Computer Networks 36 (2001) 659–670
COMPUTER NETWORKS
www.elsevier.com/locate/comnet

Location aware, dependable multicast for mobile ad hoc networks

Stefano Basagni *, Imrich Chlamtac, Violet R. Syrotiuk

Center for Advanced Telecommunications Systems and Services (CATSS), Erik Jonsson School of Engineering and Computer Science, The University of Texas at Dallas, Dallas, TX 75080, USA

Received 6 December 2000; accepted 22 February 2001

Responsible Editor: I.A. Akyildiz

Abstract

This paper introduces *dynamic source multicast* (DSM), a new protocol for multi-hop wireless (i.e., *ad hoc*) networks for the *multicast* of a data packet from a source node to a group of mobile nodes in the network. The protocol assumes that, through the use of positioning system devices, each node knows its own geographic location and the current (global) time, and it is able to efficiently spread these measures to all other nodes. When a packet is to be multicast, the source node first locally computes a snapshot of the complete network topology from the collected node measures. A *Steiner* (i.e., multicast) tree for the addressed multicast group is then computed *locally* based on the snapshot, rather than maintained in a distributed manner. The resulting Steiner tree is then optimally encoded by using its unique *Prüfer sequence* and is included in the packet header as in, and extending the length of the header by no more than, the header of packets in source routing (unicast) techniques. We show that all the local computations are executed in polynomial time. More specifically, the time complexity of the local operation of finding a Steiner tree, and the encoding/decoding procedures of the related Prüfer sequence, is proven to be $O(n^2)$, where n is the number of nodes in the network. The protocol has been simulated in ad hoc networks with 30 and 60 nodes and with different multicast group sizes. We show that DSM delivers packets to all the nodes in a destination group in more than 90% of the cases. Furthermore, compared to flooding, DSM achieves improvements of up to 50% on multicast completion delay. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Mobile ad hoc networks; Multicast routing; Global positioning system

1. Introduction

Multicast in general refers to forms of communication with multiple participants. In this paper, we consider the special case of point to multipoint communication, or *source multicast*, wherein the same packet is sent from one node (the *source*) to a specified subset of nodes of the network (the *multicast group*). This paper proposes a new multicast protocol for multi-hop wireless radio networks, often called *ad hoc* networks, which are networks in which all the nodes are mobile. Such networks have no fixed infrastructure (i.e., no fixed base stations) and thus nodes rely on each other to

* Corresponding author. Tel.: +1-9728832216; fax: +1-9728836204.

E-mail addresses: basagni@utdallas.edu (S. Basagni), chlamtac@utdallas.edu (I. Chlamtac), syrotiuk@utdallas.edu (V.R. Syrotiuk).

1389-1286/01/$ - see front matter © 2001 Elsevier Science B.V. All rights reserved.
PII: S 1 3 8 9 - 1 2 8 6 (0 1) 0 0 1 8 2 - 7

forward packets (hence multi-hop) to destinations not directly in their transmission range. While ad hoc networks are still primarily used in military applications (tactical networks), they are starting to be used in commercial environments when a wireline network is not operational or unavailable (emergency response, search and rescue, wildlife management, etc.) since they are rapidly deployable and can extend the range of the fixed network infrastructure, both wireline and cellular.

As more and more real-time multimedia applications emerge on wireless networks the need to support multicast has grown in importance. Examples of such applications include video conferencing, interactive television, and the support of virtual private networks, all of which are characterized by groups with common interests. In addition, other applications such as resource discovery and distributed/replicated database management make use of multicast to query sites and update "mirror" sites, respectively. The specific challenges for multicast in an ad hoc network are the rapidly changing environment, as well as the limited transmission bandwidth.

Many existing protocols for multicast are based on *multicast routing trees*, both in wireline and in ad hoc networks (see, e.g., [1,7,9]). However, these solutions, which include the recent "multicast mesh" solution proposed in [11] and the multicast protocols presented in [6] and [16] rely on the existence and correct operation of an underlying ad hoc routing protocol. Other solutions, instead, depend on the existence of a hierarchical organization in the ad hoc network (e.g., clustering) [5]. Thus, whether based on routing or clustering, many existing multicast protocols for ad hoc networks expend great effort to distributively maintain a "multicast routing structure," and this can heavily affect the overall performance of the multicast.

In this paper we propose a new multicast protocol for ad hoc networks which neither assumes any routing scheme as a basis nor builds and maintains any multicast routing structure. The proposed protocol is based on the expectation that nodes of ad hoc networks are aware of their location. Due to the commercial proliferation of positioning system devices, we assume that all nodes in our network are equipped with low cost positioning receivers. For ease of description, and given their widespread availability, in this paper we consider the use, at each node, of global positioning system (GPS) devices, through which each node can compute not only its three-dimensional position (latitude, longitude and altitude), but also its current velocity and the current time (which can be considered "global", in the sense that it is the same for all nodes in the network).

Using a dissemination mechanism specifically tuned to the system requirements of ad hoc networks such as the one introduced and described in [3,4], a node's *GPS measures* and current transmission radius are spread throughout the network. Thus, each node can maintain a cache of tuples (entries that contain the node identifier, its position at a certain time, velocity, transmission radius, etc.) for each other node in the network, and, whenever it is needed, the network topology is easily constructed from these cached tuples.

