Multicast Delivery Using Opportunistic Routing in Wireless Mesh Networks

Amir Darehshoorzadeh and Llorenç Cerdà-Alabern
[amir,llorenc]@ac.upc.edu
Univ. Politècnica de Catalunya, Computer Architecture Dep. Barcelona, Spain

June 2013

Abstract

Opportunistic Routing (OR) has been proposed to improve the efficiency of unicast protocols in wireless networks. In OR, in contrast to traditional routing, instead of preselecting a single specific node to be the next-hop forwarder, an ordered set of nodes (referred to as candidates) is selected as the next-hop potential forwarders. In this paper, we investigate how OR can be used to improve multicast delivery. We propose a new multicast routing protocol based on opportunistic routing for wireless mesh networks, named Multicast Opportunistic Routing Protocol (MORP). MORP opportunistically employs a set of forwarders to send a packet toward all destinations. Each forwarder is responsible for sending the packet to a subset of destinations. Based on the candidates that successfully receive the packet in each transmission, MORP builds a tree on the fly. We compare our proposal with two well known ODMRP and ADMR multicast protocols. Our results demonstrate that MORP outperforms ODMRP and ADMR, reducing the number of data transmissions and increasing the delivery ratio.

1 Introduction

Multi-hop wireless networks (MWNs) [1, 2] have become a very active research field during the last years. Routing in MWNs is more challenging than in wired networks because of two fundamental differences. The first difference is the heterogeneous characteristics of wireless links. As a consequence, there can be significant differences in packet delivery probabilities across the links of a MWN network. The second difference is the broadcast nature of wireless transmissions [3]. Unlike wired networks, where links are typically point to point, when a node transmits a packet in a wireless network the neighbors of the intended destination node can overhear it.

Multicasting in wireless networks has been an active area of research for quite a long time, and a number of multicast routing protocols have been proposed. On the other hand, Opportunistic Routing (OR) has been investigated in recent years as a way to increase the performance of wireless networks by exploiting its broadcast nature. In this paper we investigate how OR can be used to improve multicast delivery. We do so by proposing MORP, a new *Multicast Opportunistic Routing Protocol*, and comparing it with two well known multicast protocols proposed in the literature.

Generally, multicast routing protocols can be classified into three types based on the multicast topology: tree-based, mesh-based and the hybrid-based [4, 5]. The tree-based multicast protocols (like [6, 7, 8, 9]), establish a single path between any two nodes in the multicast group, whereas for mesh-based multicast protocols (like [10, 11]), packets are distributed along mesh structures that are a set of interconnected nodes and multiple paths may exist between a source-destination pair. The mesh based protocols outperform tree-based ones in terms of robustness, but in an other hand, mesh based protocols suffer from a considerable amount of duplicate packets. Hybrid-based multicast routing protocols combine the advantages of both tree and mesh-based approaches [12, 13]. However, these three types of multicast protocols do not fully take advantage of the spatial characteristic of wireless communications. When a packet is transmitted, it is possible that some nodes in the neighbor nodes receive the packet while the designated next-hop does not. While there are some attempts to achieve a high performance in wireless multicast routing, the research field is still open.

Opportunistic Routing (OR), also referred to as diversity forwarding [14], cooperative forwarding [15] or any-path routing [16, 17], has recently been proposed as a way to increase the performance of unicast in multi-hop wireless networks by taking advantage of its broadcast nature. It improves the throughput and the transmission reliability in the face of unreliable wireless links [18]. In OR, in contrast to traditional routing, instead of preselecting a single specific node to be the next-hop forwarder, an ordered set of nodes (referred to as candidates) is selected as the next-hop potential forwarders. More specifically, when the current node transmits a packet, all the candidates that receive the packet successfully will coordinate with each other to determine which one would actually forward the packet according to some criteria, while the other nodes will simply discard the packet.

By using OR if a certain wireless forwarder fails or moves out of the radio range during the transmission, other possible paths may be used. As a result, OR can better cope with lossy, unreliable and time varying link quality. It can significantly reduce the number of transmissions necessary to deliver a packet to the destination and greatly increases the transmission reliability and the network throughput by taking advantage of the broadcast nature of the wireless medium.

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

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

We compare MORP with two well known protocols: ODMRP multicast mesh protocol [19, 10] and ADMR multicast tree protocol [8, 20]. An additional reason to choose these protocols for comparison is that ODMRP is implemented in the simulation tool used to obtain numerical results (GloMoSim [21]). The source code of ADMR for GloMoSim was kindly provided by the authors in [20].

In summary, the main contributions of this paper are:

- We investigate the advantages of using OR to support multicast by proposing MORP.
- In contrast to the most of previous works which used the two-ray ground or some simple loss propagation models, we use the shadowing propagation model for the packet loss of all algorithms under study.
- Our main conclusion is that OR can be an effective mechanism to achieve reliable multicast delivery in wireless mesh networks.

The rest of this paper is organized as follows. We briefly review the related work on multicast and OR in Section 2. MORP description is presented in Sections 3 and 4. Section 5 and 6 briefly describe ODMRP and ADMR, respectively. Section 7 explains the evaluation methodology. MORP's performance is evaluated in Section 8 and concluding remarks are given in Section 9.

2 Related work

2.1 Multicast Routing

Multicast routing protocols come into play when a host needs to send the same message or data stream to multiple destinations. Due to the unique characteristics of the wireless networks such as limited resources and unreliable channels, traditional multicast protocols in the wired networks do not perform well in wireless, and new protocols have been proposed. One of the most popular methods to classify multicast routing protocols is based on how distribution paths among group members are constructed. According to this method, existing multicast routing approaches can be classified into tree-based, mesh-based and hybrid protocols [4, 22, 5].

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

Tree-based proposals are also divided into two sub-categories: source-based tree and shared-based tree approaches. A source-based tree maintains an individual route towards all the multicast receivers for each multicast group. Some source-based multicast protocols are Differential Destination Multicast (DDM) [23], Preferred Link Based Multicast (PLBM) [24], Adaptive Demand-driven Multicast Routing [8] and probabilistically reliable on-demand (PROD) [9].

Since the construction of a separate tree for each source is costly, some tree-based multicast protocols use a shared-based (core-based) tree to distribute the multicast messages. In shared-based tree a single tree is constructed to support the whole groups. Since the shared-based multicast tree only permits the multicast traffic to be sent out from the root to the multicast receivers, each multicast source must forward its multicast traffic to the root initially. Multicast traffic of each source is then forwarded along the shared tree. Ad-hoc Multicast Routing utilizing Increasing ID numbers (AMRIS) [25], Multicast Ad-hoc On-demand Distance Vector routing (MAODV) [26], Multicast Zone Routing (MZRP) [27] and Adaptive Core based Multicast routing (ACMP) [28] are some popular shared-based tree multicast routing protocols.

The main advantage of a tree as the underlying forwarding structure is that the number of forwarding nodes tends to be reduced. However, they generally suffer from fragile tree structure [22]. Besides the previous problem, source-based tree proposals also suffer from large memory space requirements and wasteful usage of limited bandwidth because each source constructs its own tree. But, it performs better than shared-based tree proposals at heavy loads due to efficient distribution of trees. Although shared-based tree proposals are more scalable, they have the vulnerability of the single core problem [29].

In a mesh-based multicast routing protocol, multiple routes may exist between any pair of source and destination, which is intended to enrich the connectivity among group members. The major difference between the tree-based and mesh-based protocols lies in the manner in which a multicast message is relayed. In tree-based protocols, each intermediate node on the tree has a well-defined list of the next-hop nodes for a specific multicast session. It will send a copy of the received multicast message to only the neighboring nodes on its next-hop list. In mesh-based protocols, each node on the mesh will broadcast the message upon its first reception of the message. Mesh-based multicast routing protocols generally are robust due to the penalty of multiple paths between different nodes. But many of these proposals suffer from excessive control overhead which will affect on scalability and utilization of limited bandwidth. Examples of mesh based multicast routing protocols include On-Demand Multicast Routing (ODMRP) [19, 10] and its variations (PatchODMRP [30], PoolODMRP [31], PDAODMRP [32], EnhancedODMRP [33], Resilient ODMRP [34] and limited flooding ODMRP [35]), Forwarding Group Multicast Core-Assisted Mesh (CAMP) [36], Clustered Group Multicast (CGM) [37], Neighbor-Supporting Multicast (NSMP) [38], Dynamic Core based Multicast routing (DCMP) [39] and link stability based multicast routing in MANETs (LSMRM) [40].

