

# A Location-Based Routing Method for Mobile Ad Hoc Networks

Ljubica Blazevic, *Member, IEEE*, Jean-Yves Le Boudec, *Fellow, IEEE*, and Silvia Giordano, *Member, IEEE*

**Abstract**—Using location information to help routing is often proposed as a means to achieve scalability in large mobile ad hoc networks. However, location-based routing is difficult when there are holes in the network topology and nodes are mobile or frequently disconnected to save battery. Terminode routing, presented here, addresses these issues. It uses a combination of location-based routing (Terminode Remote Routing, TRR), used when the destination is far, and link state routing (Terminode Local Routing, TLR), used when the destination is close. TRR uses anchored paths, a list of geographic points (not nodes) used as loose source routing information. Anchored paths are discovered and managed by sources, using one of two low overhead protocols: Friend Assisted Path Discovery and Geographical Map-based Path Discovery. Our simulation results show that terminode routing performs well in networks of various sizes. In smaller networks, the performance is comparable to MANET routing protocols. In larger networks that are not uniformly populated with nodes, terminode routing outperforms existing location-based or MANET routing protocols.

**Index Terms**—Restricted random waypoint, mobility model, ad hoc network, scalable routing, location-based routing method, robustness to location inaccuracy.

## 1 INTRODUCTION

**Location-Based Routing for Scalability.** Many existing routing protocols (DSDV [30], WRP [27], OLSR [21], FSR [20], LANDMAR [12], DSR [8], AODV [29], TORA [28], CBRP [22]), proposed within the MANET [26] working group of IETF, are designed to scale in networks of a few hundred nodes. They rely on state concerning all links in the network or links on a route between a source and a destination. This may result in poor scaling properties in larger mobile ad hoc networks or when nodes frequently disconnect to save battery. More recently, there has been a growing focus on a class of routing algorithms that rely largely, or completely, on location (and possibly mobility) information. These algorithms improve network scalability by reducing the total routing overhead. The idea is to use location information in order to reduce propagation of control messages (LAR [37]), to control packet flooding (DREAM [2]), to reduce intermediate system functions or to make simplified packet forwarding decisions (GPSR [23], GFG [7], and GRA [33]).

**Issues with Existing Location Based Routing.** LAR is an on-demand routing protocol where location information is used to reduce the search space for a desired route. The source uses the last known destination location in order to estimate the zone in which the destination is expected to be found. This is used to determine a request zone, as a set of

nodes that should forward route requests. DREAM proactively maintains location information at each node in routing tables and data packets are partially flooded to nodes in the direction of the destination. GPSR [23], GFG [7], and GRA [33] use only neighbor location information for forwarding data packets. Routing is done in a greedy way by forwarding the packet to a neighbor closer to the physical location of the destination. This local optimal choice repeats at each intermediate node until the destination is reached. When the greedy process fails, GPSR and GFG route the packet around the problem region using *perimeter* mode packet forwarding. Perimeter mode forwards the packet using a planar graph traversal. The knowledge of locations of its one-hop neighbors is sufficient for a node to determine its local view of the planar graph. An issue with perimeter mode is that it may give a very bad path in large networks when the source and destination are not well connected along a straight line.

With GRA, when the greedy method fails, a distributed breadth-first or depth-first route discovery method is invoked to find an acyclic path to the destination. The problem with this method is that the discovery and maintenance of such paths can result in large overhead for large mobile ad hoc networks.

Further, in location-based routing protocols, sources should know destination locations accurately enough for packets to reach, or come close to their destination. However, it is very difficult for the location management service to maintain accurate location information at all times. This is especially true if nodes are close and their relative locations change frequently. Existing location-based routing protocols do not address how to cope with location management inaccuracies.

**Our Proposed Approach.** We present a routing protocol, called terminode routing, which aims at keeping the scalability benefits of location-based routing, while addressing the two issues of irregular topology and node mobility. It is shown in [15] that the per node capacity

- L. Blazevic is with the Broadband Wireless LAN Group, Advanced System Technology, STMicroelectronics, 39, ch. du Champ des Filles, CH-1228 Geneva, Switzerland. E-mail: ljubica.blazevic@st.com.
- J.-Y. Le Boudec is with the Institute for Communication Systems, EPFL, CH-1015, Lausanne-EPFL, Switzerland.  
E-mail: jean-yves.leboudec@epfl.ch.
- S. Giordano is with the University of Applied Science-SUPSI, Galleria 2-Via Cantonale, 6928 Manno-Switzerland.  
E-mail: silvia.giordano@die.supsi.ch.

Manuscript received 18 Sept. 2002; revised 23 May 2003; accepted 10 Feb. 2004; published online 27 Jan. 2005.