When a packet for the multicast group M is created at a source node S, S first constructs a graph G representing the current network topology. Each *link* connecting two nodes in the *topology graph* G represents that the two nodes are in transmission range of one another (i.e., a radio link exists between them), and is carefully considered by taking into account node movement. S then *locally* computes a multicast tree (called also a *Steiner tree*, in graph theoretic terms) for G and M rooted at S. The source node S then includes a coded representation of the paths (i.e., the tree) the packet is to follow as a *Prüfer sequence* [15] in the packet header, and broadcasts the packet. Each child node that is the root of a subtree of the source will forward the packet to each of its children until finally the path is exhausted, at which point the multicast is complete. Since the packet travels hop-by-hop following the paths in the tree similar to the Dynamic Source Routing protocol [13] we call our multicast protocol *dynamic source multicast* (DSM).

We show that each step of our multicast protocol can be solved by an efficient local computation. Namely, each of the problems of:

1. computing a graph corresponding to the network topology,

2. the selection of "usable" links within the topology graph,
3. the computation of the multicast tree,
4. the encoding and decoding of the multicast tree

can all be solved in polynomial time. In particular, the time complexity of each of these local tasks is proven to be $O(n^2)$, where n is the number of nodes in the network.

Overall, the DSM protocol achieves the following desirable properties:

- DSM is easy to implement, relying only on a bandwidth and energy efficient dissemination mechanism, rather than on an underlying routing or clustering protocol. As well, since the computations are performed locally, no complex coherent data structure (such as a distributed multicast tree or mesh) needs to be maintained distributively among the nodes.
- Each of the four subproblems solved locally does not impose a significant overhead on the multicast. In particular, this is true about the computation of the multicast tree, since a polynomial time approximation algorithm can be used to compute a Steiner tree in the network topology graph. Thus, the local complexity of DSM is linked to the best (possibly yet to come!) algorithm for computing a Steiner tree in a network graph.
- Due to optimal coding through a tree's unique Prüfer sequence, there is minimal overhead associated with transmitting the encoded tree in the header of a data packet, with respect to ad hoc source routing solutions since a tree can be uniquely encoded by a sequence whose length is at most the length of the longest route between two nodes.
- DSM does not impose restrictions on the number of multicast groups, on the number of nodes in each group and on the number of groups with which each node can be affiliated.
- The possibility for a node to change (join/leave) groups, is easily and efficiently supported in DSM by disseminating group identifiers along with the GPS measures.

The dependability of DSM for successfully completing multicast is demonstrated through the use of simulation. The obtained results show that in an ad hoc network with 30 and 60 nodes all moving at a velocity from 6 to 20 m/s, independently of the size of the multicast groups, all the nodes in the addressed group receive the packet more than 90% of the time (this percentage is actually >97% in the case of networks with 30 nodes). In the remaining 10% of the cases, more than 85% of the nodes in the addressed multicast group always receive the packet, leaving only fewer than 15% of the nodes not receiving the packet.

Furthermore, in both cases (30 and 60 nodes), the delay associated with the completion of the multicast (computed as the difference between the arrival time of the packet at the last node of the destination group that received it and the time it was sent at the source) is up to 50% better than the delay associated with the multicast obtained by simply flooding the packet through the network.

The rest of this paper is organized as follows. Section 2 explains the dissemination mechanism that allows each node to efficiently communicate its GPS measures. Section 3 describes the multicast protocol in detail, emphasizing the efficiency of the implementation of each of the local operations. Section 4 demonstrates the effectiveness of our protocol in delivering a packet to all the nodes of the multicast group. Section 5 concludes the paper.

2. Dissemination of GPS data

Our DSM protocol assumes that each node is aware of its own geographic location which can be obtained by equipping a node with a global positioning system (GPS) receiver. Such a device allows a node to receive GPS broadcasts and compute its three-dimensional position (latitude, longitude, and altitude), velocity and time (these node parameters, together with its transmission radius, will be called a node's GPS *measures* in the sequel) with a precision to within a few hundred meters for position and 340 ns for time. (A nice account of GPS features can be found by visiting the website [10].)

Since ad hoc networks lack a fixed infrastructure, every node in the network is responsible for disseminating its GPS measures to all other nodes. This is obtained by flooding the network with a packet containing the GPS measures of the node.

Upon reception of a "GPS packet" from a node B, a node A updates its *GPS cache* that stores, for each other node, its most updated GPS measures. This dissemination mechanism is especially tailored to meet the requirements of ad hoc networks, where the minimization of bandwidth and energy usage are important goals.

These goals are first addressed by the size of the packets used to disseminate the GPS measures. A node's latitude and longitude require no more than 16 bytes, the time the measures were taken requires 2 bytes, and, depending on the size of the network, only a few bytes are required for the node identifier. Finally, the node velocity and transmission radius, require an additional 4 bytes. Thus, packets containing the GPS measures are very small, requiring very little of a node's available bandwidth and associated energy to transmit.

A second way we address the goal of minimizing bandwidth and energy usage is in how the dissemination mechanism itself operates. Instead of each node periodically flooding the network with its GPS measures, we note that the frequency a node needs to disseminate its GPS measures should depend on its velocity since it is clear that a node's position changes more rapidly at higher speed. Thus, each node can locally adjust its dissemination frequency according to its mobility rate. (For instance, if a node is stationary it stops the dissemination entirely.)

Incorporating these observations into the dissemination mechanism results in extremely efficient use of node bandwidth and energy.

