Chapter 3

Opportunistic Routing in Wireless Mesh Networks

Amir Darehshoorzadeh, Llorenç Cerdà-Alabern and Vicent Pla

Contents

3.1 Introduction

In recent years, multi-hop wireless networks (MWN) have already become very popular and are receiving an increasing amount of attention by the research community. Compared to wired networks, routing in MWN 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 MWN network. 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, this can simultaneously be received by several neighboring nodes.

Traditional routing protocols proposed for wireless networks perform best path routing, i.e., preselect one fixed route before transmissions starts. Each node in a route uses a fixed neighbor to forward to. 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 do not adapt well to the dynamic wireless environment where transmission failures occur frequently.

Opportunistic Routing (OR), also referred to as diversity forwarding [24], cooperative forwarding [17] or any-path routing [16], is being investigated to increase the performance of MWNs by taking advantage of its broadcast nature. In OR, in contrast to traditional routing, instead of preselecting a single specific node to be the next-hop as a forwarder for a packet, an ordered set of nodes (referred to as *candidates*) are selected as the potential next-hop forwarders. We shall refer to the ordered set of candidate of a node as its 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 3.1 (the example has been taken from [5]). 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.

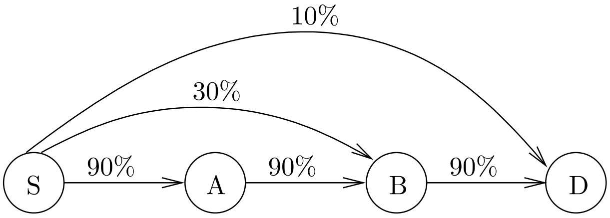


Figure 3.1: An example of Opportunistic Routing (OR).

Assume that S is the source and 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, OR takes advantage of any 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 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 [18]. As shown in figure 3.2 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 transmissions 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.

3.1.1 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 trans-

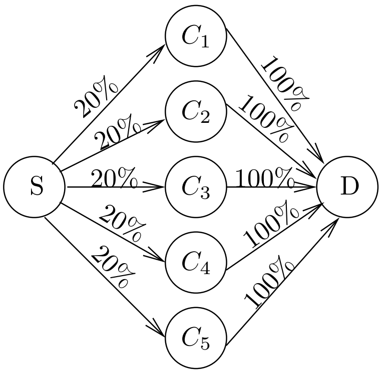


Figure 3.2: An illustration of virtual link in OR.

missions from the source to the destination. In section 3.4 some noteworthy candidate selection algorithms are described.

- **OR metric** In order to accurately select and prioritize the 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 3.2.

- **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.

  With perfect coordination among candidates, the larger is the number of candidates the lower is the expected number of transmissions from the source to the destination. 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 often been neglected in candidate selection algorithms proposed in the literature. Perfect coordination and no signaling overhead has been assumed and the algorithms have been designed to select all possible candidates to reduce the expected number of transmissions.

3.1.2 Research Directions in Opportunistic Routing

In this section we give an overview of the main research contributions in OR. Table 3.1 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 (see section 3.2).
- **Coord.**: Coordination method used in the paper (see section 3.3). 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 (see section 3.4): *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 3.1 are sorted in chronological order. The table shows the increasing interest that has emerged related with OR in the last decade.

Table 3.1: Classification of research works in opportunistic routing protocol.

Protocol	**Year**	**Type**	**Topic**	**Metric**	**Coord.**	**Cand. Sel.**
SDF [25]	2001	S	Candidate coordination	ETX	Ack	Topology
GeRaF [50]	2003	A/S	Candidate coordination	Geo.	RTS-CTS	Location
ExOR ver-1 [6]	2004	S	Candidate selection	ETX	Ack	Topology
ExOR ver-2 [5]	2005	E	Candidate Coordination	ETX	Timer	Topology
NA [36]	2005	A/S	Sensor networks	Geo.	RTS-CTS	Location
COPE [20, 21]	2005	E	Network coding	ETX	Net. coding	Topology
OAPF [49]	2006	S	Candidate selection	ETX/EAX	Ack	Topology
LCOR [15]	2007	S	Candidate selection	EAX	NA	Topology
MORE [11]	2007	E	Network coding	ETX	Net. coding	Topology
GOR [47]	2007	S	Candidate selection	Geo.	Timer	Location
NA [33]	2008	A	Analytical	Geo.	NA	Location
NA [4]	2008	A/S	Analytical	Geo.	NA	Location
NA [17]	2008	E	Candidate selection	Cloud	Ack	Topology
CORE [43, 42]	2008	S	Network coding	Geo.	Timer	Location
MTS [29]	2009	S	Candidate selection	EAX	Timer	Topology
POR [44]	2009	S	Candidate selection	Geo.	Timer	Location
SOAR [35]	2009	S/E	Candidate selection	ETX	Timer	Topology
Pacifier [22]	2009	S	Multicast	ETX	Net. coding	Topology
NA [9]	2010	A	Maximum performance	EAX	NA	Location
MSTOR [28]	2010	S	Multicast	EAX/ETX	Ack	Topology
MORP [13]	2011	S	Multicast	ETX	Ack	Topology
NA [12]	2011	A	Analytical/cand. selec.	ETX/EAX	NA	Topology

Most of the research in OR is related with the issues described in section 3.1.1, 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 3.2.

- Candidate coordination, section 3.3.
- Candidate selection algorithms, section 3.4.
- Network coding, section 3.5.
- Geographic OR, section 3.6.
- Multicast OR, section 3.7.
- Sensor networks.

Each of this 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 [36] in table 3.1. 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.

After the sections explaining the work in the research areas listed above, we continue the chapter by describing analytical models of OR in section 3.8. In section 3.9 we presented some numerical results that illustrate the performance achieved with some relevant candidate selection algorithms described in section 3.4. Finally, some concluding remarks are given in section 3.10.

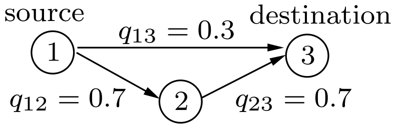
3.2 Routing Metrics

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 candidate new hop forwarders each node uses and and priority order of the candidates has 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 [46, 39, 19], geographic-distance [50, 48] (Geo-Distance), expected number of transmissions (ETX) [14], expected any-path transmission (EAX) [49] 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 [31, 37]. Below, we describe the two usual metrics ETX and EAX that have been used in the literature.

Expected Transmission Count (ETX) [14]: 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:

$$ETX(i,j) = \frac{1}{p_{ij}} \tag{3.1}$$



Network topology.

node	candidates to 3
1	3, 2
2	3

Candidates set of each node.

Figure 3.3: Illustration of the EAX metric.

