Mobility-Aware Mesh Construction Algorithm for Low Data-Overhead Multicast Ad Hoc Routing

Pedro M. Ruiz and Antonio F. Gomez-Skarmeta

Abstract: **We study the problem of controlling data overhead of mesh-based multicast ad hoc routing protocols by adaptively adding redundancy to the minimal data overhead multicast mesh as required by the network conditions. We show that the computation of the minimal data overhead multicast mesh is NP-complete, and we propose an heuristic approximation algorithm inspired on epidemic algorithms. In addition, we propose a mobility-aware and adaptive mesh construction algorithm based on a probabilistic path selection being able to adapt the reliability of the multicast mesh to the mobility of the network. Our simulation results show that the proposed approach, when implemented into ODMRP, is able to offer similar performance results and a lower average latency while reducing data overhead between 25 to 50% compared to the original ODMRP.**

Index Terms: **Ad hoc multicast routing, mesh-based multicast, minimal data overhead, mobility-aware mesh construction.**

I. INTRODUCTION

A mobile ad hoc network consists of a set of mobile nodes which are free to move and are interconnected through wireless interfaces. Nodes which are not able to communicate directly, use multihop paths using other intermediate nodes in the network as relays. So, a mobile ad hoc node can act both as a mobile router and a mobile host. In addition, the continuous topology changes, the strong requirements in minimizing battery consumption as well as the limited network resources, make the problem of routing in these networks a real challenge. However, their completely distributed nature and their ability to operate without depending upon the deployment of any infrastructure, makes them an ideal component of future mobile computing scenarios. These scenarios include among others emergency situations, battlefield assistance and search and rescue operations. Hence, the interest in mobile ad hoc networks is expected to increase in the future.

Multicast is one of the areas in mobile ad hoc networks which is to play a key role in future wireless networks. Key to this is the fact that most of the application scenarios for mobile ad hoc networks are strongly based on many-to-many interactions and they require a high degree of collaboration among terminals. Many services such as multimedia applications, service discovery, and many other bandwidth-avid applications can strongly benefit from the underlying support of efficient multicast communications.

The problem of the efficient distribution of traffic from a set of senders to a group of receivers in a datagram network was initially studied by Deering [1]. Several multicast routing protocols like DVMRP [2], MOSPF [3], CBT [4], and PIM [5] have been proposed for IP multicast routing in fixed networks. However, these protocols are not able to perform well in highly mobile and topology changing scenarios such as ad hoc networks. The main reason is that their forwarding structures are fragile. Thus, the cost in terms of the control overhead which they would require to recompute their multicast trees whenever one of the links in the tree breaks, makes unreasonable their deployment in an mobile ad hoc network.

Several multicast routing solutions specifically designed for ad hoc networks have been proposed in the literature [6]. In general, these protocols can be classified into two groups: Tree-based and mesh-based approaches. Tree-based schemes construct a multicast tree from each of the sources to all the receivers. Examples of protocols following this approach are AMRIS [7], MAODV [8], LAM [9], and ADMR [10]. 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 when there is a high mobility in the network. Mesh-based approaches like ODMRP [11] and CAMP [12], by using additional links in their underlying forwarding structure, manage to deal with mobility very efficiently. The main drawback of using a mesh is that, due to the additional paths which are created, duplicated data packets can make an excessive consumption of network resources even if a duplicate detection is used.

The usual metrics used in the literature to assess the effectiveness of a multicast ad hoc routing protocol are usually related to the control packet overhead. In the author's opinion previous studies have neglected the huge impact that the data overhead may have in the overall performance of a protocol. In fact, if we take into account that data traffic consumes more bandwidth than control traffic, the overhead due to the selection of sub-optimal routes may easily become more expensive in terms of bandwidth and energy consumption than the cost of sending a few more control packets.

We analyze the problem of creating multicast forwarding structures minimizing data overhead. We show that forwarding structures used by existing ad hoc multicast routing protocols such as shortest path trees (SPT), multicast meshes or minimal steiner trees (MST) do not offer minimal data overhead. In particular, we focus on mesh-based multicast routing protocols because of their suitability for mobile scenarios. We demonstrate that the problem of finding a multicast mesh which yields the minimal data overhead is NP-complete, and we propose a heuristic to approximate minimal data overhead forwarding meshes. The proposed heuristic is based on the epidemic propagation of the number of forwarding nodes during the process of creating multicast paths. In this way, ad hoc nodes are pro-

Manuscript received July 30, 2004.
The authors are with the Depto. de Ingeniería de la Información y las Comunicaciones, Universidad de Murcia, Spain, email: {pedrom, skarmeta}@dif.um.es.

1229-2370/04/$10.00 © 2004 KICS

vided with the information required to select between a shortest path or a low data-overhead one. In addition, we present an adaptive mobility-aware mesh construction algorithm based on a probabilistic path selection function. The function is defined so that low data-overhead paths are selected with a higher probability when there are enough backup links in the current mobility conditions of the network. When more reliability is needed the function assigns a lower probability to the low data-overhead paths.

