Modeling and Comparison of Candidate Selection Algorithms in Opportunistic Routing

Amir Darehshoorzadeh[a], Llorenç Cerdà-Alabern[a,*], Vicent Pla[b]

[a] *Univ. Politècnica de Catalunya, Computer Architecture Dep. Barcelona, Spain*
[b] *Univ. Politècnica de València, Dept. of Communications Valencia, Spain*

Abstract

Opportunistic Routing (OR) has been investigated in recent years as a way to increase the performance of multihop wireless networks by exploiting its broadcast nature. In contrast to traditional routing, where traffic is sent along pre-determined paths, in OR an ordered set of candidates is selected for each next-hop. Upon each transmission, the candidates coordinate such that the most priority one receiving the packet actually forwards it. Most of the research in OR has been addressed to investigate candidate selection algorithms. In this paper we propose a discrete time Markov chain to assess the improvement that may be achieved using opportunistic routing. We use our model to compare a selected group of candidate selection algorithms that have been proposed in the literature. Our main conclusion is that optimality is obtained at a high computational cost, with a performance gain very similar to that of much simpler but non optimal algorithms. Therefore, we conclude that fast and simple OR candidate selection algorithms may be preferable in dynamic networks, where the candidates sets are likely to be updated frequently.

Keywords: Wireless Networks, Opportunistic routing, Performance modeling, Markov Chain, Candidate selection algorithms

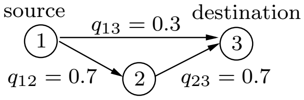
1. Introduction

During the last years multi-hop wireless networks (MWN) have become very popular, 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 potentially be received by its neighbors.

Routing protocols in MWN have traditionally coped with heterogeneous characteristics of the wireless links using distributed protocols that at each node choose for every destination

[*] Corresponding Author

Email addresses: amir@ac.upc.edu (Amir Darehshoorzadeh), llorenc@ac.upc.edu (Llorenç Cerdà-Alabern), vpla@dcom.upv.es (Vicent Pla)



node	next-hop to 3	candidates to 3
1	2	3, 2
2	3	3

(a) Uni-path (b) OR

Figure 1: Example topology.

the best link. Doing this way, in the routing table of every node participating in the routing between a source and a destination, there is the link to a neighbor (referred to as *next-hop*), over which packets addressed to the destination will be forwarded. 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.

Opportunistic Routing (OR), also referred to as diversity forwarding [1], cooperative forwarding [2] or any-path routing [3], has been proposed 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. 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 would actually forward it, while the others will simply discard the packet.

The following example illustrates how OR works, and its potential to improve the network performance. Consider the network topology of figure 1. 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. Table (a) in figure 1 shows the entries that would be added to the routing tables of nodes 1 and 2 to reach node 3 by a traditional uni-path routing protocol (e.g. using *shortest path first* minimizing the expected number of transmissions). Table (b) of figure 1 shows the entries with the candidates that would be added using OR. Note that in the candidate list of node 1, node 3 has more 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 uni-path routing. We now compute the expected number of transmissions from node 1 to the destination for uni-path routing (E_1^{uni}) and OR (E_1^{OR}). Clearly, $E_1^{uni} = 1/q_{12} + 1/q_{23} \approx 2.86$. On the other hand, using 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 $E_1^{OR} \approx 2.15$. By comparing with E_1^{uni}, we obtain that using OR in this simple example could reduce the expected number of transmissions about 25 %.

One of the important issues of OR is candidate selection and relay priority assignment. 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.* Numerous routing protocols building on the idea of OR have been proposed [4, 5, 6, 7, 8]. Apart from other implementation aspects, these solutions specify how to select and prioritize the candidates. The aim of candidate selection algorithms is to minimize the expected number of transmissions from the source to the destination.

Another key issue in 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.

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, i.e., the algorithms have been designed to select all candidates, without limit, that can reduce the expected number of transmissions.

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

In the second contribution we apply our model to compare four candidate selection algorithms that have been proposed in the literature. They range from sub-optimal, but simple, to optimal, but with a high computational cost. We assume a shadowing propagation model and use our proposed model to study the expected number of transmissions and variance of the number of transmissions for the different candidate selection algorithms in a variety of scenarios. We propose modifications to the candidate selection algorithms under study in order to limit the maximum number of candidates per node. We address the questions: is there a substantial difference in performance between the simple and the optimal algorithms? What is their computational cost? So, under which conditions it is worth using an optimal algorithm?

The remainder of this paper is organized as follows. Section 2 surveys the related work. The proposed DTMC model is described in Section 3. The description of the four candidate selection algorithms under study is done in Section 4. Section 5 introduces the propagation model that we have used in our numerical experiments. The numerical results are presented and discussed in Section 6 and concluding remarks are given in Section 7.

2. Related work

Biswas and Morris proposed ExOR [9, 4], one of the firsts and most referenced OR protocols. The selection of candidates is based on the Expected Transmission Count (ETX) metric. ETX is the average number of transmissions required to reliably send a packet across a link or a uni-path route, including retransmissions [10]. The candidates are assigned a priority order according to their position for forwarding packets to the destination.

In opportunistic routing, 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. So using the ETX to compute the expected number of transmissions does not give an accurate metric for OR.