The accuracy of such a dissemination mechanism, as well as the effectiveness in supporting routing in ad hoc networks, has been studied and presented in [3] and [4], respectively.

3. DSM

Assume that a node S needs to send a packet to a subset M of nodes of the network, $|M| \leqslant n$. The node S is the *source* of the packet and the destinations in M form the *multicast group*. This is a completely general situation that includes multicast to a predefined, static multicast group, or the case in which the membership of a node in a group can vary over time. [1]

Every time a node needs to multicast a packet to the nodes of a specific group M, it computes from its GPS cache the graph G representing the "current" network topology. Then, it applies to G, *locally*, a (centralized) algorithm for the determination of a *minimum cost multicast tree*, i.e., the acyclic subgraph of G that spans all the nodes in the addressed group and that minimizes the total cost associated with the links. In our case, the cost associated with each link of an ad hoc network is 1. In this way, the total cost represents the total number of transmissions (hops) a packet takes to reach all nodes in the multicast group. Therefore, a minimum cost multicast tree minimizes the overall transmission time, the related energy consumption and the overall bandwidth needed.

Once the multicast tree is computed, a packet is processed in a manner similar to the dynamic source routing (DSR) protocol for routing in ad hoc networks [13]. Namely, the obtained tree is included in the header of the data packet, and the packet is transmitted in a hop-by-hop fashion to all and only those nodes in the tree (provided, of course, that each of these nodes is reachable). The similarities with the DSR protocol suggests the name for our multicast protocol: DSM.

In the rest of this section we address the four main problems that must be solved in order to implement DSM in ad hoc networks, namely:

1. Obtain a snapshot of the network topology, i.e., the graph G that represents the topology of the ad hoc network as seen by the source node according to its GPS cache.
2. Define a criterion to select an edge in the graph G so that the wireless link connecting the two nodes corresponding to that edge is still likely to be available.
3. Compute the multicast tree such that only negligible overhead is associated with the sending of a multicast packet.

[1] For the sake of presentation, we consider here that each node belongs to only one multicast group. The proposed multicast protocol also supports "multi-group" multicast, i.e., a node may belong to any number of groups.

4. Encode the multicast tree so that when the encoding is included in the header of the corresponding multicast packet it does not significantly affect the transmission time of the packet itself.

3.1. Obtaining network topology from GPS measures

Through the use of its GPS cache, each node knows the geographic location and the transmission radius of each other node at the time those measures were transmitted. Thus, a node can compute which nodes are in the transmission range of each node in the network, i.e., it can easily obtain a snapshot of the entire network topology: where all the nodes are located, and how they are (bidirectionally) linked.

In graph theoretic terms, this means that a source node S can construct from the GPS cache the *undirected graph* $G = (V, E)$ that corresponds to the network topology, where V is the set of network nodes, $|V| = n$, and E is the set of bidirectional radio links. A link e in E between two nodes A and B in V means that, according to the measures stored in S's GPS cache for A and B, the nodes A and B are in the transmission range of one another. As an example, let us assume that Fig. 1 depicts the content of A and B's entries in S's GPS cache at the time S wants to send a packet. (With $\mathrm{lat}(\cdot)$, $\mathrm{lon}(\cdot)$ and $\mathrm{tx}(\cdot)$ we indicate the entries for the latitude, the longitude and the transmission radius of a node, respectively.) In the topology graph G there is an edge between A and B if and only if

$$\mathrm{dist}(A, B) < \min\{\mathrm{tx}(A), \mathrm{tx}(B)\}, \tag{1}$$

where the *distance* function $\mathrm{dist}(\cdot,\cdot)$ depends on $\mathrm{lat}(\cdot)$, $\mathrm{lon}(\cdot)$ and possibly the GPS measure for altitude if something other than a flat topography is considered. Therefore, there is an edge between nodes A and B if and only if the (for instance, Euclidean) distance between A and B is less than the smaller of A and B's transmission radii.

We notice that the time complexity of the construction of the network topology graph G from the GPS cache is polynomial in n, the number of nodes in the network, and thus it only imposes a negligible overhead for a node. More precisely, since the links are bidirectional, for each node entry i, $i = 1, \ldots, n-1$, of the GPS cache, it is enough to check node entries $j > i$ whether inequality (1) is satisfied. Only $n(n-1)/2$ of these checks are needed, namely, the time complexity is $O(n^2)$.

3.2. Defining a suitable topology graph G

The measures contained in a node's GPS cache are time-stamped with the time they were taken at the node that originated them. This means that, when the source S of a multicast packet needs to retrieve the topology of the network, a node A may no longer be in the location stored in S's cache since nodes are mobile. This implies that the communication link between two nodes may no longer exist. Let us consider the situation shown in Fig. 2.

The filled circles represent the location of nodes A and B according to S's GPS cache. Suppose that in these positions, the nodes are within the transmission radius of each other denoted by the line connecting them. Now, let us assume that, when S

A	....	lat(A)	lon(A)	tx(A)	....
⋮			⋮		⋮
B	....	lat(B)	lon(B)	tx(B)	....

Fig. 1. The entries for nodes A and B in S's GPS cache.