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 [15, 29] have shown that using ETX may give suboptimal selection of candidates and in [32] it was shown that OR in combination with ETX could degrade the performance of the network. Because of that Zhong et al. [49] proposed another metric which has bee widely adopted in OR.

Expected Any-path Transmission (EAX) [49]: 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 other authors [15, 29, 10]. In section 3.9.1 we present a model for OR that can be used to calculate the expected number of transmissions from source to the destination.

The following simple example illustrates the meaning of EAX. Consider the network topology and the CS of each node in figure 3.3. 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 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 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 $E_1^{OR} \approx 2.15$. Although ETX is much simple to compute than EAX, it does not accurately compute the expected number of transmissions under OR.

3.3 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 RTS-CTS Coordination. In the following subsections we briefly describe these approaches.

3.3.1 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 [24] 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 ExOR [6], 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 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 3.4 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 3.4 we suppose that the acknowledgment from A does not receive by B, but node C does hear the A's ACK (see figure 3.4). Suppose further that node B hears node Cs 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.

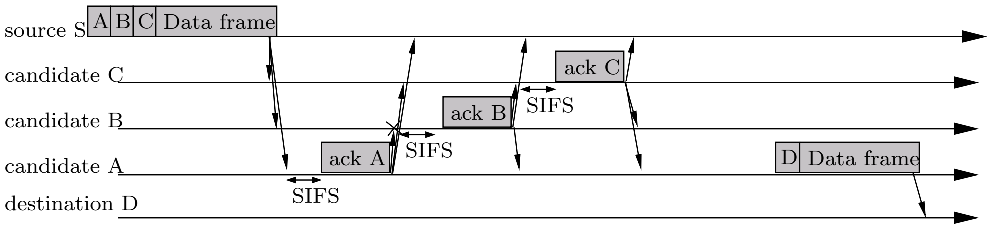


Figure 3.4: Acknowledgment-based coordination using a modified 802.11 MAC.

3.3.2 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 word, forwarding a data packet by a candidate will prevent the lower priority ones to forward it. In the example of figure 3.1, 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 [18].

3.3.3 Network Coding Coordination

Another approaches to prevent duplicate transmission is combining OR with network coding (NC) [3] which provides an elegant method for candidate coordination [11, 7, 43]. 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 3.5. Assume that source S transmits two native packets a and b using CS $\{C_1, C_2\}$. It generates two coded packet which are linear combination of a and b and broadcasts them. Assume that C_2 received both coded packet but C_1 received only one of them. Two candidates generate coded packets and transmit to the destination D without any coordination. When D received transmitted packets from C_1 and C_2, it can decode and restore the original packets.

However, using network coding 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 [7].

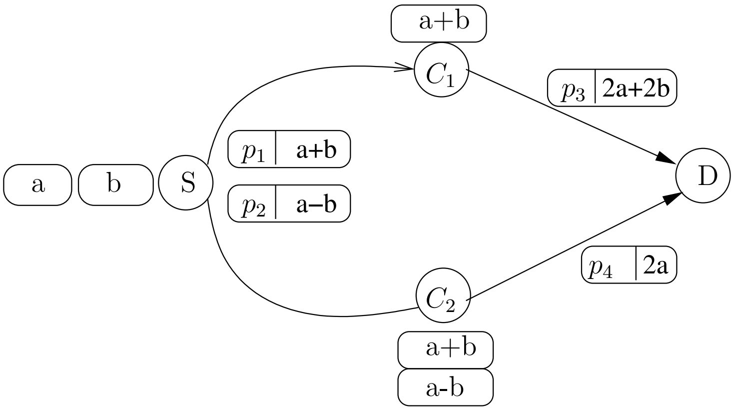


Figure 3.5: Network coding coordination approach [18].

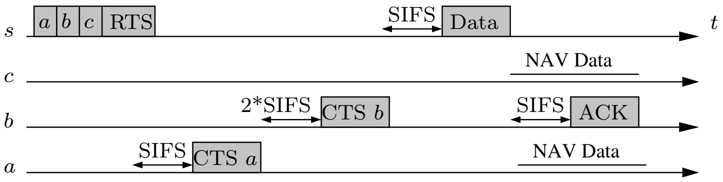


Figure 3.6: RTS-CTS coordination approach.

3.3.4 RTS-CTS Coordination

Some other mechanisms like [19, 50] 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 NAV until the end of ACK period.

Figure 3.6 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 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.

3.4 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 [50], each node maintains a limited state information and independently determines its own CS along the path to the intended destination. Topology-based selections [6, 5] 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 [31].

In this section, we describe four different 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. These algorithms are: Extremely Opportunistic Routing (ExOR) [6]; Opportunistic Any-Path Forwarding (OAPF) [49]; Least-Cost Opportunistic Routing (LCOR) [15]; and Minimum Transmission Selection (MTS) [29].

ExOR is one of the firsts 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 optimal sets of candidates (i.e. the CSs that minimize the expected number of transmissions). Finally, we have chosen LCOR and MTS which select the optimal sets of candidates.

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

- *ncand* is the maximum number of candidates per node.
- $ETX(v, d)$ is the uni-path ETX between the two nodes v and d.
- $EAX(C_{v,d}, v, d)$ is the EAX between node v and d by using $C_{v,d}$ as the CS of v to reach node d.
- $N(v)$ is the set of all neighbors of node v.
- $|S|$ is the cardinality of the set S.

In the following subsections we describe each algorithm with more details. For the sake of being precise, we shall give a pseudo-code for each algorithm.

To show the differences between ExOR, OAPF, LCOR and MTS, we use a simple example shown in figure 3.7. In this example node S is the source; D is the destination; the number on each link indicates its packet delivery probability (symmetric) and the maximum number of candidates for each node is set to 2 (*ncand*=2). The expected number of transmissions (ETX) using uni-path routing from each node to the destination D is shown in table 3.2.

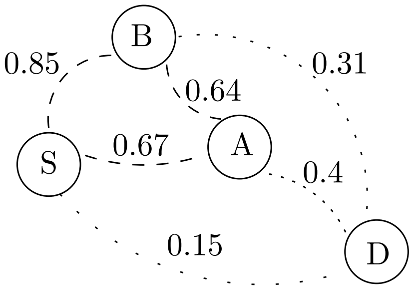


Figure 3.7: An example of Candidate selection.