In [11], Zhong et al. proposed a new metric —*expected any-path transmission (EAX)*— that generalizes ETX to an OR framework and proposed a candidate selection and prioritization scheme based on the new metric. They analyzed the efficacy of OR by using this metric and did a comparison using the link-level measurement trace of MIT Roofnet [12].In [7, 3] a distributed algorithm for computing minimum cost opportunistic routes, which is a generalization of the well-known Bellman-Ford algorithm, is presented. The authors also alert about the risk of using too many relay candidates. In [13] Li et al. introduced a new metric —*Successful Transmission Rate (STR)*— for choosing the forwarder list. They considered multi-links contribution instead of one best link information used in [11, 14]. They proposed a fair OR (FORLC) protocol that used STR as a metric to select the forwarder list. In [8] the key problem of how to optimally select the forwarder list is addressed, and an optimal algorithm that minimizes the expected total number of transmissions is developed.

One of the important issues of opportunistic routing is the coordination between candidates in order to prevent duplicate retransmissions. Different coordination schemes have been proposed which normally rely on establishing some priority order and exchanging state information between candidates. In some of them like [9], an acknowledgment based scheme as the one used in traditional 802.11 is employed. This scheme requires each candidate who has received the data packet to broadcast an ACK in different time slots according to its priority. Each ACK contains the ID of the highest-priority successful recipient known to the ACK's sender. All the candidates listen to all ACKs before deciding whether to forward the data packet. In [15], for the coordination, explicit control packets are exchanged immediately before or after a data transmission. A RTS/CTS scheme is used to select the best next-hop forwarder between the potential ones which respond in a priority order.

In contrast to the aforementioned mechanism for coordination in conventional opportunistic routing, network coding may provide an elegant method for candidate coordination [16, 17, 18]. Each candidate can create and broadcast coded packets. MORE [17] is a MAC independent protocol that uses both the idea of OR and network coding. It avoids duplicate transmissions 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 candidates set. Each receiving node discards the packet if it is not linearly independent from the the other packets received before, or if its ID does not appear in the candidate list. Otherwise, it linearly combines the received coded packets and rebroadcasts the new packet.

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 [16]. In general, design of efficient coding-aware OR protocols needs to consider topology, traffic pattern and reduction of extra overhead [19].

There are some papers which propose analytical models to study the performance of OR. Baccelli et al. [20] 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 [21] 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. The authors of [22] proposed an utility-based model for opportunistic routing 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 [23] an algebraic approach is applied to study the interaction of OR routing algorithms and routing metrics. Zubow et al. in [24] 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. In [25] we proposed a Markov model to assess the improvement that may be achieved using opportunistic routing. This model is the basis of the analytic approach described in Section 3. At the same time, Li and Zhang published an analytical framework to estimate the transmission costs of packet forwarding in wireless networks [26]. 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 [26] transmission costs are computed using spectral graph theory. Finally, in [27] we have derived the equations that yield the distances of the candidates in opportunistic routing (OR) such that the per transmission progress towards the destination is maximized. There, we have proposed a lower bound to the expected number of transmissions needed to send a packet using OR.

3. 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 2).

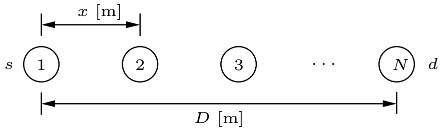


Figure 2: Linear network topology.

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:

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

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

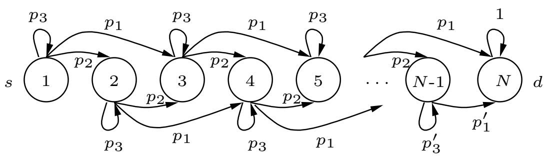


Figure 3: Opportunistic routing model with 2 candidates.

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. The transition probabilities are given by:

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

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 is readily extended to an arbitrary network. The only ingredients needed to build the transition probability matrix are the candidates sets involved in the routing from s to d, and the delivery probabilities to reach them. Notice that these candidates sets are: the candidates of node s towards d, the candidates of these candidates towards d, and so on until d (whose candidates is the empty set). This is explained in the following.

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

$$p_{ij} = p(i,j), \qquad i \neq d,\ j = c_i(1) \tag{3}$$

$$p_{ij} = p(i,j) \prod_{l=1}^{k-1} (1 - p(i, c_i(l))), \qquad i \neq d,\ j = c_i(k),\ k = 2, \cdots, n_i \tag{4}$$

$$p_{ii} = 1 - \sum_{l \in C_{i,d}} p_{il} = \prod_{l=1}^{n_i} (1 - p(i, c_i(l))),\ i \neq d \tag{5}$$

$$p_{ii} = 1, \qquad i = d\,. \tag{6}$$

Note that the two expressions given for p_{ii} in equation (5) follow from the stochastic nature of the transition matrix (the first one), and because p_{ii} is the probability that none of the candidates ($C_{i,d}$) receives the packet (the second one).

Without loss of generality, we can number the nodes such that the source and the destination are respectively 1 and N, and for for any node i, its candidates satisfy: $c_i(l \in C_{i,d}) > i$.

