On-Demand Location Aware Multicast (OLAM) for Ad Hoc Networks*

Stefano Basagni Imrich Chlamtac Violet R. Syrotiuk Rodeen Talebi

Center for Advanced Telecommunications Systems and Services (CATSS)
Erik Jonsson School of Engineering and Computer Science
The University of Texas at Dallas
{basagni,chlamtac,syrotiuk,rodeen}@utdallas.edu

Abstract- **This paper introduces *OLAM*, a novel *On-demand Location Aware Multicast* protocol for ad hoc networks. The protocol assumes that, through the use of positioning system devices, such as Global Positioning System (GPS) devices, each node knows its own position and the current (global) time, and it is able to efficiently distribute these measures, including its current transmission radius, to all other nodes. As the measures are received, each node updates its local *snapshot* of the complete network topology. When a packet is to be multicast to a group, a heuristic is then used to locally compute the *Steiner* (i.e., multicast) tree for the addressed multicast group based on the snapshot rather than maintaining the tree in a distributed manner. The resulting Steiner tree is then optimally encoded by using its unique *Prüfer sequence* and included along with the packet, extending the length of the header by no more than the header of packets in source routing (unicast) techniques. All local computations are executed using efficient (i.e., polynomial time) algorithms. The protocol has been simulated in ad hoc networks with 30 and 60 nodes and with different multicast group sizes. We show that OLAM delivers packets to all the nodes in a destination group in more than 85% of the cases. Furthermore, compared to flooding, OLAM achieves improvements of up to 50% on multicast completion delay.**

I. INTRODUCTION

Among the basic network operations, *multicast* refers to forms of communication with multiple participants (generally more than two). In this paper we consider the special case of one-to-many communication (*source multicast*), wherein the same packet is sent from one node (the *source*) concurrently to a specified subset of nodes of the network (the *multicast group*).

We consider networks in which all nodes can be mobile, called mobile multi-hop radio networks (or *ad hoc* networks). Unlike cellular networks, in which a mobile node communicates by radio one hop through a fixed base station interconnected by a wired backbone network, ad hoc networks have no fixed infrastructure and may need to communicate by radio for multiple hops. In this latter case, each node relies on its neighbors to forward packets to destinations not directly in its transmission range, thus also acting as a router.

Many existing protocols for multicast are based on shared *multicast routing trees*, both in wireline and in ad hoc networks (see, e.g., [1, 2, 3]). However, these solutions, which include the recent "multicast mesh" solution proposed in [4], the on-demand multicast presented in [5], and the multicast operation of the AODV routing protocol [6], all have a high dependence on the existence and correct operation of an underlying ad hoc routing protocol. Other solutions, instead, depend on a hierarchical organization of the ad hoc network (e.g., clustering), as in the algorithm proposed in [7]. Thus, whether based on ad hoc routing or clustering, these multicast protocols expend great effort to distributively maintain the "multicast routing structure," and this can heavily affect the overall multicast performance.

In this paper we propose a new multicast protocol for ad hoc networks which neither assumes any ad hoc routing scheme, nor builds and maintains any distributed data structure. The proposed protocol is based on the expectation that nodes of ad hoc networks will be equipped with positioning system devices, such as Global Positioning System (GPS) devices (possibly integrated with some inertial positioning devices, to permit use indoors) whose commercial availability is as common as any other peripheral device. Through the use of GPS, each node is aware not only of its three-dimensional position (latitude, longitude and altitude), but also of its current velocity and the current global time.

Using a dissemination mechanism specifically tuned to the system requirements of ad hoc networks such as the one introduced and described in [8] and [9], a node's GPS measures and current transmission radius are continually distributed throughout the network as the node moves. On receipt of such a "GPS packet," a snapshot of the "cur-

* This work was supported in part by the Army Research Office (DARPA) under contract No. DAAG55-97-1-0312.

rent" network topology can be maintained locally as an undirected graph G in which two nodes are neighbors if their distance is less than the minimum of their transmission radii.

When a source node S receives a packet to multicast to a group M, S uses a heuristic to *locally* compute a multicast tree (called also a *Steiner tree*, in graph theoretic terms) for G and M rooted at S. The source node S then transmits the packet, that includes a coded representation of the paths (i.e., the tree) the packet is to follow, to each of its child nodes. Each child node is the root of a subtree and will forward the packet to each of its children until finally the path is exhausted, at which point the multicast is complete.