Algorithm 1: Candidate.selection.ExOR(s, d, $ncand$).

```
$G_{tmp}$ ← temporal copy of the network topology
$cost(s)$ ← $ETX(s, d)$ in $G_{tmp}$
$C_{s,d} \leftarrow \emptyset$
while $|C_{s,d}| < ncand$ & (s, d) connected in $G_{tmp}$ do
    $cand$ ← 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
    $G_{tmp}$ ← delete $edge(s, cand)$ in $G_{tmp}$
end
$C_{s,d} \leftarrow C_{s,d}$ ordered by $cost$
```

3.4.1 Extremely Opportunistic Routing (ExOR)

Biswas and Morris proposed ExOR [6], one of the firsts and most referenced OR protocols. The selection of candidates is based on the Expected Transmission Count (ETX) metric. The basic idea of ExOR is running the *Shortest Path First* (SPF) with $weight = 1/q_{ij}$, where q_{ij} is the delivery probability of the link between the two nodes i and j (i.e. the weights are the ETXs of the links).

The algorithm for selection of candidates in ExOR is shown in Algorithm 1. Every node s except the destination d runs this algorithm. The first node after s in the shortest path is selected as candidate ($cand$) if its ETX to the destination (ETX($cand, d$)) is less than ETX(s, d). Then the link between s and $cand$ is removed, and this process is repeated until no more paths to d are available, or the maximum number of candidates is reached ($|C_{s,d}| = ncand$). Finally, ETX($cand, d$) (or 0 if the $cand$ is the destination) is used to sort the CS.

Assume that node S in figure 3.7 want to find its CS using ExOR. According to ExOR's algorithm (see algorithm 1), node S finds the shortest path first (SPF) to D which is S-A-D

with ETX=3.99 (see table 3.2). Therefore, node A is the first 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 is S-B-D with ETX=4.40 and B is selected as the second candidate. Finally, the ETX of each candidate to the destination d is used to sort the CS. The final candidates set 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 for the fact that packets are delivered by the candidates under opportunistic forwarding.

Table 3.2: Expected number of transmissions of each node to D in figure 3.7

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

There is a second version of ExOR [5] proposed in 2005. It is designed for batch forwarding. The source node includes in each packet a CS prioritized by closeness to the destination. Each packet has a Bitmap, which marks those packets that have been received by the sending node or nodes with higher priorities. A candidate transmits a packet only if no forwarder with higher priority has explicitly acknowledged receipt of it, as indicated in Bitmap position for this packet. Therefore, the coordination is done using timer-based coordination (see section 3.3.2). The performance of this protocal was evaluated on Roofnet [2], an outdoor roof-top 802.11b network.

SOAR [35] 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 CS. SOAR uses timer-based approach for candidate coordination.

3.4.2 Opportunistic Any-Path Forwarding (OAPF)

OAPF [49] is an OR protocol which is based on ETX and EAX. In [49] the authors analyzed the efficacy of OR by using EAX metric and did a comparison using the link-level measurement trace of MIT Roofnet [2]. The pseudo-code of OAPF is shown in Algorithm 2.

In OAPF the selection of candidates can be performed as follows at a node s for a specific destination d. First, a set of initial candidates $\hat{C}_{s,d}$ is determined based on the best path ETX. A neighbor v is included in the initial CS ($\hat{C}_{s,d}$) only if $\mathrm{ETX}(v, d) < \mathrm{ETX}(s, d)$. Then, a subset of $\hat{C}_{s,d}$ is selected as the actual CS $C_{s,d}$. Note that, all nodes in the initial CS must select their CSs before s.

After initiating the CS, s selects the best candidate among the nodes in its initial CS. Here,

Algorithm 2: Candidate.selection.OAPF(s, d, $ncand$).

```
1  $C_{s,d} \leftarrow \emptyset$
2  $\hat{C}_{s,d} \leftarrow \emptyset$; $m_p \leftarrow \infty$
3  if $s == d$ then
4      $cost(s) \leftarrow 0$
5      return
6  end
7  forall the $v \in N(s)$ do
8      if $ETX(v,d) < ETX(s,d)$ then
9          $\hat{C}_{s,d} \leftarrow \hat{C}_{s,d} \cup \{v\}$
10     end
11 end
12 while $|C_{s,d}| < ncand$ // search for the best candidate
13 do
14     $cand \leftarrow \arg\min_{c \in \hat{C}_{s,d}} EAX(C_{s,d} \cup \{c\}, s, d)$
15     $m_c \leftarrow EAX(C_{s,d} \cup \{cand\}, s, d)$
16     if $m_c < m_p$ then
17         $C_{s,d} \leftarrow C_{s,d} \cup \{cand\}$
18         $\hat{C}_{s,d} \leftarrow \hat{C}_{s,d} \setminus \{cand\}$
19         $m_p \leftarrow m_c$
20     else
21         $cost(s) \leftarrow m_p$
22         $break$
23     end
24 end
25 $C_{s,d} \leftarrow C_{s,d}$ ordered by $cost$
```

the best candidate is the one that reduces the most the expected number of transmissions from s to the destination (line 14). Node s adds the best candidate to its actual CS ($C_{s,d}$) and removes it from its initial set. Note that to find the best candidate in each iteration, candidates should be ordered according to their EAX. It tries again to find the best node from its new initial CS. This process is repeated until there is not any other suitable node to be included in the candidates set of s, or the number of candidates in the $C_{s,d}$ reaches the maximum number of candidates ($ncand$). Finally, the CS is ordered by EAX of each candidate.

Now assume that node S in figure 3.7 wants to find its CS using OAPF. First, it creates its initial CS $\hat{C}_{S,D}$. Since the ETX of all its neighbors (A, B and D) to the destination D is less than ETX(S,D) (see table 3.2) 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 3.3 we summarize the CS and related expected number of transmissions for node A and B.

Table 3.4 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 section iteration of while-loop in algorithm 2 (line 12-24), source looks for the second candidate from the remaining potential

Table 3.3: Candidates set of A and B in figure 3.7 using OAPF

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

candidates in $\hat{C}_{S,D} = \{A, D\}$. As we can see in table 3.4, 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.

3.4.3 Least-Cost Opportunistic Routing (LCOR)

The goal of this algorithm is to find the optimal CSs. Recall that the optimal CSs are the sets that minimize the expected number of transmissions from the source to the destination. LCOR [15] uses EAX as the metric to select candidates as shown in Algorithm 3. It works similar to the classical distributed Bellman-Ford algorithm.

The algorithm proceeds iteratively and at each iteration an exhaustive search over all possible CSs is carried out. It starts by initializing the cost (EAX) of each node v to reach the destination d (lines 1–4). Since in the initializing phase the 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 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 not exceeding $ncand$ (line 9). The algorithm terminates when the cost to reach the destination does not change for all nodes in two consecutive iterations (lines 12–17).

In each iteration the algorithm checks for all the nodes but the destination, all subsets of their neighbors with cardinality $\leq ncand$. Therefore, for dense networks the computational cost of the algorithm increases extremely fast due to the combinatorial explosion of the exhaustive search of line 9. In section 3.9.8, we will carry out an experimental evaluation of the computational time.