Note that, neglecting self-transitions, the former condition implies that the graph is loop free. This condition holds assuming that the candidate selection algorithm uses some kind of strict order, i.e., for a node j to be included into the set of candidates of i it must be strictly *closer* to the destination than i is. 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:

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & \cdots & p_{1N} \\ 0 & p_{22} & p_{23} & \cdots & p_{2N} \\ 0 & 0 & p_{33} & \cdots & p_{3N} \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{T} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \quad . \tag{7}$$

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

Let X_1 be the random variable equal to the number of transitions from the source (node 1) until absorption. Note that in our model this is the number of transmissions since the source first transmits the packet, until it is received by the destination. The DTMC obtained in our model represents a discrete phase-type distribution [28]. Thus, the point probabilities and factorial moments of X_1 are given by:

$$P\{X_1 = n\} = \boldsymbol{\tau}\, \mathbf{T}^{n-1}\, \mathbf{t},\ n \geq 1 \tag{8}$$

$$\mathrm{E}[X_1(X_1-1)\cdots(X_1-k+1)] = n!\, \boldsymbol{\tau}\, (\mathbf{I} - \mathbf{T})^{-n}\, \mathbf{T}^{n-1}\, \mathbf{1} \tag{9}$$

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

3.1. Expected Number of Transmissions

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

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

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

$$\mathrm{E}[X_i] = \frac{1 + \sum_{l=1}^{n_i} p_{il}\, \mathrm{E}[X_l]}{1 - p_{ii}}, \quad i \neq d\,. \tag{10}$$

Taking $\mathrm{E}[X_d] = 0$, the equation (10) can be used to compute the expected number of transmissions needed to send a packet from the source s to the destination d by using $C_{s,d}$ as the candidates set of s to reach node d. Note that the loop free property of the chain guarantees that the recursive equation (10) is finite.

Equation (10) has been obtained by other methods in [3] and [11], 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 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. Due to this reason, some authors have shown that using ETX may give suboptimal selection of candidates (see e.g. [7, 8]).

4. Candidate Selection Algorithms

In this section, we briefly describe the candidate selection algorithms under study. These algorithms are: extremely Opportunistic Routing (ExOR) [9]; Opportunistic Any-Path Forwarding (OAPF) [11]; Least-Cost Opportunistic Routing (LCOR) [7]; and Minimum Transmission Selection (MTS) [8].

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

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

- *ncand* is the maximum number of candidates per node. We shall refer as $ncand = \infty$ to the case when the maximum number of candidates is not limited. We shall also use the notation ExOR(n) to refer to ExOR with $ncand = n$, and similarly for the other algorithms under study (see the legend of figures 5–8).
- $ETX(v, d)$ 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 candidates set 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 the implementation that we have done for each of the selection algorithms under study. For the sake of being precise, we shall give a pseudocode summarizing our implementations.

4.1. ExOR

ExOR [9] uses ETX as the metric for selecting candidates. Algorithm 1 shows our implementation of ExOR. 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). In Algorithm 1 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 candidates set.

Every node s except the destination d runs this algorithm.

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

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

4.2. OAPF

This algorithm [11] is a hop-by-hop opportunistic routing which is based on ETX and EAX. The pseudocode of OAPF is shown in Algorithm 2. Assume that node s wants to select its candidates set to reach the destination d. It creates an initial candidates set ($\hat{C}_{s,d}$). A neighbor v of s is included in the initial candidates set only if $\text{ETX}(v,d) < \text{ETX}(s,d)$. The actual candidates set of s will be a subset of the initial candidates set. Note that, all nodes in the initial candidates set must select their candidates sets before s. This is done by the recursive call in line 15.

After initiating the candidates set, s selects the best candidate among the nodes in the initial candidates set. Here, the best candidate is the one that reduces the most the expected number of transmissions from s to the destination (line 19). Node s adds the best candidate to its actual candidates set ($C_{s,d}$) and removes it from its initial set. It tries again to find the best node from its new initial candidates set. 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 candidates set is ordered by EAX of each candidate.

4.3. LCOR

The goal of this algorithm is to find the optimal candidates sets. Recall that the optimal candidates sets are the sets that minimize the expected number of transmissions from source to the destination. It works similar to the classical distributed Bellman-Ford algorithm. LCOR [7] uses EAX as the metric to select candidates as shown in Algorithm 3.

The algorithm proceeds iteratively and at each iteration an exhaustive search over all possible candidates sets 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 candidates sets for all nodes are empty, the cost to reach the destination for all nodes is equal to ∞ ($cost_{curr}(v) \leftarrow \infty$). Note that the cost for the destination d is always equal to 0 ($cost_{curr}(d) \leftarrow 0$).

To find the optimal candidates sets 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

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

```
Require: $flag(v) = FALSE$ for all nodes $v$ in the network.
 1: $C_{s,d} \leftarrow \emptyset$
 2: if $s == d$ then
 3:     $cost(s) \leftarrow 0$
 4:     $flag(s) = TRUE$
 5:     return
 6: end if
 7: $\hat{C}_{s,d} \leftarrow \emptyset$; $m_p \leftarrow \infty$
 8: for all $v \in N(s)$ do {Initialization}
 9:     if $ETX(v,d) < ETX(s,d)$ then
10:         $\hat{C}_{s,d} \leftarrow \hat{C}_{s,d} \cup \{v\}$
11:     end if
12: end for
13: for all $v \in \hat{C}_{s,d}$ do
14:     if $flag(v) == FALSE$ then
15:         CALL Candidate.selection.OAPF($v$, $d$, $ncand$)
16:     end if
17: end for
18: while $|C_{s,d}| < ncand$ do {search for the best candidate}
19:     $cand \leftarrow \arg\min_{c \in \hat{C}_{s,d}} EAX(C_{s,d} \cup \{c\}, s, d)$
20:     $m_c \leftarrow EAX(C_{s,d} \cup \{cand\}, s, d)$
21:     if $m_c < m_p$ then
22:         $C_{s,d} \leftarrow C_{s,d} \cup \{cand\}$
23:         $\hat{C}_{s,d} \leftarrow \hat{C}_{s,d} \setminus \{cand\}$
24:         $m_p \leftarrow m_c$
25:     else
26:         $cost(s) \leftarrow m_p$
27:         $flag(s) = TRUE$
28:         break
29:     end if
30: end while
31: $C_{s,d} \leftarrow C_{s,d}$ ordered by cost
```

cardinality not exceeding *ncand* (line 8). The algorithm terminates when the cost to reach the destination does not change for all nodes in two consecutive iterations (lines 11-16). LCOR will be finished after at most N iterations [7], where N is the number of nodes in the network.