Hybrid multicast routing protocols combine the advantages of both tree-based and mesh-based multicast approaches, i.e., the robustness of the mesh-based multicast routing protocols and low overhead of tree-based protocols. Therefore, the hybrid multicast routing protocols are able to address both efficiency and robustness issues. Multicast Core-Extraction Distributed Ad Hoc Routing (MCEDAR) [41], Ad-hoc Multicast Routing (AMRoute) [7] and Efficient Hybrid Multicast Routing (EHMRP) [12] are some well-known hybrid multicast routing protocols.

2.2 Opportunistic Routing

The majority of previous studies in opportunistic routing do not use it for multicast routing, and most of them are devoted to the selection of the candidates, the way of acknowledging packet reception and how to prevent, or at least reduce, duplicate transmissions.

Biswas and Morris proposed ExOR [42, 18], one of the firsts and most referenced OR protocols. The selection of candidates in ExOR is based on the *Expected Transmission Count (ETX)* [43] metric. In [44], Zhong et al. proposed a new metric –*expected any-path transmission (EAX)*– that generalizes ETX to an OR framework. They analyzed the efficacy of OR by using this metric and did a comparison using link-level measurements at MIT Roofnet project [45]. In [17, 16] a distributed algorithm for computing minimum cost opportunistic routes is presented. The authors also alert about the risk of using too many relay candidates. In [46] 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. In [47] different OR candidate selection algorithms have been compared.

One of the important issues of opportunistic routing is the coordination between candidates in order to prevent duplicate transmissions. Different coordination schemes have been proposed which normally rely on establishing some priority order and exchanging state information between candidates. In [14] 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. The coordination used in MORP follows a similar approach. In [42], an acknowledgment based scheme as the one used in traditional 802.11 is employed. This scheme requires each candidate which has received the data packet to broadcast an ACK in different time slots according to its priority. All the candidates listen to all ACKs before deciding whether to forward the data packet. Other approaches combine OR with network coding, providing an elegant method for candidate coordination [48, 49, 50, 51]. 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 [49].

There are some papers which propose analytical models to study the performance of OR. Baccelli et al. [52] 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 [53] 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 [54] 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 [55] an algebraic approach is applied to study the interaction of OR algorithms and routing metrics. Zubow et al. in [56] 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 [57, 58] a Markov model to assess the improvement that may be achieved using opportunistic routing was proposed. At the same time, Li and Zhang published an analytical framework to estimate the transmission costs of packet forwarding in wireless networks [59]. Both approaches are similar in their formulation, although differ in the way the model is solved: our model leads to a discrete phase-type distribution, while in [59] transmission costs are computed using spectral graph theory. In [60], the issue of optimal candidates set selection in the OR has been addressed. They provide an analytical framework to model the problem of selecting the optimal candidates set for both the constrained (limited number of candidates) and unconstrained (unlimited number of candidates) candidates set selection. They proposed two algorithms for optimal candidates set selection, one for the constrained and one for the unconstrained case. Finally, in [61] some equations that yield the distances of the candidates in OR such that the per transmission progress towards the destination is maximized have been derived. There, we have proposed a lower bound to the expected number of transmissions needed to send a packet using OR.

There are few works that have been made to adapt OR in multicast. MORE [50] 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 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. In [62] 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 [63] used a Steiner tree based on ETX and sent data packets through the links using OR. Their protocol constrains the nodes involved in routing a packet to be near the default multicast tree. The average EAX of each candidate to reach a sub-group of destinations is used as the cost of reaching to multiple destinations. The authors in [64] proposed a Multicast OR (MOR) algorithm. It opportunistically employs a set of forwarders to push a packet closer to all receivers round-by-round. They proposed a new metric –*expect transmission advancement (ETA)*– which is the expected number of OR transmissions achieved after one transmission from a source node toward the destination using the candidates set of source. Based on packet receptions at the end of each round, a new forwarder set is constructed to maximize the expect transmission advancement towards all destinations. They developed an event-driven simulator to measure the performance of their proposal. For the propagation model they used a simple packet loss which is only related to the geographic distance between two nodes. They believe that implementing of MOR using packet-level simulators is not straightforward. The recent work from [65] proposes 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. They employed unicast OR on each link of the tree. Their protocol does not exploit opportunistic receptions cross different links in the tree.

MORP differentiate from these proposals by the candidate selection and the coordination mechanism between candidates. MORP uses a three-way-handshaking where the sending node selects the candidates, and towards which destinations they have to forward the packet. By doing this, MORP aims to achieve a high delivery ratio with a low number of data packet transmissions.

3 Multicast Opportunistic Routing Protocol (MORP)

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

3.1 Network Model

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

Denote $\mathcal{C}^{i,d_j}_{ncand} = \{c_1, c_2, \cdots, c_{ncand}\}$ as the candidates set of node i with at most $ncand$ candidates to reach a destination d_j using unicast OR (c_1 the highest priority candidate, and c_{ncand} the least one). In this paper we have used $ncand = 2$ and 10. From this point forward we shall call $\mathcal{C}^{i,d_j}_2$ and $\mathcal{C}^{i,d_j}_{10}$ the "small candidates set" and "large candidates set" of node i to reach destination d_j, respectively. Each node in the network must compute these candidates sets using one of the candidates selection algorithms that have been proposed in the literature for unicast OR, like ExOR [42]. All this information (small and large candidates sets) is stored in a *Candidate-Table*.

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

$$\mathcal{C}^{s,\mathcal{D}} = \bigcup_{d_j \in \mathcal{D}} \mathcal{C}^{s,d_j}_2 \tag{1}$$

Equation (1) uses the small candidates sets instead of the large candidates set in order to maintain the cardinality of $C^{s,\mathcal{D}}$ as small as possible. The reason is that the lower is the cardinality of $C^{s,\mathcal{D}}$, the less nodes are involved in the packet delivery, and thus, the lower is the signaling overhead.

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

3.2 Description of MORP

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

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

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

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

3.3 Forwarding Set

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

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

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

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

To create the non-redundant and redundant sets of candidates and destinations, node s uses its large candidate set, $\mathcal{C}_{10}^{s,d_j}$, $d_j \in \mathcal{D}$, defined in section 3.1. Here, the large candidates set is used instead of the small one in order to increase the chance of reaching all destinations with the minimum number of candidates. For example, it may happen that a candidate c_i does not appear in the small candidates set to reach destination d_j, $c_i \notin C_2^{s,d_j}$, but it is in the small candidate set of another destination d_k, $c_i \in C_2^{s,d_k}$. If c_i receives the packet and appears in the large candidate set of d_j ($c_i \in \mathcal{C}_{10}^{s,d_j}$), then node s can also use c_i to reach destination d_j.

Algorithm 1 Computation of the *Forwarding-Set* and its *Bind-Destinations* by node s.

```
Require: $\mathcal{D}_s$, Bind-Destinations of node $s$.
 1: Find RCandSet, RDestSet, NRCandSet and NRDestSet
 2: for all $d_j \in$ NRDestSet do
 3:     $c \leftarrow c_i \in$ NRCandSet and $c_i \in \mathcal{C}_{10}^{s,d_j}$
 4:     Add $c$ to the Forwarding-Set
 5:     Add $d_j$ as the Bind-Destinations of $c$
 6: end for
 7: $\mathcal{S} \leftarrow$ RCandSet
 8: while TRUE do
 9:     $C \leftarrow CostFunc(\mathcal{S})$
10:     $\mathcal{R} \leftarrow \arg\min_{\mathcal{T}=\mathcal{S}\setminus c_i} CostFunc(\mathcal{T})$
11:     $C' \leftarrow CostFunc(\mathcal{R})$
12:     if $(C' - C)/C >$ Threshold then
13:         break
14:     else
15:         $\mathcal{S} \leftarrow \mathcal{R}$
16:     end if
17: end while
18: for all $d_j \in$ RDestSet do
19:     $c \leftarrow \arg\min_{c_i \in \mathcal{S} \ \& \ c_i \in \mathcal{C}_{10}^{s,d_j}} ETX(c_i, d_j)$
20:     Add $c$ to the Forwarding-Set
21:     Add $d_j$ as the Bind-Destinations of $c$
22: end for
```

Algorithm 1 shows the pseudocode used by a node to compute the *Forwarding-Set* and its *Bind-Destinations*. The general aim of algorithm 1 is to select few and good candidates to reach all destinations such that the expected number of transmissions is minimized. The algorithm works as follows: First node s creates the Non-Redundant-Set and Redundant-Set for both candidates and destinations (*NRCandSet*, *NRDestSet*, *RCandSet* and *RDestSet*). For each destination $d_j \in$ *NRDestSet* the algorithm assigns the only possible candidate $c_i \in$ *NRCandSet* (lines 2-6). Recall that *NRDestSet* is the set of destinations d_j reachable by only one candidate. Therefore for each destination in the *Non-Redundant-Destinations-Set* there is only one possible choice from *Non-Redundant-Candidates-Set* to add to the *Forwarding-Set*.