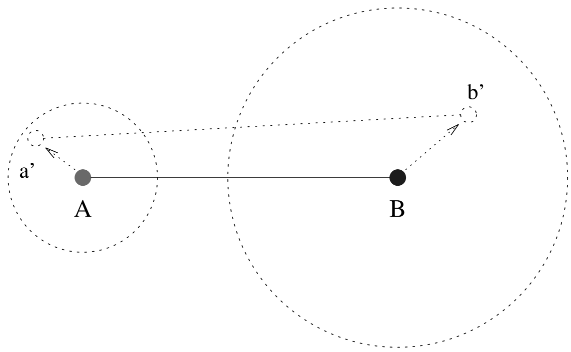

Fig. 2. The possible movement areas of nodes A and B with respect to their positions stored in S's GPS cache.

wants to compute the topology graph G, the nodes A and B have moved to the positions marked with a' and b' and that these new positions *do not* satisfy inequality (1). Thus, S's GPS cache leads us to consider a link that is no longer available. In general, the topology graph G resulting from a node's GPS cache may contain such links, i.e., links that are possibly no longer available for the transmission of the multicast packet. If these links are actually missing, they compromise the success of the multicast.

It is easy to see that the closer the distance separating two nodes A and B is to the minimum of their transmission radii (i.e., at the boundary of the transmission radius, see inequality (1)), the more likely that the link between A and B will no longer be available (as in Fig. 2). This depends on the fact that the graph G is built considering inequality (1), i.e., simply considering the transmission radius of the nodes.

In order to reduce the dependence on these possibly missing links, we introduce a protocol parameter δ such that $\delta < \min_{A\in V} \{\mathrm{tx}(A)\}$.

The condition of existence of a link between two nodes can now be redefined so that there is a link between two nodes A and B if and only if

$$\mathrm{dist}(A,B) < \delta. \tag{2}$$

In other words, by selecting a suitable δ we select only those links that are more likely to be *available* when a packet has to be forwarded to the nodes of a multicast group. The resulting graph G should have only those links that satisfy inequality (2). (These links are less likely to be disrupted by node movement as compared to the links obtained by using inequality (1).)

The main problem in this case is to be careful not to exclude those links that do not satisfy inequality (2), but that are necessary to maintain the network connectivity, i.e., we want to ensure the possibility of having at least one route to each node in the multicast group M. A connected topology graph G can be obtained as follows:

1. Obtain the network topology graph $G = (V,E)$, as described in the previous section, namely, by considering inequality (1). (It is assumed that the graph obtained in this step is connected.)
2. Select all the links that do not satisfy the "δ condition" (inequality (2)).
3. Among the selected links (Step 2), keep only those whose deletion compromises the connectivity of G.

The third step, namely, deciding if the removal of a link disconnects G (i.e., in graph theoretic terms deciding if that link is a *bridge*) can be efficiently determined by using, e.g., a depth-first search algorithm in G (see, e.g. [8]). The time complexity of depth-first search is $O(n+m)$ where $n = |V|$ and $m = |E|$. Since we consider connected networks (we always want the possibility to perform the multicast to any multicast group M), namely, $n \leqslant m$, the time complexity of the third step is $O(m)$. Thus, the overall complexity of producing a "suitable" topology graph G is $O(n^2)$.

The definition of δ is of course a crucial step of this phase of our protocol. An analytical definition that takes into account the time elapsed since the last position update, the original position and the speed of the nodes and the areas of their possible movements (the large dotted circles centered around the filled circles in Fig. 2) is beyond the scope of this paper. For formal definitions, results and further considerations the reader is referred to [2] where the concept of *link availability* is introduced for the selection of the "most available" among the routes between any two nodes.

3.3. Constructing a multicast tree

Once the source of the multicast has generated the topology graph G, a *multicast tree*, i.e., a minimal set of routes to all the nodes in the multicast group, is locally computed.

Constructing a minimum cost multicast tree, also called a *Steiner tree*, for the nodes of a given group in a generic network is a well known NP-hard optimization problem (see, among many others, [12]). This implies that exact algorithms for generating the tree of minimum cost to all nodes in a given group requires computational time which would detrimentally affect the delivery of the packet to the group. Therefore, a number of heuristic algorithms have been proposed that allow the construction of a Steiner tree in a time which is polynomial in n, the number of nodes in the net-

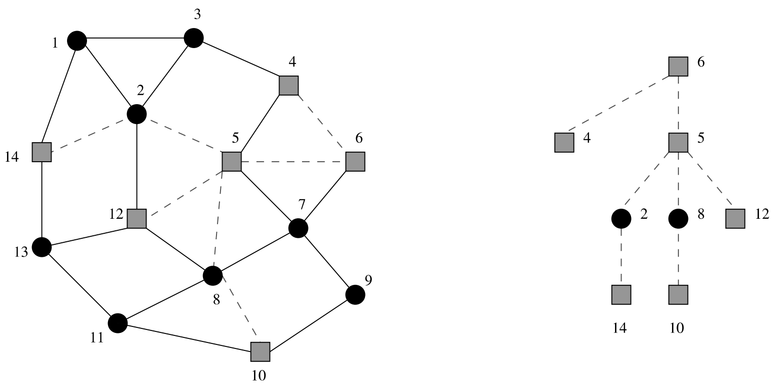

Fig. 3. The network topology according node 6's GPS cache (left), and the Steiner tree for M, rooted at 6 (right).

work, and m the number of bidirectional links. Furthermore, for many of these algorithms it is possible to prove an error ratio with respect to an optimal solution which is at most $2-(2/|M|)$, where M is the addressed group of nodes, thus guaranteeing a bounded distance from the best possible solution. (Extensive overviews of and references to these heuristics can be found in [12] and [14], respectively.)