We show that, by using efficient algorithms for the local computation of the multicast tree, and by optimally coding the tree through its unique *Prüfer sequence* [10], our On-demand Location-Aware Multicast (OLAM) protocol achieves the following desirable properties:

1. OLAM is easy to implement, relying only on a bandwidth and energy efficient dissemination mechanism, rather than underlying routing or clustering protocols. As well, since the computations are performed locally, no complex coherent distributed data structure (such as a distributed multicast tree) needs to be maintained among the nodes.

2. The local computation of the multicast tree does not impose a significant overhead to the multicast, since a polynomial time (i.e., computationally efficient) approximation algorithm can be used to compute a Steiner tree in the network topology graph. Furthermore, the local complexity of OLAM is linked to the best (possibly yet to come!) algorithm for computing a Steiner tree in a network graph. As well, no packet looping can occur since the tree is computed locally.

3. Due to optimal coding through a tree's unique Prüfer sequence, there is no overhead associated with transmitting the encoded tree along with a data packet, with respect to ad hoc source routing solutions (like the on-demand DSR presented in [11]), since a tree can be uniquely encoded by a sequence whose length is at most the length of the longest route between two nodes.

4. OLAM 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.

5. "Dynamic multicast," i.e., the possibility for a node to change (join/leave) groups, is easily and efficiently supported in OLAM by sending the identifier of a node's new group along with its GPS measures.

The effectiveness of OLAM in successfully completing multicast is demonstrated through the use of simulation. The obtained results show that in ad hoc networks with 30 and 60 nodes 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 in more than 85% of the time (this percentage is actually $> 97\%$ in the case of networks with 30 nodes). In the remaining 15% (3%) of the cases, always more than 80% of the nodes in the addressed multicast group receive the packet, leaving only less than 20% of the nodes in the group 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 the paper is organized as follows. Section II explains how a graph representing the network topology can be constructed from the GPS measures. Section III describes the multicast protocol in detail, while Section IV demonstrates the effectiveness of our protocol in delivering a packet to all nodes of the addressed group. Section V presents conclusions and future work.

II. Obtaining Network Topology from GPS Measures

The OLAM protocol assumes that each node is aware of its own geographical position which can be easily obtained by equipping a node with a Global Positioning System (GPS) receiver. Such devices allow the nodes to receive GPS broadcasts and compute their three-dimensional position (latitude, longitude, and altitude), velocity and time (from now on, together with a node's transmission radius, called GPS *measures*) with a precision to within 100 meters horizontal, 156 meters vertical, and 340 nanoseconds time.

Since nodes in an ad hoc network lack a fixed infrastructure, every node in the network is responsible for disseminating its GPS measures to all the other nodes. This is obtained by flooding the network with a packet containing its 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. Its accuracy in disseminating GPS measures, as well as the effectiveness in supporting routing in ad hoc networks, has been studied and presented in [9] and [8], respectively.

Through the use of the dissemination mechanism, each node knows the geographic position 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 to whom they are (bidirectionally) linked. In graph theoretic terms, this means to construct from the "GPS packets" the *undirected graph* $G = (V, E)$ of the network topology, where V is the set of network nodes, and E is the set of bidirectional radio links. (A link e in E between two nodes A and B in V means that the nodes A and B are in the transmission range of one another.) This local network topology graph G can be easily maintained in an on-line fashion: every time a GPS packet is received, G is modified accordingly (if needed). The (time) complexity of this update operation is clearly linear in n, the number of the nodes in the network, thus imposing a negligible overhead on the node. Given the adaptation to mobility of the dissemination mechanism used [9], the network topology graph at each node provides quite a faithful snapshot of the "current" network topology, and can be effectively used to compute the routes to the nodes in a given multicast group.