Applying LCOR on the topology in figure 3.7 yields as a result $C_{S,D} = \{D, A\}$ with the expected number of transmissions equal to 3.36.

Table 3.4: 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 3: Candidate.selection.LCOR(s, d, $ncand$).

```
forall the $v$ in the network $\setminus\{d\}$ do
    $cost_{curr}(v) \leftarrow \infty$; $cost_{prev}(v) \leftarrow \infty$
end
$cost_{curr}(d) \leftarrow 0$
repeat
    flag ← TRUE
    forall the $v$ in the network $\setminus\{d\}$// search for the best CS
    do
        $C_{v,d} \leftarrow \underset{S \in 2^{|N(v)|}, 0<|S| \leq ncand}{\arg\min} EAX(S, v, d)$
        $cost_{curr}(v) \leftarrow EAX(C_{v,d}, v, d)$
    end
    forall the $v$ in the network $\setminus\{d\}$ do
        if $cost_{curr}(v) \neq cost_{prev}(v)$ then
            $cost_{prev}(v) \leftarrow cost_{curr}(v)$
            flag ← FALSE
        end
    end
until $flag == TRUE$
$C_{s,d} \leftarrow C_{s,d}$ ordered by $cost_{curr}$
```

3.4.4 Minimum Transmission Selection (MTS)

MTS [29, 45] is another algorithm which selects the optimal CSs for any node to a given destination d that minimizes the total expected number of transmissions. Like LCOR, this algorithm proceeds iteratively and uses EAX as the metric for selecting the candidates sets.

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. Note that, the closest node to the destination has the least EAX. MTS uses 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. The algorithm starts by initializing the cost (EAX) of each node v to reach the destination d (lines 2–10 in Algorithm 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 minimum cost in terms of the expected number of transmissions to the destination (line 12). The neighbors of *minnode*, $N(minnode)$, add *minnode* and its candidates to their CS and *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).

