Route Selection in Mobile Multimedia Ad Hoc Networks

Stefano Basagni Imrich Chlamtac András Faragó 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
e-mail: {basagni,chlamtac,farago,syrotiuk,rodeen}@utdallas.edu

Abstract–**The problem of routing packets in ad hoc networks is complex due to the lack of network infrastructure, shifting the reliance of packet forwarding onto the nodes of the network, all of which are mobile. In order to improve the robustness and adaptation of protocols for ad hoc networks to node mobility, in this paper we propose a new metric that bases route selection on the probability of the existence of the route (route availability). Specifically, we derive a simple closed form expression for the computation of the availability of a route, that takes into account node mobility and dependencies between links, which can then be used to satisfy the requirements of multimedia applications. In particular, we show how by efficiently collecting measures about the geographic location of other nodes, each node of an ad hoc network can locally and efficiently compute multiple paths to a given destination node, and, based on the introduced metric, choose the route that best meets the strict requirements of multimedia applications.**

I. INTRODUCTION

Mobile multi-hop wireless radio networks, also called *ad hoc* networks, are networks that have no fixed infrastructure such as underground cabling or base stations. While the primary use of ad hoc networks is in military applications (tactical networks), these networks are being considered to extend the range of commercial wired and cellular networks in the event of network failure or unavailability (e.g., emergency response, search and rescue) since they can be deployed quickly and are self configuring.

Since *all* the nodes of an ad hoc network are mobile, when a node routes a packet to another node not directly in its radio transmission range, it must rely on the other nodes of the network to cooperatively forward the packet to the destination. The highly dynamic network topology that arises from node mobility makes the selection of a route to an intended destination an especially challenging problem.

Most of the existing protocols for ad hoc networks—from routing protocols to the more recent solutions for multipoint communication (multicast and broadcast)—are concerned with the problem of *finding* the route(s) to the destination(s). This problem is dealt with by either exchanging route information while the topology changes (as in *proactive* or *table-driven* solutions) or by "discovering" routes to the destination node(s) whenever it is needed (*reactive* or *on-demand* protocols) or by maintaining network structures (i.e., a multicast tree or a multicast mesh, or a clustering of the ad hoc network) to support a given protocol. (A comprehensive account of ad hoc routing is given in [1]. Multicast protocols are presented, e.g., in [2], [3] and [4] where further references can be found.) Other solutions have been proposed that assume the use at each node of positioning system devices such as Global Positioning System (GPS) receivers [5], [6], [7], [8], [9]. In this case, the problem of finding a route is dealt with by using the knowledge at each node of the geographic location of the other nodes. The packet is then either forwarded along routes in the *direction* of the destination [5], [6], [7], [9], or a specific route to the destination is locally computed and the packet is forwarded along that route [8].

A common characteristic of these existing protocols is that whatever the strategy followed to find a route, there is no metric associated with the *selection* of the route itself that can provide either information about the actual *availability* of the route as a packet is going to follow it, or any information that can help in the selection of the "most suitable" route for the demands of a specific application prior to the packet transmission.

In this paper we introduce a new metric that can be used for route selection *before* the actual forwarding of the packet along that route takes place. In order to define our metric, we make use of a dissemination mechanism that distributes GPS location information and other nodes attributes (such as transmission radius and velocity) throughout the network allowing nodes to become aware of their current location and of the location of all the other nodes at a given time (see, e.g., [6] and [7]). Based on the location information available locally, when a source node has to transmit a packet to a given destination (or more, depending if a routing protocol is considered, or one-to-many communication), it will compute one or more routes to that destination and then it will associate with each route a value that bounds the probability that the route exists for the duration that it is traversed by the packet (*route availability*). Based on this probability, any source node can then select the route that best serves the purposes of a specific application.

The local computation of the route availability is based on a simple closed form expression introduced in this paper. We start by determining the probability that a wireless radio link exists between any two mobile nodes in a computed route, i.e., the probability that the link between those two nodes is *active*. Using this probability the source node can then compute the probability that a route exists to any given destination—the route availability.

Overall, the proposed metric has the following desirable

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

0-7803-5904-6/99/$10.00 © 1999 IEEE

characteristics:

- It is based on real geographic measures obtained by using GPS devices and disseminated to the nodes through an efficient and mobility adaptive mechanism. We do not assume any abstract network model, and the obtained measures are immediately available to efficiently retrieve a "snapshot" of the network topology.
- It is easily locally computable, being based on standard geometric-based formulas for which basic library functions can be used.
- It takes into account both the mobility of the nodes, and the natural dependence (of the probability of existence) of adjacent links.
- It increases route stability, since node movement is factored into the computation thereby reducing the chances of rerouting.