Then the algorithm chooses the candidates from *RCandSet* to reach the destinations in the *RDestSet*. For these destinations there are multiple choices of candidates. The optimum choice would minimize the expected number of transmissions to reach all destinations. However, even for a single destination, computing the expected number of transmissions is an equation with a high computational cost (see e.g. [16]). For multiple destinations there has not been proposed any exact equation to compute the expected number of transmissions, and in any case, the computational cost would be extremely high. Additionally, in [61] was shown that the performance results are not very sensitive to the selection of best candidates. Therefore, MORP builds the *Forwarding-Set* using the following simple cost function as an estimation of the expected number of transmissions to reach all destinations in *RDestSet*, using the candidates in the set $\mathcal{S}$:

$$CostFunc(\mathcal{S}) = \sum_{d_j \in RDestSet} \min_{c_i \in \mathcal{S}} ETX(c_i, d_j) \qquad (2)$$

where $ETX(c_i, d_j)$ is the expected transmissions count [43] from candidate c_i to the destination d_j. Note that equation 2 gives the expected number of transmissions that would be obtained using unicast delivery to each destination, choosing the candidate in $\mathcal{S}$ that is closest to each destination in *RDestSet*. Therefore, this will be an upper-bound to the expected number of transmissions obtained using OR.

Lines 8-22 of algorithm 1 show the selection of the candidates for the destinations in *RDestSet*. In each iteration of the while-loop, the algorithm runs an exhaustive search over all possible subsets of the set $\mathcal{S}$ by removing one candidate. The algorithm uses equation (2) to choose the subset having the minimum cost (line 10). If the difference between the cost of new set (C') and the previous one (C) to reach the *Redundant-Destinations-Set* is not very large (e.g., Threshold=1), the algorithm will continue with the new set to eliminate more candidates.

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

3.4 Candidate Coordination and Data forwarding

After running algorithm 1, the source puts the *Forwarding-Set* and its *Bind-Destinations* in the *Forwarding-Packet* and broadcasts it. Each node i receiving the *Forwarding-Packet* having its ID in the *Forwarding-Set* will forward the data packet stored in its buffer to its *Bind-Destinations*. The candidates with IDs not included in the *Forwarding-Packet* will simply discard the packet. This process will be continued until the forwarding nodes directly deliver the data packet to their *Bind-Destinations*.

3.5 Data Structures

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

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

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

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

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

3.6 An Example of MORP

We finish the description of MORP by means of a simple example. Consider the network topology shown in Figure 1. Assume that the delivery probability is a function of the distance between the nodes shown in the figure. The source node is s and the destinations set is $\mathcal{D} = \{d_1, d_2, d_3, d_4\}$. An unicast OR candidates selection algorithm (e.g. ExOR) is used by all nodes to compute the small and large candidates sets. Table 1 shows these sets for node s. In each row, candidates are ordered in descending priority from left to right.

When s wants to send a packet, it puts its multicast candidates set (see equation (1)), which is $\mathcal{C}^{s,\mathcal{D}} = \{a, b, c, d_3, d_4, f\}$ in the data packet and sends

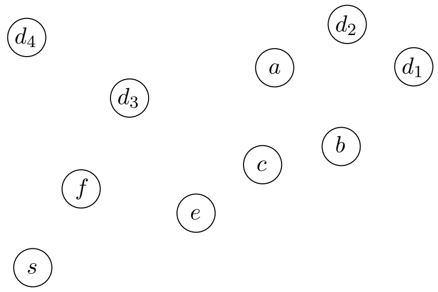

Figure 1: Example of MORP.

it. The source sets the timer T_{ACK} and waits for the ACKs from the candidates that have received the packet successfully. Assume that only the candidates a, b and d_3 receive the data and send back an ACK to the source.

When s receives ACK from a, b and d_3, it stores their ID in its *Ack-Table*. After T_{ACK} expires in node s, it runs the algorithm 1 to find the candidate which should forward the packet. Since one destination, d_3, has received the packet, node s looks for the candidates to reach destinations d_1, d_2 and d_4. First, it finds the non-redundant and redundant sets of candidates and destinations. As we mentioned in section 3.3, the algorithm 1 uses the large candidates set to create the non-redundant and redundant sets. The only candidate which has received the packet and can reach the destination d_4 is d_3 (see large candidate set in Table 1). Therefore the *Non-Redundant-Destinations-Set* (*NRDestSet*) is $\{d_4\}$, and the candidate d_3 will be added to the *Forwarding-Set* with destination d_4 as its *Bind-Destination*.

Table 1: Small and large candidates sets of s

(a) Small and large candidates sets

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

(b) ETX Table

node	d_1	d_2
a	4.3	4.1
b	4.8	3.8

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

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

In the first iteration of the while-loop of algorithm 1, the cost of reaching *RDestSet* $= \{d_1, d_2\}$ using $\mathcal{S} = \{a, b\}$ is estimated as: $C = ETX(a, d_1) + ETX(b, d_2) = 8.1$ (see equation 2). Then it reduces the number of candidates in the *RCandSet* and uses formula 2 again to find the set with the minimum cost (line 10 in algorithm 1). This is given by the set $\mathcal{R} = \{a\}$ with cost $C' = 8.4$. Since the relative difference between new cost and the previous one ($C = 8.1$) is small, the algorithm takes the new set $\mathcal{S} = \{a\}$. Then the while-loop finishes.

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

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

4 Implementation of MORP

As explained in section 3.1, MORP computes the candidates sets using one of the candidates selection algorithms that have been proposed in the literature for unicast OR. To do so, the nodes need to be aware of the network topology and the delivery probability of the wireless links. This information can be gathered in different ways. One possible implementation could be the method described in ExOR [18], where nodes collects measurements and send them to a central server which distributes the required information to all nodes. Distributed algorithms similar to the topology discovery mechanism used by OLSR [66] would also be possible.

MORP could be implemented at link or network layer. A link layer implementation would permit the design of an efficient signaling protocol. For instance, the three-way-handshaking used by MORP (see section 3.2) could be implemented using a modified 802.11 MAC as shown in Figure 2. In this figure the *Multicast Candidates Set* consists of the nodes $\{a, b, c\}$. The candidates send back an ACK which is immediately followed by the *Forwarding-Packet*. A similar proposal to send the ACKs was proposed in [42].

A network layer implementation would allow using current off-the-shelf 802.11 network cards. In this case ACKs and *Forwarding-Packets* would be sent using unicast 802.11 data frames, thus, increasing the overhead and delays of the three-way-handshaking used by MORP. Nevertheless, for the sake of investi-

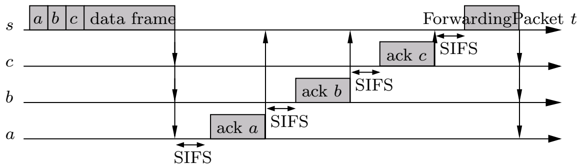

Figure 2: Three-way-handshaking of MORP using a modified 802.11 MAC.

gating the feasibility to implement MORP with current hardware, in the numerical results presented in section 8 we have assumed a network layer implementation using standard 802.11 cards.

5 Summary of the ODMRP Protocol

The On Demand Multicast Routing Protocol (ODMRP) is a mesh based multicast protocol where group membership and multicast routes are established and updated by the source on demand [10, 29, 67]. It introduces the concept of *forwarding groups.* A multicast source will transmit packets to the destinations via the *forwarding group.* The *forwarding group* is a set of nodes in charge of forwarding multicast packets. When a multicast source has data packets to send, but there is no route to the multicast group, it broadcasts a *Join-Query* control packet to the entire network. This control packet is periodically sent every *REFRESH INTERVAL*, e.g., every 3 seconds to refresh the membership information and update routes. When a node receives a non-duplicate *Join-Query*, it stores the upstream node ID and rebroadcasts the packet.

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

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

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

6 Summary of the ADMR Protocol

Adaptive Demand-Driven Multicast Routing (ADMR) [8, 68] protocol is an on demand protocol like ODMRP. It creates a source-based forwarding tree connecting the source with the destinations of the multicast group. Each multicast packet is dynamically forwarded from the source along the shortest delay path through the tree to the destinations of the multicast group. In ADMR, packet forwarding is based on two types of flooding: *tree flood* and *network flood.* In the tree flooding the packets are constrained to the nodes in the multicast tree, while network flooding is the flooding among all nodes in the network. Note that the tree flooding in ADMR is similar to the *forwarding group* concept in ODMRP.