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 8. In Section 6.4, we will carry out an experimental evaluation of the computational time.

4.4. MTS

MTS [8] is another algorithm which selects the optimal candidates set for any node to a given destination d. Like LCOR, this algorithm proceeds iteratively and uses EAX as the metric for selecting the candidate 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, and using the following principle: if u and

Algorithm 3 Candidate selection LCOR(s, d, $ncand$).

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

v are neighbors and $EAX(C_{u,d}, u, d) < EAX(C_{v,d}, v, d)$, then adding u and its candidates to the candidates set of node v will reduce the expected number of transmissions from v to d, i.e. $EAX(C_{v,d} \cup \{u\} \cup C_{u,d}, v, d) < EAX(C_{v,d}, v, d)$.

The algorithm starts by initializing the cost (EAX) of each node v to reach the destination d (lines 1-11 in Algorithm 4). If d is one of the neighbors of v, then v adds the destination to its candidates set 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 $C_{v,d} = \{d\}$).

In each iteration the algorithm looks for the node with minimum cost to the destination (*minnode* in Algorithm 4). The neighbors of *minnode*, $N(minnode)$, add *minnode* and its candidates to their candidates set. This process is done by means of the function *merge*, which combines both candidates sets and order them in increasing order of their cost (EAX). In the original version of MTS [8], even the candidates of *minnode* that are not neighbors of v are added as candidates of v. In our implementation, the candidates sets of any node v are populated only with nodes that belong to its neighborhood, $N(v)$. Note that proceeding this way, MTS finishes in $N - 1$ iterations, where N is the number of nodes in the network.

In order to limit the maximum number of candidates, maintaining the optimality of the algorithm, we have added the lines 20–23. Here the nodes are visited in increasing order of their cost, and an exhaustive search is done over all subsets of the candidates sets with cardinality $\leq$ *ncand*. Since we first find the optimal candidates sets 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 candidates sets will be the optimal candidates sets.

Algorithm 4 Candidate selection MTS (s, d, $ncand$).

```
$cost(d) \leftarrow 0$
$S \leftarrow$ all nodes in the network $\setminus\{d\}$
for all $v \in S$ do
    if $v \in N(d)$ then
        $cost(v) \leftarrow \frac{1}{q_{vd}}$
        $C_{v,d} \leftarrow d$
    else
        $C_{v,d} \leftarrow \emptyset$
        $cost(v) \leftarrow \infty$
    end if
end for
while $S$ is not empty do
    $minnode \leftarrow \arg\min_v cost(v)$
    $S \leftarrow S \setminus \{minnode\}$
    for all $v \in N(minnode)$ do
        $C_{v,d} \leftarrow merge(C_{v,d}, minnode, C_{minnode,d})$
        $cost(v) \leftarrow EAX(C_{v,d}, v, d)$
    end for
end while
$S \leftarrow$ all nodes in the network $\setminus\{d\}$ ordered by $cost$
for all $v \in S$ do
    $C_{v,d} \leftarrow \underset{T \in C_{v,d}, |T| \leq ncand}{\arg\min} EAX(T, v, d)$
end for
```

5. 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:

$$P_r(x)|_{dB} = 10\,\log_{10}\left(\frac{P_t\,G_t\,G_r\,\lambda^2}{L\,(4\,\pi)^2\,x^\beta}\right) + X_{dB} \tag{11}$$

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 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:

$$p(x) = Prob(P_r(x)|_{dB} \geq 10\,\log_{10}(\texttt{RXThresh})) = \\ Q\left(\frac{1}{\sigma_{dB}} 10\,\log_{10}\left(\frac{\texttt{RXThresh}\,L\,(4\,\pi)^2\,x^\beta}{P_t\,G_t\,G_r\,\lambda^2}\right)\right) \tag{12}$$

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) [29], given in table 1.

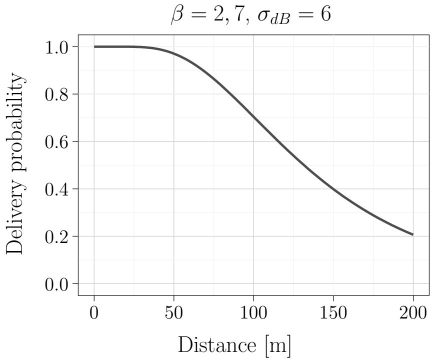


Figure 4: Delivery probability versus distance for a path loss exponent $\beta = 2,7$ and standard deviation $\sigma_{dB} = 6$ dBs.

Figure 4 depicts the delivery probability varying the distance, for a path loss exponent $\beta = 2,7$ and a standard deviation $\sigma_{dB} = 6$ dBs. We shall use these values in the numerical results presented in Section 6.

Table 1: 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

6. Results and Discussion

In this section we use the proposed model (see Section 3) to study the performance of the candidate selection algorithms described in Section 4. 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 ($10 \leq N \leq 50$).
- Then the shadowing propagation model described in Section 5 is used to assess the delivery probabilities of the links (q_{ij} in the model described in Section 3). 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 4 to compute the candidates sets for the given destination.
- Finally, we use the DTMC model described in Section 3 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 [30]. 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, 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 candidates sets in each scenario. These times have been obtained running the algorithms on a computer with an Intel Xeon Dual-Core 2 with 4 MB cache and 12 GB of memory.

