11, 12, 16, 17, 19