Our simulation results show that the proposed approach, when incorporated into the ODMRP mesh-based ad hoc routing protocol, yields around 20 to 50% improvement in the forwarding efficiency. In addition, it achieves a reduction of the mean delivery latency due to the reduced link layer contention while still offering similar packet delivery ratios than the original ODMRP.

The remainder of the paper is organized as follows: Section II discusses the problem of data overhead minimization and shows how previous multicast tree construction approaches do not minimize it. Section III 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 IV. In Section V, we show the need for a mobility-aware mesh construction and propose a modified heuristic. Section VI evaluates and analyzes the simulation results of those heuristics. A description of related work is given in Section VII. Finally, Section VIII provides some discussion and conclusions.

II. DATA OVERHEAD IN AD HOC MULTICAST ROUTING

The goal of multicast routing protocols in ad hoc networks is finding a set of relay nodes so that data packets sent out by multicast sources can be delivered to multicast receivers. Obviously, the number of those relay nodes is lower than the number of nodes except in the case of flooding. The paths defined by the union of all these nodes 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. We define forwarding nodes as those nodes which are in the path between any source and any receiver. Note that even a source or a receiver can also be a forwarding node. Fig. 1 shows a multicast tree in which we identify forwarding nodes by a double circle. Wider lines represent the forwarding structure induced by the forwarding nodes.

There are two basic approaches to build 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 transmissions required or the number of forwarding nodes which are required. So, they do not provide an optimal bandwidth consumption.

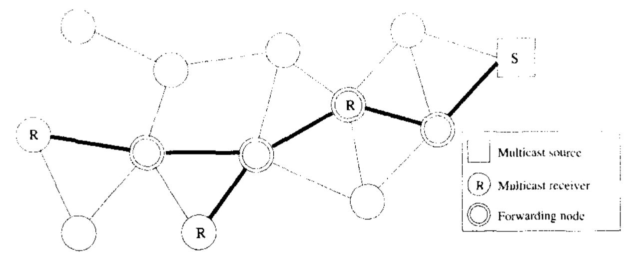


Fig. 1. Example of tree-based multicast forwarding structure.

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 have been proposed in protocols like ODMRP [11] and CAMP [12]. These structures are particularly interesting to deal with mobility of the nodes, but obviously they do not minimize the cost of multicast forwarding.

The problem of finding a minimum cost multicast tree is well-known as the minimum Steiner tree problem. Karp [19] demonstrated that this problem is NP-complete even when every link has the same cost, by a transformation from the exact cover by 3-sets. 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.

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. 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, needs k transmissions to send a multicast data packet to k of its neighbors. However, in a broadcast medium, the transmission of a multicast data packet from a given node v to any number of its 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.

Fig. 2 shows different multicast trees connecting the source (S) to the set of receivers (R). As it is depicted in Fig. 2(a), the union of the shortest path trees result in 3 forwarding nodes. Thus, number of transmissions is 4, one performed by the multicast source and one for each of the forwarding nodes. Similarly, the number of transmissions for the Steiner tree (see Fig. 2(b)) is also 4. We can see from the figure, how the Steiner tree tries to

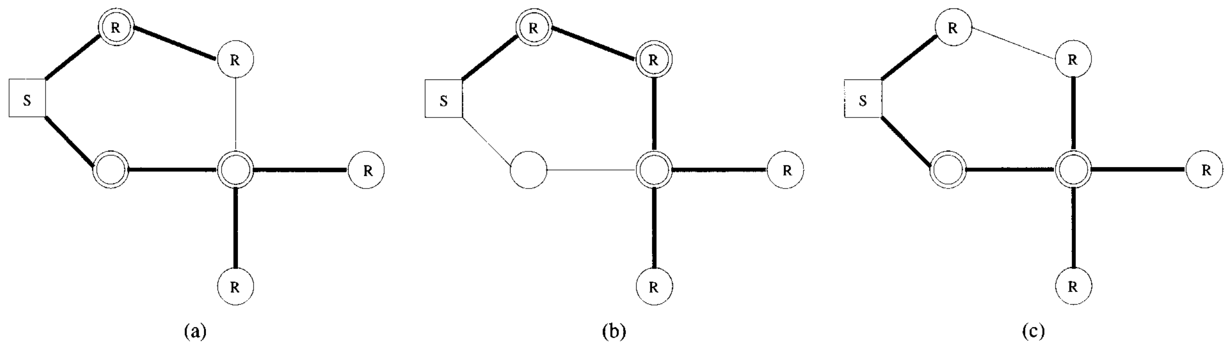


Fig. 2. Differences in cost for several multicast trees over the same ad hoc network: (a) Shortest path tree, $C(T) = 4$. (b) Steiner tree, $C(T) = 4$, (c) min. data overhead tree, $C(T) = 3$.

minimize the number of non-terminal nodes (i.e., nodes which are not either senders or receivers) which take part in the tree. These nodes are also known as Steiner nodes. Finally, we see in Fig. 2(c) that the minimal data overhead tree, which takes into account the broadcast medium of the network, is able to cover all the receivers with only 3 transmissions. This shows that the Steiner tree problem fails to provide the minimal cost multicast tree in these networks.

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.

III. NETWORK MODEL AND PROBLEM FORMULATION

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 $C : T \rightarrow Z^+$. 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.