6.1. 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 5. 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 5, 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 candidates sets, and thus, the curves are the same. Additionally, for $ncand = \infty$ the expected number of transmissions for LCOR and MTS are the same, so we show only one of the curves obtained with LCOR(∞) and MTS(∞) (indicated as Opt(∞)). About OAPF, we can see that the expected number of transmissions 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 [27] 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 candidates set. On the other hand, recall that OAPF incrementally adds the nodes to the candidates set that are most effective at reducing the expected number of transmissions (EAX). Although this does not guarantee choosing the optimal candidates sets, we can see from the figure 5 that the results are very close to the optimal algorithm.

Regarding the scenario with $ncand = \infty$, figure 5 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 6, this is at cost of using a large number of candidates.

As mentioned in section 2, coordination between candidates is one of the important parts of opportunistic routing protocols. It might require signaling between the candidates, and imperfect coordination may cause duplicate transmission of packets. The authors in [11, 14] claimed that the number of candidates should be kept moderate in order to reduce the number of transmissions. In fact, our results in [25] show that when the maximum number of candidates increases from 2 to 5, the reduction obtained in terms of the expected number

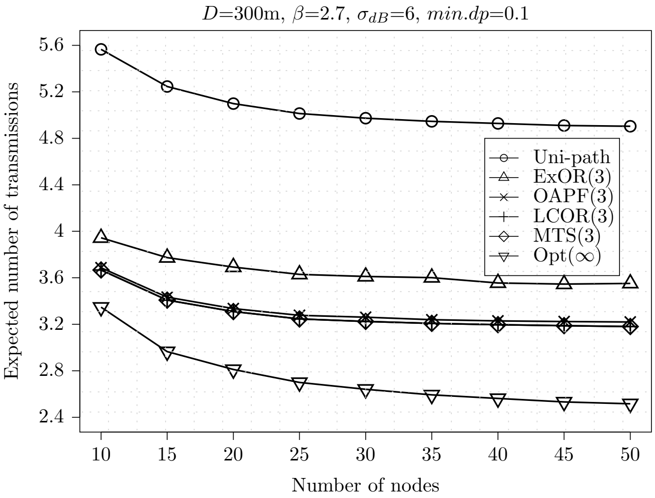


Figure 5: Expected number of transmissions in the case $ncand = 3$.

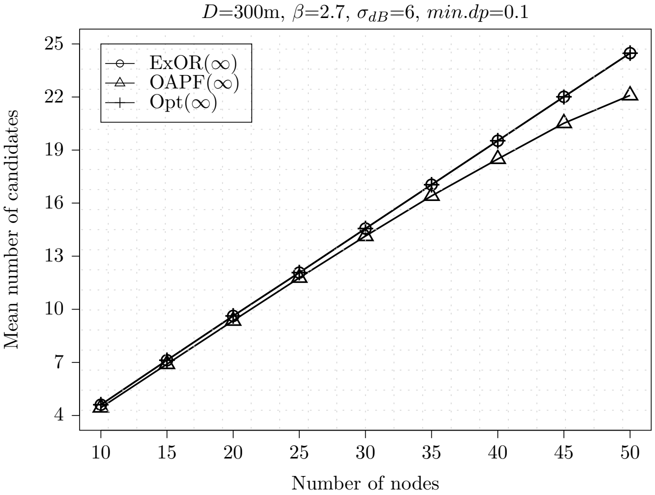


Figure 6: Mean number of candidates in the case $ncand = \infty$.

of transmissions is not remarkable. In addition, implementing an OR protocol with a large number of candidates is difficult in practice. This is because the coordination between them increases the signaling overhead and also the possibility of having duplicated transmissions, which reduces the efficency of the protocol. Therefore, we suggest that using a small number of candidates, say 2 or 3, may be a sensible choice.

For other scenarios we have obtained similar results. For instance, figures 7 and 8 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

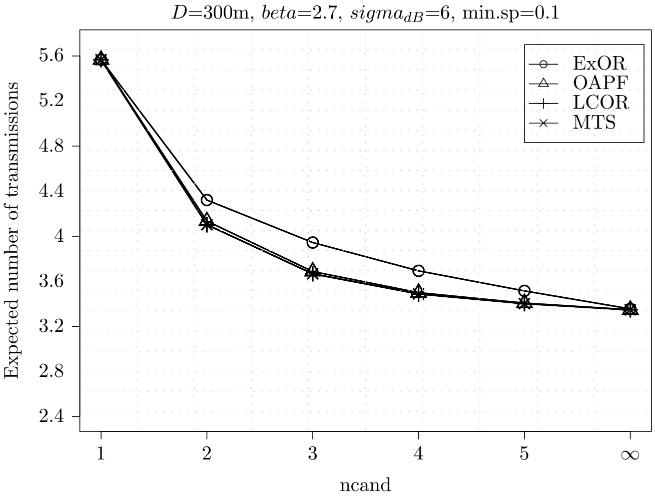


Figure 7: Expected number of transmissions for the random topology with $N = 10$ nodes varying the maximum number of candidates.

