Multicast Ad hoc Routing Through Mobility-Aware Steiner Tree Meshes with Consistency Across Different Mobility Models

Pedro M. Ruiz
Dept. Information and Communications Engineering
University of Murcia
Fac. Informatica, Campus de Espinardo
E-30071, Murcia, Spain
E:pedrom@dif.um.es

Antonio F. Gomez-Skarmeta
Dept. Information and Communications Engineering
University of Murcia
E-30001, Aptdo. 4021, Murcia, Spain
E: skarmeta@dif.um.es

***Abstract*— We study the problem of reducing the data overhead of mesh-based multicast ad hoc routing protocols while maintaining high packet delivery ratios. Our proposed approach adaptively controls the redundancy added by the routing protocol to the minimal data overhead multicast mesh. We show that the problem of finding the minimal data overhead multicast mesh is NP-complete, and we offer a heuristic algorithm to approximate such a mesh. Based on the heuristic algorithm, we propose and evaluate several mesh construction algorithms. We show that the mobility-aware variant being able to control the reliability of the mesh depending upon the mobility of the network offers the best performance. Our simulation results show that our proposed approach offers similar packet delivery ratios than ODMRP at a much lower data overhead. In addition, the simulations show that our mobility metric provides consistent results across different mobility models.**

I. Introduction

Ad hoc wireless networks are formed by a set of mobile nodes which use their wireless interfaces to communicate. Two nodes separated by a distance longer than their transmission ranges, can use intermediate nodes as relays providing multi-hop paths. Nodes are usually free to move, which means that the topology of the network may constantly change. Although routing in these networks is usually more difficult than in traditional wired networks, their completely distributed nature makes them ideal for scenarios in which it is hard (or impossible) to deploy a fixed wireless infrastructure such as catastrophic areas, battlefields or space missions.

Many of these scenarios commonly require an efficient support for many-to-many communications. In particular, multicast is forseen as one of the key areas to support those interactions in mobile ad hoc networks. Key to this is the fact, that the broadcast nature of wireless ad hoc networks is certainly a very good substratum for multicast protocols to achieve important bandwidth savings. In addition, given the power limitations and scarcity of bandwidth in these networks, the efficient support of multicast communications will become a must.

The problem of multicast routing in fixed networks was initially studied by Deering [1]. Several multicast routing protocols like DVMRP [2], MOSPF [3], CBT [4] and PIM [5]) has been proposed for IP multicast routing. However, these protocols are not suited for ad hoc networks because they were engineered for hierarchical IP networks. In addition, they are not able to support constant topological changes at a reasonable cost in terms of control overhead. This impelled the creation of new multicast routing protocols specifically designed for ad hoc networks [6].

In general, ad hoc multicast routing protocols can be classified into tree-based and mesh-based approaches. Tree-based schemes construct a multicast tree from each of the sources to all the receivers, being the source path tree (SPT) the most common alternative. The main advantage of using a tree as the underlying forwarding structure is that the number of forwarding nodes tends to be reduced (although not necessarily optimized). However, a tree is very fragile to topological changes and they often require tree repairing mechanisms. Mesh-based approaches by using additional links in their underlying forwarding structure, manage to deal with mobility very efficiently just by using alternate paths when some of the links break. The main drawback associated to the use of a mesh, is the unnecessary transmission of data packets that may occur even if duplicates are detected and eliminated.

In the author's opinion previous studies, by considering the control overhead as one of the key metrics, have neglected the huge impact that the data overhead may have in the overall performance of a multicast protocol. In multicast routing the path selection has a stronger influence on the overall performance than the control overhead. In fact, if we take into account that data traffic consumes more bandwidth than control traffic, the overhead due to the selection of suboptimal routes may easily become more expensive in terms of bandwidth and energy consumption than the cost of sending a few more control packets.

In this paper, we study how to achieve efficient ad hoc multicast routing by reducing data overhead while maintaining high packet delivery ratios. To satisfy the packet delivery ratio requirements we focus our study on mesh-based multicast ad hoc routing protocols. In particular, we propose a mobility-aware mesh construction algorithm being able to control the redundancy added to the minimal data-overhead mesh. However, we show that the computation of such a minimal data-overhead mesh is an NP-complete problem, and we propose a heuristic and totally distributed approximation algorithm inspired on epidemic algorithms. The proposed algorithm probabilistically selects the paths which increase or reduce the number of forwarding nodes depending upon the network mobility and the existing reliability provided by the multicast mesh. According to our simulations, the proposed approach is able to offer similar packet delivery ratios than existing mesh based approaches while reducing data overhead by 20 to 50%. In addition, our simulations show that the mobility-aware algorithm offers a consistent performance across different mobility models.

The remainder of the paper is organized as follows: section II provides a literature review. Section III discusses the problem of data overhead mitigation and shows how previous multicast tree construction approaches does not minimize it. Section IV describes our network model and the problem formulation. The demonstration of the NP-completeness of the problem and the description of the proposed heuristic is given in section V. In section VI, we show the need for a mobility-aware mesh construction and propose a modified heuristic. Section VII evaluates and analyzes the simulation results of those heuristics across different mobility models. Finally, section VIII provides some discussion and conclusions.

II. Related Work

Several protocols have been proposed for multicast routing in mobile ad hoc networks. They can be classified into tree or mesh-based depending upon the underlying forwarding structure that they use. Tree-based schemes ([7], [8], [9], [10], [13]) construct a multicast tree from each of the sources to all the receivers using any of the tree computations schemes which are discussed in section III. Mesh-based approaches ([11], [12]), compute several paths among senders and destinations. Thus, when the mobility rate increases they are able to tolerate link breaks better than tree-based protocols. Hybrid approaches ([14], [15]) try to combine the robustness of mesh-based ad hoc routing and the low overhead of tree-based protocols. Finally, there are stateless multicast protocols ([16], [17]) in which there is no need to maintain a forwarding state on the nodes. For instance, if the nodes to traverse are included in the data packets themselves. Our proposed heuristic is mainly targeted to mesh-based multicast routing protocols and we therefore focus our discussion on the protocols falling in this category.

ODMRP [11] and CAMP [12] are the best-known mesh-based multicast ad hoc routing protocols. CAMP was designed as an extension of the ”Core Based Trees” (CBT [4]) protocol, offering multiple paths, and relieving the core nodes from doing data forwarding. On the other hand, ODMRP introduces the concept of the forwarding group (FG) as the set of intermediate nodes taking part in the multicast mesh. Although both of them are able to achieve very high packet delivery ratios, and they both guarantee that the shortest paths are included in the multicast mesh. However, they do not attempt to minimize the data overhead incurred by not selecting minimal cost paths.

Most of the works in the literature dealing with the problem of minimizing the cost of multicast trees are related to wired multicast routing. There are many works which propose approximations to Steiner trees. For instance, Jia [27] proposed a distributed heuristic for the Steiner tree problem being able to provide suboptimal multicast trees satisfying certain delay bounds. Waxman [28] also provided two approximation algorithms to the Steiner tree problem in the static case. Chen et al. [29] proposed approximation algorithms to minimize the number of Steiner nodes. However, these proposed approximations are not useful for mobile ad hoc networks because minimal Steiner trees are very fragile to topology changes, and many of the heuristics either are centralized or are not able to work with local information.

For ad hoc networks, most of the works in the literature devoted to the improvement of multi-point forwarding efficiency for routing protocols have been related to the particular case of flooding (i.e. the broadcast storm problem). Only a few papers study those mechanisms for multicast ad hoc routing. Lim and Kim [30] analyzed the problem of minimal multicast trees in ad hoc networks, but they defined several heuristics based on the minimum connected dominating set (MCDS) which are only valid for flooding. Lee and Kim [31] worked on a solution to reduce the number of forwarding nodes using a probabilistic approach. However, the overhead reductions where lower than the results we have obtained, their non-adaptive path selection probability makes their proposal unable to perform well under different network conditions and mobility rates.