Definition 3: Given a multicast tree T, we define the data overhead of T as $\omega_d = C(T) - C(T^*)$. It is immediate from this definition that $\omega_d(T^*) = 0$. In addition, it is also easy to show that the minimal multicast tree in terms of number of transmissions is also the minimal data overhead multicast 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 $C' : M \rightarrow Z^+$. 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 1: 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: Let's 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 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

$$1 + |F| \geq C(T^*_{\max}). \quad (1)$$

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 $1 + |F| \geq C(T^*_{\max}) \geq C(T_i^*), i = 1, \cdots, |S|$. □

Using 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 $T^* = \{T_1^*, T_2^*, \cdots, T_i^*, \cdots, T_{|S|}^*\}$ 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

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

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 proved by Theorem 1).

In addition, combining Definition 4 and Theorem 1 we can also introduce the following corollary.

Corollary 1: 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. However, it is of paramount importance being able to control their data overhead. Thus, it makes sense to consider the problem of finding minimal data overhead multicast meshes. 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 forwardings. 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 2: 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. □

IV. 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.

A. Proposed Heuristic

From the definition the data overhead of a multicast mesh given in Definition 5, it is clear that 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 reducing 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

Algorithm 1 Minimal data overhead mesh heuristic

```
BestRREQ ← null
loop
  Receive route request packet RREQ
  if RREQ.seqno > BestRREQ.seqno then
    BestRREQ ← RREQ
    if forwarder node and not receiver then
      NewRREEQ.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
    Schedule sending of NewRREQ
  else if RREQ.seqno = BestRREQ.seqno and
  RREQ.FNCount < BestRREQ.FNCount then
    if NewRREQ not sent out yet 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
```

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.

As Algorithm 1 shows, the way in which an intermediate node updates the FNCount field is very simple and intuitive:

- If the node is not a forwarding node, then increment the counter.
- If the node is a forwarding node, do not increment the counter.
- If the node is a receiver and a multicast forwarder, set the counter to zero. If it is only a receiver then set the counter to one.

The latter case in which the node is a receiver has been defined in that way to reflect that the receiver will have to join the source through its lower FNCount route anyway. So, every intermediate node in that route will become a multicast forwarder. Thus, the multicast receiver acts as if he would have received a best route with an FNCount of zero.

The use of the proposed heuristic produces multicast meshes in which there is a very low redundancy. Although that is excellent from the point of view of the data overhead reduction, the performance in mobile environments may suffer a noticeable degradation because the minimal mesh is less resilient to topological changes. However, our interest in the minimal data overhead multicast mesh stems from the possibility of being able to control the reliability of the underlying mesh, by simply adding or reducing links to that minimal mesh depending on the mobility of the network.

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.

V. MOBILITY-AWARE MESH CONSTRUCTION ALGORITHM

Improving the forwarding efficiency of a multicast ad hoc routing protocol without an impairment in the protocol's performance requires a trade-off between reliability and data overhead. 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. However, in scenarios with a lower mobility, having a big number of forwarding nodes is usually 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. Network Metrics under Consideration

An adaptive routing scheme requires some feedback about the network conditions to properly adapt its behavior. 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 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

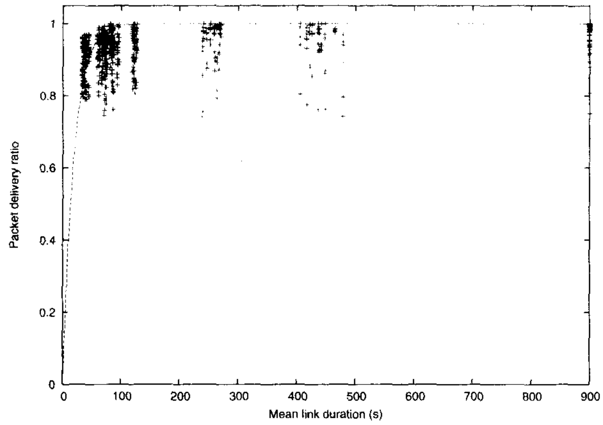


Fig. 3. Packet delivery ratio vs. mean link duration.

we have selected the mean link duration as the mobility metric to use.

One of the key parameters affecting the computation of 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 the feedback 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, 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 assess 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 consider 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 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.

The key difference with the unicast case (and the results of Boleng *et al.*) are due to the additional reliability which a multicast mesh provides. 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. It is very easy to compute, and we have found that a much better coefficient of determination can be obtained when considering independently 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 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. Concretely, we estimate probability value which influences the probabilistic selection of the paths. We explain the details about the probabilistic path selection in the next subsection.

B. Adaptive Mesh Construction

The design of a deterministic mesh construction algorithm being able to balance the data overhead according to the network conditions, is very difficult. It requires the consideration of scenario-specific or overall network metrics which are usually difficult or expensive to obtain. In these cases, it is often more interesting the use of probabilistic schemes which can work with lossy or partial information and still provide good results in terms of performance.

This is our case 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 (RR), instead of selecting every time the route with the lower FNCount, it will select 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.