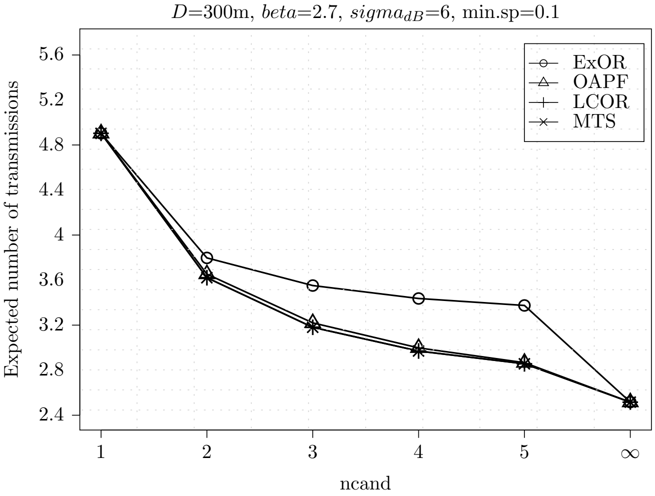


Figure 8: Expected number of transmissions for the random topology with $N = 50$ nodes varying maximum the number of candidates.

candidates sets. Note that since ExOR uses ETX as the metric to select candidates, the order of candidates may be different compared with the candidates sets in the other algorithms. Therefore the expected number of transmissions in the case of ExOR with $ncand = \infty$ has a very small difference compared with the other algorithms.

By comparing figures 7 and 8 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 candidates sets. 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.

6.2. 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 9 and 10 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 9 and 10 show that using OR the variance of the expected number of transmissions is significantly reduced compared with uni-path routing. It is also observed that while the variance decreases with the value of *ncand*, just a small 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 pronounced when the candidate selection algorithm employed is ExOR. Furthermore, although as it was shown above ExOR is the algorithm that yields the highest mean number of transmissions it produces the lowest variance.

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

6.3. Probability distribution of the number of transmissions

For having a more detailed comparison, we have included the probability distribution of the number of transmissions for $ncand = 1, 3$ and ∞ for a small number of nodes $N = 10$ and a large one $N = 50$, in figures 11 and 12, respectively.

The probability curves for the $ncand = 1$ case (uni-path routing) in both figures 11 and 12 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 11 and 12 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 12 by using these algorithms in the $ncand = 3$ case about 20% and 50% of packets reach the destination with 2 and 3 transmissions, respectively.

By comparing the figures 11 and 12 we can see that the probabilities change significantly for the $ncand = \infty$ case. For instance, in figure 12 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 6 we can see that, the $ncand = \infty$ case uses 25 candidates in a dense network ($N = 50$). With such a large number of candidates it is likely that some candidate close to the destination will receive the packet, thus, allowing the delivery to the destination with only two transmissions.

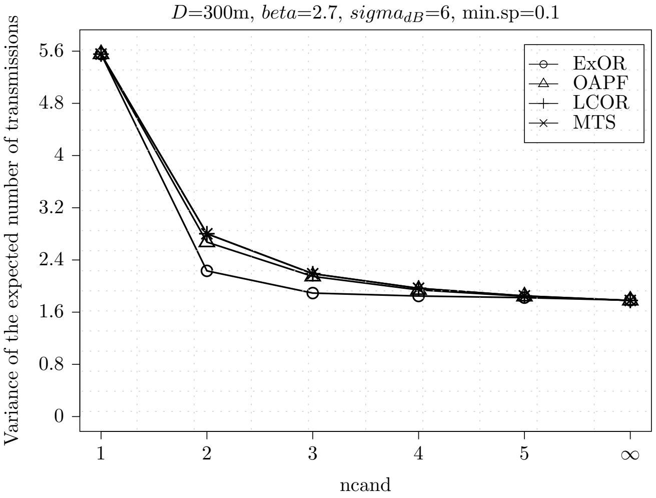


Figure 9: Variance of the expected number of transmissions for the random topology with $N = 10$ nodes varying the maximum number of candidates.

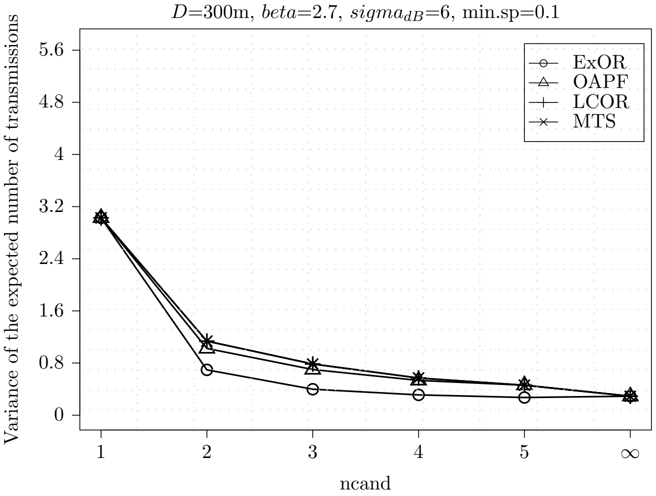


Figure 10: Variance of the expected number of transmissions for the random topology with $N = 50$ nodes varying the maximum number of candidates.

6.4. Execution Time

In this section we estimate the computational cost of the algorithms under study by measuring the *execution time* it takes to compute the candidates sets towards the destination necessary to solve the DTMC model described in Section 3. Recall that these candidates sets 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 candidates sets of the source requires the computation of all the necessary candidates sets. This comes from the fact that the other algorithms are based on the EAX metric, which requires the candidates sets. Therefore, for the algorithms OAPF, LCOR and MTS, the execution time is time it takes calling only once the algorithms 2, 3 and 4, respectively.