III. On-Demand Multicast in Ad Hoc Networks

We assume that the n nodes of the network are partitioned into *k multicast groups*. Multicast can then be defined as the sending of a packet from one node, the multicast *source*, to all the nodes of a specific group.[1]

Every time a node has to multicast a packet to the nodes of a specific group, it applies a heuristic 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) to its local network topology graph G. For our purposes, the cost associated with each link of an ad hoc network is 1. Thus, the total cost represents the total number of transmissions (hops) a packet takes to reach all the nodes in the multicast group. Therefore, a minimum cost multicast tree minimizes the overall transmission time, the related energy consumption and the overall needed bandwidth.

Once the multicast tree is computed, a packet is processed in a manner similar to any source routing protocol. 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 the nodes in the tree (provided that each of these nodes is reachable).

The resulting multicast is thus *on-demand*, since the multicast tree is computed only when needed, and no distributed data structure (i.e., a tree, or any other structure) is built and maintained in order to multicast a packet.

Two main problems have to be addressed in order to implement the described multicast protocol in ad hoc networks:

1. How to efficiently compute the multicast tree, so that only negligible overhead is associated with the sending of a multicast packet?, and

2. How to code a multicast tree so that the corresponding header of the multicast packet does not significantly affect the transmission time of the packet itself?

In the remaining part of this section we describe how these problems can be solved efficiently.

[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.

1. 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 means that exact algorithms for generating the tree of minimum cost to all the nodes in a given group requires computational time that detrimentally affects the delivery of the packet to the group. Therefore, many 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 network, 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 $2 - \frac{2}{h}$, where h is the number of nodes in the addressed group, thus guaranteeing a bounded distance from the best possible solution. (Extensive overviews of these heuristics can be found in, e.g., [12] and [13].)

The OLAM protocol imposes no limitations on the choice of the algorithm for the local computation of a Steiner tree. Thus, the efficiency of our solution is connected to 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 $M = \{4, 5, 6, 10, 12, 14\}$. The left half of Figure 1 shows G, the network topology graph that represents a snapshot of the network topology as seen by node 6 at the time it receives the packet to multicast. (The square vertices represent the nodes in M.) Node 6 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 [14], and are summed up in [12]. The right half of Figure 1 shows the resulting Steiner tree. Note that a multicast tree always has group members as leaves.

2. Any finite tree with j nodes (and $j-1$ links) can be optimally encoded as a sequence of $j-2$ integers (the node identifiers) using a *Prüfer sequence* (see, e.g., [10]). 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.[2] More importantly, the encoding and decoding of a tree are obtainable in time which is polynomial in j, the size of the tree. In our case, we use Prüfer sequences for the computed Steiner tree with $j \leq n$ nodes. We notice that this encoding is as efficient as specifying the longest source route needed for

[2] A detailed description of how to generate the Prüfer sequence of a tree, and vice versa, is beyond the scope of this paper. To better understand the example, it is enough to know that the Prüfer sequence of a tree encodes in the sequence *only* interior, i.e., non-leaf, nodes. Further details can be found in [10].

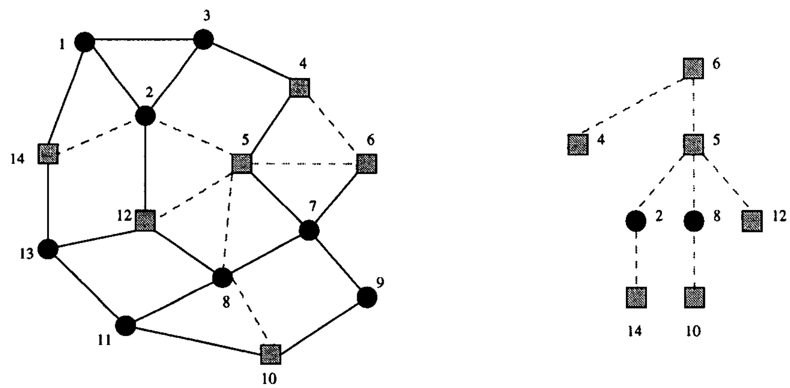

Figure 1: The network topology according node 6 (left), and a Steiner tree for M, rooted at 6 (right).

routing in ad hoc networks according to on-demand protocols such as the DSR protocol [11]. Thus, from the perspective of header size of a data packet, the proposed multicast needs as much space as an on-demand routing protocol. As a consequence, the encoded Steiner tree incurs very little overhead along with the transmission of the packet and, as for routing, it reduces with each hop subsequently taken by the packet.