The DSM protocol imposes no limitations on the choice of the algorithm for the local computation of a Steiner tree. Thus, the efficiency of the local computation of our solution is determined by the best possible heuristic for computing Steiner trees. In any case, a solution is obtained in polynomial time, i.e., efficiently, and it is provably "not far" from an optimal (minimum cost) solution.

As an example, suppose that the multicast source node 6 receives a packet to multicast to the group $M = \{4, 5, 6, 10, 12, 14\}$. Node 6 first constructs the network topology graph G from its GPS cache (as described in the previous two sections). The left half of Fig. 3 shows G, with the square vertices representing the nodes in M. Node 6 now computes a multicast tree for G and M, rooted at 6. The algorithm selected for this example (and for the simulation in the next section) is a minimum spanning tree based heuristic that produces a multicast tree in a time proportional to $m + n \log n$. The details of the algorithm can be found in [17], and are summed up in [12]. The right half of Fig. 3 shows the resulting Steiner tree. Note that a multicast tree always has group members as leaves, and interior nodes may or may not be group members.

3.4. Optimal coding of trees

Finally, once a tree that describes the route to all nodes in the multicast group has been obtained, we need to encode the tree and include it in the header of the multicast packet. Although it may seem expensive in terms of bits, and thus, in terms of transmission time, we show that a multicast tree can be encoded using space comparable to the space needed to encode a route according to the DSR routing protocol.

Any finite tree with j nodes (and thus $j-1$ links) can be optimally encoded as a sequence of $j-2$ integers (the node identifiers) using a *Prüfer sequence* (see, e.g., [15]). A Prüfer sequence uniquely characterizes a tree, in the sense that there is a one-to-one correspondence between the set of all finite trees and their Prüfer sequences. This correspondence is defined as follows. Given any tree T of order (number of nodes) $j \geqslant 3$ whose vertices are labeled from 1 to j, the following is a polynomial time algorithm that returns T's Prüfer sequence [15]:

Step 1. Find the node v in T of degree 1 (a *leaf*) such that $v = \min_{k \in J=\{1,\ldots,j\}} \{k\}$.

Step 2. Record the identifier of the node adjacent to v (there is only one since v is a leaf). Stop if $j-2$ identifiers have been recorded, otherwise, go to Step 3.

Step 3. Delete v from T (and from the set J of Step 1) and return to Step 1 with the tree $T - v$.

It is easy to see that the time complexity of this algorithm is $O(j^2)$. (Assuming an array representation of the tree, where each entry i of the array, $i = 1, \dots, j$, contains the identifier of the parent of node i and the number of children of node i, then all of the steps of the algorithm above can be executed all together in linear time at most $j - 2$ times.) Thus, in the worst case (a tree that spans the whole topology graph G, i.e., $j = n$), the time complexity of coding a tree into its Prüfer sequence is $O(n^2)$.

We notice that this encoding is as efficient as specifying the longest source route needed for routing in ad hoc networks according to on-demand protocols such as the DSR protocol [13]. Thus, from the perspective of the size of the header of a data packet, our proposed multicast requires only as much space as an on-demand routing protocol. As a consequence, the encoded Steiner tree induces very little overhead along with the transmission of the packet and, as in routing, it reduces with each hop subsequently taken by the packet.

In our example (see previous subsection), node 6 encodes the Steiner tree with eight nodes (Fig. 3, right) as the Prüfer sequence {6,5,8,5,5,2} of length 6, and broadcasts the encoding along with the packet, that also carries the identifier of the group (in our case, M). (We assume the transmission medium is broadcast.)

A node A that receives a multicast packet uses the following polynomial algorithm to retrieve the multicast tree included in the header, thus obtaining information about whether it is in the tree or not. The key of the decoding procedure is the observation that a Prüfer sequence of a tree encodes in the sequence *only* interior, i.e., non-leaf, nodes. Thus, a node that receives the packet *and* that belongs to the multicast group *and* whose identifier is *not* in the sequence is a leaf and does not have to forward the packet:

Step 0. If A belongs to the multicast group and if its identifier is not in the received Prüfer sequence (i.e., A is a leaf node) then receive the packet and stop; otherwise go to Step 1.

Step 1. Make a list of the leaves of the encoded tree, i.e., make a list of all the nodes whose identifier is not in the received sequence *and* belongs to the addressed group M; [2]

Step 2. Find the smallest identifier, say k, in the list of leaves. This is the identifier of the first leaf deleted from the original tree to obtain this Prüfer sequence. Let h be the first identifier in the sequence. Then, the nodes with identifiers k and h must be adjacent in the tree, and the corresponding link is part of our tree. (We thus record this link.)

Step 3. Delete the identifier k from the list of leaves. If the Prüfer sequence has only one element h, then there will remain only one element, say g, in the list of leaves. Then, the nodes with identifiers g and h must also be adjacent in the tree. We record the corresponding link and stop the decoding. Otherwise, delete the first element h of the sequence. If h is repeated elsewhere in this shortened Prüfer sequence, return to Step 2. Otherwise, add h to the list of leaves and return to Step 2.