In this paper we illustrate how, by using the snapshot of the network topology (the *network topology graph*), a source node can locally compute a set of routes to an intended destination. We use, as an example, a well known and efficient algorithm for finding the k shortest paths between the source and one or multiple destinations [10]. These paths—which correspond to routes through the network—can now each be evaluated for route availability. Once again, this computation is performed locally at the source node. The source node can then use the route availability results as part of its route selection criteria to match the needs of the specific application traffic.

The rest of this paper is organized as follows. Section II reviews how nodes in the network become location aware through the dissemination of GPS location packets. Section III describes the local computation that takes place at a source node when a packet has to be sent to a given destination. More precisely, we start by showing how a graph that represents the network topology can be constructed from the source node location awareness (i.e., the collection of GPS attributes that it maintains locally). We then briefly describe the main characteristics of the k shortest paths algorithm that is used to compute routes to the intended destination node. We finally proceed by introducing our new metric for route availability and describe the simple mechanism for route selection based on the new metric. Conclusions and suggestions for further work can be found in Section IV.

II. NETWORK LEVEL LOCATION AWARENESS

A central assumption of our research is that each node is aware of its own geographic location. This can be easily obtained by equipping a node with a Global Positioning System (GPS) receiver, a device which is nowadays reliable and commercially available. Such devices allow 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 *attributes* in the sequel) with a precision to within a few hundred meters for position and 340 nanoseconds for time. (A nice account of GPS features can be found by visiting the website [11].)

Since ad hoc networks lack a fixed infrastructure, every node in the network is responsible for disseminating its GPS attributes to all other nodes. This is obtained by flooding the network with a packet containing the GPS attributes 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 attributes. 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 attributes. A node's latitude and longitude require no more than 16 bytes, the time these attributes 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 attributes are very small, requiring very little of a node's available bandwidth and associated energy to transmit.

A second way to 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 packet, we note that the frequency a node needs to disseminate its GPS packet 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 presented and applied in [6], [7] and [8], respectively.

III. ROUTE SELECTION IN MOBILE MULTIMEDIA NETWORKS

In this section we introduce the new metric for route selection. In order to illustrate the use of the new metric, here we describe in detail the computation that takes place at a source node S when a packet has to be sent to a given destination node D. In particular:

A. We describe how to obtain a "snapshot" of the network topology from the GPS attributes that S maintains locally. This *network topology graph* represents an approximation of the current network topology at the source node.

B. Given the network topology graph, S locally computes on that graph up to k routes from S to D. Here we describe the use of a well known and computationally efficient algorithm that finds the k shortest paths in the graph, i.e, the k shortest routes in the network from S to D.

C. We finally define how the availability of each link in a route can be computed, i.e., how to compute the probability that a link between two nodes in any of the routes between S and D is still active (link availability metric). This probability is then used to compute the availability of an entire route from S to D, based on which S can select the route(s) along which to send the packet that best meet the requirements of the application generating the packets.

A. Obtaining Network Topology from GPS Attributes

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 attributes 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, $|E| = m$. A link e in E between two nodes A and B in V means that, according to the attributes 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, 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 $lat(\cdot)$, $lon(\cdot)$ and $tx(\cdot)$ we indicate the entries for the latitude, the longitude and the transmission radius of a node, and $time(\cdot)$ records the time the measure was taken at its source node, respectively.)

In the *network topology graph G* there is a link between A and B if and only if:

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

where the *distance* function $dist(\cdot, \cdot)$ depends on $lat(\cdot)$, $lon(\cdot)$ and possibly the GPS attribute for altitude if something other than a flat topography is considered. Therefore, there is a link 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 (1) is satisfied. Only $n(n-1)/2$ of these checks are needed, namely, the time complexity is $O(n^2)$.

The network topology graph G can also be constantly *maintained* at each node, and updated in an on-line fashion each time a new location packet is received. In this case, the time complexity of updating G is clearly linear in n.