For information on obtaining reprints of this article, please send e-mail to: [tmc@computer.org](mailto:tmc@computer.org), and reference IEEECS Log Number 16-092002.

asymptotically tends to zero as the number of nodes goes to infinity. Thus, we should not expect to support networks of extremely large sizes. However, for networks of 500 to 1,000 nodes, we verified by simulations that we are not in the asymptotic regime of [15]. We also found that our routing method does perform better than the existing MANET and location-based routing protocols we compared it to. Nonuniform topologies are likely to appear in metropolitan areas with mountains or lakes, like the Lake of Geneva area.

Terminode routing uses the following ingredients to achieve its goal. First, it combines a location-based routing method with a link state-based mechanism. Second, it uses a special form of restricted search mode (Restricted Local Flooding, RLF). These first two ingredients solve problems due to the inaccuracy of location information, in particular for control packets. Third, it introduces the concept of anchors, which are geographical points imagined by sources for routing to specific destinations. This helps efficiently route around connectivity holes. An overview of terminode routing is given in Section 2, and a detailed description in Sections 3 and 4 in the form of protocol walkthrough.

We evaluated the performance of our protocol by detailed simulations and its scalability by analysis (Section 5). The results show that it performs well in networks of different sizes. In order for the comparison to be fair to MANET protocols, we implemented an ad hoc location management scheme. In smaller ad hoc networks, we compared terminode routing to some existing MANET-like routing protocols (AODV and LAR1) and found similar performance. In larger mobile ad hoc networks of 500 nodes, MANET-like routing protocols do not perform well (except when mobility is small), while our routing protocol still performs well. In networks that are regularly populated with nodes, terminode routing performs comparable to GPSR when the location management accuracy is high; however, terminode routing performs better when the location information accuracy is low. We also consider irregular networks with holes in node distribution. Here, too, we find that terminode routing outperforms GPSR. Finally, we show by simulation the effectiveness of TLR and RLF at solving location inaccuracies. In all cases, terminode routing is characterized by low routing overhead, even when we include the overhead of location management.

Reference [6] contains support material that could not find its place in this version.

#### List of Acronyms:

- EUI: End-system Unique Identifier (permanent address),
- FAPD: Friend Assisted Path Discovery,
- GMPD: Geographic Map-based Path Discovery,
- LDA: Location Dependent Address (geographical coordinates)
- RLF: Restricted Local Flooding,
- TLR: Terminode Local Routing, and
- TRR: Terminode Remote Routing.

## 2 OVERVIEW OF TERMINODE ROUTING

### 2.1 Combination of Local and Remote Routing

Terminode routing uses a combination of location-based routing (Terminode Remote Routing, TRR), used when the destination is far, and link state routing (Terminode Local

Routing, TLR), used when the destination is close. TLR uses location independent addresses only. TRR uses a combination of direct paths, perimeter mode, and anchors, as described in the rest of this section.

A **direct path** is an approximation of the straight line, and is built as follows: Assume that the source  $S$  knows an approximate location of the destination  $D$  (see Section 2.5 for details).  $S$  sends the packet to a neighbor that brings the packet closer to the assumed location of  $D$ , and this is repeated by intermediate nodes, as long as it works. Fig. 1a shows an example of where the direct path works well.

**TLR:** When a packet has arrived up to two hops away from the destination, a link state approach is used, which does not use location. In Fig. 1a, some intermediate node on the direct path finds that  $D$  is one or two hops away, using its TLR reachability information (which is based on permanent addresses, not location).

The combination of TLR and TRR is able to keep the scalability benefits of location-based routing, while avoiding problems due to mobility. However, combining TLR and TRR in one protocol poses a number of design challenges (in particular, avoiding loops), which we solved by using the mechanisms described in Section 3.

**Perimeter Mode:** Fig. 1b shows a case where the direct path does not work well: The packet may be “stuck” at a node that does not have a neighbor closer to the destination than self. Here, TRR uses **perimeter** mode to circumvent the topology hole, similar to GFG [7] and GPSR [23]. Perimeter mode consists of turning around the obstacle. This is achieved by planar graph traversal [23]. This goes on until a node is found that reduces the distance to the destination, from whereon the packet is forwarded using a direct path, as in the previous case.

Perimeter mode may give very long suboptimal paths. Furthermore, it can cause frequent routing loops in mobile ad hoc networks [5]. Thus, we restrict the use of perimeter mode to discovery phases, when a better mode is not available to the source.

### 2.2 Anchored Paths in TRR

In order to avoid perimeter mode, we introduce the concept of anchors, which are imaginary locations used to assist in routing. In Fig. 1c, source  $S$  uses three anchors to route the packet to  $D$ . The anchors are geographical locations, not nodes. The list of anchors is written by the source into the packet header, similar to IP loose source routing information. The packet is sent by intermediate nodes in the direction of the next anchor in the list until it reaches a node close to an anchor, at which point the next anchor becomes the following in the list. The location of the final destination takes the role of the last anchor. TLR is used when the packet comes close to the final destination, as previously shown.