Step 0 is clearly performed in linear time, i.e., its time complexity is $O(j)$. Implementing the list of leaves by an array with n entries (marking, e.g., with a 1 the nodes that are leaves of the Steiner tree and with a 0 the nodes that are either interior nodes of the tree or that do not belong to the tree) it is easy to perform Step 1 in linear time also ($O(j)$). As for Step 2, to find the leaf with the smallest identifier, with the described implementation, requires at most $O(n)$ time steps. In Step 3, deleting the identifier k requires constant time, and checking if h is repeated elsewhere in the sequence requires $O(j)$ time. Since the procedure terminates after all the $j - 2$ elements of the sequence have been considered (and removed), the decoding procedure requires $O(nj)$ time steps, i.e., its overall time complexity is $O(n^2)$.

Let us consider our example again. Node 6 broadcasts the Prüfer sequence {6,5,8,5,5,2} to its neighbors, nodes 4, 5 and 7. According to the decoding algorithm, since node 4 is not in the sequence and belongs to M, it realizes that it is a leaf

[2] We assume here that each node knows the nodes in M. In the case in which this information is not locally available, it can be transmitted along with the packet as a n-bit long string.

node, and thus it receives the packet, but does not forward it any further (notice that no decoding of the sequence is necessary). Since the radio transmission from node 6 is a broadcast, node 7 receives the packet too. But node 7 is not in the Prüfer sequence, and since it does not belong to M it is not involved in the forwarding of the packet and simply discards the packet (again, no decoding of the sequence is necessary). Node 5 is in M and so it receives the packet, and since it is also in the Prüfer sequence, it realizes that it is an interior node. Thus, it decodes the multicast tree, and forwards the packet with the encoding of its corresponding subtree ($\{8,5,5,2\}$ of length $6-2=4$). The process continues until all leaf nodes, if possible, are reached.

4. Simulations results

We have simulated the DSM protocol mainly to demonstrate its effectiveness in delivering multicast packets. A discrete-event simulator of an ad hoc network, implemented in C++, was used to count the number of "successful multicasts," namely, the number of multicasts that deliver the packet to all the nodes in the addressed group. If some node of the group (even one) does not receive the packet, the multicast is considered unsuccessful. In this latter case, we have measured the percentage of the nodes of the addressed group that have received the packet. [3]

The n nodes of the ad hoc network can freely move around in a rectangular region (modeled as a grid) according to the following mobility model. (To ease the modeling, the node movements are discretized to grid units with a grid unit = 1 m.) Each time it moves, a node determines its direction randomly, by choosing between its current direction (with 75% probability) and uniformly among all other directions (with 25% probability). The node then moves in the chosen direction according to its current speed. When a node reaches a grid boundary, it bounces back into the region with an angle determined by the incoming direction.

Each node has a fixed transmission range of 350 m (we found this value resulted in good network connectivity, i.e., more than 98% of the time, after network topology changes, the network was connected). Each node is modeled by a store-and-forward queuing station, and is characterized by parameters such as buffer space which is assumed to be adequate for packets that are awaiting transmission. Each link is modeled by a FCFS queue with service time as the packet transmission time characterized by a bandwidth of 1 Mbits/s. Control packets containing the GPS measures and multicast (data) packets share the same transmission channel (which implies that the accuracy of the dissemination mechanism may be affected by the network load, and that the transmission of data multicasts may be slowed down by the transmission of the GPS measures).

Each control packet contains time-stamped, node identified, position coordinates and the current transmission radius of a node, which in the current experiments is considered the same for each node. These packets are generated every time a node moves (i.e., at a frequency that is a function of the node velocity; see also [3]).

Multicast packets contain a payload that is 512 bytes in size, the identifier of the source node and that of the addressed group, as well as the encoded Steiner tree. In our simulations we have considered a "heavy" network load: the multicast packets arrivals are distributed exponentially with a mean of 10 ms for networks with $n=30$ nodes and 50 ms in networks with $n=60$ nodes.

Fig. 4 refers to an ad hoc network with $n=30$ nodes in a grid of 1000 m $\times$ 1000 m. It shows the percentage of successful multicasts for nodes whose velocity varies from 6 to 20 m/s, i.e., from around 20 to around 70 km/h. For multicast group sizes between $n/10$ and $n/3$ (here only $(n/10)=3$, $(n/6)=5$ and $(n/3)=10$ are plotted), all the nodes of the addressed multicast groups received the packet more than 97% of the time (i.e., more than 97% of the multicasts were successful).

Furthermore, of the unsuccessful multicasts, more than 85% of the nodes received the packet.

[3] Currently, our study is concerned only with network-layer details. Thus no physical-layer is modeled, and the link-layer is taken into account only with respect to the delay: we have randomly augmented up to 50% the transmission time of a packet.

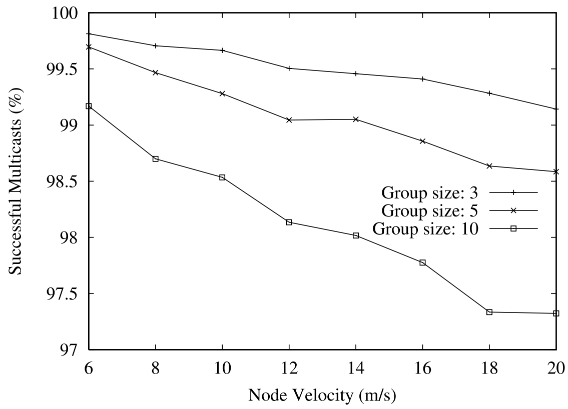

Fig. 4. Average percentage of successful multicasts in networks with 30 nodes.