B. Finding the k Shortest Paths

The k *shortest paths* problem in a graph $G = (V, E)$, where $|V| = n$, $|E| = m$ and weights ≥ 0 are associated with each link in E, is to list the k paths connecting a given source-destination pair in G with minimum total weight. This problem is a very well investigated problem. Here we use the recent result proposed by Eppstein in [10], where several further references as well as an historical account on this problem can be found.

The main characteristics of Eppstein's algorithm that make it suitable for being used in our approach are the following:

1. It produces the k shortest paths between any two nodes according to any non-negative weight function associated with the set E of links of G. Here we consider G unweighted (i.e., for each link e in E, its weight is 1), which implies that the shortest path refers to the "hop"-based distance between two nodes. Of course, generic weights can be considered that represent other link parameters, such as the average delay over a given link, the available bandwidth, etc.
2. Its time complexity is proven to be optimal. More precisely, an implicit representation of the k shortest paths is found in $\theta(m + n \log n + kn)$ time. From this representation, each of the k paths between the selected source-destination pair can be retrieved in constant time and its links can be listed in time that is proportional to the number of links in the path. In the same paper [10], the author also presents a basic-version of the optimal algorithm which is suitable for practical implementation while losing only a logarithmic factor in time complexity.
3. The same time complexity, $O(m + n \log n + kn)$, is needed to compute the k shortest paths between a given source node and *all* the other nodes. This may be useful for finding multicast and broadcast (spanning) trees, for implementing the corresponding protocols.
4. The k paths found are not necessarily disjoint. This is a desirable property, since a critical link, such as a *bridge* which would disconnect the network graph if removed, can be considered for all the paths in which it is needed.
5. It can be used in directed graphs. Although this is not the case in the present paper, where we consider bidirectional links, a directed graph would represent the topology of an ad hoc network with asymmetric links.

The main idea of Eppstein's solution is to extend the problem of finding the k shortest paths between any two nodes s and t to the problem of finding the k shortest paths from s to *any* other node in the network. One can then find paths from s to t by simply concatenating two paths, one from s to any node r and then a shortest path from r to t. The core idea is to build a binary heap for each node, listing the links that are not part of the shortest path tree routed at s and that can be reached

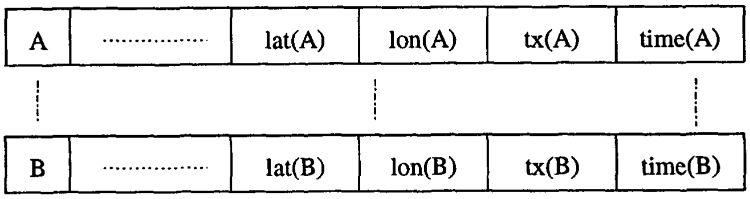

Fig. 1. The entries for nodes A and B in S's GPS cache.

from that node by shortest-path-tree links. In the basic version of the algorithm, suitable for efficient implementation, this collection of heaps form a bounded-degree graph with $O(m + n \log n)$ nodes. From this graph shortest paths to the final destination are then efficiently computed.

A detailed description of Eppstein's algorithm is beyond the scope of this paper, considering also that efficiency is there obtained by using techniques and data structures that are too lengthy and tedious to report here. Therefore, the interested reader is referred to the original paper [10].

C. Computing Route Existence Probability

Let $\mathcal{P} = n_1 n_2 \ldots n_h$ be a sequence of $h > 1$ nodes representing a path (route) from node n_1 (the source) to node n_h (the destination node). We assume that the route is known at the source node (we described in the previous section how a source node can locally compute up to k routes to the destination). In this section, we show how to compute the probability that all of the links $\ell_i = (n_i, n_{i+1})$, $i = 1, \ldots, h-1$, in the path exist (i.e., that a route from n_1 to n_h exists).

We start by observing that the probability for a link ℓ_i, $P(\ell_i)$, to be active is dependent on the probability of existence of links that share with ℓ_i an endpoint. For instance, the fact that a link exists between node n_i and node n_{i+1} due to the movement of n_{i+1} towards n_i, may imply that the link between node n_{i+1} and node n_{i+2} is now disrupted as a consequence of that movement, $i = 1, \ldots, h-2$. On the other hand, the probability of existence of links that do not share an endpoint are independent of each other. Therefore, the probability of existence of the whole path $\mathcal{P}$, which by definition is:

$$\begin{aligned} P(\mathcal{P}) &= P(\ell_1 \cap \ell_2 \cap \ldots \cap \ell_{h-1}) \\ &= P(\ell_1) \cdot \frac{P(\ell_1 \cap \ell_2)}{P(\ell_1)} \cdot \frac{P(\ell_1 \cap \ell_2 \cap \ell_3)}{P(\ell_1 \cap \ell_2)} \cdot \ldots \cdot \\ &\quad \frac{P(\ell_1 \cap \ell_2 \cap \ldots \cap \ell_{h-1})}{P(\ell_1 \cap \ell_2 \cap \ldots \cap \ell_{h-2})} \\ &= P(\ell_1) \cdot P(\ell_2 | \ell_1) \cdot P(\ell_3 | \ell_1 \cap \ell_2) \cdot \ldots \cdot \\ &\quad P(\ell_{h-1} | \ell_1 \cap \ell_2 \cap \ldots \cap \ell_{h-2}), \end{aligned}$$

here reduces to:

$$P(\ell_1) \cdot P(\ell_2 | \ell_1) \cdot P(\ell_3 | \ell_2) \cdot \ldots \cdot P(\ell_{h-1} | \ell_{h-2}). \qquad (2)$$

For each $i = 2, \ldots, h-1$, we compute

$$P(\ell_i | \ell_{i-1}) = \frac{P(\ell_{i-1} \cap \ell_i)}{P(\ell_{i-1})}$$

geometrically in two steps.

In the **first step**, we compute $P(\ell_{i-1})$ by intersecting two circles (Fig. 2). The first circle is centered at node n_{i-1} and has radius R_{i-1} defined to be the transmission radius of node n_{i-1}. We call this the *transmission circle* of n_{i-1}, and denote it by $C_{tx}(n_{i-1})$. (This is the larger circle in Fig. 2.)

To compute the probability for a link to be available, we consider node n_{i-1} to be stationary and centered at the origin $(0,0)$ of a Cartesian system such that node n_i is centered at $(d, 0)$, where d is the (e.g., Euclidean) distance between n_{i-1} and n_i (see also Section III-A). We then consider that node n_i can move in any direction at a maximum velocity of $v = v_{i-1} + v_i$ that takes into account the fact that the two nodes can be moving away from each other. Here, of course, v_{i-1} and v_i are the velocities of nodes n_{i-1} and n_i, respectively. (These values can be computed at the source node through the GPS attributes, or, as mentioned, they can be transmitted by each node along with the GPS information and stored in the local GPS cache.) Thus, node n_i can be found anywhere inside the circle whose radius R_i is easily computed as the product of v and the time elapsed between the time the last GPS packet was received at the source from n_i (*time*(n_i), see Section III-A) and the current time. We call this the *movement circle* of n_i relative to n_{i-1} and denote it by $C_{mv}(n_i)$.

The area A_{i-1} of the (usually) asymmetric "lens" (the shaded area in Fig. 2) in which the two circles $C_{tx}(n_{i-1})$ and $C_{mv}(n_i)$ intersect defines where node n_i must be in order for nodes n_{i-1} and n_i to be in the transmission range of each other given that each node is moving in a random direction at velocity v_{i-1} and v_i, respectively. Thus, the area of the lens divided by the area of the movement circle of n_i is a geometric approach to computing $P(\ell_{i-1})$, i.e.:

$$P(\ell_{i-1}) = \frac{A_{i-1}}{\pi R_i^2}. \qquad (3)$$

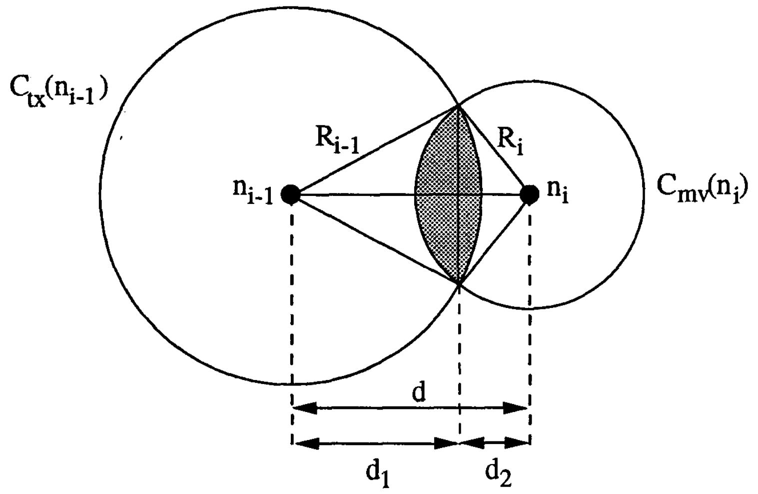

Fig. 2. The two circles, $C_{tx}(n_{i-1})$ and $C_{mv}(n_i)$, of radii R_{i-1} and R_i, respectively, intersect in a lens shaped region. This is the area in which node n_i must be in order to receive packets from node n_{i-1}.