Algorithm 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          $cost(v) \leftarrow \frac{1}{q_{vd}}$
5          $C_{v,d} \leftarrow d$
6      else
7          $C_{v,d} \leftarrow \emptyset$
8          $cost(v) \leftarrow \infty$
9      end
10 end
11 while $\mathbb{S}$ is not empty do
12     $minnode \leftarrow \arg\min_v cost(v)$
13     $\mathbb{S} \leftarrow \mathbb{S} \setminus \{minnode\}$
14     forall the $v \in N(minnode)$ do
15         $C_{v,d} \leftarrow merge(C_{v,d}, minnode, C_{minnode,d})$
16         $cost(v) \leftarrow EAX(C_{v,d}, v, d)$
17     end
18 end
19 $\mathbb{S} \leftarrow$ all nodes in the network $\setminus\{d\}$ ordered by $cost$
20 forall the $v \in \mathbb{S}$ do
21     $C_{v,d} \leftarrow \arg\min_{T \in C_{v,d}, |T| \leq ncand} EAX(T, v, d)$
22 end
```

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 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 19–22. 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 MTS first find the optimal CSs in the case of *infinite candidates* (i.e. all possible nodes can be selected as candidates), and then we look for the best subset of candidates sets with at most *ncand* elements, the final CSs will be the optimal CSs.

Like the previous algorithms, we apply MTS in the example of figure 3.7 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 3.5, 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 removes 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 3.5). Note that in each iteration the EAX of each node is updated according to the new CS. In

Table 3.5: Candidates Set selection for the figure 3.7 using MTS

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

Table 3.6: Candidates set and EAX of each node in figure 3.7 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 second iteration 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 3.7 using different algorithms under study in table 3.6. The first item in each cell is the CS for the corresponding node, and the second item is the expected number of transmissions (EAX) of the corresponding node using the said set. As we can see in table 3.6 the algorithms that use EAX to select CSs have better expected number of transmissions than ExOR which uses ETX for selection of candidates.

3.5 Network Coding Opportunistic Routing

MORE [11] is an OR protocol that can be used in both unicast and multicast scenarios. It deploys the advantages of network coding 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 [20, 21] is a practical network coding 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 [20]. 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 [41, 27], which are aimed at improving the network throughput by combining routing with inter-flow network coding. However, neither COPE nor its variants consider the opportunistic characteristic of a wireless channel in packet delivery and the dynamic availability of network coding opportunities at a network node, which can be caused by traffic and network dynamics. This largely limits their capabilities for improving network throughput.

CORE [43, 42] is a coding-aware OR mechanism that combines opportunistic forwarding and localized inter-flow network coding for improving the throughput performance of a MWN network. Through opportunistic forwarding, CORE allows the next-hop node with the most coding gain to continue the packet forwarding. Through localized network coding, 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.

3.6 Geographic Opportunistic Routing

Geographic Random Forwarding (GeRaF) [50] 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 CS and prioritization can easily be implemented via an RTS-CTS dialog at the MAC layer, which also ensures that a single forwarder can be chosen.

GOR [47] 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. [44] 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 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.

3.7 Multicast Opportunistic Routing

Multicast is an important communications paradigm in wireless networks. 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.

In [22] 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 network coding to improve multicast efficiency and simplify node coordination.

The authors in [38] 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.

In a recent work, Le and Liu [28] propose an overlay multicast to adapt OR in wireless network. They construct a minimum overlay Steiner tree, and map it into unicast OR relay path connecting the source with all destinations. Their protocol does not exploit opportunistic receptions across different links in that tree. They employed unicast OR on each link of the tree.

In [13], we propose a new multicast routing protocol based on OR named *Multicast Opportunistic Routing Protocol*, MORP. It opportunistically employs a set of forwarders to send a packet toward all destinations. MORP uses a three-way-handshaking where the sending node selects the candidates and towards which destinations they have to forward the packet. The basic idea of MORP is to form a CS to reach the destinations and based on the candidates which successfully receive the packet, selects a set of candidates as the forwarders to reach all destinations. Each forwarder is responsible for sending the packet to a subset of destinations. Indeed, based on the candidates that successfully receive the packet in each transmission, MORP builds a multicast tree on the fly using OR and forwards the packet through the tree.

3.8 Opportunistic Routing Analytical Models

There are some papers which propose analytical models to study the performance of OR. Baccelli et al. [4] 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 [33] 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.

Zubow et al. in [23] 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 [40] 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 [32] 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 [10] we proposed a Markov model to assess the improvement that may be achieved using OR. This model is the basis of the analytic approach described in section 3.9.1. At the same time, Li and Zhang published an analytical framework to estimate the transmission costs of packet forwarding in wireless networks [30]. 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 [30] transmission costs are computed using spectral graph theory. In [9] we have derived the equations that yield the distances of the candidates in OR such that the per transmission progress towards the destination is maximized. There, we have proposed a lower bound to the expected number of transmissions needed to send a packet using OR.

In [8], 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.

3.9 Performance Evaluation