In our example, node 6 encodes the Steiner tree with 8 nodes (Figure 1, right) as the Prüfer sequence $\{6, 5, 8, 5, 5, 2\}$ of length 6, and broadcasts the encoding along with the packet, that carries also the group identifier (in our case, M). According to the Prüfer decoding algorithm, since node 4 is not in the sequence and belongs to M, it realizes that it is a leaf node, and thus it receives the packet, but does not forward it any further. 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 (node 7 receives the packet because we assume a broadcast type of transmission at the MAC layer, i.e., all the current neighbors of a node v eventually receive a packet transmitted by v). Node 5 is in M and so 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.

IV. Simulations Results

We have simulated the OLAM protocol to demonstrate its effectiveness in delivering multicast packets. A 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 meter.) 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 hits 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 350m (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 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 [9]).

Multicast packets contain a payload that is 1024 bytes

[3] Currently, our study is limited to network-layer details, thus no link- or physical-layer are modeled.

in size, the identifier of the source node and that of the addressed group, as well as the encoded Steiner tree. For each packet, the source node and the destination group are chosen randomly and uniformly among all the nodes of the network. In our simulations we have considered a "heavy" network load: the multicast packets arrivals are distributed exponentially with a mean of 10ms for networks with $n = 30$ nodes and 50ms in networks with $n = 60$ nodes. (This corresponds to an average of 3 packets/node per second when $n = 30$ and 1 packet/node every 3 seconds.)

Figure 2 (a) refers to an ad hoc network with $n = 30$ nodes in a 1000m $\times$1000m grid. It shows the percentage of successful multicasts for nodes whose velocity varies from 6m/s to 20m/s, i.e., from around 20 km/h to around 70 km/h. For multicast group sizes between $\frac{n}{10}$ and $\frac{n}{3}$ (here only $\frac{n}{10} = 3$, $\frac{n}{6} = 5$ and $\frac{n}{3} = 10$ 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. 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 1000m $\times$2600m.[4] The percentage of successful multicasts for the various velocities and groups with sizes $\frac{n}{10} = 6$, $\frac{n}{6} = 10$ and $\frac{n}{3} = 20$ is shown in Figure 2 (b). In this case, more than 85% of the multicasts are successful, and, in the case of unsuccessful multicast, more than 80% of the nodes in a group receive the packet.

Figure 2 (a) shows that in small ad hoc networks OLAM is basically insensitive to both velocity and network load. The same is true for networks with up to 60 nodes with velocity up to 14 m/s (in this case more than 93% of the multicast are successful; see Figure 2 (b)).

In both cases (networks with 30 and 60 nodes) we have also compared our protocol with global flooding, the simplest multicast protocol that, as OLAM, does not assume the construction and maintenance of any underlying distributed data structure. As expected, simulations show that OLAM improves on the average delay of (successful) multicast completion up to 51% (see Table 1 below, where $|M|$ indicates the group size).
All the simulations run for a time long enough to achieve a confidence level of 95% with a precision within 5%.

V. Conclusions and Future Research

In this paper we have described how, using location awareness through GPS devices and efficient dissemination of GPS measures, an On-demand Location-Aware Multicast (OLAM) protocol can be designed for ad hoc networks. Different from previously proposed multicast protocols, OLAM does not assume, construct or maintain any distributed data structure, nor does it use an ad hoc routing or clustering protocol as a basis. Thus, node and network resources can be effectively used for the transmission of the multicast packets. The protocol is executed locally at each node, where a multicast tree is computed in polynomial time on the current network topology graph. The resulting tree is then optimally encoded and transmitted along with the packet with the same "space" overhead in the packet header as ad hoc source routing.

Simulation results show that our protocol is effective in delivering multicast packets in networks up to 60 nodes, with high network loads and regardless of the nodes' velocity and group sizes. The behavior of OLAM 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 for 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 [8]). We intend also 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.

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

References

(1) T. Ballardie, P. Francis, and 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. Also Computer Communication Review, vol.23, no.4, Oct. 1993*, pages 85–95, San Francisco, CA, 13–17 September 1993.

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

(3) C.-C. Chiang, M. Gerla, and 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.