III. Multicast Forwarding Structures

Multicast ad hoc routing protocols are in charge of selecting a subset of relay nodes from the network for each of the multicast sources and groups. They are elected so that they form a connected subgraph including the source and all the receivers. That connected subgraph may resemble different forwarding structures such as shortest path trees, shared trees, minimal steiner trees, acyclic meshes, etc. In general, the underlying forwarding structure is protocol-specific because it strongly depends on the path creation process implemented by that particular protocol. In general, those relay nodes are called *forwarding nodes*. Note that even a source or a receiver can also be a forwarding node.

There are two basic structures used by multicast trees: shortest path trees and shared trees [18]. Given a multicast source s, a shortest path tree is formed by the aggregation of the shortest paths from any receiver r to s. The main advantage of this kind of trees is that each destination receives multicast data through its best route, which usually means that the latency from s to each r is also minimized. However, these trees are not optimal in terms of the overall number of data transmissions required or the number of forwarding

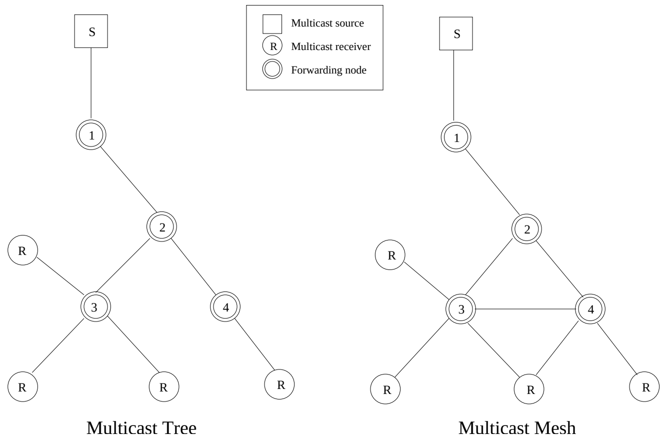


Fig. 1. Tree-based vs. mesh-based multicast forwarding structure

nodes which are required. So, they do not provided an optimal bandwidth consumption.

A second variant are the so-called shared trees. Shared trees try to reduce the cost of the multicast tree by reducing the number of links which are required to connect sources and receivers. This is done by selecting the links in the tree which can be used by a bigger number of receivers. Of course, in the resulting tree individual paths from sources to receivers might not be optimal. For the particular case of ad hoc networks, other approaches like the use of multicast meshes with redundant links has been proposed in protocols like ODMRP [11] and CAMP [12]. These structures are particularly interesting to deal with the mobility of the nodes, but obviously they do not minimize the cost of multicast forwarding. Fig. 1 shows a multicast tree and a multicast mesh. Forwarding nodes are identified by a double circle. As the figure depicts, in a tree every node has a single parent towards the source. However, in multicast meshes, several parents are allowed. Thus, they can offer a higher reliability on the advent of changes in the topology, at the cost of possibly duplicating data packets in cases in which it is not required because there is no change in the topology.

The problem of finding a minimum cost multicast tree is well-known as the minimum Steiner tree problem. R. Karp [19] demonstrated by a transformation from the exact cover by 3-sets problem that this problem is NP-complete even when every link has the same cost. There are some heuristic algorithms to compute minimal Steiner trees. For instance, the MST algorithm ([20], [21]) provides a 2-approximation, and Zelikovsky [22] proposed an algorithm which obtains a 11/6-approximation. However, given the complexity of computing this kind of trees in a distributed way, most of the existing multicast routing protocols use shortest path trees, which can be easily computed in polynomial time. For mesh-based forwarding, the multicast mesh is usually computed as the union of various shortest path trees.

In addition, the problem of minimizing the cost of a multicast tree in an ad hoc network needs to be re-formulated in terms of minimizing the number of data transmissions, rather than the usually-considered edge-cost. Given the broadcast nature of wireless ad hoc networks, a minimum Steiner tree does not minimize the cost of the multicast tree. The cost assignment function used in wired networks is not well-defined for ad hoc networks. That is, by assigning a cost to each link of the graph, existing formulations have implicitly assumed that a given node v, it needs k transmissions to send a multicast data packet to k of its neighbors. However, in a broadcast medium, the replication of a multicast data packet from a given node v to any number of his neighbors can be done with a single data transmission. Thus, in ad hoc networks the minimum cost tree is the one which connects sources and receivers by issuing a minimum number of transmissions, rather than the tree with the lower edge cost. This can be easily observed in the example in Fig. 2.

In the next section we provide formal definitions of data overhead and minimal data overhead trees. In addition, we give a formulation for the problem of finding a minimal data overhead multicast mesh, and we demonstrate that the problem is NP-complete.

IV. Minimal Data Overhead Multicast Mesh

A. Network model

We represent the ad hoc network as an undirected graph $G(V, E)$ where V is the set of vertices and E is the set of edges. We assume that the network is two dimensional (every node $v \in V$ is embedded in the plane) and mobile nodes are represented by vertices of the graph. Each node $v \in V$ has a transmission range r. Let $dist(v_1, v_2)$ be the distance between two vertices $v_1, v_2 \in V$. An edge between two nodes $v_1, v_2 \in V$ exists iff $dist(v_1, v_2) \leq r$ (i.e. v_1 and v_2 are able to communicate directly). In wireless mobile ad hoc networks some links may be unidirectional due to different transmission ranges. However, given that lower layers can detect and hide those unidirectional links to the network layer, we only consider bidirectional links. That is, $(v_1, v_2) \in E$ iff $(v_2, v_1) \in E$.

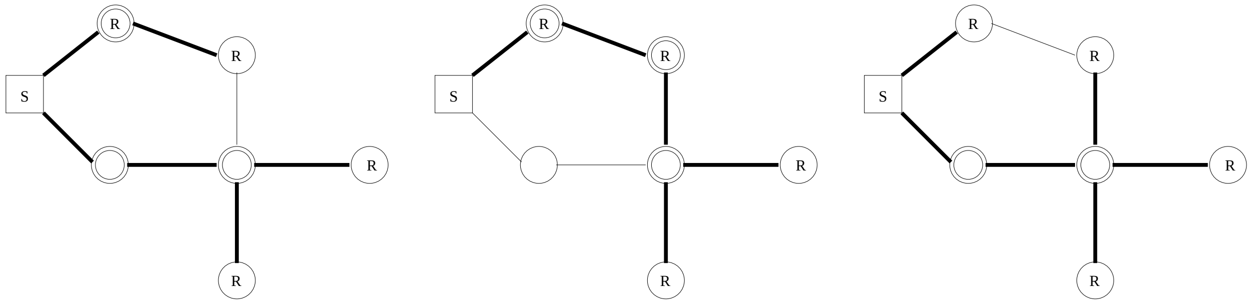
The set of all multicast sources and receivers is denoted by V' ($V' \subseteq V$). More precisely, V' is defined as $R \cup S$ where R is the set of multicast receivers and S is the set of multicast sources.

B. Defining data overhead

Before formulating the problem of computing the minimal data overhead multicast mesh, we give some definitions.

Definition 1: Given a multicast tree T, we can define the number of transmissions required to deliver a multicast datagram from the source to all the receivers in T as a function $C : T \rightarrow \mathbb{Z}^+$. We denote by $C(T)$ the number of such transmissions.

Definition 2: Given a graph $G = (V, E)$, a multicast source $\{s\} \in V$, and a set of receivers $R \subseteq V$, we denote by $T^* \subseteq G$ the multicast tree satisfying that $C(T^*) \leq C(T)$ for every possible multicast tree $T \subseteq G$. T^* is the multicast tree requiring the minimal number of transmissions.



(a) Shortest Path Tree, $C(T) = 4$ (b) Steiner Tree, $C(T) = 4$ (c) Min. Data Overhead, $C(T) = 3$

Fig. 2. A comparison of an SPT and an Steiner tree over the same ad hoc network

Definition 3: Given a multicast tree T we define the data overhead of T as $\omega_d = C(T) - C(T^*)$. With this definition we can prove the following theorem. It is immediate from this definition that $\omega_d(T^*) = 0$.

Theorem 1: The minimal-data overhead multicast tree, is the one with the minimum number of forwarding nodes.