B.1 Finding a Function for $\pi_1(|S|)$

If we consider the minimal data overhead multicast mesh, we are interested in analyzing the rate at which the number of new forwarding nodes increases as the number of sources is augmented. Let $S = \{s_1, s_2, \cdots, s_n\}$ 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. Then, 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 increases, 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

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

B.2 Finding a Function for $\pi_2(ld)$

From our simulation 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 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 found the function in (4) to be appropriate. In fact, this function ,which is plotted in Fig. 4, meets all our requirements. As the number of sources increases, $\pi(ld, |S|)$ decreases. As the link duration decreases $\pi(ld, |S|)$ increases. And finally, when the number of sources provides enough reliability the contribution from the mobility metric is lower than when there are a few sources.

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

The integration of this approach into ODMRP is very straightforward, and the increase in control overhead is negligible. The only extensions are: (i) Adding a new field to JOIN_QUERY messages in which the *FNCount* field is being updated, (ii) changing the route selection algorithm to use $\pi(ld, |S|)$ according to (4), and (iii) make a node wait J_QUERY_AGG time to eventually receive a better data-overhead route before propagating JOIN_QUERY messages.

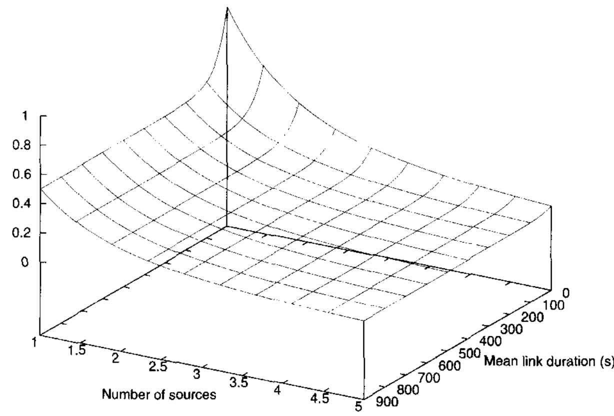


Fig. 4. $\pi(ld, |S|)$ values for in the probabilistic path selection.

Table 1. ODMRP parameters for simulation.

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

Table 2. Simulation parameters.

MAC layer	IEEE 802.11b DCF at 2 Mb/s
Comm. range	250 m
Sim. time	900 s
Traffic	1, 2, and 5 CBR sources, 330 bytes/pkt, 5 pkt/s
Receivers	5, 15, and 30

VI. 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.

Table 1 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.

A. Simulation Methodology and Performance Metrics

The simulated scenario consists of 100 mobile nodes randomly distributed over an area of 1200×800 m^2 , moving according to a random waypoint model at a speed uniformly distributed between 0 and 20 m/s. The rest of the simulation parameters used, are described in Table 2.

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

- Packet delivery ratio. Defined as the number of data packet successfully delivered over the number of data packets generated by the sources.
- Normalized packet overhead. Defined as the total number of control and data packets sent and forwarded normalized by the total number of packets successfully delivered.

- Forwarding efficiency. 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. 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.

B. *Performance Evaluation*

The data labeled as 'ODMRP' 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'.

The packet delivery ratio as a function of the pause time is depicted in Fig. 5(a) for the four variants in the 1 source and 15 receivers scenario. As expected, the minimal data overhead mesh heuristic yields around a 10% lower packet delivery ratio compared to ODMRP. That is due to the lack of redundant links. However, both ODMRP and the adaptive variants are able to deliver around 99% of the packets. ODMRP delivers around a 0.5% more packets than the mobility-aware approach. However, to achieve this packet delivery ratio, the original ODMRP version requires around a 20% more forwarding nodes (see Fig. 5(c)). In addition, the proposed alternative has a lower overhead. As we can see, the minimal data overhead mesh heuristic uses a 60% less overhead than ODMRP. This is because by reducing the number of forwarding nodes, the data overhead is also reduced. An additional benefit of reducing the data overhead is the reduction of the mean latency between the source and the receivers. The expected behavior is that ODMRP would have offered a lower latency because it uses the shortest path tree. However, the proposed approach by reducing the number of forwarding nodes also reduces the MAC-layer contention and collisions among forwarding nodes. Thus, the overall average latency can be reduced.

It is also interesting to point out, that the mobility-aware heuristic by considering the mean link durations is able to offer higher packet delivery ratios than the same heuristic considering only the number of sources. This is particularly noticeable for the scenarios with lower pause times.

The cause for the important difference in the number of forwarding nodes between the different variants is that in the original ODMRP the randomness in the access to the MAC layer can make the shortest path routes to change very quickly even if the topology is relatively static. 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. The proposed schemes are based in the lower *FNCount* metric, which does not change as fast as the shortest path. Thus, the number of nodes can be reduced. In fact, in the case of the mobility-aware algorithm, the data overhead can be controlled by the selection of appropriate π_s for the probabilistic path selection. That is why 'ODMRP-LD' has a bigger data overhead when the network is less static (lower pause times) and the number of forwarding nodes reduces as the pause times increase.

The performance results are very similar in the scenarios for 5 and 30 receivers. In general, the higher the number of receivers the lower the differences in the packet delivery ratio and also the lower the differences in data overhead. This is because as the number of receivers increases, so does the number of forwarding nodes which are really needed.