(4) J. J. Garcia-Luna-Aceves and E. L. Madruga. The core-assisted mesh protocol. *IEEE Journal on Selected Areas In Communications, Special Issue on Wireless Ad Hoc Networks*, 17(8):1380–1394, August 1999.

(5) C.-C. Chiang and M. Gerla. On-demand multicast in mobile wireless networks. In *Proceedings of IEEE ICNP'98*, Austin, TX, 1998.

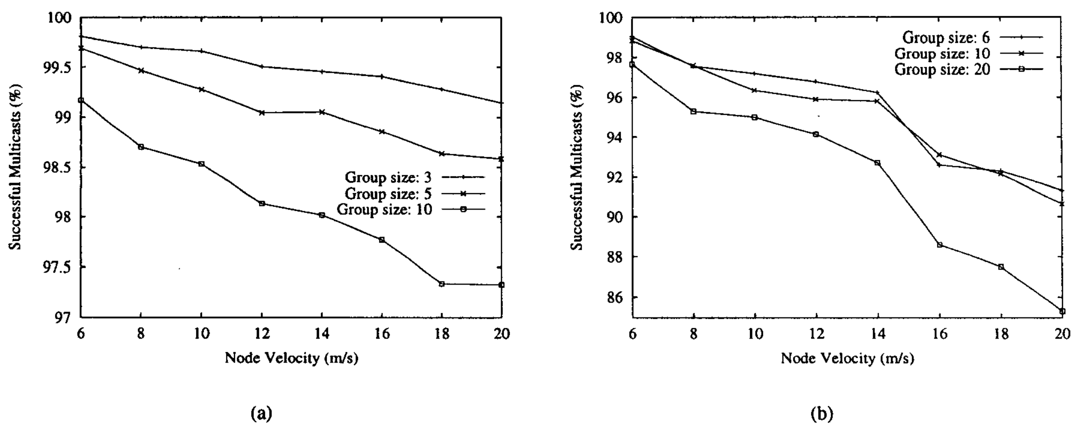

Figure 2: Average percentage of successful multicasts in networks with (a) 30 nodes and (b) 60 nodes.

Node Velocity (m/s):	6	8	10	12	14	16	18	20
$n = 30, \|M\| = 3$	51	51	51	50	49	50	50	51
$n = 30, \|M\| = 5$	45	47	46	45	42	45	44	44
$n = 30, \|M\| = 10$	36	33	35	33	29	25	24	25
$n = 60, \|M\| = 6$	26	24	25	26	25	25	26	25
$n = 60, \|M\| = 10$	23	21	22	21	24	24	24	24
$n = 60, \|M\| = 20$	20	18	19	18	20	18	20	20

Table 1: OLAM delay improvement with respect to flooding (%).

(6) E. M. Royer and 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, 15–20 August 1999.

(7) C.-C. Chiang and 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*, volume 2, pages 557–561, San Diego, CA, 12–16 October 1997.

(8) S. Basagni, I. Chlamtac, V. R. Syrotiuk, and 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*, pages 76–84, Dallas, TX, October 25–30 1998.

(9) S. Basagni, I. Chlamtac, and V. R. Syrotiuk. Geographic messaging in wireless ad hoc networks. In *Proceedings of the IEEE* 49*th Annual International Vehicular Technology Conference*, volume 3, pages 1957–1961, Houston, TX, May 16–20 1999.

(10) E. M. Palmer. *Graphical Evolutions: An Introduction to the Theory of Random Graphs*. Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons, New York, 1985.

(11) D. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In T. Imielinski and H. F. Korth, editors, *Mobile Computing*, chapter 5, pages 153–181. Kluwer Academic Publishers, Dordrecht, The Netherlands, February 1996.

(12) F. K. Hwang, D. S. Richards, and P. Winter. *The Steiner Tree Problem*. Number 53 in Annals of Discrete Matematics. North-Holland, The Netherlands, 1992.

(13) K. Makki, N Pissinou, and O. Frieder. Efficient solutions to multicast routing in communication networks. *Mobile Networks and Applications (MONET)*, 1(2):221–232, 1996.

(14) H. Takahashi and A. Matsuyama. An approximate solution for the steiner problem in graphs. *Math. Jap.*, 24:573–577, 1980.