In this section we propose a simple Markov chain model to study the performance of the candidate selection algorithms described in section 3.4. Our model can be used to compute the probability distribution and moments of the number of transmissions needed to send a packet from the source to the destination in a variety of scenarios. 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.

3.9.1 Markov Model

We will consider 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 $x = D/(N-1)$, being D the distance between the source s and the destination d of the tagged connection (see figure 3.8).

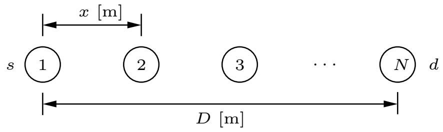


Figure 3.8: Linear network topology.

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 transmissions N_t in uni-path routing is given by:

$$N_t = \frac{N-1}{p(x)}. \tag{3.2}$$

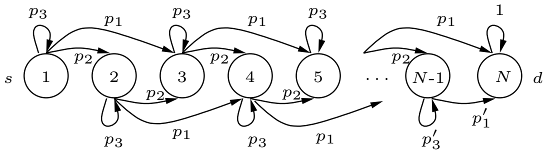


Figure 3.9: Opportunistic routing model with 2 candidates.

Assume now that 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 towards the destination. We can model this routing by means of the absorbing DTMC depicted in figure 3.9. The transition probabilities are given by:

$$\begin{aligned} p_1 &= p(2\,x) \\ p_2 &= p(x)\,(1-p_1) \\ p_3 &= 1-(p_1+p_2) \\ p'_1 &= p(x) \\ p'_3 &= 1-p'_1. \end{aligned} \tag{3.3}$$

The DTMC models the progress of a packet from source to destination. The initial state is 1 and a transition occurs at each transmission shot. When the DTMC is in state i it represents that the packet has progressed up to node i. Eventually the packet reaches its destination when the DTMC is absorbed at state N.

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

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:

$$p_{ij} = p(i, j), \qquad i \neq d,\ j = c_i(1) \tag{3.4}$$

$$p_{ij} = p(i, j) \prod_{l=1}^{k-1} (1 - p(i, c_i(l))), \qquad i \neq d,\ j = c_i(k),\ k = 2, \cdots, n_i \tag{3.5}$$

$$p_{ii} = 1 - \sum_{l \in C_{i,d}} p_{il} = \prod_{l=1}^{n_i} (1 - p(i, c_i(l))),\ i \neq d \tag{3.6}$$

$$p_{ii} = 1, \qquad i = d\,. \tag{3.7}$$

Note that the two expressions given for p_{ii} in equation (3.6) 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: $c_i > i \quad \forall c_i \in C_{i,d}$. 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 $i = j_0 \to j_1 \to j_2 \to \cdots \to j_n = i$ (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 itself. With these assumptions, the transition matrix of the resulting chain has the triangular form:

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & \cdots & p_{1N} \\ 0 & p_{22} & p_{23} & \cdots & p_{2N} \\ 0 & 0 & p_{33} & \cdots & p_{3N} \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{T} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \quad . \tag{3.8}$$

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 [26]. Thus, the point probabilities and factorial moments of X_1 are given by:

$$P\,\{X_1 = n\} = \boldsymbol{\tau}\,\mathbf{T}^{n-1}\,\mathbf{t},\; n \geq 1 \tag{3.9}$$

$$\mathrm{E}[X_1(X_1-1)\cdots(X_1-k+1)] = n!\,\boldsymbol{\tau}\,(\mathbf{I}-\mathbf{T})^{-n}\,\mathbf{T}^{n-1}\,\mathbf{1} \tag{3.10}$$

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.

3.9.2 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:

$$\mathrm{E}[X_i] = \frac{1 + \sum_{l=1}^{n_i} p_{il}\,\mathrm{E}[X_l]}{1 - p_{ii}}, \quad i \neq d\,. \tag{3.11}$$

Taking $\mathrm{E}[X_d] = 0$, the equation (3.11) 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 CS of s to reach node d. Note that the loop free property of the chain guarantees that the recursive equation (3.11) is finite.

Equation (3.11) has been obtained by other methods in [16] and [49], where it is referred to as *least cost any-path* and *expected any-path transmissions* (EAX) respectively. We shall adopt the acronym EAX to refer to it. As explained in section 3.4, 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.

3.9.3 Propagation Model

In order to assess the delivery probabilities we will assume that the channel impediments are characterized by a shadowing propagation model: the power received at a distance x is given by:

$$P_r(x)|_{dB} = 10\,\log_{10}\left(\frac{P_t\,G_t\,G_r\,\lambda^2}{L\,(4\,\pi)^2\,x^\beta}\right) + X_{dB} \tag{3.12}$$

where P_t is the transmitted power, 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), β

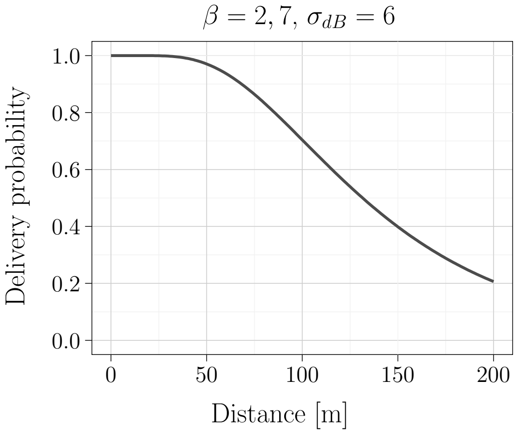


Figure 3.10: Delivery probability versus distance for a path loss exponent $\beta = 2,7$ and standard deviation $\sigma_{dB} = 6$ dBs.

is a path-loss exponent and X_{dB} is a Gaussian random variable with zero mean and standard deviation σ_{dB}.

Packets are correctly delivered if the received power is greater than or equal to `RXThresh`. Note that we shall not consider collisions in our model. Thus, the delivery probability at a distance x ($p(x)$) is given by:

$$p(x) = Prob(P_r(x)|_{dB} \geq 10\,\log_{10}(\texttt{RXThresh})) = \\ Q\left(\frac{1}{\sigma_{dB}} 10\,\log_{10}\left(\frac{\texttt{RXThresh}\,L\,(4\,\pi)^2\,x^{\beta}}{P_t\,G_t\,G_r\,\lambda^2}\right)\right) \tag{3.13}$$

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

We have set the model parameters to the default values used by the network simulator (ns-2) [1], given in table 3.7. We shall use these values in the numerical results presented in section 3.9. With these parameters the link delivery probability is approximately 40% at the distance of 150 m.

Table 3.7: Default values in *ns-2* 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

3.9.4 Evaluation Methodology

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 3.11–3.19).

We have proceed has 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 ($10 \leq N \leq 50$).
- The shadowing propagation model described in section 3.9.3 is used to assess the delivery probabilities of the links (q_{ij} in the model described in section 3.9.1). 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 section 3.4 to compute the CSs for the given destination.
- Finally, we use the DTMC model described in section 3.9.1 to compute the following performance measures: expected number of transmissions, variance of the expected number transmissions, probability of the number of transmissions.

We have done this evaluation using the R numerical tool [34]. 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 section 3.9.1, 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 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 3.3 GHz, FSB 1333 MHz, with 4 MB cache and 12 GB of memory.

3.9.5 Numerical Results

Expected Number of Transmissions

First, we we examine in detail the case with at most 3 candidates for each node ($ncand = 3$), as shown in figure 3.11. 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.