The evaluation of the scenarios with 2 sources and 15 receivers in Fig. 6 shows a similar trend. The packet delivery ratio of the mobility-aware approach is still around 0.5% lower than ODMRP. Moreover, the proposed approach becomes even more efficient compared to the original ODMRP. In Fig. 6(c) it is shown that the proposed approach manages to improve the forwarding efficiency by around a 50%. Given that in these scenarios the traffic load is doubled, both approaches experience a slight increase in the average latency. This is due to the higher contention at the MAC-layer. However, as Fig. 6(d) depicts, the proposed approach improves even more its average latency compared to the one of the original ODMRP. This is explained by the 50% less forwarding nodes required by the proposed approach. Similar results were obtained for the case of 5 and 30 receivers. For the case of the minimal data overhead heuristic scheme, the differences in the packet delivery ratio compared to ODMRP are reduced to a 2.5% for the scenarios with a higher mobility and around a 1% for more static scenarios. This corroborates our studies in the previous section about the additional reliability automatically added to the multicast mesh by increasing the number of sources. In addition, by comparing 'ODMRP-LD' and 'ODMRP-NS' we can see that in this case the mobility-aware approach stills performs better under high mobility, but the differences in overhead are much lower. This is because the definition of $\pi(ld,|S|)$ takes advantage of the additional reliability provided by having two sources.

In the scenarios with 5 multicast sources, we can see that the traffic load is so high, that the ad hoc network starts getting congested. This can be shown by the reduction in the packet delivery ratio shown in Fig. 7(a) and the increments in latency shown in Fig. 7(d). The congestion for ODMRP is so high for the case of 15 and 30 receivers that we focus here in the case of 5 receivers which is more representative. In fact, we can notice that the proposed approach improves the packet delivery ratio by 3.5% using 50% less of data transmissions (see Fig. 7).

VII. 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]– [10], [13] construct a multicast tree from each of the sources to all the receivers using any of the tree computations schemes which we discussed in Section II. 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