Proof: Let T^* be the minimal-data overhead multicast tree, and let F^* be the set of forwarding nodes of T^*. We will show that there is no tree with a higher number of forwarding nodes producing a lower data-overhead.

Lets consider another minimal data-overhead tree $T^{'}$, with associated forwarding set $F^{'}$, and lets assume that $|F^{'}| > |F^*|$. Each multicast data packet is only relayed by forwarding nodes in any of the trees. Thus, the number of times such a message is transmitted in $T^{'}$ is $C(T^{'}) = 1 + |F^{'}|$. Similarly, $C(T^*) = 1 + |F^*|$. So, given that we assumed that $|F^{'}| > |F^*|$, it is clear that $C(T^{'}) > C(T^*)$, which means that $T^{'}$ cannot be a minimal-data overhead tree. ■

The previous definition can also be extended for multicast meshes. However, before that, we need an additional definition and the following theorem.

Definition 4: Given a graph $G = (V, E)$ and a multicast mesh $M \subseteq G$ (i.e. a subgraph of G formed by the sources, the receivers and the forwarding nodes), we can define the number of transmissions required to deliver a multicast datagram from each of the sources to all the receivers in a mesh M as a function $C^{'} : M \to \mathbb{Z}^+$. We denote by $C^{'}(M)$ the number of such required transmissions in the mesh M. In addition, given the set of forwarding nodes $F \subseteq V$, and the set of sources $S \subseteq V$, $C^{'}(M) = |S| \times (1 + |F|)$. That is, provided that in a multicast mesh a forwarding node sends each multicast data packet only one time, the number of transmissions for each source is the one for the source itself plus one for each forwarding node.

Theorem 2: Given a graph $G = (V, E)$, a set of multicast sources $S \subseteq V$, and a set of receivers $R \subseteq V$, the minimal number of transmissions required to deliver a datagram from each of the sources to all the receivers is $\sum_{i=1}^{|S|} C(T_i^*)$ being T_i^* the minimal data overhead multicast tree for each source $i \in S$.

Proof: Lets assume that there is a minimal number of transmissions mesh (or shared tree) $T^{'}$ which connects all sources and receivers so that

$$C(T^{'}) < \sum_{i=1}^{|S|} C(T_i^*)$$

Let F be the set of forwarding nodes in $T^{'}$. Then, by definition 3.4, $C(T^{'}) = |S| \times (1 + |F|)$.

Let T_{max}^* be minimal data overhead tree with the bigger number of transmissions. That is, $C(T_{max}^*) \geq C(T_i^*)$ for every source $i \in S$. The number of forwarding nodes in T_{max}^* is then $C(T_{max}^*) - 1$. Given that T_{max}^* is the minimum number of transmissions tree for one of the sources, and provided that $T^{'}$ also contains that source, then $|F| \geq C(T_{max}^*) - 1$. This means that the following relation holds:

$$1 + |F| \geq C(T_{max}^*) \tag{1}$$

However, by definition of $T^{'}$ our initial assumption should be satisfied. This is equivalent to say that the following relation should hold

$$|S| \times (1 + |F|) < \sum_{i=1}^{|S|} C(T_i^*)$$

But this can only happen if $1 + |F| < C(T_i^*)$ for every source $i \in S$. Which is a contradiction provided that $1 + |F| \geq C(T_{max}^*) \geq C(T_i^*), i = 1 \ldots |S|$. ■

Based on the previous theorem, we can give the following definition of data overhead for a multicast mesh.

Definition 5: Let $G = (V, E)$ be a graph, $S \subseteq V$ be a set of sources, and $R \subseteq V$ be a set of receivers. Let $T^* = \{T_1^*, T_2^*, \ldots, T_i^*, \ldots, T_{|S|}^*\}$ be the set of trees containing the minimal data-overhead multicast tree for each of the sources. Let $M \subseteq G$ be a multicast mesh and let F be the set of forwarding nodes in M. The data overhead of M can be defined as:

$$\omega_d(M) = |S| \times (1 + |F|) - \sum_{i=1}^{|S|} C(T_i^*) \tag{2}$$

The lefthand term of the subtraction in (2) is obtained from the fact that in a multicast mesh every forwarding node makes a single transmission of each data packet generated by any of the sources. The righthand term is simply the minimal number of transmissions required to deliver a message from each of the sources to all the receivers (as shown by theorem 2).

In addition, combining definition 4 and theorem 2 we can also introduce the following corollary.

Corollary 3: A multicast mesh M connecting several sources and receivers does not offer the minimal data overhead. The proof is immediate by substituting the inequality (1) in (2) to show that $\omega_d(M) \geq 0$.

C. Problem formulation

Although a multicast mesh does not offer the minimal number of transmissions, multicast meshes are very interesting for multicast ad hoc routing due to their reliability. Thus, it makes sense for our adaptive approach to compute such a minimal data-overhead mesh to add additional redundancy only as required. This particular problem can be formulated as follows:

Given a graph $G = (V, E)$ and a set of nodes $V' = R \cup S$ so that $V' \subseteq V$, find the smallest set of nodes $X \subset G$ such that the following conditions are satisfied:

1) $X \supseteq S$
2) G_X is connected
3) G_X is a vertex cover for R

Condition 1) means that X must contain at least all the sources. This is because we want X to be the set of all nodes performing multicast transmissions or forwarding. Thus, at least all the sources should be in X and eventually, other nodes required to connect sources and receivers will also be in X. Condition 2) requires that the subgraph induced by the set of nodes X in G, denoted by G_X is connected. This means that there is at least one path in G_X connecting every source and every receiver. Finally, condition 3) states that given any node $r \in R$, there is a node $x \in X$ so that $dist(x, r) \leq 1$. In other words, it means that every receiver is at no more than 1 hop from one of the nodes sending or forwarding multicast packets. Note that in the particular case in which a receiver $r_j \in R$ belongs to X the condition still holds because in that case $dist(r_j, r_j) = 0 < 1$.

Note that by finding the smallest set of nodes X we are inherently asking for the induced subgraph with the minimal $C(G_X)$. In addition, as G_X is a mesh, that also means minimizing $\omega_d(G_X)$.

D. NP-completeness

We demonstrate through the following theorem that the problem of finding a minimal data overhead multicast mesh is NP-complete.

Theorem 4: Given $G = (V, E)$ and $V' \subseteq V$ so that $V' = R \cup S$, the problem of finding the smallest X so that $X \supseteq S$, G_X is connected and G_X is a vertex cover of R is NP-complete.

Proof: We consider a special case of our problem in which $R = V - S$. For this particular case given $S \subseteq V$, we are interested in finding the smallest $X \supseteq S$ which covers $(V - S)$. This is equivalent to say that we are looking for the smallest $X \supseteq S$ such that for any vertex $v \in V$, $dist(v, X) \leq 1$.

Given a graph $G = (V, E)$, the problem of finding the smallest $X \subseteq V$ such that $dist(v, S) \leq 1$ for any $v \in V$ is the well-known vertex cover problem. This problem, which is NP-complete [23], is thus a special case of our problem when $R = V - S$. Then, we can conclude that our original problem is also NP-complete. ■

V. Heuristic for Minimal Data Overhead Meshes

Given the NP-completeness of the problem of finding the minimal data-overhead multicast mesh, in this section we propose an heuristic algorithm. In addition, we justify the validity of the proposed approach.

According to Eq. (2), minimizing the data overhead of a multicast mesh, requires the minimization of the expression $|S| \times (1 + |F|)$. The number of sources ($|S|$) is something the routing protocol cannot change. So, the only way to reduce the overhead of a multicast mesh is computing multicast routes which reduce the number of forwarding nodes ($|F|$) which are used.