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

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

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

7 Evaluation Methodology

To evaluate the performance of MORP we compare it with ODMRP and ADMR, which have been shown to perform well in previous studies. The simulation code has been implemented within the Global Mobile Simulation (GloMoSim) library [21]. The number of multicast groups and sources is set to one in all scenarios. Members join the multicast group at the start of the simulation and remain throughout the simulation. The simulation field consists of a square with diagonal equal to 500 m. We have run simulations varying the number of nodes in the range $20 \leq N \leq 100$. One node is the source, and it is located in a square corner, the others are placed randomly inside the square. The destinations of the multicast group are chosen randomly among the nodes inside the square. Each simulation runs for 300 seconds of simulation time. Each point in our performance graphs represents the average of 20 simulation runs. For this number of runs we obtained reasonably small confidence intervals. The IEEE 802.11 Distributed Coordination Function was used as the medium access control protocol.

The multicast application-layer source in our simulations generates Constant Bit Rate (CBR) traffic with 4 packet per second and 64 bytes of payload. This sending rate was chosen to challenge the routing protocols' abilities to successfully deliver data packets in a wireless network. It was not chosen to represent any particular or class of applications, although it could be considered to abstractly model a very simple broadcast audio distribution application [8].

For a more realistic simulation of an 802.11 network, we have considered that packets can be transmitted at two different transmission rates: a data rate of 11 Mbps, and a basic rate of 2 Mbps. Most of previous works used the two-ray ground or some simple loss propagation models [64, 65, 63], we use the shadowing propagation model (below shadowing propagation model is explained in more detail) for the packet loss of all algorithms under study. Packets transmitted at the data rate are subject to a shadowing propagation model, which introduces random transmission losses . Packets transmitted at the basic rate does not suffer transmission losses. We have assumed that data packets are always transmitted at data rate. However, the protocols can transmit signaling packets using the basic rate to prevent losses due to impairments of the radio channel. More specifically, we have assumed that in MORP, all signaling packets (i.e. ACKs and *Forwarding-Packets*) are transmitted at the basic rate. In ODMRP, *Join-Query* packets are sent at the data rate. This is because these packets are used to build the routing tables, and thus, they need to have the same transmission properties over the wireless links as those of data packets. For the same reason, *Source Information* and *Multicast Solicitation* packets are sent at the data rate in ADMR, although *Receiver Join* packets are sent at the basic rate.

We have assumed that in MORP nodes are aware of the network topology and the delivery probability of the wireless links, due to the shadowing propagation model of the radio channel. MORP uses this information and applies ExOR [42] to compute the candidates sets.

In the shadowing propagation model the wireless links between nodes are not reliable. The power received at a distance d, in terms of the transmitted power is given by:

$$P_r(d)|_{dB} = 10\,\log_{10}\left(\frac{P_t\,G_t\,G_r\,\lambda^2}{L\,(4\,\pi)^2\,d^\beta}\right) + X_{dB} \qquad (3)$$

Where $P_r(d)$ is the power received at a distance d and P_t is the transmitted power. The 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 delivered correctly if the received power is greater than or equal to a threshold *RxThresh*. Thus, the delivery probability owing to the propagation model at a distance d is given by:

$$p(d) = Prob(P_r(d)|_{dB} \geq 10\,\log_{10}(RxThresh)) \qquad (4)$$

Table 2: Default GloMoSim values for the shadowing propagation model.

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

We have set the model parameters to the default values used by the GloMoSim, given in Table 2. Figure 3 depicts the delivery probability varying the distance, for a path loss exponent with parameters

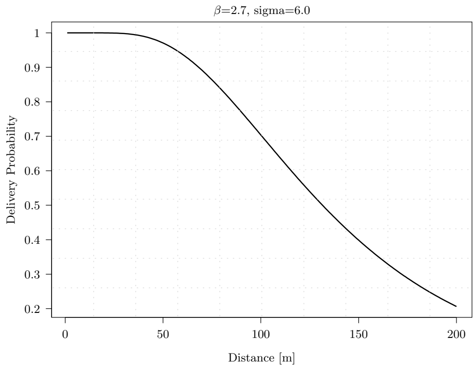

Figure 3: Delivery probability versus distance for a path loss exponent $\beta = 2.7$ and standard deviation $\sigma_{dB} = 6$ dBs.

$\beta = 2.7$ and $\sigma_{dB} = 6$ dBs. With these parameters the link delivery probability is approximately 40% at the distance of 135 m. To find the candidates sets using ExOR, we have assumed that a link between any two nodes exists only if the delivery probability between them is greater (or equal) than $min.dp = 0.1$ (see [57] for more details).

For the other parameters it was used $G_t = G_r = L = 1$, $f = 914$ MHz and $RxThresh = 281$ mw. We have used these values in our simulations. With these parameters the link delivery probability is approximately 40% at the distance of 150 m. To find the candidates sets using ExOR, we have assumed that a link between any two nodes exists only if the delivery probability between them is greater (or equal) than $min.dp = 0.1$ (see [57] for more details).

7.1 Protocols Parameters

We have evaluated two different variations of the ODMRP parameters. The "ODMRP-3-9" variation represents ODMRP using the parameter values chosen by ODMRP's designers: 3 seconds for the *Join-Query* flooding interval (*REFRESH INTERVAL*=3 seconds) and a forwarding state lifetime of 3 times of this interval (a total of 9 seconds). The "ODMRP-3-3.3" variation reduces the forwarding state lifetime to 1.1 times of the *Join-Query* flooding interval; it shows the effect of reducing the forwarding redundancy of ODMRP (see section 5). For ADMR parameters we have used the default values which are used in [8]: 30 seconds for the periodic data flood interval and 2 missing packets to trigger disconnection detection procedure.

In MORP we have used ExOR [42] as the candidate selection algorithm. Authors in [57] have shown that using a small number of candidates (like 2) is a sensible choice. Therefore, we have fixed the maximum number of candidates for the small and large candidates sets to $ncand = 2$ and 10, respectively. In our protocol we have assumed that all candidates that successfully receive the packet send ACK to the source. To do so, we have used 12 milliseconds, which is long enough to receive all ACKs from the candidates set ($T_{ACK} = 12$ ms). The legend MORP-ExOR(n) in the following sections refers to MORP with $MAX_{ReTx} = n$.

7.2 Performance Metrics

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

- Packet delivery ratio: The ratio of the number of data packets delivered to the destinations versus the number of data packets supposed to be received.

- Multicast group reachability: Let X be a random variable equal to the number of destinations of the multicast group receiving a given data packet. We have computed the empirical complementary cumulative distribution function (EC-CDF) of X. This gives a measure of the number of destinations of the multicast group receiving data packets.

- Forwarding cost: Total number of data packets transmitted by all nodes in the network over the total number of data packets sent by the source. This metric represents the delivery cost in terms of transmissions of each multicast packet. Note that, to make the comparisons more clear, in this metric we take as the reference originated instead of delivered packets.

- Normalized packet overhead: The total number of all data and control packets transmitted by any node in the network (either originated or forwarded), divided by the total number of all data packets received across all multicast receivers.

- End-to-End delay: Average end-to-end delay of all data packets received by the destinations.

8 Numerical Results

8.1 Packet delivery ratio

One important parameter of MORP is the maximum number of retransmissions (MAX_{ReTx}). Recall that if the forwarder does not receive enough ACKs from its candidates, it retransmits the data packet up to MAX_{ReTx} times before it is forwarded. To see the effect of this parameter, Figures 4 and 5 depict the delivery ratio varying MAX_{ReTx} from 1 to 5. The

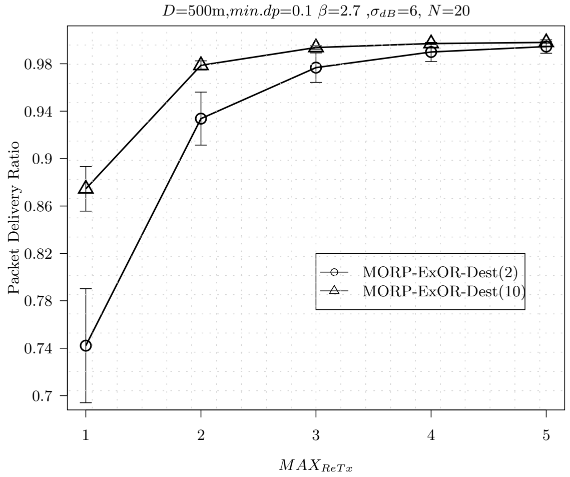

Figure 4: Packet delivery ratio of MORP in a sparse network as a function of MAX_{ReTx}.