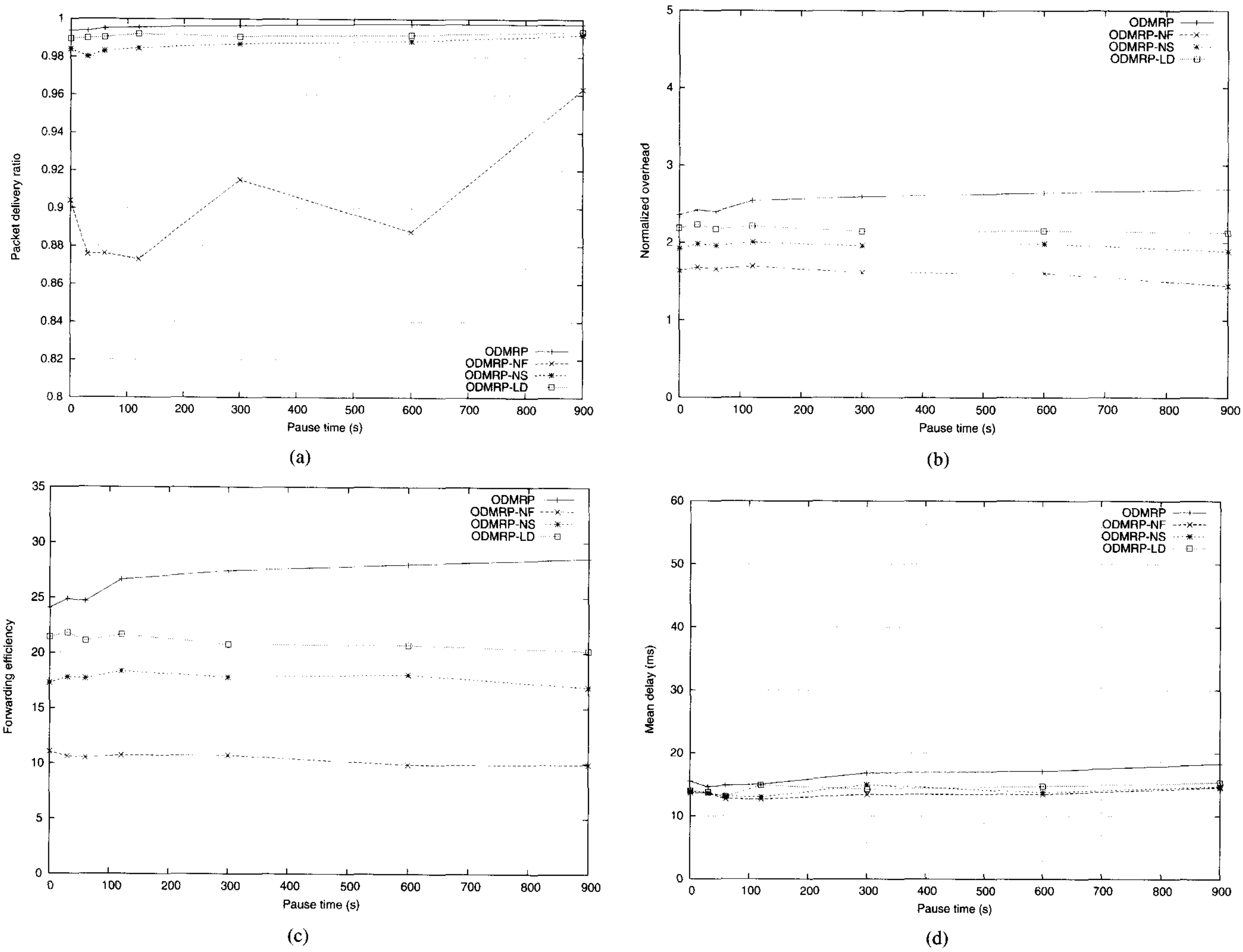


Fig. 5. 1 source and 15 receivers: (a) Packet delivery ratio, (b) normalized packet overhead, (c) forwarding efficiency, (d) latency.

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 costs 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.

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 were lower than the results we have obtained, and their fixed path selection probabil-

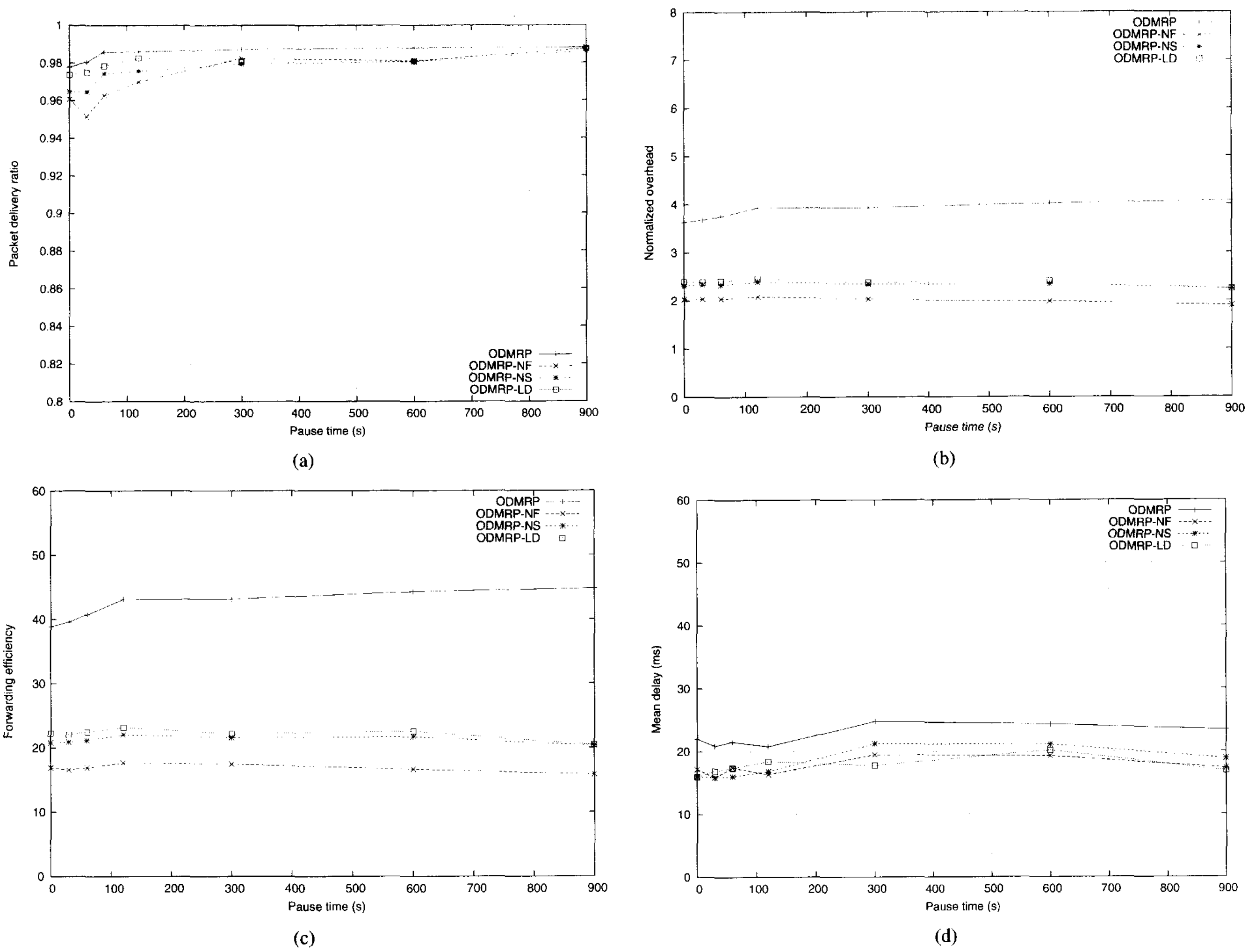


Fig. 6. 2 sources and 15 receivers: (a) Packet delivery ratio, (b) normalized packet overhead, (c) forwarding efficiency, (d) latency.

ity makes their proposal unable to perform well under different mobility rates.