From this observation, we provide a distributed heuristic to approximate the minimal data overhead multicast mesh by reducing the number of forwarding nodes which are required to connect multicast sources and receivers. The proposed algorithm (see algorithm 1) is a distributed counting process inspired on epidemic algorithms. Basically, a counter is added to route request (RREQ) messages. A multicast source initially sets this counter to zero. During the propagation of RREQ messages throughout the ad hoc network, this counter is modified by intermediate nodes to reflect the number of non-forwarding nodes in their path to that multicast source. So, this counter (denoted by FNCount) computes the number of new forwarding nodes which would be added to the multicast mesh if that particular path is selected. By giving more preference to those routes with the lower FNCount, ad hoc nodes can reduce the data overhead and the resulting multicast mesh will be an approximation to the minimal data overhead multicast mesh.

However, this process itself is not enough to guarantee a good approximation, because the resulting paths would be like shortest path trees in which the metric is changed from the number of hops to the number of new forwarding nodes. A much better approximation can be obtained if we compute Steiner trees using the FNCount metric, and then we combine them together to build the multicast mesh. The big problem is that the Steiner tree problem is also NP-complete, and we require a lightweight, distributed and localized algorithm.

However, we found that by a simple modification to the initial idea we can approximate such Steiner trees. The idea is to simply make receivers to reset the FNCount when they propagate RREQ messages. By doing that, ad hoc nodes will join the tree through the FNCount path to any source or any receiver. This has exactly the same effect that computing the minimum spanning tree (MST) on the metric closure of the graph consisting only of sources and receivers. In other words, we propose a very lightweight, localized and simple algorithm similar to the MST heuristic for Steiner trees, which unfortunately is a centralized algorithm. The details of the algorithm are presented in algorithm V.

Algorithm 1 Minimal Data Overhead Mesh Heuristic

```
loop
  Receive route request packet RREQ
  if RREQ.seqno < LastSeqno then
    continue
  else if RREQ.seqno > BestRREQ.seqno or RREQ.FNCount < BestRREQ.FNCount then
    if RREQ.seqno > BestRREQ.seqno then
      LastSeqno ← RREQ.seqno
      NewRREQ ← RREQ
      Schedule sending of RREQ
    end if
    if NotSent(NewRREQ) then
      BestRREQ ← RREQ
      if forwarder node and not receiver then
        NewRREQ.FNCount ← RREQ.FNCount
      else if forwarder node and receiver then
        NewRREQ.FNCount ← 0
      else if not forwarder and receiver then
        NewRREQ.FNCount ← 1
      else
        NewRREQ.FNCount ← RREQ.FNCount+1
      end if
    end if
  end if
end loop
```

In the next section we study and propose an adaptive and mobility-aware mesh construction algorithm based on the principle of adjusting the reliability of the minimal data overhead multicast mesh to the changing conditions of the network. Our goal is to add reliability to the mesh computed by this algorithm as required.

VI. Adaptive Mesh Construction

As we mentioned before, achieving efficient multicast ad hoc communications requires a good trade-off between forwarding efficiency and reliability. A higher number of forwarding nodes produces a higher reliability but it worsens the forwarding efficiency. Our goal is to provide the ad hoc protocol with enough information to adaptively decide when to increase or reduce the data overhead. In addition, we believe that the mobility of the network is an important parameter to consider. When the mobility rate in the network is high, it is interesting to count on the additional paths created when the number of forwarding nodes is increased to alleviate the effect of link breaks. However, in scenarios with a lower mobility, having a big number of forwarding nodes usually is a waste of resources provided that there are few topological changes and most of the additional paths are not required. Thus, we believe that a mesh-based multicast ad hoc routing protocol can benefit from being able to adaptively adjust the number of forwarding nodes which are created according to the network conditions. To achieve that, we introduce a probabilistic route selection scheme based on the heuristic described in the previous section. Depending on the network conditions, we adjust the probability of selecting the paths producing a lower data overhead or those incrementing the reliability of the mesh.

A. Mobility metric and reliability estimation

One of the key network parameters which influences the protocol's performance is the stability of the network. There are several mobility metrics which can be considered. Examples of those are the mean duration of the paths, the link change rate and the link duration among others. The duration of a link is defined as the mean number of time units that the link is up during a certain time window. Thus, the mean link duration is defined as average of the links durations of all the links of a particular node. This metric is particularly interesting as it can be easily computed by a node in real scenarios using only local information (i.e. and without the need for additional equipment or location information such as GPS). In addition, Boleng et al. [24] showed that link duration is much more interesting than previous mobility metrics used in the literature. They showed that the link duration and the packet delivery ratio for unicast ad hoc routing protocols are strongly correlated. In fact, other metrics such as the link break rate were not able to achieve such a big degree of correlation. That is why we have selected the mean link duration as the mobility metric to use.

An important parameter to compute the link duration is the time window under consideration. If the time window is too large, old links may influence very much the mean link duration, causing adaptive approach not to be very responsive. If it is too small, the link duration might fluctuate very much, without capturing the stability effect of long lived links. The determination of a proper time window was not addressed in [24]. However, from our simulation results we have found a value of 120 seconds to be a good trade-off. In addition, unlike previous works in the literature, we do not use any additional control messages (e.g. beacons or HELLOs) to compute mean link durations. We just use control and data messages that the routing protocol would send anyway to asses the status of individual links. So, there is no extra overhead when using this approach, and the measurements are accurate enough.

Unfortunately, the studies in [24] did not considered the case of multicast ad hoc routing. Moreover, they did not consider the case of mesh-based routing in which additional paths may help at reducing the effects of mobility. So, we evaluated ourselves the relation between the packet delivery ratio and the link duration when using a multicast mesh. To make that evaluation we used ODMRP challenged with the minimal data overhead mesh heuristic introduced before as the routing protocol. The results in Fig. 3 show that trying to

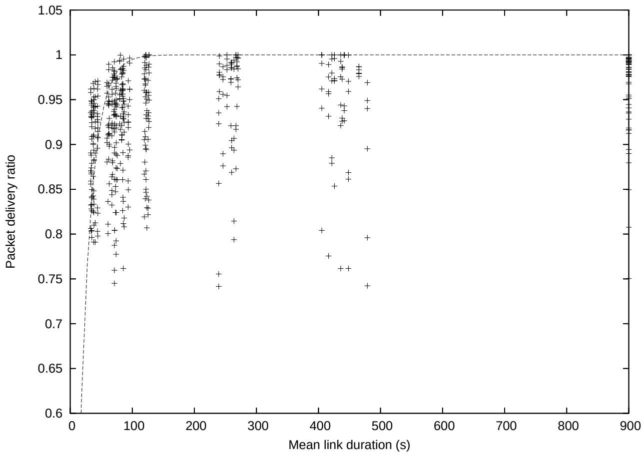


Fig. 3. Packet delivery ratio vs. mean link duration

fit a curve doing regression as the authors did in [24] gives a considerable fitting error, compared to the 98.5% coefficient of determination which they obtained. This indicates that there are other parameters in addition to the link duration which influence the packet delivery ratio in mesh based ad hoc routing.

This is because in mesh-based multicast routing in which redundant links or paths may exist, the link duration alone is not responsible for the differences in packet delivery ratios. For instance, given two scenarios with the same link durations, the one with the higher reliability can obtain a bigger packet delivery ratio. So, in addition to the link duration we need to consider another easy-to-compute metric to estimate the existing reliability that the multicast mesh already has.

We have found that the number of multicast sources has a strong correlation with the number of forwarding nodes. In addition, is very easy to compute locally without additional overhead. In fact, we have found that a much better coefficient of determination can be obtained when considering independently the the link duration correlation with the packet delivery ratio for each number of multicast sources. This is explained by the fact that as the number of sources increase, so does it the number of forwarding nodes. Thus, the greater the number of sources, the higher the reliability of the multicast mesh.

As we show in the next subsection, by considering both the link duration and the packet delivery ratio, we can estimate the amount of additional reliability which the minimal data multicast mesh requires. So, our probability function which influences the probabilistic path selection will relay on these two metrics.

B. Adaptive mesh construction

The design of a deterministic mesh construction algorithm being able to control data-overhead requires the consideration of scenario-specific or overall network metrics. These metrics are usually difficult or expensive (in terms of control overhead) to obtain. Thus, we have decided to use a probabilistic scheme being able to work with local and lossy information while still achieving a good performance.