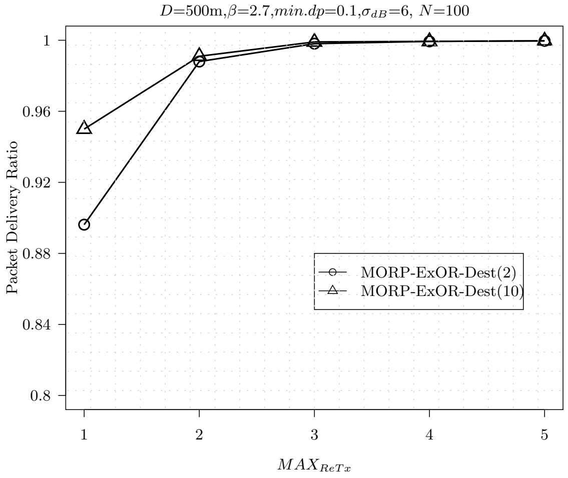

Figure 5: Packet delivery ratio of MORP in a dense network as a function of MAX_{ReTx}.

two curves correspond to a number of destinations of the multicast group equal to 2 and 10. The legend MORP-ExOR-Dest(n) in these two figures refers to MORP with number of destinations equal to n. These figures have been obtained with a total number of nodes equal to $N = 20$ (Figure 4) and $N = 100$ (Figure 5). In the rest of this paper we shall refer to the scenarios having these number of nodes as *sparse* and *dense* networks, respectively. The 95% confidence intervals have been added in Figure 4. It can be observed that the intervals are relatively small, and the same was obtained for the other figures. So, for the sake of clarity, confidence intervals are not depicted in the rest of the figures.

As expected, Figures 4 and 5 show that the higher is MAX_{ReTx}, the higher is the delivery ratio. Additionally, we observe that the maximum delivery ratio improvement is obtained when MAX_{ReTx} is increased from 1 to 2. For instance, in the sparse network (Figure 4) we can see that the delivery ratio of MORP for 2 destinations with $MAX_{ReTx} = 1$ is about 74%, while it improves to 94% with $MAX_{ReTx} = 2$ (improvement around 27%). Increasing from $MAX_{ReTx} = 2$ to 3 yields a delivery ratio of 98% (improvement around 4%).

Comparing Figures 4 and 5 we can see that packet delivery ratio is always higher in a dense than in a sparse network. This comes from the fact that in the dense network, MORP uses better candidates than in the sparse network. For instance, the packet delivery ratio of MORP in a sparse network with 2 destinations and $MAX_{ReTx} = 1$ is about 74%, while it increases to 90% in a dense network.

Figure 6 shows the packet delivery ratio of MORP in comparison with ODMRP and ADMR. The curves are obtained varying the number of nodes from 20 to 100. In this figure the number of destinations has been set to 5 (NumDest = 5). The results of MORP are shown for MAX_{ReTx} is set to 1 and 2 (MORP-ExOR(1) and MORP-ExOR(2), respectively).

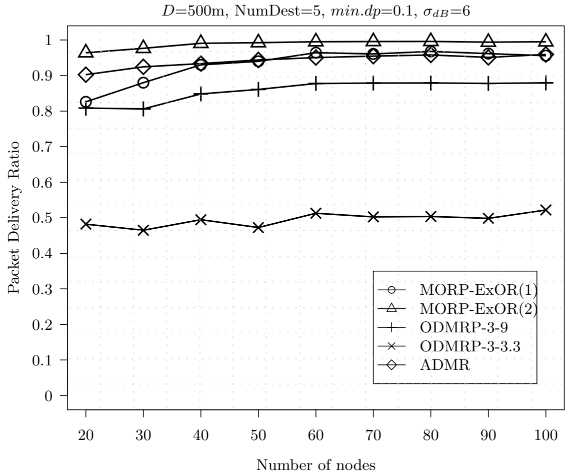

Figure 6: Packet delivery ratio for 5 destinations as a function of number of nodes in the network.

As we can see in Figure 6, MORP with any MAX_{ReTx} outperforms both ODMRP (ODMRP-3-9 and ODMRP-3-3.3) and ADMR. For instance, even with $MAX_{ReTx} = 1$, MORP has about 92% packet delivery ratio, while ODMRP-3-9, ODMRP3-3.3 and ADMR have about 83%, 48% and 89%, respectively. This comes from the fact that the construction of the routes in ODMRP and ADMR are subject to the random losses that may have the *Join-Query* packets in ODMRP and the *Source Information* and *Multicast Solicitation* packets in ADMR. On the other hand, MORP takes routing decisions “on the fly” (when the forwarding nodes are chosen), and thus, adapts faster to random losses.

Figure 6 shows that the packet delivery ratio of ODMRP-3-3.3 is significantly lower than ODMRP-3.9 (about 35%). As we described in section 5, ODMRP creates forwarding groups within nodes in the network that expires after a fixed timeout. In ODMRP-3-3.3 the forwarding state timeout (3.3 seconds) is shorter than in ODMRP-3-9 (9 seconds).

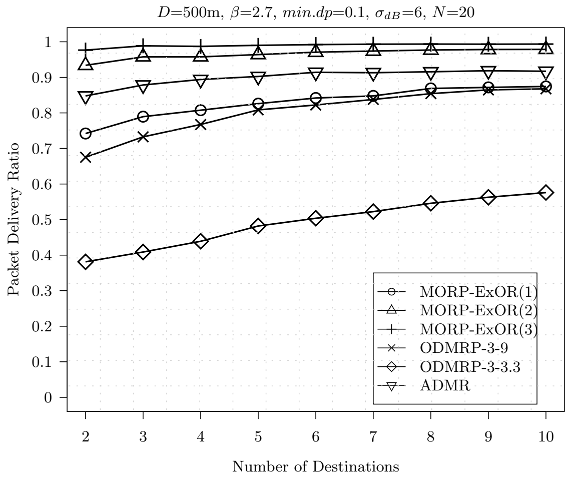

Figure 7: Packet delivery ratio in a sparse network as a function of number of destinations.

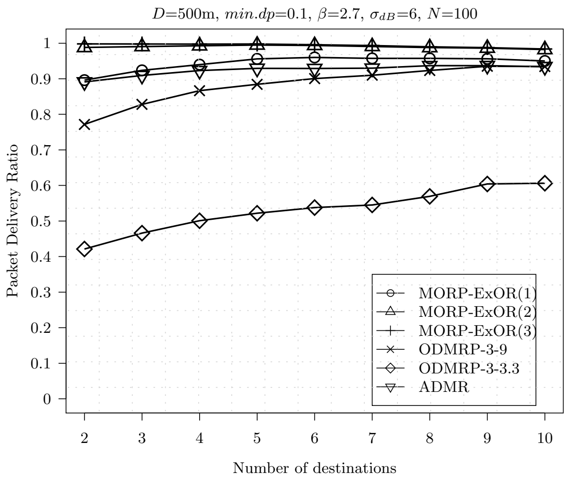

Figure 8: Packet delivery ratio in a dense network as a function of number of destinations.

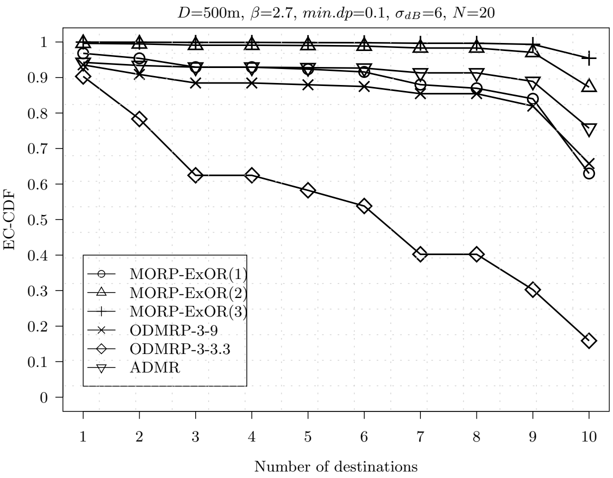

Figure 9: Distribution of received packets for 10 destinations and 20 nodes.