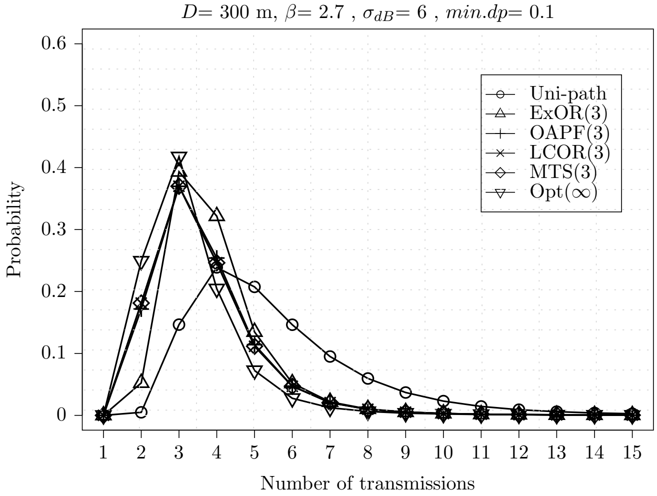


Figure 11: Probability of the number of transmissions for the random topology with $N = 10$ nodes

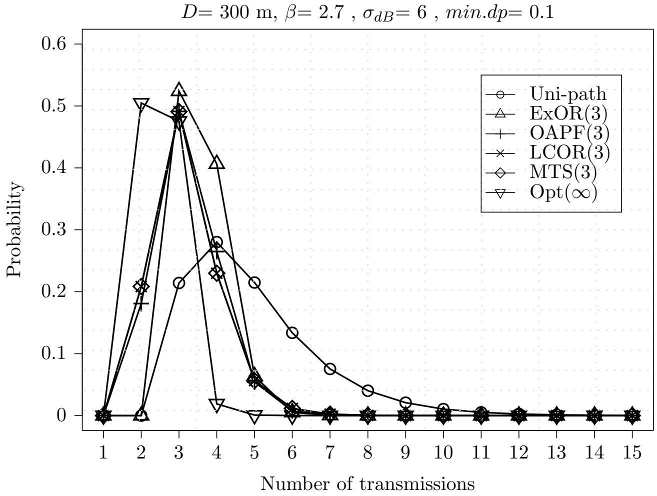


Figure 12: Probability of the number of transmissions for the random topology with $N = 50$ nodes

Figure 13 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 13 have been obtained by averaging over the 100 runs of the corresponding points in figure 5. 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 larger. 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

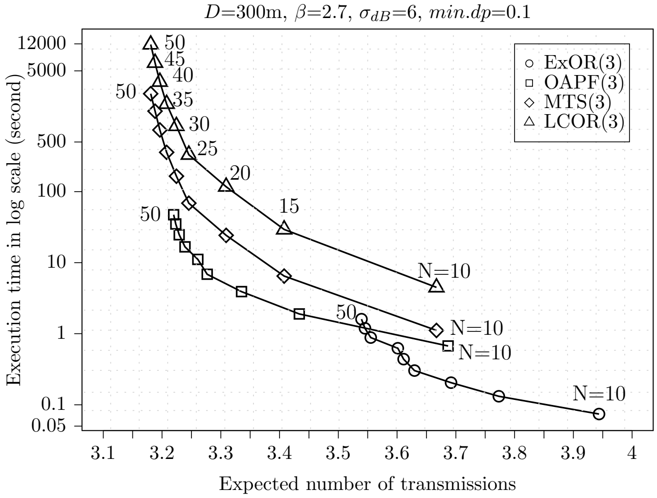


Figure 13: Expected number of transmissions and execution time of all algorithms.

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, 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 lower than the optimal.

7. Conclusions

In this paper we have proposed a discrete time Markov chain to analyze the performance gain that may be achieved by using opportunistic routing. In our model the nodes are the states of the chain, and the state transitions model how the packet progresses through the network. 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 proposed in the literature for the candidate selection in OR: extremely Opportunistic Routing (ExOR); Opportunistic Any-Path Forwarding (OAPF); Least-Cost Opportunistic Routing (LCOR); and Minimum Transmission Selection (MTS). They range from the simplicity of ExOR, the intermediate computational complexity of OAPF, to the optimal but high computational cost of LCOR and MTS. We have modified the algorithms such that the maximum number of candidates can be limited.

We have used our model to obtain numerical results for a shadowing propagation model and different candidate selection algorithms. We have compared different scenarios in terms of the expected number of transmissions needed to send a packet from source to the destination, variance of expected number of transmissions and the probability of the number of transmissions needed to reach the destination. We also computed the execution time to construct the candidates sets.

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

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

Acknowledgments

This work was supported by the Spanish government through projects TIN2010-21378-C02-01 and TIN2010-21378-C02-02, by the Generalitat de Catalunya through project 2009-SGR-1167, and by the European Commission through the NoE EuroNF.

References

1. P. Larsson, Selection diversity forwarding in a multihop packet radio network with fading channel and capture, SIGMOBILE Mob. Comput. Commun. Rev. 5 (4) (2001) 47–54.

2. D. Fuste-Vilella, J. Garcia-Vidal, J. Morillo-Pozo, Cooperative forwarding in IEEE 802.11-based manets, in: Wireless Days, 2008. WD '08. 1st IFIP, 2008, pp. 1 –5.