Our goal is to build the multicast mesh by extending the minimal data overhead mesh according to the mobility of the network and considering the innate reliability that the minimal mesh already has. So, we can differentiate two different components affecting the probabilistic path selection: the mobility and the existing reliability in the minimal mesh. For the first component we use the mean link duration as a relevant metric. The number of multicast sources is used for the latter component.

The probabilistic mesh construction is based on the minimal data overhead mesh construction algorithm depicted in algorithm 1. When a node receives a route request (RREQ), instead of always selecting every time the route with the lower FNCount, it selects the route with the lower FNCount with a certain probability π_s. By adjusting the value of π_s we can control whether the selected paths are those adding reliability or those reducing data overhead. A bigger value of π_s reduces data overhead while a lower value of π_s increases reliability. Key to this, is to find a suitable probability assignment function π defined in the interval $[0, 1]$ such that given a link duration ld and given a number of multicast sources $|S|$, $\pi(ld, |S|) = \pi_s$ ends up generating the required amount of redundancy.

In order to compute a suitable function π, we focus on finding appropriate functions $\pi_1(|S|)$ and $\pi_2(ld)$, which we shall later combine to find the probability assignment function π.

1) Finding a function for $\pi_1(|S|)$: Let $S = \{s_1, s_2, \ldots s_n\}$ be the set of multicast sources. Let M_i be the forwarding mesh obtained when the source s_i is considered individually, and let F_i be the set of forwarding nodes of M_i. If we consider the overall multicast mesh produced when considering all the n sources, $M^n = M_1 \oplus M_2 \oplus \cdots \oplus M_n$, the number of forwarding nodes in M^n denoted as $|F^n|$ can be recursively defined as

$$|F^n| = |F_n| + |F^{n-1}| - |F_n \cap F^{n-1}|$$

Obviously, the new number of forwarding nodes added is

$$|F^n| - |F^{n-1}| = |F_n| - |F_n \cap F^{n-1}|$$

So, given a fixed set of receivers R, a multicast mesh has a higher resilience as the number of sources increase, but the increasing rate is not linear. The bigger the number of sources, the lower will be the number of new forwarding nodes added. This is because for each forwarding node $f \in F_i$ the probability of the event that that node was already considered in the mesh for any other source $M_j, j \neq i$ is increased. So, $\pi_1(|S|)$ should be monotonously decreasing with a lower decreasing rate as $|S|$ increases. By simulating different possible functions we found that a good approximation was given by (3).

$$\pi_1(|S|) = \frac{1}{1 + |S|^2} \quad (3)$$

2) Finding a function for $\pi_2(ld)$: From our simulations results of the minimal data overhead mesh, we computed the packet loss ratio for different link durations. This ratio yields an approximation of the additional reliability which

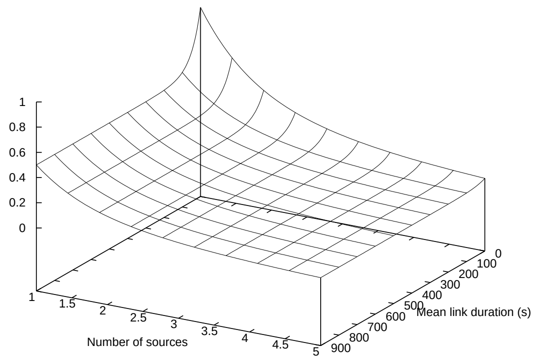


Fig. 4. $\pi(ld, |S|)$ values for in the probabilistic path selection

would have been required to deliver all the messages. This function, which is the inverse of the fitting function in Fig. 3, showed the exponential nature of the target function, being $\pi_2(ld) \sim e^{-\alpha \times ld}$. Through simulation we found that the value of α was, as expected, dependent on the number of sources. For a higher number of sources, the needed reliability was lower because of the additional paths in the mesh.

So, by further simulating combinations of $\pi_1(|S|)$ and $\pi_2(ld)$ with varying α, we find the function in (4) to be appropriate. This function (which is plotted in Fig. 4) mimics the expected behavior. The higher the number of sources, the lower $\pi(ld, |S|)$ And as the link duration decreases $\pi(ld, |S|)$ increases. Of course, for a fixed ld, $\pi(ld, |S|)$ is lower as the number of sources increase.

$$\pi(ld, |S|) = \frac{1}{1 + |S|^2} \times \left(1 + e^{\frac{-ld}{50}}\right) \quad (4)$$

VII. Simulation Results

In order to assess the effectiveness of the proposed schemes, we have implemented the proposed changes into the original ODMRP code from the Monarch extensions [26] to the NS-2 [25]. ODMRP has been selected as the mesh-based multicast routing protocol because it is very well-documented in the literature where it is shown to offer very good performance results.

A. Integrating the heuristics in ODMRP

For the integration with ODMRP we have had to consider three main aspects: the integration of algorithm 1 in ODMRP's control messages, the changes to the route selection process and the avoidance of cycles.

The integration with ODMRP's control messages has been almost immediate. The only change has been the introduction of a new field in JOIN_QUERY messages in which the $FNCount$ field is being updated. In addition, we introduced an "aggregation time" so that an ODMRP node can wait for a short period of time before propagating a JOIN_QUERY. This is done so that the node can receive other routes producing a lower data overhead than the shortest path. A node selects the route with the lower $FNCount$. If no other route is received in that time then the shortest path route is propagated.

In addition, it may happen that two nodes select one each other as the next hop towards a source because both of them has the same cost (e.g. if both are receivers). So, a tie resolution method has been proposed by which the node with the lowest ID accepts being a forwarding node for the other node, and selects another neighbor towards the source. To detect that situation, a new field is added to the JOIN_QUERY message which includes the neighbor that the sender of the JOIN_QUERY message selected as its next hop towards the source. This was not a problem in the original specification of ODMRP because alternative routes (different from the shortest path) were detected as duplicate control packets and were not processed.

For the case of the mobility-aware mesh construction algorithm, the route selection process has been changed to the probabilistically route selection explained before. That is, a node always select a route with a greater sequence number rather than an old one. But, in the case of receiving two equally fresh routes with different cost, it selects the one with the lower cost with a probability $\pi(ld, |S|)$ according to (4). For the scheme which only uses the number of sources, the route selection is the same but the probability is $\pi_1(|S|)$ according to (3).

Table I summarizes ODMRP parameters which have been used for the simulations. They are the standard values except for the J_QUERY_AGG parameter which has been introduced only for the particular case of the proposed modifications.

TABLE I
ODMRP Parameters for simulation

Parameter	Meaning	value
REF_INTERVAL	Time between JQ fboding	3s
FG_TIMEOUT	Timeout for the FG_GLAG	9s
J_REPLY_RET	Max. # of J. REPLY retransmissions	3
J_REPLY_AGG	Timeout for aggregation of J. REPLYs	0.025s
J_QUERY_AGG	Timeout for aggregation of J. QUERYs	0.015s

B. Simulation methodology and performance metrics

The simulated scenario consists of 100 mobile nodes randomly distributed over an area of 1200x800m. We used three different mobility models: random waypoint, Gauss-Markov and Manhattan. In all the cases the nodes were configured to move at an speed uniformly distributed between 0 and 20 m/s, with variations according to the mobility models. The MAC layer used is IEEE 802.11b DCF at 2 Mb/s, and a communication range of 250 m. The simulation time is 900 seconds, and the traffic load consisted of 1, 2 and 5 CBR sources at 5 packets per second, and 330 bytes per packet. The number of receivers used were 5, 15 and 30.

To assess the effectiveness of the different protocols, we have used the following performance metrics:

- Packet delivery ratio (PDR). Defined as the number of data packet successfully delivered over the number of data packets generated by the sources.
- Normalized packet overhead (NOH). Defined as the total number of control and data packets sent and forwarded normalized by the total number of packets successfully delivered.
- Forwarding Efficiency (FEF). The mean number of times that a multicast data packet was forwarded by the routing protocol. This metric represents the efficiency of the underlying forwarding structure.
- Mean delivery latency (DLY). The mean of the latencies for a data packet at the different receivers. The mean delivery latency is then the average of these mean latencies over all the data packets.