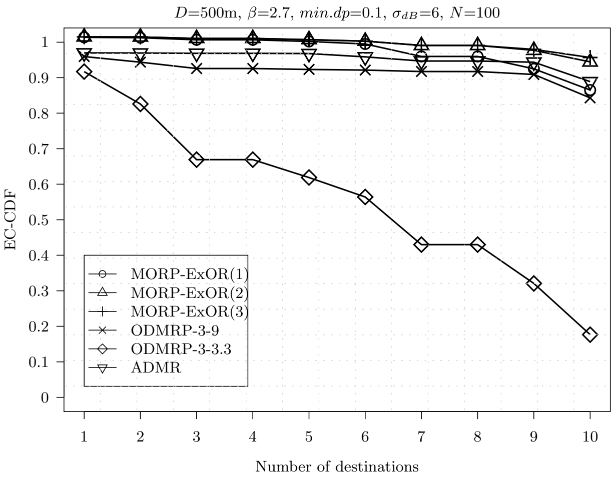

Figure 10: Distribution of received packets for 10 destinations and 100 nodes.

Therefore, ODMRP-3-3.3 has less number of nodes in the forwarding group than in ODMRP-3-9, resulting in a lower delivery ratio.

Figures 7 and 8 show the delivery ratio of the protocols under study, varying the number of destinations. In these figures we can see that all protocols achieve a higher delivery ratio in the dense scenario than in the sparse network. In the sparse network (Figure 7) the packet delivery ratio of MORP with any maximum number of retransmissions (MAX_{ReTx}) outperforms both variations of ODMRP. The same figure shows that ADMR has a delivery ratio about 6% better than MORP-ExOR(1). However, in section 8.3 we will see that this small improvement is at cost of having much more data transmissions than MORP. Nevertheless, Figure 7 shows that MORP outperforms ADMR when MAX_{ReTx} is increased to 2 and 3.

For a dense network, we can see in Figure 8 that the packet delivery ratio of MORP with any MAX_{ReTx} is higher than ODMRP and ADMR. This comes from the fact that in a dense network, MORP can choose better candidates to forward the packets. For instance, the packet delivery ratio of MORP-ExOR(1) and MORP-ExOR(2) in the dense network is about 94% and 98%, respectively. Although the delivery ratio of ADMR in the case of a dense network is close to MORP-ExOR(1), we will see in section 8.3 that it is at cost of a large amount of data transmissions.

8.2 Multicast group reachability

In this section we investigate the number of destinations of the multicast group receiving data packets. To do so, we have calculated its empirical complementary cumulative distribution function (EC-CDF). This is shown in Figures 9 and 10 in a scenario with 10 destinations for the sparse and dense networks (with 20 and 100 nodes), respectively.

Figure 9 shows that ODMRP-3-3.3 performs much worst than the others: the probability of reaching the multicast group decreases sharply with increasing the number of destinations, and less than 20% of the packets reach all destinations.

Regarding the other protocols, Figure 9 shows

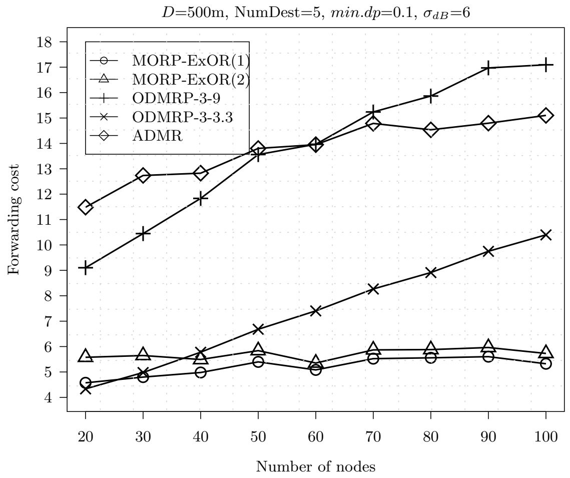

Figure 11: Forwarding cost for 5 destinations as a function of number of nodes in the network.

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

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

Finally, the same conclusions can be derived from the dense network scenario shown in Figure 10. Of course, the group reachability increases, due to the higher proximity between the nodes.

8.3 Forwarding cost

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

Figure 11 shows the forwarding cost of the protocols varying the number of nodes from 20 to 100 in the case of 5 destinations. The results of MORP, like in Figure 6, are obtained for $MAX_{ReTx} = 1$ and 2.

Figure 11 shows that the forwarding cost of MORP outperforms ODMRP-3-9 and ADMR. In fact, the forwarding cost of both variations of ODMRP and ADMR is rather sensitive to the number of nodes, while in MORP is not. This is because using opportunistic routing, as in MORP, only some useful nodes are selected as candidates to forward the packets. Figure 11 also shows that there is only a slight increase of the forwarding cost when MORP increases MAX_{ReTx} from 1 to 2.

As described in section 5, ODMRP periodically floods a data packet together with a *Join-Query* packet. I.e., it piggybacks the *Join-Query* information on the data packet periodically to update the routes. Because of this, new nodes may become forwarders, while forwarders created during a previous periodic flood still have a set forwarding flag. Consequently, redundant routes are created, and the number of data transmissions increases with increasing the network density. In fact, the forwarding cost of both variations of ODMRP is dominated by the flooding packets and forwarding state timeout. Since in ODMRP-3-9 the forwarding state timeout (9 seconds) is longer than in ODMRP-3-3.3, there are more nodes with the forwarding flag set in ODMRP-3-9 than in ODMRP-3-3.3. Therefore, the forwarding cost of ODMRP-3-9 is much higher than in ODMRP-3-3.3.

The construction of the routes in ADMR is subject to the random losses that may have the *Source Information* and *Multicast Solicitation* packets. Recall that the absence of some data packets in ADMR is an indication of forwarding tree disconnection and the local or global repair procedure is triggered to repair the path. As we mentioned in section 7.1, we used 2 missing data packets to trigger disconnection detection. When a node on the tree does not receive 2 consecutive data packets, it starts repairing algorithm, which may add new nodes to the tree. This is exacerbated with increasing the density of the network, thus, increasing the forwarding cost.

Figure 11 shows that for $N = 20$ ODMRP-3-3.3 is slightly better than MORP. However, recall from Figure 6 that in this scenario the delivery ratio of ODMRP-3-3.3 is much lower than in MORP.

In section 8.1 we have shown that MORP outperforms ODMRP and ADMR in terms of packet delivery ratio for different number of destinations. Figures 12 and 13 give the forwarding cost for the same scenarios. Figure 12 shows that the forwarding cost of MORP in the sparse network, and with any MAX_{ReTx}, is much lower than ADMR and ODMRP-3-9. The figure shows that only ODMRP-3-3.3 is slightly better than MORP. However, as we showed in Figure 7, the delivery ratio of ODMRP-3-3.3 is much lower than MORP.

Regarding ADMR, Figure 7 in section 8.1 showed that the delivery ratio in a sparse network is slightly better than MORP-ExOR(1). However, we observe in Figure 12 that this is at cost of ADMR having a forwarding cost of about 3 times higher than MORP-ExOR(1).

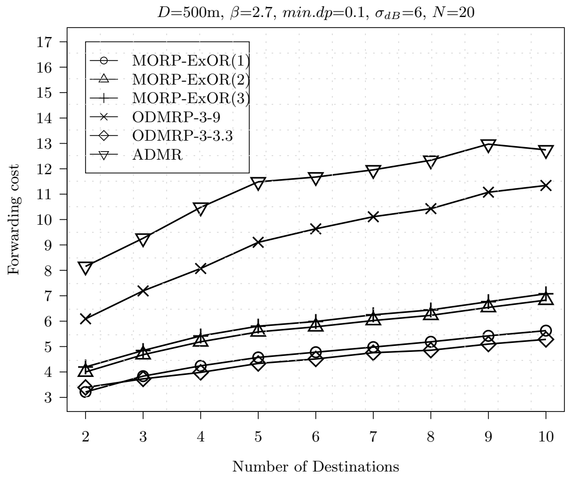

Figure 12: Forwarding cost in a sparse network as a function of number of destinations.

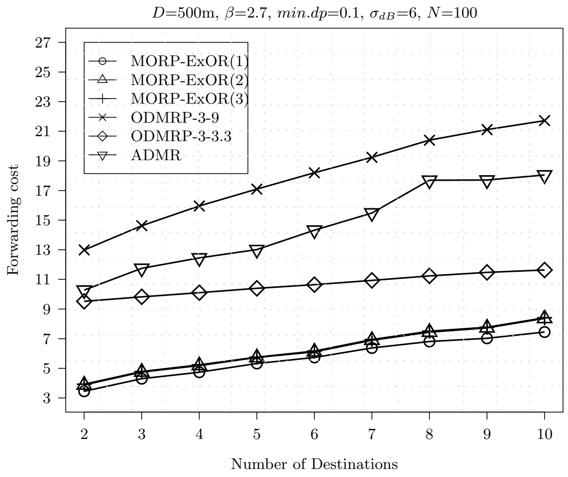

Figure 13: Forwarding cost in a dense network as a function of number of destinations.