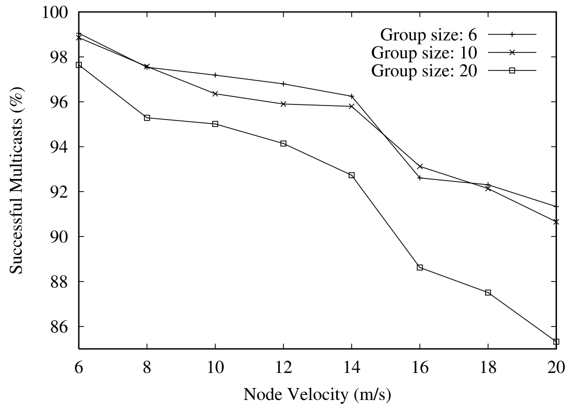

Fig. 5. Average percentage of successful multicasts in networks with 60 nodes.

Thus, even though the multicast packet failed to reach all the nodes in the addressed group, only a few nodes (less than 15%) did not receive the packet.

Our second set of simulations concerned networks with $n = 60$ nodes in a 1000 m $\times$ 2600 m grid. [4] The percentage of successful multicasts for the various velocities and groups with sizes $(n/10) = 6$, $(n/6) = 10$ and $(n/3) = 20$ is shown in Fig. 5. In this case, more than 90% of the multicasts are successful, and, in the case of an unsuccessful multicast, more than the 80% of the nodes in the group receive the packet.

In both cases (networks with 30 and 60 nodes) we have also compared our protocol with global flooding, the simplest multicast protocol that, as DSM, does not assume the construction and maintenance of any underlying network structure. As expected, simulations show that DSM improves on the average delay of (successful) multicast completion up to 50%.

All the simulations ran for a time long enough to achieve a confidence level of 95% with a precision within 5%.

[4] As in the case with 30 nodes, these values for the grid sides and the selected transmission radius guarantee that after movements occur the network is connected.

5. Conclusions

In this paper we have described how, using location awareness through positioning devices and efficient dissemination of geographic measures, a DSM protocol can be designed for ad hoc networks. Different from previously proposed multicast protocols, DSM does not assume, construct or maintain any network structure, nor does it use an ad hoc routing or clustering protocol as a basis. Thus, node and network resources can be basically entirely used for the transmission of the multicast packets. The protocol is executed locally at each node, where, according to the cached geographic measures, a multicast tree is computed in polynomial time. The resulting tree is then optimally encoded locally and transmitted along with the packet extending the length of the packet header by no more than ad hoc source routing. The time complexity of all the local operations is shown to be $O(n^2)$, i.e., the proportional to the square of the number of nodes in the network.

Simulation results show that our protocol is effective and dependable for the delivery of multicast packets in networks up to 60 nodes, with high network loads and regardless the nodes' velocity and group sizes. The behavior of DSM has to be investigated when the size of the network grows. In this case, it is expected that both the dissemination mechanism and the protocol should be optimized to obtain scalability. In particular, our research will be directed towards implementing the dissemination mechanism so that a GPS measure is not flooded throughout the possibly huge network, but instead only to those nodes that are mostly affected by the movements of the sending node, i.e., the "closer nodes" (see also [4]). We also intend to investigate the use of tree-caching and/or tree recomputation at intermediate nodes to improve the number of successful multicasts when the multicast tree is very large.

References

[1] T. Ballardie, P. Francis, J. Crowcroft, Core based trees (CBT): An architecture for scalable inter-domain multicast routing, in: Proceedings of the ACM SIGCOMM'93, Communications Architectures, Protocols and Applications, San Francisco, CA, September 13–17, 1993, Comput. Commun. Rev. 23 (4) (1993) 85–95.

[2] S. Basagni, I. Chlamtac, A. Faragó, V.R. Syrotiuk, R. Talebi, Route selection in mobile multimedia ad hoc networks, in: Proceedings of the Sixth IEEE International Workshop on Mobile Multimedia Communications, MOMUC'99, San Diego, CA, November 15–17, 1999, pp. 97–103.

[3] S. Basagni, I. Chlamtac, V.R. Syrotiuk, Geographic messaging in wireless ad hoc networks, in: Proceedings of the IEEE 49th Annual International Vehicular Technology Conference, vol. 3, Houston, TX, May 16–20, 1999, pp. 1957–1961.

[4] S. Basagni, I. Chlamtac, V.R. Syrotiuk, B.A. Woodward, A distance routing effect algorithm for mobility (DREAM), in: Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking, MobiCom'98, Dallas, TX, October 25–30, 1998, pp. 76–84.

[5] C.-C. Chiang, M. Gerla, Routing and multicast in mobile wireless networks, in: Proceedings of 1997 IEEE 6th International Conference on Universal Person Communications Record, Bridging the Way to the 21st Century, ICUPC'97, vol. 2, San Diego, CA, October 12–16, 1997, pp. 557–561.

[6] C.-C. Chiang, M. Gerla. On-demand multicast in mobile wireless networks, in: Proceedings of IEEE ICNP'98, Austin, TX, 1998.

[7] C.-C. Chiang, M. Gerla, L. Zhang, Adaptive shared tree multicast in mobile wireless networks, in: Proceedings of Globecom'98, IEEE Global Telecommunications Conference, Sydney, Australia, November 8–12, 1998.