To show the validity of the proposed approach we assess its performance from two different perspectives. Firstly we study its ability to react to mobility, and secondly we show that those results are consistent across different mobility models. The data labeled as 'ODMRP' in the plots corresponds to the original ODMRP protocol. The minimal data overhead heuristic is labeled as 'ODMRP-NF'. The adaptive mesh construction using only the number of sources is labeled as 'ODMRP-NS' and the mobility-aware mesh construction approach is labeled as 'ODMRP-LD'.

C. Adaptation to mobility of the network

For these experiments we consider the random waypoint mobility model in which we vary the pause times from 0 to 900 seconds. The experiments with a pause time of 0 seconds correspond to dynamic scenarios whereas a pause time of 900 seconds represent a totally static scenario. We evaluate how do the different approaches perform at different pause times.

The packet delivery ratio as a function of the pause time is depicted in Fig. 5(a) for the 1 source and 15 receivers scenario, and in Fig. 5(c) for the scenario with 2 sources. As we can see, all the protocols (except the minimal data overhead) obtain a very high packet delivery ratio. However, it is interesting to note that the mobility-aware approach (ODMRP-LD) is able to offer a higher PDR when the mobility of the network is medium and high compared to ODMRP-NS, which only considers the number of sources. This is because by reacting to the mobility of the network, the ODMRP-LD approach manages to add additional reliability just when it is needed.

The minimal data-overhead mesh, yields a 10% lower packet delivery ratio in the case of 1 source due to the lack of redundant links, which limits its reliability. As predicted by our analysis in section VI, in the case of 2 sources, the packet delivery ratio offered by the minimal data-overhead mesh is much higher, due to the additional reliability introduced by the second source.

So, although ODMRP delivers around a 0.5% more packets than the mobility-aware approach, it requires around a 20 to 50% more data overhead (see Fig. 5(b) and Fig. 5(d)) to achieve this packet delivery ratio. The proposed alternatives have a lower overhead due to the reduction in the number of forwarding nodes. In addition, is it is depicted in the figures, the mobility-aware mesh construction approach, by using the link duration, is able to very the number of forwarding nodes depending on the network mobility. For instance, in Fig. 5(b), it is clearly shown how ODMRP-LD reduces the number of forwarding nodes as the network becomes more static. In addition, by comparing 'ODMRP-LD' and 'ODMRP-NS' in the scenario with 2 sources (see Fig. 5(d)), we can perceive that the differences in overhead are much lower than in the case of having a single source. This is because the definition of $\pi(ld, |S|)$ takes advantage of the additional reliability provided by having two sources, and allows ODMRP-LD to reduce the reliability added due to mobility because part of it, has been already compensated by the reliability added by the second source.

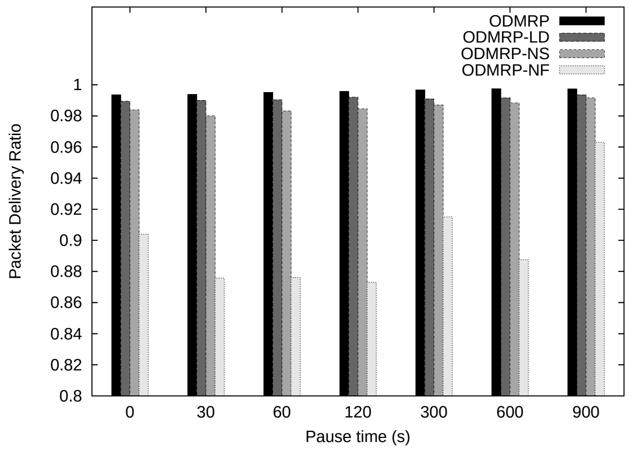
An additional benefit of reducing the data overhead is the reduction of the mean latency between the source and the receivers, by means of reducing the MAC-layer contention and collisions among forwarding nodes. Thus, the overall average latency offered by our proposed schemes is lower than the one the original ODMRP offers (see Figs. 6(j), 6(k) and 6(l)). In the next subsection we show that, according to our simulations, the benefits offered by the proposed approaches are also preserved across different mobility models.

D. Consistency across mobility models

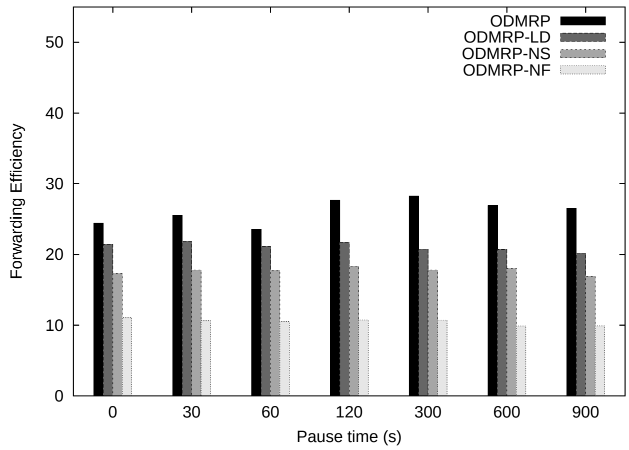
Another important aspect to evaluate is whether the link duration as mobility metric used by our approach is able to offer a consistent view of the network mobility when we consider different mobility models. For these experiments we will evaluate the same approaches under the "random waypoint", "Gauss-Markov" and "Manhattan" mobility models.

Regarding the packet delivery ratio, Fig. 6(a), 6(b) and 6(c) show that the mobility-aware heuristic is able to maintain very similar packet delivery ratio regardless of the mobility models. This is because even if the mobility rates are different across models, the link duration still provides a good feedback that allows the protocol to add redundancy. This can be easily observed by the increase of the forwarding efficiency (see Fig. 6(a), 6(b) and 6(c)) that ODMRP-LD has in the cases of Gauss-Markov and Manhattan mobility models. So, by adding additional reliability ODMRP-LD manages to offer a sustained packet delivery ratio across mobility models.

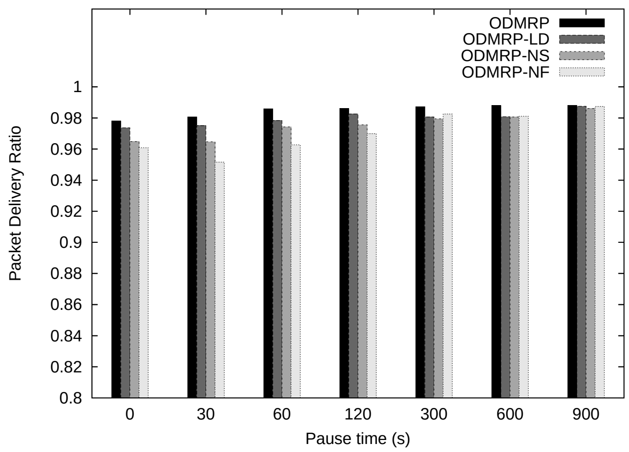
In the case of ODMRP, it also manages to maintain similar packet delivery ratios for different mobility models, at the cost of a very high data overhead. This is because the randomness in the access to the MAC layer can make the shortest path routes to change very quickly.This is because ODMRP considers the path from which the JOIN_QUERY is first received to be the shortest path. Thus, provided that after a new route has been selected the forwarding nodes in the old path will still remain active for two additional refresh intervals, the number of forwarding nodes quickly increases. This means that ODMRP has a higher reliability which is obtained at the cost of increasing data overhead. This is clearly shown in Fig. 6(a), 6(b) and 6(c).