3. H. Dubois-Ferriére, M. Grossglauser, M. Vetterli, Valuable detours: Least-cost anypath routing, Networking, IEEE/ACM Transactions on 19 (2) (2011) 333 –346.

4. S. Biswas, R. Morris, ExOR: opportunistic multi-hop routing for wireless networks, ACM SIGCOMM Computer Communication Review 35 (4) (2005) 133–144.

5. Y. Yuan, H. Yang, S. Wong, S. Lu, W. Arbaugh, ROMER: Resilient opportunistic mesh routing for wireless mesh networks, in: IEEE Workshop on Wireless Mesh Networks (WiMesh), 2005.

6. E. Rozner, J. Seshadri, Y. Mehta, L. Qiu, Simple opportunistic routing protocol for wireless mesh networks, in: 2nd IEEE Workshop on Wireless Mesh Networks, WiMesh, 2006, pp. 48–54.

7. H. Dubois-Ferriere, M. Grossglauser, M. Vetterli, Least-cost opportunistic routing, in: Proceedings of 2007 Allerton Conference on Communication, Control, and Computing, 2007.

8. Y. Li, W. Chen, Z.-L. Zhang, Optimal forwarder list selection in opportunistic routing, in: Mobile Adhoc and Sensor Systems, 2009. MASS '09. IEEE 6th International Conference on, 2009, pp. 670 –675.

9. S. Biswas, R. Morris, Opportunistic routing in multi-hop wireless networks, ACM SIGCOMM Computer Communication Review 34 (1) (2004) 69–74.

10. D. S. J. De Couto, D. Aguayo, J. Bicket, R. Morris, A high-throughput path metric for multi-hop wireless routing, Wireless Networks 11 (4) (2005) 419–434.

11. Z. Zhong, J. Wang, S. Nelakuditi, G.-H. Lu, On selection of candidates for opportunistic anypath forwarding, SIGMOBILE Mob. Comput. Commun. Rev. 10 (4) (2006) 1–2.

12. MIT roofnet, http://pdos.csail.mit.edu/roofnet.

13. Y. Li, Y. an Liu, P. Luo, Link probability based opportunistic routing metric in wireless network, in: Communications and Mobile Computing, 2009. CMC '09. WRI International Conference on, Vol. 2, 2009, pp. 308–312.

14. Z. Zhong, S. Nelakuditi, On the efficacy of opportunistic routing, in: SECON '07, 2007, pp. 441–450.

15. M. Zorzi, R. Rao, Geographic random forwarding (geraf) for ad hoc and sensor networks: multihop performance, Mobile Computing, IEEE Transactions on 2 (4) (2003) 337–348.

16. R. Bruno, M. Nurchis, Survey on diversity-based routing in wireless mesh networks: Challenges and solutions, Computer Communications 33 (3) (2010) 269–282.

17. S. Chachulski, M. Jennings, S. Katti, D. Katabi, Trading structure for randomness in wireless opportunistic routing, in: SIGCOMM, ACM, New York, NY, USA, 2007, pp. 169–180.

18. Y. Yan, B. Zhang, H. Mouftah, J. Ma, Practical coding-aware mechanism for opportunistic routing in wireless mesh networks, in: Communications, 2008. ICC '08. IEEE International Conference on, 2008, pp. 2871 –2876.

19. H. Liu, B. Zhang, H. Mouftah, X. Shen, J. Ma, Opportunistic routing for wireless ad hoc and sensor networks: Present and future directions, Communications Magazine, IEEE 47 (12) (2009) 103 –109. doi:10.1109/MCOM.2009.5350376.

20. F. Baccelli, B. Blaszczyszyn, P. Muhlethaler, On the performance of time-space opportunistic routing in multihop mobile ad hoc networks, in: WiOpt, 2008, pp. 307–316.

21. C.-P. Luk, W.-C. Lau, O.-C. Yue, An analysis of opportunistic routing in wireless mesh network, in: Communications, 2008. ICC '08. IEEE International Conference on, 2008, pp. 2877–2883.

22. J. Wu, M. Lu, F. Li, Utility-based opportunistic routing in multi-hop wireless networks, in: Distributed Computing Systems, 2008. ICDCS'08. The 28th International Conference on, 2008, pp. 470–477.

23. M. Lu, J. Wu, Opportunistic routing algebra and its applications, in: IEEE INFOCOM, 2009, pp. 2374–2382.

24. M. Kurth, A. Zubow, J.-P. Redlich, Cooperative opportunistic routing using transmit diversity in wireless mesh networks, in: IEEE INFOCOM, 2008, pp. 1310–1318.

25. L. Cerdà-Alabern, V. Pla, A. Darehshoorzadeh, On the performance modeling of opportunistic routing, in: MobiOpp '10: Proceedings of the Second International Workshop on Mobile Opportunistic Networking, ACM, New York, NY, USA, 2010, pp. 15–21.

26. Y. Li, Z.-L. Zhang, Random walks on digraphs: a theoretical framework for estimating transmission costs in wireless routing, in: 29th conference on Information communications, INFOCOM'10, IEEE Press, 2010, pp. 2775–2783.

27. L. Cerdà-Alabern, A. Darehshoorzadeh, V. Pla, On the maximum performance in opportunistic routing, in: World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a, 2010, pp. 1 –8.

28. G. Latouche, V. Ramaswami, Introduction to Matrix Analytic Methods in Stochastic Modeling, ASA-SIAM, 1999.

29. The Network Simulator ns-2.
URL http://www.isi.edu/nsnam/ns

30. R. D. C. Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0 (2008).
URL http://www.R-project.org