To compute the area of the lens in which two circles intersect, we use the formula for the circular segment of radius r and triangular height t (basic definitions and proofs of the following and the next formulas can be found, e.g., in [12]. For a more in-depth treatment, the reader is also referred to [13]):

$$C(r,t) = r^2 \cos^{-1}\left(\frac{t}{r}\right) - t\sqrt{r^2 - t^2}$$

twice, once for each half of the lens. Noting that the heights of the two segment triangles, as shown in Fig. 2, are:

$$\begin{aligned} d_1 &= \frac{d^2 + R_{i-1}^2 - R_i^2}{2d} \text{ and} \\ d_2 &= \frac{d^2 + R_i^2 - R_{i-1}^2}{2d}, \end{aligned}$$

the resulting area A_{i-1} of the lens is:

$$\begin{aligned} A_{i-1} &= C(R_{i-1}, d_1) + C(R_i, d_2) \\ &= R_{i-1}^2 \cos^{-1}\left(\frac{d^2 + R_{i-1}^2 - {R_i}^2}{2dR_{i-1}}\right) + \\ &\quad R_i^2 \cos^{-1}\left(\frac{d^2 + R_i^2 - {R_{i-1}}^2}{2dR_i}\right) - \\ &\quad \frac{1}{2}\sqrt{(d - R_i - R_{i-1})(d + R_i - R_{i-1})} \cdot \\ &\quad \sqrt{(d - R_i + R_{i-1})(d + R_i + R_{i-1})}. \end{aligned}$$

We notice that the previous expression is a closed formula (all the values involved are known to the source node), and only uses standard library functions.

Now, in order to compute $P(\ell_i | \ell_{i-1})$, the **second step** requires us to compute $P(\ell_{i-1} \cap \ell_i)$. An exact computation would integrate over all points in the lens, intersecting the transmission circle of each point with the movement circle $C_{mv}(n_{i+1})$ of n_{i+1} relative to n_i. Since it would be computationally difficult to define the area of integration (the lens) and since the local computation of integrals would be performed by using iterative (and approximate) methods anyway, we perform this computation by choosing, randomly and uniformly, z points in the lens, and we then average over these z random samples, thus approximating the lens by some of its points. (The selection of the random points within the lens can be efficiently performed using standard techniques of computational geometry.) Of course, the more points z taken, the more accurate is $P(\ell_{i-1} \cap \ell_i)$.

Assume that node n_i has moved into the area of the lens, which is where it must be in order for the link $\ell_{i-1} = (n_{i-1}, n_i)$ to be active. Fig. 3 shows the transmission circle $C_{tx}(n_i)$ of a single point n_i, chosen randomly and uniformly within the lens (lightly shaded area) intersected with the movement circle $C_{mv}(n_{i+1})$ of n_{i+1} resulting in the darkly shaded area. As before, this computation reduces to the intersection of two circles where n_i is considered to be stationary at the origin of a Cartesian system with the position of node n_{i+1} translated to the x axis, and the velocity of n_{i+1} being v_{i+1} (this velocity is used to compute R_{i+1}).

Thus, by using (3) and averaging over the z random positions on the lens, we obtain:

$$P(\ell_i \cap \ell_{i-1}) = \frac{1}{\pi R_{i+1}^2} \frac{1}{z} \sum_{j=1}^{z} A_j,$$

where for each of the sample points j, A_j is computed as

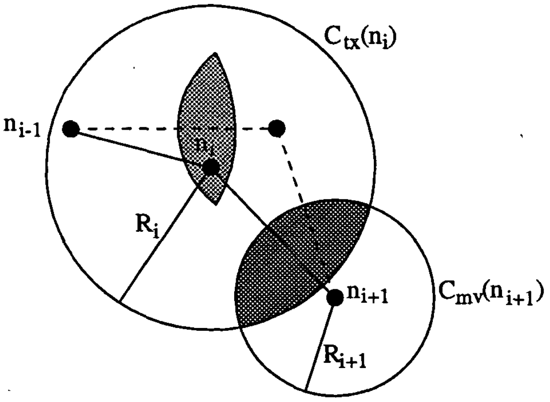

Fig. 3. The darker area indicates where node n_{i+1} must be for the link ℓ_i to exist when n_i is in the area (lens) that guarantees the existence of link ℓ_{i-1}.