As a first observation in figure 3.11, 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.

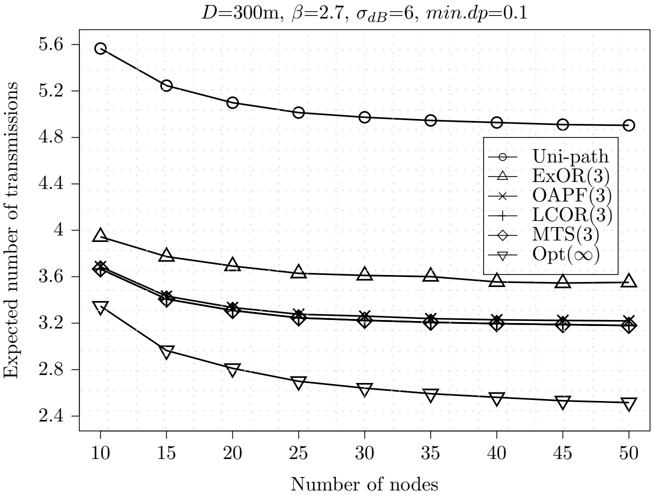


Figure 3.11: Expected number of transmissions in the case $ncand = 3$.

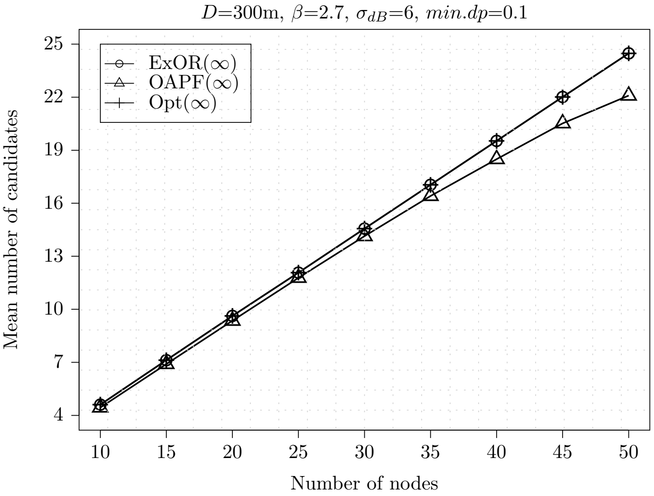


Figure 3.12: Mean number of candidates in the case $ncand = \infty$.

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 [9] 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 number of transmissions (EAX). Although this does not guarantee choosing the optimal CSs, we can see from the figure 3.11 that the results are very close to the optimal algorithm.

Regarding the scenario with $ncand = \infty$, figure 3.11 shows that it achieves a noticeable reduction of the expected number of transmissions compared to the scenario with $ncand = 3$. However, as shown in figure 3.12, this is at cost of using a large number of candidates. Note that 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 3.11.

For other scenarios we have obtained similar results. For instance, figures 3.13 and 3.14 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 3.13 and 3.14 we can see that the difference between ExOR and the other algorithms is higher in a dense network ($N = 50$). 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 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.

3.9.6 Variance of Expected Number of Transmissions

One of the metrics which can also be calculated with our model is the variance of expected number of transmissions from source to the destination. Figures 3.15 and 3.16 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 3.15 and 3.16 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

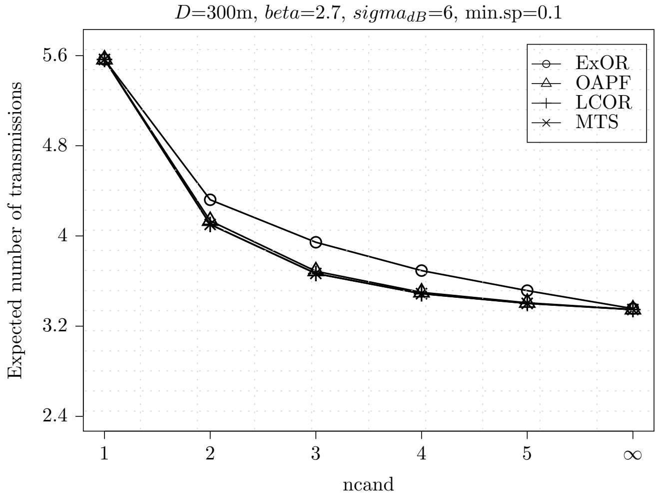


Figure 3.13: Expected number of transmissions for the random topology with $N = 10$ nodes varying the maximum number of candidates.

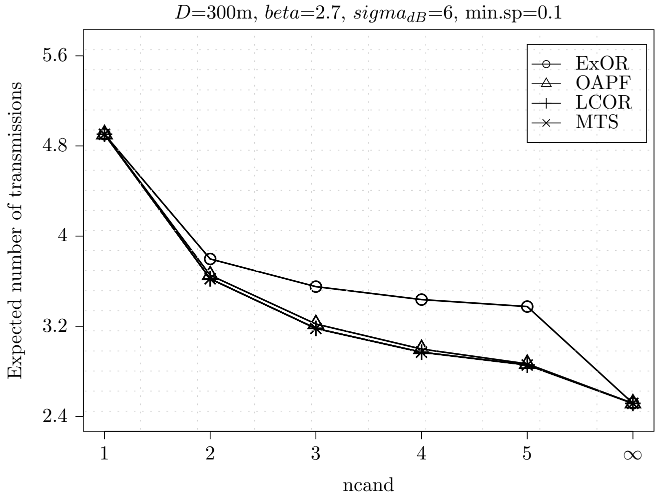


Figure 3.14: Expected number of transmissions for the random topology with $N = 50$ nodes varying maximum the number of candidates.

while the variance decreases with the value of *ncand*, just a small of value (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 unipath 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.

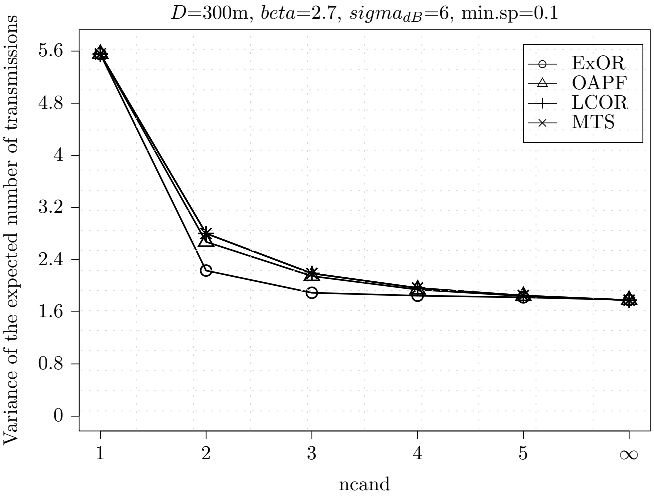


Figure 3.15: Variance of the expected number of transmissions for the random topology with $N = 10$ nodes varying the maximum number of candidates.

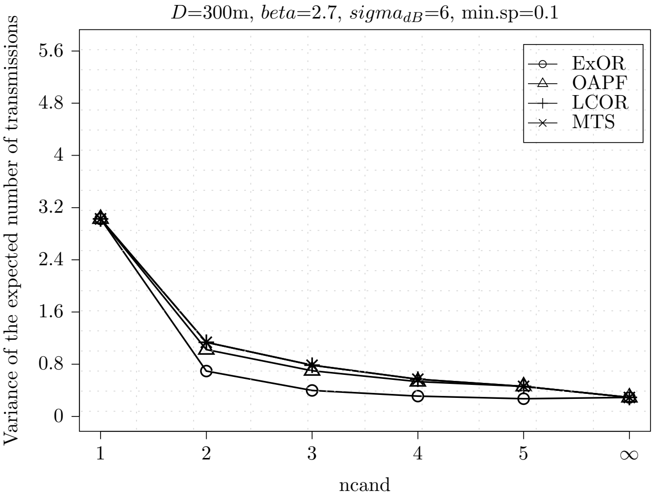


Figure 3.16: Variance of the expected number of transmissions for the random topology with $N = 50$ nodes varying the maximum number of candidates.

3.9.7 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 3.17 and 3.18, respectively.

The probability curves for the $ncand = 1$ case (uni-path routing) in both figures 3.17 and 3.18 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 3.17 and 3.18 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 3.18 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 3.17 and 3.18 we can see that the probabilities change significantly for the $ncand = \infty$ case. For instance, in figure 3.18 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 3.12 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.

3.9.8 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 necessary to solve the DTMC model described in section 3.9.1. 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 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 2, 3 and 4, respectively.

Figure 3.19 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 3.19 have been obtained by averaging over the 100 runs of the corresponding points in figure 3.11. The values next to the points represent the number of nodes of 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 candidates sets without limiting the maximum number of candidates,

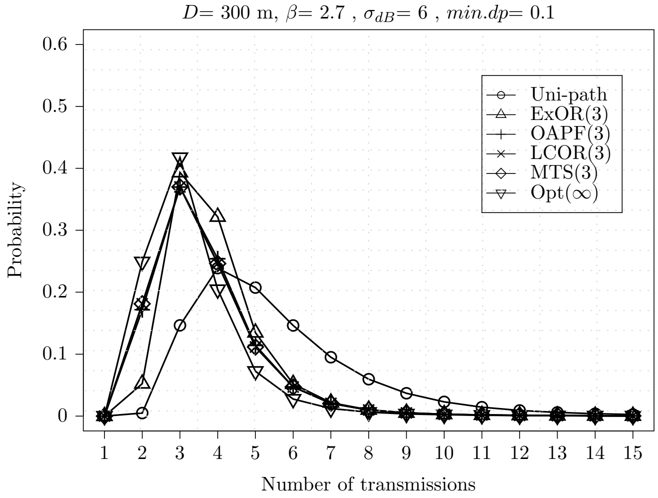


Figure 3.17: Probability of the number of transmissions for the random topology with $N = 10$ nodes

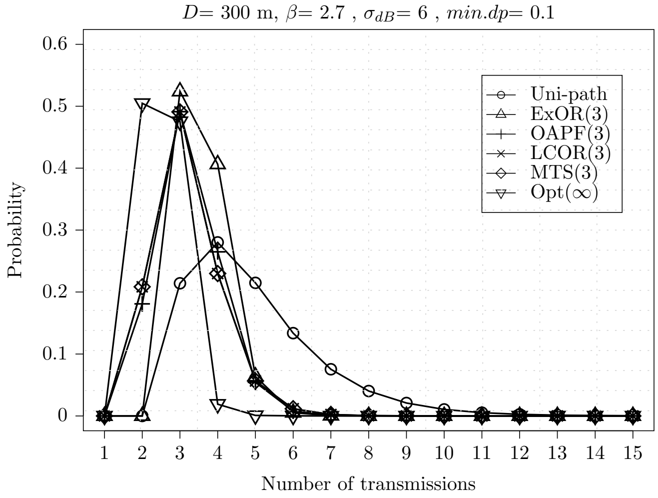


Figure 3.18: Probability of the number of transmissions for the random topology with $N = 50$ nodes

and then the candidates sets 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 with a much lower execution time. With simple algorithms as ExOR, the performance may be significantly poorer than the optimal.

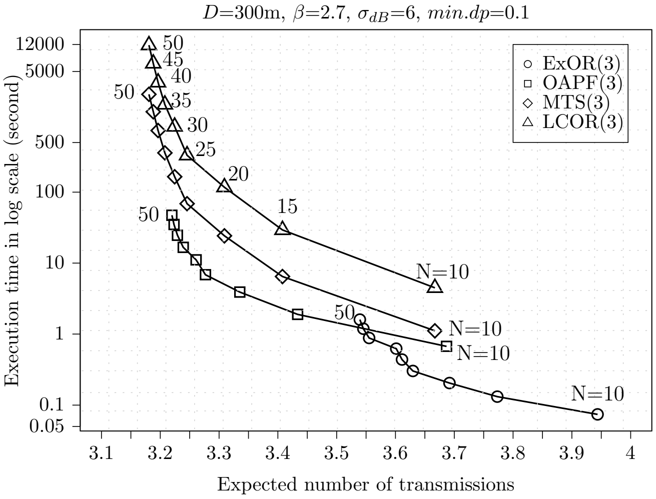


Figure 3.19: Expected number of transmissions and execution time of all algorithms.

3.10 Conclusions

In this chapter we have described the meaning of Opportunistic Routing (OR) which has been introduced as a way of using the broadcast nature of multi-hop wireless networks. We have classified different research areas in OR: routing metrics, candidate section, candidate coordination, geographic OR and multicast OR. Then, we have surveyed the main research contributions in each category.

The two usual metrics that used in the literature of OR, ETX and EAX, have been discussed in detailed. Although ETX is simpler to compute than EAX, it does not accurately compute the expected number of transmissions under OR. The other important issues of OR is the candidate selection. We have described in detail four different 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. To show the differences between each algorithm under study we have used a simple example running for each algorithm. Regarding the different candidate coordination approaches we have described the four most used methods of coordination in OR: acknowledgment-based, timer-based, network coding and RTS-CTS; and have explained the advantages and disadvantages of each approach.

Applying the OR paradigm to multicast routing is another new research direction. The availability of multiple destinations can make the selection of CS and coordination among them complicated. There are few works that have tried to adapt OR to multicast settings. We have briefly describe different protocols that apply OR to multicast routing. Most of them first create the shortest path tree to the destinations and then send the packet through the tree using OR.

After the general overview and introduction of OR we have focused on its performance analysis. First, we have surveyed the existing performance studies that are based on analytical models. Then, we have introduced our own contribution, a discrete time Markov chain model to analyze the performance gain that may be achieved by using OR. In our model the nodes are represented by the states of the chain, and the state transitions model how the packet progresses through the network. The only ingredients needed 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 have applied our model to compare four relevant algorithms that have been described in the candidate selection section. We have compared different scenarios in terms of the expected number, the variance and the probability of the number transmissions needed to send a packet from source to the destination. The algorithms have also been compared from the perspective of the the execution time which is needed to construct the CSs.