[8] T.H. Cormen, C.E. Leiserson, R.L. Rivest, Introduction to Algorithms, The MIT Press and McGraw-Hill, Cambridge, MA and New York, 1990.

[9] S.M. Corson, S.G. Batsell, A reservation-based multicast (RBM) routing protocol for mobile networks: Initial route construction phase, Wireless Networks 1 (4) (1995) 427–450.

[10] P.H. Dana, The Geographer's Craft Project, Department of Geography, The University of Texas at Austin, Austin, TX, 1998, http://www.utexas.edu/depts/grg/gcraft/notes/gps/gps.html.

[11] J.J. Garcia-Luna-Aceves, E.L. Madruga, The core-assisted mesh protocol, IEEE J. Selected Areas Commun. (Special Issue on Wireless Ad Hoc Networks) 17 (8) (1999) 1380–1394.

[12] F.K. Hwang, D.S. Richards, P. Winter, The Steiner Tree Problem, Annals of Discrete Matematics, vol. 53, North-Holland, Amsterdam, 1992.

[13] D. Johnson, D.A. Maltz, Dynamic source routing in ad hoc wireless networks, in: T. Imielinski, H.F. Korth, (Eds.), Mobile Computing, Kluwer Academic Publishers, Dordrecht, February 1996, pp. 153–181 (Chapter 5).

[14] K. Makki, N. Pissinou, O. Frieder, Efficient solutions to multicast routing in communication networks, Mobile Networks Appl. (MONET) 1 (2) (1996) 221–232.

[15] E.M. Palmer, Graphical Evolutions: An Introduction to the Theory of Random Graphs, Wiley-Interscience Series in Discrete Mathematics, Wiley, New York, 1985.

[16] E.M. Royer, C. E. Perkins, Multicast using ad-hoc on-demand distance vector routing, in: Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking, MobiCom'99, Seattle, WA, August 15–20, 1999, pp. 207–218.

[17] H. Takahashi, A. Matsuyama, An approximate solution for the steiner problem in graphs, Math. Jap. 24 (1980) 573–577.

Stefano Basagni holds a Ph.D. degree in computer science from the University of Milano, Italy (May 1998). He is currently an Assistant Professor in the Department of Computer Science in the Erik Jonsson School of Engineering and Computer Science at the University of Texas at Dallas (UTD). Since 1997 he is also a faculty member of the Center for Advanced Telecommunications Systems and Services (CATSS) at The University of Texas at Dallas. Dr. Basagni current research interests concern theoretical and implementation aspects of mobile networks and wireless communication systems, network protocols and theoretical and practical aspects of distributed algorithms. Dr. Basagni served as a guest editor of the special issues of the ACM/Kluwer Journal on Special Topics in Mobile Networking and Applications (MONET) on Multipoint Communication in Wireless Mobile Networks, as well as of the special issue on Mobile Ad Hoc Networks of the Wireless Communications & Mobile Computing journal (Wiley). He served on several Organizing and Technical Program Committees, including the ACM/IEEE MobiCom OC and TPC, IFIP Networking OC, and has served as session chair and organizer at IEEE, ACM and IASTED conferences.

Imrich Chlamtac holds a Ph.D. degree in computer science from the University of Minnesota, and B.Sc. and M.Sc. degrees awarded with the Highest Distinction from Tel Aviv University. Since 1997 he holds the Distinguished Chair in Telecommunications at the University of Texas at Dallas, where he is also the Director of CATSS, the Center for Advanced Telecommunications Systems and Services. Prior to joining UTD, Dr. Chlamtac was on faculty at Technion, The Israel Institute of Technology and the University of Massachusetts in Amherst. Dr. Chlamtac is also the recipient of the titles of Senior Professor at Tel Aviv University, The Bruno Kessler Honorary Professor at the University of Trento, Italy, and of University Professor at the Technical University of Budapest. Dr. Chlamtac is a Fellow of the IEEE and ACM societies for numerous contributions to networking, including the concept of lightpath or wavelength routing and contention free channel access methods. He is also an IEEE Distinguished Lecturer and was a Fulbright Scholar. Dr. Chlamtac published over 250 papers in refereed journals and conferences and is the co-author of the first textbook on Local Area Networks and of Mobile and Wireless Networks Protocols and Services (Wiley). Dr. Chlamtac serves as the founding Editor in Chief of the ACM/URSI/Kluwer Wireless Networks (WINET), Mobile Networks and Applications (MONET) journals, and the SPIE/Kluwer Optical Networks Magazine (ONM) and has served on the editorial boards of most major publications in telecommunications. Dr. Chlamtac is the founder of ACM/IEEE MobiCom and SPIE/IEEE/ACM OptiComm conferences.

Violet R. Syrotiuk received the Ph.D. degree in Computer Science from the University of Waterloo (Canada) in 1992. She is currently an Assistant Professor in the Department of Computer Science in the Erik Jonsson School of Engineering and Computer Science at the University of Texas at Dallas. From 1997 to 1999 she was involved in a DARPA Global Mobile Information Systems (GloMo) project working to make the mobile environment a "first class citizen" in the defense information infrastructure. Her current research interests include MAC and network layer protocols for wireless mobile networks, network simulation, and distributed algorithms and systems. Dr. Syrotiuk is a member of IEEE and ACM.