illustrated above (first step). For each $i = 2, \ldots, h-1$ we finally have:

$$P(\ell_i|\ell_{i-1}) = \frac{P(\ell_i \cap \ell_{i-1})}{P(\ell_{i-1})} = \frac{R_i^2}{A_{i-1}R_{i+1}^2}\frac{1}{z}\sum_{j=1}^{z} A_j$$

based on which and by using (2) we can compute, for each given route $\mathcal{P}$, the corresponding route availability $P(\mathcal{P})$.

We notice that the described computation can be considerably simplified if the *direction* of the motion of a node is taken into account. The direction of any node A can be easily retrieved at each other node by comparing two successive GPS packets. Alternatively, node A can transmit its own direction along with the other GPS attributes. In any case, the method described above is completely general, i.e., it is independent of the (possibly changing) direction of a node.

In this section we have illustrated how a source node can compute routes to a given destination and how, based on simple information related to the location of the other nodes and their mobility, it can associate with each route a measure of its availability. Based on this measure, and depending on the specific application that requires the transmission of a packet, the source node is thus able to select the route that best meets the requirements of the given application. Once the route is selected, it is piggybacked as a sequence of nodes to the packet and the packet itself is then routed along the selected route (as in, e.g., [8] and [14]).

IV. CONCLUSIONS

This paper presented a new metric that allows nodes in an ad hoc network to select routes to given destinations based on the probability that the route is still available when a packet has to be sent through it. The introduced metric is based on the knowledge at each node of the geographic position of all the other nodes in the network and takes realistically into account both the mobility of the nodes and the dependencies between links in a computed route, thus being suitable for route selection when the needs of applications, such as real-time and multimedia applications, have to be met.

To illustrate the way in which any source node S can select a specific route depending on its availability, we have described here the basic steps that S has to perform in order to retrieve network topology information, to efficiently compute up to k routes to the intended destination and finally to associate with each route the probability if its existence at the time a packet is going to follow it.

The computation of the link and route availability is of course independent on how the route has been obtained by the source node. For instance, local computation can be performed at the source node that finds one or more broadcast or multicast trees instead of multiple simple routes to the destination nodes. The introduced metric can then be used to evaluate the availability of those routing structures before the actual sending of the packet.

Further research is going to be pursued in this direction, as well as in the direction of combining link/route availability with other measures related to each route, such as delay, bandwidth, local conditions at each node of the route, etc. Experimental tests have yet to be performed in order to demonstrate the effectiveness of the proposed metric with respect to variations in the most common parameters of ad hoc networks, such as, for instance, different mobility patterns, node velocity, the possibility of node faults, etc.

REFERENCES

[1] E. M. Royer and C.-K. Toh. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, 6(2):46–55, April 1999.

[2] 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.

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

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

[5] K. N. Amouris, S. Papavassiliou, and M. Li. A position-based multi-zone routing protocol for wide area mobile ad-hoc networks. In *Proceedings of the IEEE 49th Annual International Vehicular Technology Conference*, volume 2, pages 1365–1369, Houston, TX, May 16–20 1999.

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

[7] 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.

[8] S. Basagni, I. Chlamtac, and V. R. Syrotiuk. Dynamic source routing for ad hoc networks using the global positioning system. In *Proceedings of the IEEE Wireless Communications and Networking Conference 1999 (WCNC'99)*, New Orleans, LA, September 21–24 1999.

[9] Y.-B. Ko and N. H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 66–75, Dallas, TX, October 25–30 1998.

[10] D. Eppstein. Finding the k shortest paths. *SIAM Journal on Computing*, 28(2):652–673, February 1999.

[11] 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.

[12] E. W. Weisstein. *CRC Concise Encyclopedia of Mathematics*. CRC Press, Boca Raton, FL, 1998. (See also: http://www.treasure-troves.com/math/).

[13] D. Pedoe. *Circles: A Mathematical View*. Math. Ass. Amer., Washington, DC, revised edition, 1995.

[14] 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.