Our numerical results have shown 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 obtain 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 probably duplicate transmissions. When the maximum number of candidates is limited, our results have shown that the expected number of transmissions required by ExOR is larger than that of the other OR algorithms. This is because of the coarse selection of the CSs of ExOR, which relies on ETX. On the other hand, the performance obtained with OAPF has proven to be very close to the optimal algorithms. We have also observed that the variance of the number of transmissions can be substantially reduced by using OR. This result is specially important in networks with real-time requirements.

Regarding the execution times, the fact that EXOR is based on ETX makes this algorithm much faster than the others. For the optimum algorithms, we have observed that MTS outperforms LCOR. However, both algorithms require extremely large execution 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 times orders of magnitude lower than the optimum algorithms (on the order of minutes) while a relatively low degradation of performance was observed. Therefore, we conclude that a fast and simple OR candidate selection algorithm such as OAPF may be preferable in dynamic networks, where the CSs are likely to be updated frequently.

Bibliography

[1] The Network Simulator ns-2. 25

[2] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris. Link-level measurements from an 802.11b mesh network. *SIGCOMM Comput. Commun. Rev.*, 34:121–132, August 2004. 13

[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. 9

[4] 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. 5, 20

[5] S. Biswas and R. Morris. ExOR: opportunistic multi-hop routing for wireless networks. *ACM SIGCOMM Computer Communication Review*, 35(4):133–144, 2005. 2, 5, 11, 13

[6] Sanjit Biswas and Robert Morris. Opportunistic routing in multi-hop wireless networks. *ACM SIGCOMM Computer Communication Review*, 34(1):69–74, 2004. 5, 8, 11, 12

[7] R. Bruno and M. Nurchis. Survey on Diversity Based Routing in Wireless Mesh Networks: Challenges and Solutions. *Computer Communications*, 33(3):269–282, 2010. 9

[8] A.S. Cacciapuoti, M. Caleffi, and L. Paura. Optimal constrained candidate selection for opportunistic routing. In *IEEE GLOBECOM*, pages 1–5, December 2010. 21

[9] Llorenç Cerdà-Alabern, Amir Darehshoorzadeh, and Vicent Pla. On the maximum performance in opportunistic routing. In *World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a*, pages 1–8, 2010. 5, 21, 27

[10] 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. 7, 21

[11] 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. 5, 9, 18

[12] Amir Darehshoorzadeh and Llorenç Cerdà-Alabern. A new multicast opportunistic routing protocol for wireless mesh networks. In Vicente Casares-Giner, Pietro Manzoni, and Ana Pont, editors, *NETWORKING 2011 Workshops*, volume 6827 of *Lecture Notes in Computer Science*, pages 62–72. Springer Berlin / Heidelberg, 2011. 5, 20

[13] 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. 5

[14] 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. 6

[15] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. Least-cost opportunistic routing. In *Proceedings of 2007 Allerton Conference on Communication, Control, and Computing*, 2007. 5, 7, 11, 15

[16] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. Valuable detours: Least-cost anypath routing. *IEEE/ACM Transactions on Networking*, 19(2):333–346, April 2011. 2, 24

[17] 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. 2, 5

[18] 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, 9, 10

[19] 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. 6, 10

[20] 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. 5, 18

[21] 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. 5, 18

[22] 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. 5, 20

[23] 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. 21

[24] 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, 8

[25] 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. 5

[26] G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. ASA-SIAM, 1999. 23

[27] 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

[28] Tan Le and Yong Liu. Opportunistic overlay multicast in wireless networks. In *GLOBECOM*, pages 1–5, 2010. 5, 20

[29] Yanhua Li, Wei Chen, and Zhi-Li Zhang. Optimal forwarder list selection in opportunistic routing. In *Mobile Adhoc and Sensor Systems, 2009. MASS '09. IEEE 6th International Conference on*, pages 670–675, October 2009. 5, 7, 11, 16

[30] 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. 21

[31] 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, December 2009. 6, 11

[32] Mingming Lu and Jie Wu. Opportunistic routing algebra and its applications. In *IEEE INFOCOM*, pages 2374–2382, 2009. 7, 21

[33] 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. 5, 20

[34] 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. 26

[35] 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. 5, 13

[36] 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. 5, 6

[37] S. Cañadas-Hurtado Triviño-Cabrera, Alicia. Survey on opportunistic routing in multihop wireless networks. *IJCNIS*, 3(2), 2011. 6

[38] 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. 20

[39] 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. 6

[40] 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. 21

[41] 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

[42] 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. 5, 19

[43] 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. 5, 9, 19

[44] Shengbo Yang, Feng Zhong, Chai Kiat Yeo, Bu Sung Lee, and J. Boleng. Position based opportunistic routing for robust data delivery in manets. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–6, December 2009. 5, 19

[45] Wei Chen Yanhua Li and Zhi-Li Zhang. Design of forwarder list selection scheme in opportunistic routing protocol. Technical-report, University of Minnesota, Minneapolis, MN, US, 2008. 16

[46] 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. 6

[47] 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. 5, 19

[48] 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. 6

[49] 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. 5, 6, 7, 11, 13, 24

[50] 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. 5, 6, 10, 11, 19