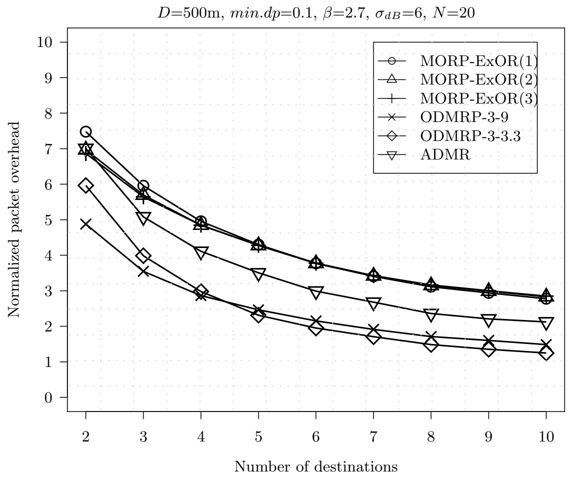

Figure 14: Packet overhead in a sparse network as a function of number of destinations.

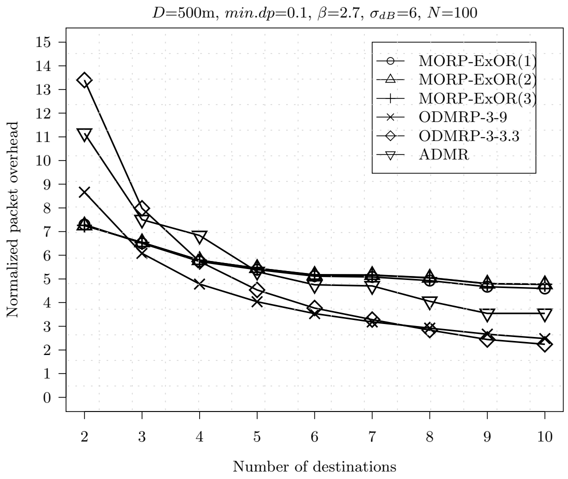

Figure 15: Packet overhead in a dense network as a function of number of destinations.

Figures 12 and 13 confirm, as in Figure 11, that the forwarding cost of ODMRP and ADMR is much higher in a dense than in a sparse network, while MORP is rather insensitive to network density. Nevertheless, it can be observed that the forwarding cost increment in MORP is slightly higher in a dense than in a sparse network. This may be counterintuitive, since MORP can choose better candidates in a dense network. However, since MORP looks for candidates to reach all destinations, in a dense network MORP chooses more candidates, thus, increasing the forwarding cost. However, recall from Figures 7 and 8, this slightly higher forwarding cost in a dense network is rewarded with a significantly higher packet delivery ratio.

8.4 Packet overhead

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

Figures 14 and 15 show the packet overhead of all protocols varying the number of destinations for a sparse and dense network, respectively. Figure 14 shows that in a sparse network MORP has a higher packet overhead than ADMR and ODMRP. This is due to the ACKs sent by the candidates in MORP. Note that in this comparison we are giving the same weight to data and control packets. Recall from Figures 12 and 13 that MORP performs better than the other protocols in terms of forwarding cost, where only data packets are considered. Thus, in case of sending data packets with large payload, the overhead considered in this section would be a pessimistic comparison with respect to MORP. Furthermore, as explained in section 4, the overhead of control packets in MORP could be reduced implementing the three-way handshaking used by MORP at MAC layer.

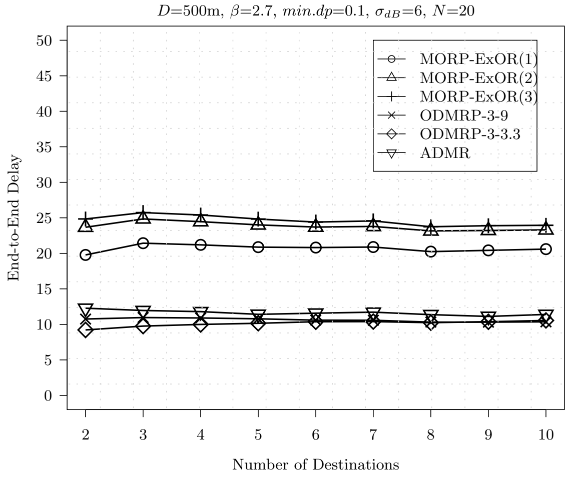

Figure 16: End-to-end delay in a sparse network as a function of number of destinations.

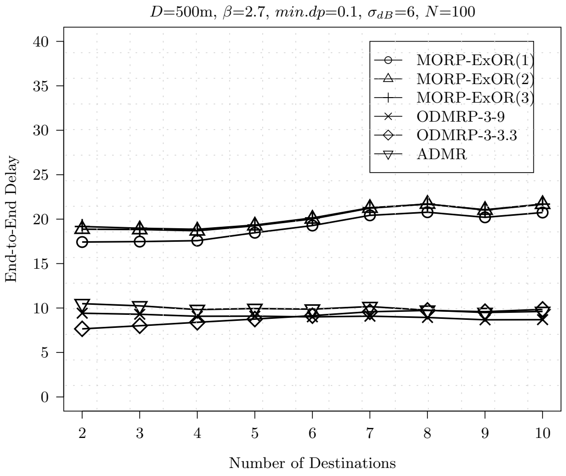

Figure 17: End-to-end delay in a dense network as a function of number of destinations.

8.5 End-To-End Delay

Figures 16 and 17 show the average end-to-end delay for different number of destinations for a sparse and dense network, respectively.

These figures show that the end-to-end delay in MORP is higher than in ODMRP and ADMR. This is because each time a node transmits a data packet in MORP, it waits for the ACKs sent by the candidates during T_{ACK} =12 ms. However, recall from section 4 that implementing the three-way handshaking used by MORP at MAC layer could reduce this delay significantly.

Comparing Figures 16 and 17 we can see that the decrease of the end-to-end delay in a dense network for MORP is much more noticeable than in the other protocols. This is because using OR in a dense network, the probability of reaching a candidate close to the destination increases, thus, reducing the average number of end-to-end hops. It can also be observed that the difference between the average delay using MORP with $MAX_{ReTx} = 1$ and $MAX_{ReTx} > 1$ is higher in a sparse network than in a dense network. This is because in the dense network the probability that the sending node receives enough ACKs to reach all destinations is higher than in the sparse network. Therefore, MORP requires less retransmissions of data packets in a dense network.

9 Conclusion

In this paper we have investigated how Opportunistic Routing (OR) can be exploited to implement a multicast protocol for wireless mesh networks. This has been done by proposing a new protocol called *Multicast Opportunistic Routing Protocol*, MORP. MORP uses a three-way-handshaking approach where candidates send ACKs to the sender node upon successfully receiving data packets. Then, the sender node partitions the set of destinations and assigns each subset to the most appropriate candidate. This information is sent to the candidates which repeat the same approach for each subset of destinations. Compared with traditional multicast protocols, MORP does not build a complete tree or mesh before the transmissions starts. Instead, MORP builds a tree on the fly, depending on the candidates that successfully receive the packet in each transmission.

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

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

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

References

[1] I. Akyildiz, X. Wang, A survey on wireless mesh networks, Communications Magazine, IEEE 43 (9) (2005) S23 – S30. doi:10.1109/MCOM.2005.1509968.

[2] I. F. Akyildiz, X. Wang, Wireless Mesh Networks, John Wiley & Sons Ltd, 2009.

[3] M. L. Kai Zeng, Wenjing Lou, Multihop Wireless Networks: Opportunistic Routing, John Wiley & Sons, 2011, 2011.

[4] L. Junhai, Y. Danxia, X. Liu, F. Mingyu, A survey of multicast routing protocols for mobile ad-hoc networks, Communications Surveys Tutorials, IEEE 11 (1) (2009) 78 –91.

[5] R. C. Biradar, S. S. Manvi, Review of multicast routing mechanisms in mobile ad hoc networks, Journal of Network and Computer Applications 35 (1) (2012) 221 – 239. doi:10.1016/j.jnca.2011.08.003.

[6] K. Chen, K. Nahrstedt, Effective location-guided tree construction algorithms for small group multicast in manet, in: INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Vol. 3, 2002, pp. 1180 – 1189 vol.3.

[7] J. Xie, R. R. Talpade, A. Mcauley, M. Liu, Amroute: ad hoc multicast routing protocol, Mob. Netw. Appl. 7 (2002) 429–439.

[8] J. G. Jetcheva, D. B. Johnson, Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks, in: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing, MobiHoc '01, ACM, New York, NY, USA, 2001, pp. 33–44.