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 increments or reduces 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 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 approach is able to achieve a higher overall network capacity. So, the proposed scheme allows mesh-based multicast as hoc routing protocols to benefit from having a low data overhead like the one of multicast trees, while still providing an excellent reliability in face of mobility.

ACKNOWLEDGMENTS

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," *Trans. Computer Sys.*, vol. 8, no. 2, pp. 85–110, May 1990.

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

[4] T. Ballardie, P. Francis, and J. Crowcroft, "Core based trees (CBT) - an architecture for scalable inter-domain multicast routing," in *Proc. ACM SIGCOMM'93*, San Francisco, CA, Oct. 1993, pp. 85–95.

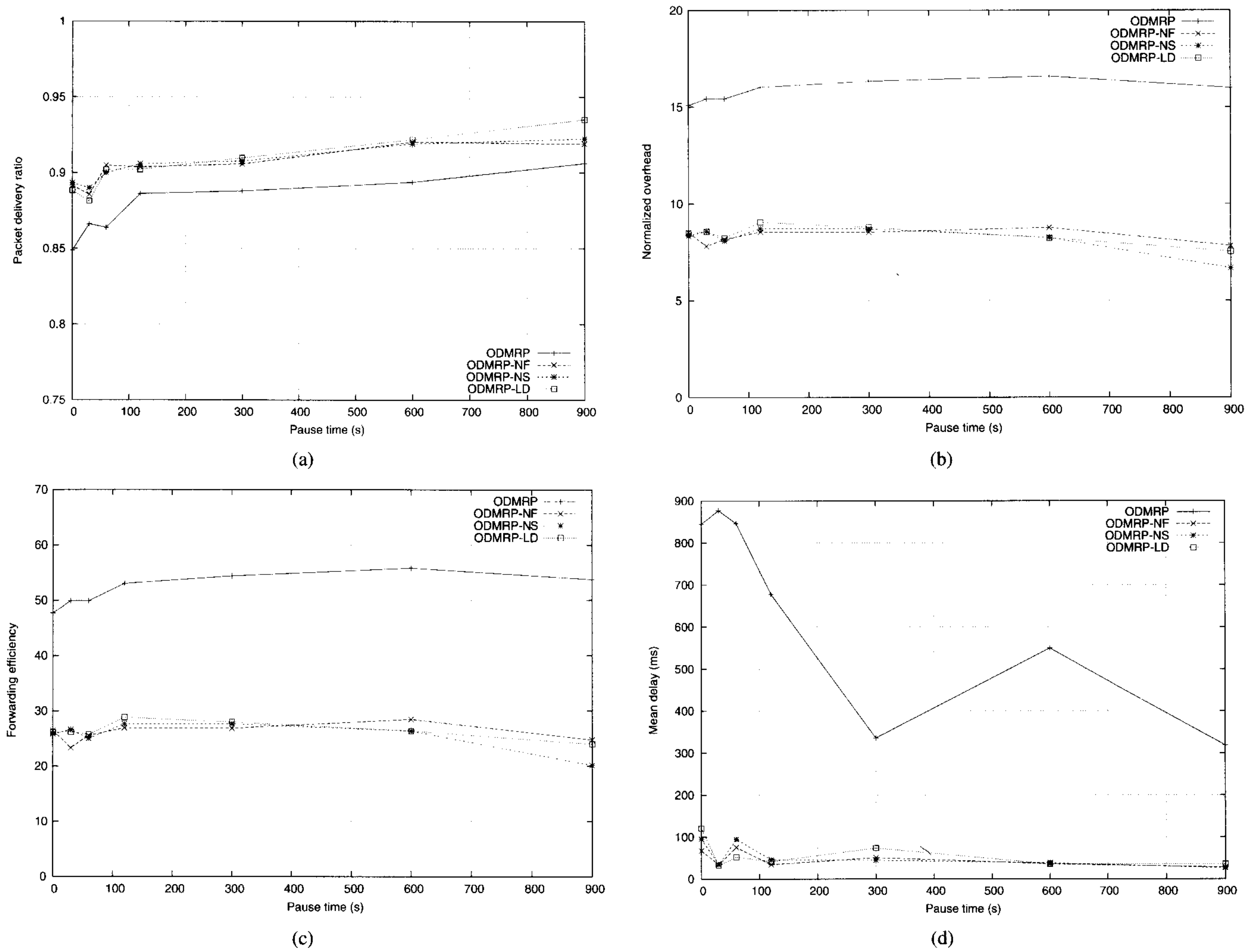


Fig. 7. 5 sources and 5 receivers: (a) Packet delivery ratio, (b) normalized packet overhead, (c) forwarding efficiency, (d) latency.

[5] S. Deering *et al.*, "The PIM architecture for wide-area multicast routing," *IEEE/ACM Trans. Networking,* vol. 4, no. 2, pp. 153–162, Apr. 1996.

[6] C. Cordeiro, H. Gossain, and D. Agrawal, "Multicast over wireless mobile ad hoc networks: Present and future directions," *IEEE Network*, no. 1, pp. 52–59, Jan. 2003.

[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, Nov. 1998.