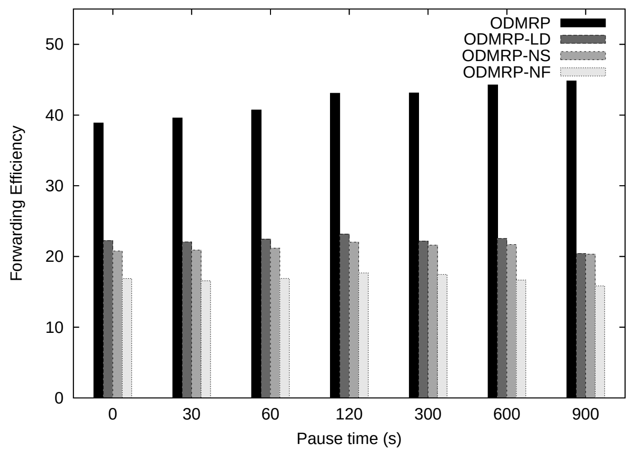
(a) PDR for 1 source and 15 receivers



(b) FEF for 1 source and 15 receivers



(c) PDR for 2 sources and 15 receivers



(d) FEF for 2 sources and 15 receivers

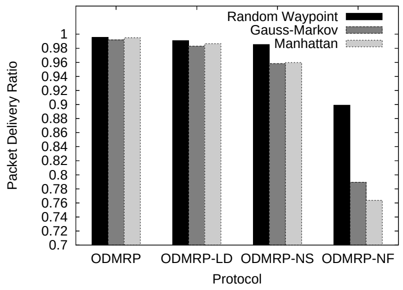
Fig. 5. Performance for different mobility rates

The minimal data-overhead mesh heuristic (ODMRP-NF) is the one whose performance varies more with mobility. This is because the minimal mesh results in nearly the same reliability in most of the cases. Thus, the final PDR obtained depends mainly on the mobility of the network, which is different across mobility models. So, from the graphs we can infer that the mobility rate is higher in the Manhattan mobility model than in the Gauss-Markov one, which is also higher than in the random waypoint model. In the case with 2 and 5 sources the difference is reduced because of the reliability automatically added by additional sources. In the case of ODMRP-NS we can observe that there is some variation across mobility models but it is not so strong as in ODMRP-NF. But in any case, we can see that the packet delivery ratio is reduced compared to the experiments performed with the random waypoint model.

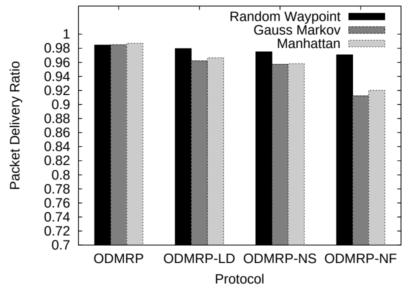
In summary, for any of the mobility models, the mobility-aware heuristic is able to offer a similar packet delivery ratio than to that of ODMRP, while still reducing very much the data overhead.

Another important metric to consider is the end-to-end delay. As we show in Fig. 6(j), 6(k) and 6(l), the original ODMRP produces a higher end-to-end delay than the other alternatives in every mobility model. Given that ODMRP uses the shortest paths, the delay should be reduced. However, the higher delay is explained by the additional channel contention due to the bigger number of forwarding nodes. This is also clearly observed in the case of 5 sources (see Fig. 6(l)) in which the network gets almost saturated with ODMRP while it is not so loaded when the other variants are used.

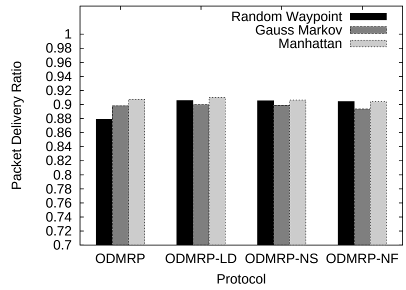
So, we can conclude that by reducing data overhead the proposed mobility-aware mesh construction algorithm manages to offer similar performance than ODMRP at a lower cost in



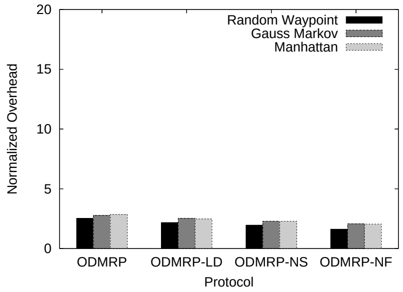
(a) PDR, 1 source, 15 receivers



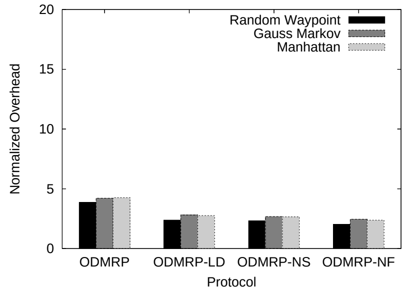
(b) PDR, 2 sources, 15 receivers



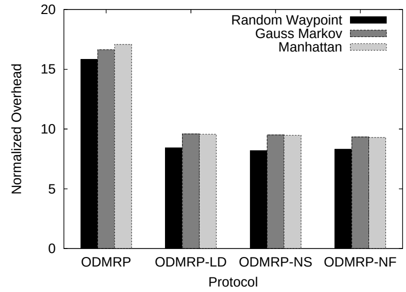
(c) PDR, 5 sources, 5 receivers



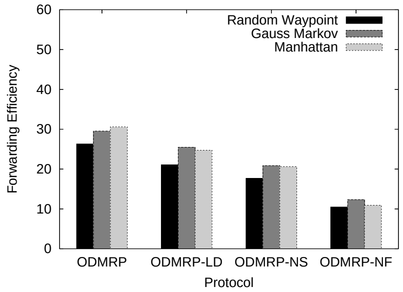
(d) Norm. overhead, 1 source, 15 receivers



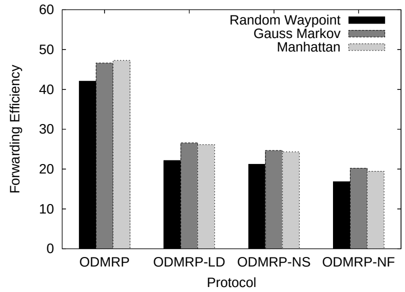
(e) Norm. overhead, 2 sources, 15 receivers



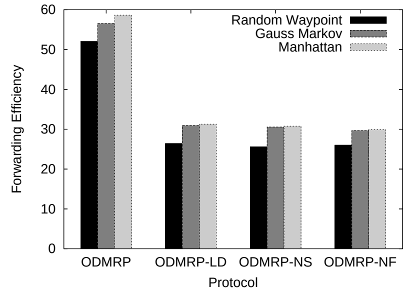
(f) Norm. overhead, 5 sources, 5 receivers



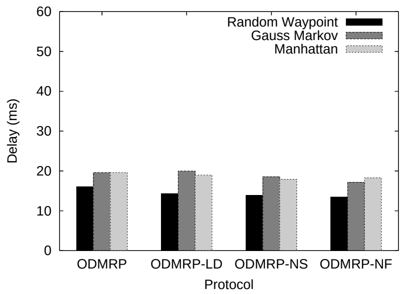
(g) FEF, 1 source, 15 receivers



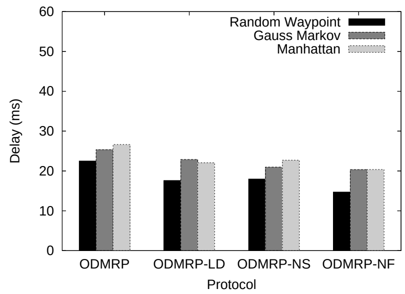
(h) FEF, 2 sources, 15 receivers



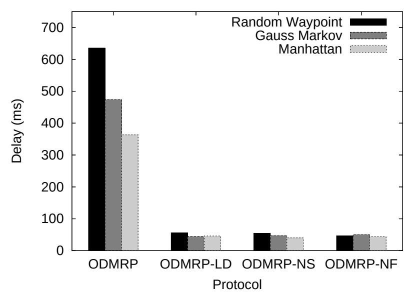
(i) FEF, 5 sources, 5 receivers



(j) Delay, 1 source, 15 receivers



(k) Delay, 2 sources, 15 receivers



(l) Delay, 5 sources, 5 receivers

Fig. 6. Protocol evaluation across mobility models

terms of forwarding efficiency and data overhead. This lower cost allows the ad hoc network to support a higher overall traffic load. In addition, the use of the link duration information allows the mobility-aware mesh construction approach to outperform the other adaptive approach across different mobility models.