[9] X. Zhao, C. T. Chou, J. Guo, S. Jha, A. Misra, Probabilistically reliable on-demand multicast in wireless mesh networks, in: World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008. 2008 International Symposium on a, 2008, pp. 1 –9.

[10] S. J. Lee, W. Su, M. Gerla, On-demand multicast routing protocol in multihop wireless mobile networks, Mob. Netw. Appl. 7 (2002) 441–453.

[11] B. Sun, L. Li, Reliable adaptive multicast protocol in wireless ad hoc networks, Journal of Systems Engineering and Electronics 17 (1) (2006) 187 – 192.

[12] J. Biswas, M. Barai, S. Nandy, Efficient hybrid multicast routing protocol for ad-hoc wireless networks, in: Local Computer Networks, 2004. 29th Annual IEEE International Conference on, 2004, pp. 180 – 187.

[13] A. Mnaouer, L. Chen, C. H. Foh, J. Tantra, Ophmr: An optimized polymorphic hybrid multicast routing protocol for manet, Mobile Computing, IEEE Transactions on 6 (5) (2007) 551 –562.

[14] 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.

[15] 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.

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

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

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

[19] S.-J. Lee, M. Gerla, C.-C. Chiang, On-demand multicast routing protocol, in: Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE, 1999, pp. 1298 –1302 vol.3.

[20] M. Pandey, D. Zappala, A scenario-based performance evaluation of multicast routing protocols for ad hoc networks, in: Proceedings of the Sixth IEEE International Symposium on World of Wireless Mobile and Multimedia Networks, WOWMOM '05, IEEE Computer Society, Washington, DC, USA, 2005, pp. 31–41.

[21] X. Zeng, R. Bagrodia, M. Gerla, GloMoSim: A Library for Parallel Simulation of Large-Scale Wireless Networks, in: Proceedings of the 12th Workshop on Parallel and Distributed Simulation (PADS'98), IEEE Computer Society, 1998, pp. 154–161.

[22] L. Junhai, X. Liu, Y. Danxia, Research on multicast routing protocols for mobile ad-hoc networks, Computer Networks 52 (5) (2008) 988 – 997. doi:10.1016/j.comnet.2007.11.016.

[23] L. Ji, M. Corson, Differential destination multicast-a manet multicast routing protocol for small groups, in: INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Vol. 2, 2001, pp. 1192 –1201 vol.2.

[24] R. Sisodia, I. Karthigeyan, B. Manoj, C. Murthy, A preferred link based multicast protocol for wireless mobile ad hoc networks, in: Communications, 2003. ICC '03. IEEE International Conference on, Vol. 3, 2003, pp. 2213 – 2217 vol.3.

[25] C. Wu, Y. Tay, Amris: a multicast protocol for ad hoc wireless networks, in: Military Communications Conference Proceedings, 1999. MILCOM 1999. IEEE, Vol. 1, 1999, pp. 25 –29 vol.1.

[26] E. M. Royer, C. E. Perkins, Multicast Ad hoc On-Demand Distance Vector (MAODV) Routing - draft-ietf-manet-maodv-00, Tech. rep., Mobile Ad Hoc Networking Working Group (2000).

[27] X. Zhang, L. Jacob", "mzrp: An extension of the zone routing protocol for multicasting in manets.", "J. Inf. Sci. Eng." (2004) 535–551.

[28] B. Kaliaperumal, A. Ebenezer, Jeyakumar, Adaptive core based scalable multicasting networks, in: INDICON, 2005 Annual IEEE, 2005, pp. 198 – 202.

[29] O. S. Badarneh, M. Kadoch, Multicast routing protocols in mobile ad hoc networks: a comparative survey and taxonomy, EURASIP J. Wirel. Commun. Netw. (2009) 26:1–26:42doi:10.1155/2009/764047.

[30] M. Lee, Y. K. Kim, Patchodmrp: an ad-hoc multicast routing protocol, in: Information Networking, 2001. Proceedings. 15th International Conference on, 2001, pp. 537 –543.

[31] S.-B. Cai, X.-Z. Yang, The performance of poolodmrp protocol., in: MMNS'03, 2003, pp. 90–101.

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

[33] S. Oh, J.-S. Park, M. Gerla, E-odmrp: enhanced odmrp with motion adaptive refresh, in: Wireless Communication Systems, 2005. 2nd International Symposium on, 2005, pp. 130 – 134.

[34] D. Pathirana, M. Kwon, Rodmrp: Resilient on-demand multicast routing protocol, Advanced Information Networking and Applications Workshops, International Conference on 2 (2007) 85–92.

[35] M.-A. Kharraz, H. Sarbazi-Azad, A. Y. Zomaya, On-demand multicast routing protocol with efficient route discovery, Journal of Network and Computer Applications 35 (3) (2012) 942 – 950, ¡ce:title¿Special Issue on Trusted Computing and Communications¡/ce:title¿. doi:10.1016/j.jnca.2011.03.012.

[36] J. Garcia-Luna-Aceves, E. Madruga, The core-assisted mesh protocol, Selected Areas in Communications, IEEE Journal on 17 (8) (1999) 1380 –1394.

[37] C. Lin, S.-W. Chao, A multicast routing protocol for multihop wireless networks, in: Global Telecommunications Conference, 1999. GLOBECOM '99, Vol. 1A, 1999, pp. 235 –239 vol.1a.

[38] S. Lee, C. Kim, Neighbor supporting ad hoc multicast routing protocol, in: Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing, MobiHoc '00, IEEE Press, Piscataway, NJ, USA, 2000, pp. 37–44.

[39] S. K. Das, B. S. B. S. Manoj, C. S. Ram Murthy, A dynamic core based multicast routing protocol for ad hoc wireless networks, in: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing, MobiHoc '02, ACM, New York, NY, USA, 2002, pp. 24–35.

[40] R. Biradar, S. Manvi, M. Reddy, Link stability based multicast routing scheme in manet, Comput. Netw. 54 (2010) 1183–1196.

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

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

[43] 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.

[44] 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.

[45] MIT roofnet, http://pdos.csail.mit.edu/roofnet.

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

[47] A. Darehshoorzadeh, L. Cerdà-Alabern, Candidate selection algorithms in opportunistic routing, in: PM2HW2N '10: Proceedings of the 5th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks, ACM, New York, NY, USA, 2010, pp. 48–54.

[48] M. A. Iqbal, B. Dai, B. Huang, A. Hassan, S. Yu, Survey of network coding-aware routing protocols in wireless networks, Journal of Network and Computer Applications 34 (6) (2011) 1956 – 1970, ¡ce:title¿Control and Optimization over Wireless Networks¡/ce:title¿. doi:10.1016/j.jnca.2011.07.012.

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

[50] 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.

[51] 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.

[52] 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.

[53] 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.

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

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

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

[57] 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.

[58] A. Darehshoorzadeh, L. CerdÃ -Alabern, V. Pla, Modeling and comparison of candidate selection algorithms in opportunistic routing, Computer Networks 55 (13) (2011) 2886 – 2898.

[59] 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.

[60] A. Cacciapuoti, M. Caleffi, L. Paura, Optimal constrained candidate selection for opportunistic routing, in: Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE, 2010, pp. 1 –5. doi:10.1109/GLOCOM.2010.5683490.

[61] L. CerdÃ -Alabern, V. Pla, A. Darehshoorzadeh, On the maximum performance in opportunistic routing, in: IEEE WoWMoM 2010, Montreal, Canada, 2010.

[62] D. Koutsonikolas, Y. Hu, C.-C. Wang, Pacifier: High-throughput, reliable multicast without "crying babies" in wireless mesh networks, in: INFOCOM 2009, IEEE, 2009, pp. 2473 –2481.

[63] Y. WenZhong, Z. ZhenYu, W. Bo, W. XiaoHong, A reliable multicast for manets based on opportunistic routing, in: Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on, 2010, pp. 1 –4.

[64] T. Le, Y. Liu, Exploring the gain of opportunistic routing in wireless multicast, Technical-report, Polytechnic Institute of New York University, 5 Metrotech Center, Brooklyn, NY 11201, US (2009).

[65] T. Le, Y. Liu, Opportunistic overlay multicast in wireless networks, in: GLOBECOM, 2010, pp. 1–5.

[66] T. Clausen, P. J. (editors), C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, L. Viennot, Optimized link state routing protocol (OLSR), RFC 3626, pages 1-75, network Working Group (October 2003).

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

[68] R. Sokullu, O. Karaca, Comparative performance study of admr and odmrp in the context of wireless lans and wireless sensor networks, in: Proceedings of the 7th WSEAS International Conference on Telecommunications and Informatics, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 2008, pp. 183–187.