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

[9] L. Ji and S. Corson, "A lightweight adaptive multicast algorithm," in *Proc. IEEE GLOBECOM'98*, Sydney, Australia, Nov. 1998, pp. 1036–1042.

[10] J.-G. Jetcheva and D.-B. Johnson, "Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks," in *Proc. ACM MobiHoc 2001*, Long Beach, CA, Oct. 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, Dec. 2002.

[12] J. J. Garcia-Luna-Aceves and E.-L. Madruga, "The core-assisted mesh protocol," *IEEE J. Select. Areas Commun.*, vol. 17, no. 8, pp. 1380–1394, Aug. 1999.

[13] C.-K. Toh, G. Guichala, and S. Bunchua, "ABAM: On-demand associativity-based multicast routing for mobile ad hoc networks," in *Proc. IEEE VTC 2000*, Boston, MA, Sept. 2000, pp. 987–993.

[14] E. Bommaiah *et al.*, "AMRoute: Ad hoc multicast routing protocol," *Internet Draft*, work in progress, draft-talpade-manet-amroute-00.txt, Aug. 1998.

[15] P. Sinha, R. Sivakumar, and V. Bharghavan, "MCEDAR: Multicast core-extraction distributed ad hoc routing," in *Proc. IEEE Wireless Commun. and Networking Conf.*, New Orleans, LA, Sept. 1999, pp. 1313–1319.

[16] L. Ji and M.-S. Corson, "Differential destination multicast: A MANET multicast routing protocol of small groups," in *Proc. IEEE INFOCOM 2001*, Anchorage, Alaska, Apr. 2001, pp. 1192–1202.

[17] J.-G. Jetcheva *et al.*, "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 Trans. Commun.*, vol. 3, no. 31, pp. 343–351, 1983.

[19] R.-M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, New York: Plenum Press, 1975, pp. 85–103.

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

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

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

[23] S. Evan, "Graph algorithm," *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," in *Proc. Int. Conf. Wireless Networks (ICWN 2002)*, Las Vegas, NV, June 2002, pp. 293–298.

[25] K. Fall and K. Varadhan, "Ns notes and documentation," The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, Nov. 2003.

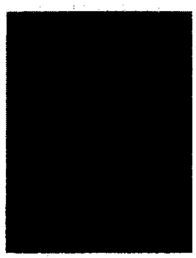
[26] The Monarch Project, "Mobile networking architectures," Rice University, available at http://www.monarch.cs.rice.edu/.

[27] X. Jia, "A distributed algorithm of delay-bounded multicast routing for multimedia applications in wide area networks," *IEEE/ACM Trans. Networking*, vol. 6, no. 6, pp. 828–837, Dec. 1998.

[28] B.-M. Waxman, "Routing of multipoint connections," *IEEE J. Select. Areas Commun.*, vol. 6, no. 9, pp. 1617–1622, Dec. 1998.
[29] D. Chen *et al.*, "Approximations for Steiner trees with minimum number of Stiner points," *Theoretical Computer Science*, no. 262, pp. 83–99, 2001.
[30] H. Lim and C. Kim, "Multicast tree construction and flooding in wireless ad hoc networks," in *Proc. 3rd ACM Int. Workshop Modeling, Analysis, and Simulation of Wireless and Mobile Systems*, Boston, MA, USA. Aug. 2000, pp. 61–68.
[31] S. Lee and C. Kim, "Neighbor supporting ad hoc multicast routing protocol," in *Proc. 1st ACM MobiHoc 2000*, Boston, MA, USA, 2000, pp. 37–44.

Pedro M. Ruiz received his B.Sc. (1999) and M.Sc. (2001) degrees in Computer Science from the University of Murcia, Spain, and a Ph.D. (2002) degree in the same University. He is a Senior Researcher in the Department of Information and Communication Engineering (DIIC) at the University of Murcia (UMU), and he is currently in a post-doctoral fellowship at the ICSI in Berkeley, CA. Prior to joining UMU in december 2003, he has worked as R&D Manager at Agora Systems S.A. He also coordinated several research activieties at RedIRIS, the Spanish National Research Network, and worked as a part-time assistant professor at the Univ. Carlos III of Madrid. During these years, he has acted as *Principal Investigator* in several projects mainly funded by the European Union, Spanish government and private companies, and has published over 50 refereed papers in international journals and conferences.

Dr. Ruiz is in the editorial board for the *International Journal on Parallel, Emergent, and Distributed Systems* starting in 2005, he is participating in the Technical Comitee of serveral conferences, and he has served as a reviewer for major IEEE journals and conferences. In fall 2003 he was awarded a *"Ramon y Cajal"* grant by the Spanish Ministry of Science and Technology. His main research interests include mobile and ad hoc wireless networks, adaptive multimedia systems, and distributed applications and services. He is a member of the IEEE Communications Society.



Antonio F. Gomez-Skarmeta received his M.S. in Computer Science from the University of Granada and his Ph.D. in Computer Science from the University of Murcia. He is an associate professor in the Department of Information and Communication Engineering (DIIC) at the University of Murcia. His research interests are fuzzy modelling and hybrid systems, softcomputing techniques, intelligent systems, intelligentagents, distributed services, network protocols, and security.