VIII. Conclusions

We have introduced a mobility-aware heuristic algorithm to reduce the data overhead of mesh-based multicast ad hoc routing protocols. The algorithm adapts the number of redundant paths of the multicast mesh to the mobility of the network. It starts with an approximation of the minimal data overhead multicast mesh, and adds or reduce the number of forwarding nodes as required. We have demonstrated the NP-completeness of finding such a minimal multicast mesh, which fully justifies the heuristic nature of the proposed scheme.

We have simulated the proposed scheme using ODMRP as the baseline protocol. The performance evaluation shows that the mobility-aware mesh construction algorithm achieves a similar packet delivery ratios than the original ODMRP protocol with a reduction between the 25 to 50% in the number of forwarding nodes and an enhancement in the average latency. The results in scenarios with a high traffic load show that the mobility-aware mesh construction is able to achieve a higher overall network capacity. So, the proposed scheme allows mesh-based multicast routing protocols to benefit from having a low data overhead like multicast trees, while still providing an excellent reliability in face of mobility.

In addition, we have shown that the link duration as a mobility metric offers a valuable information about the mobility of the network. Our simulation results show that the link duration estimator is unbiased by the use of different mobility models, making it suitable for mobility-aware adaptive approaches.

Acknowledgment

Part of this work has been funded by Spanish MCYT by means of the "Ramon y Cajal" workprogramme, the ICSI Call for Spanish Technologists and the SAM (MCYT, TIC2002-04531-C04-03) project. The authors want to thank L. Subramanian and Scott Shenker for their comments and insight.

References

[1] S. Deering, "Multicast Routing in a Datagram Internetwork," *Ph.D. Thesis, Electrical Engineering Dept., Stanford University*, Dec. 1991.

[2] S.-E. Deering and D.-R. Cheriton, "Multicast Routing in datagram internetworks and extended LANs," *Transactions on Computer Systems*, vol.8, no.2, May 1990, pp. 85–110.

[3] J. Moy, "Multicast routing extensions for OSPF," *Computer communications of the ACM*, vol.37, no.8, August 1994, pp.61–66.

[4] T. Ballardie, P. Francis and J. Crowcroft, "Core Based Trees (CBT) – An architecture for scalable inter-domain multicast routing, " *Proc. of ACM SIGCOMM'93*, San Francisco, CA, October 1993, pp.85–95.

[5] S. Deering, D.-L. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu and L. Wei, "The PIM architecture for wide-area multicast routing," *IEEE/ACM Transactions on Networking*, vol.4, no.2, April 1996, pp. 153–162.

[6] C. Cordeiro, H. Gossain and D. Agrawal, "Multicast over Wireless Mobile Ad Hoc Networks: Present and Future Directions" *IEEE Network*, no. 1, Jan 2003, pp. 52–59.

[7] C.-W. Wu, Y.-C. Tay and C.-K. Toh, "Ad Hoc Multicast Routing Protocol Utilizing Increasing id-numberS (AMRIS) Functional Specification," Internet-draft, work in progress, draft-ietf-manet-amris-spec-00.txt, November 1998.

[8] E.-M. Royer and C.-E. Perkins, "Multicast operation of the ad-hoc on-demand distance vector routing protocol," *Proceedings of ACM/IEEE MOBICOM'99*, Seattle, WA, August 1999, pp. 207–218.

[9] L. Ji and S. Corson, "A Lightweight Adaptive Multicast Algorithm," *Proceedings of IEEE GLOBECOM'98*, Sydney, Australia, November 1998, pp. 1036–1042.

[10] J.-G. Jetcheva, D.-B. Johnson, "Adaptive Demand-Driven Multicast Routing in Multi-Hop Wireless Ad Hoc Networks," *Proceedings of ACM MobiHoc'01*, Long Beach, CA, October, 2001, pp. 33–44.

[11] S.-J. Lee, W. Su, and M. Gerla, On-demand multicast routing protocol in multihop wireless mobile networks, *ACM/Kluwer Mobile Networks and Applications*, vol. 7, no. 6, pp. 441–452, December 2002.

[12] J.J. Garcia-Luna-Aceves and E.-L. Madruga, "The Core-Assisted Mesh Protocol," *IEEE Journal on Selected Areas in Communications*, vol.17, no.8, August 1999, pp. 1380–1394.

[13] C.-K. Toh, G. Guichala, S. Bunchua, "ABAM: On-Demand Associativity-Based Multicast Routing for Mobile Ad hoc Networks," *Proceedings of IEEE VTC-2000*, Boston, MA,September 2000, pp.987–993.

[14] E. Bommaiah, M. Liu, A. MacAuley and R. Talpade, "AMRoute: Ad hoc Multicast Routing Protocol," Internet-draft, work in progress, draft-talpade-manet-amroute-00.txt, August 1998.

[15] P. Sinha, R. Sivakumar and V. Bharghavan, "MCEDAR: Multicast Core-Extraction Distributed Ad hoc Routing," *Proc. of IEEE Wireless Commun. and Networking Conf.*, New Orleans, LA, September 1999, pp.1313–1319.

[16] L. Ji, and M.-S. Corson, "Differential Destination Multicast: A MANET Multicast Routing Protocol of Small Groups," *Proc. of IEEE INFOCOM'01*, Anchorage, Alaska, April 2001, pp. 1192-1202.

[17] J.-G. Jetcheva, Yih-Chun Hu, David A. Maltz and David B. Johnson, "A Simple Protocol for Multicast and Broadcast in Mobile Ad Hoc Networks," Internet-draft, work in progress, draft-ietf-manet-simple-mbcast-01.txt, July 2001.

[18] K..Bharath-Kumar and J.-M. Jaffe, "Routing to Multiple Destinations in Computer Networks," *IEEE Transactions on Communications*, no.31 Vol.3, 1983, pp. 343–351.

[19] R.-M. Karp, "Reducibility among combinatorial problems," *In Complexity of computer computations*, Plenum Press, New York, 1975, pp.85–103.

[20] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for Steiner trees," *Acta Informatica*, no. 15, vol. 2, 1981, pp. 141–145.

[21] J. Plesnik, "The complexity of designing a network with minimum diameter," *Networks*, no. 11, 1981, pp. 77–85.

[22] A. Zelikovsky, "An 11/6-approximation algorithm for the network Steiner problem," *Algorithmica*, no. 9, 1993, pp.463–470.

[23] S. Even, "Graph Algorithms," *Computer Science Press*, 1979, pp. 204–209.

[24] J. Boleng, W. Navidi and T. Camp, "Metrics to Enable Adaptive Protocols for Mobile Ad Hoc Networks", *Proc. International Conference on Wireless Networks (ICWN)*, Las Vegas, NV, June 2002, pp. 293–298.

[25] K. Fall, K. Varadhan, "ns, Notes and Documentation", The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, November 2003.

[26] The Monarch Project, mobile networking architectures, Rice University. [On-line]. Available: http://www.monarch.cs.rice.edu/

[27] X. Jia, "A Distributed Algorithm of Delay-Bounded Multicast Routing for Multimedia Applications in Wide Area Networks," *IEEE/ACM Transactions on Networking*, vol.6, no.6, December 1998, pp. 828–837.

[28] B.-M. Waxman, "Routing of Multipoint Connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, December 1998, pp. 1617–1622.

[29] D. Chen, D.-Z. Du, X.-D. Hu, G.-H. Lin, L. Wang, G. Xue, "Approximations for Steiner trees with minimum number of Stiner points," *Theoretical Computer Science* no. 262, 2001, pp. 83–99.

[30] H. Lim and C. Kim, "Multicast Tree Construction and Flooding in Wireless Ad Hoc Networks," *Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, Boston, MA, USA. August, 2000, pp. 61–68.

[31] S. Lee, C. Kim, "Neighbor Supporting Ad hoc Multicast Routing Protocol," *Proceedings of the 1st ACM MobiHoc'*, Boston, MA, USA, 2000, pp. 37–44.