The use of well-chosen anchors greatly reduces the number of hops taken by the packet compared to perimeter forwarding, for two reasons. First, anchors may lead to an overall better routing strategy. Second, even when perimeter and anchored paths use similar directions, perimeter mode paths tend to be more contorted and use more hops (because they are constantly trying to escape the problem area).

We include a method for a source to detect whether anchors should be used. It is based on a novel method for the source to find the distribution of the number of hops along the direct (nonanchored) path. The source sends some packets using TRR without anchors and receives the feedback about the number of hops it took the packet to reach the destination. The source decides that anchors are

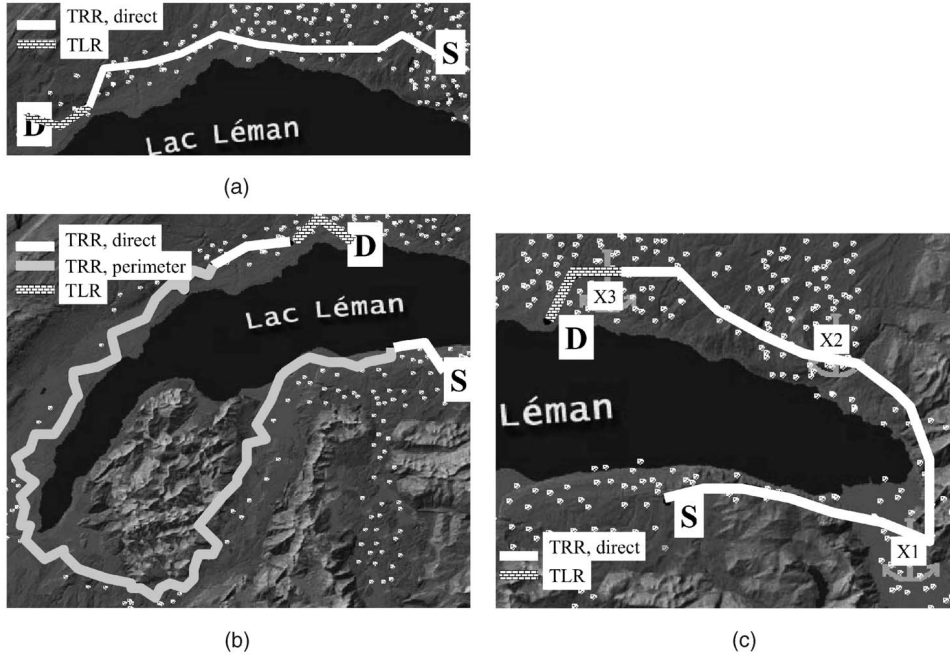


Fig. 1. (a) Packet forwarding from S to D with TRR and TLR along a direct path, no anchors (Section 2.1). (b) Direct path does not work, perimeter mode is used instead (Section 2.1). (c) Direct path does not work, anchors  $X1$  to  $X3$  are used, thus avoiding perimeter mode (Section 2.2).

needed if the packet path is significantly longer than estimated from the distribution of the number of hops along the greedy path.

### 2.3 Computing Anchors

Anchored paths, however, come at the price of computing good anchors. We propose two methods. They are always implemented at sources:

- Friend Assisted Path Discovery (FAPD, Section 4.1) assumes that some nodes (FAPD responders) are able to provide assistance to others, typically because they have a stable view of the network density. FAPD responders help find anchors, but are not used in the data path.
- Geographical Map-based Path Discovery (GMPD, Section 4.2) assumes that network density maps are available to a source node. This is for an ad hoc network where all nodes are individually mobile, but the node density can still be predicted—a common assumption for car networks. We find that GMPD performs better, but requires the overhead of map distribution; methods for distribution of density maps are left outside the scope of this paper.

### 2.4 Restricted Local Flooding

We account for situations where the accuracy of location management is low and TLR alone is not sufficient to cope with it. Our novel method, called Restricted Local Flooding (RLF), sends four to six packet duplicates in the region where the destination is expected to be, thus increasing the probability of reaching the destination. RLF recovers from location inaccuracies when the destination is within several transmission ranges from the node that starts RLF. In large networks, sending duplicates always has considerably less overhead than flooding. RLF is used for two types of discoveries: 1) search a limited area for a given node (Section 3.5) or for a node type (FAPD

responder, Section 4.1.2) and 2) establish long distance relations.

### 2.5 Assumptions on Addressing and Location Services

Terminode routing assumes that each node has a permanent address or End-system Unique Identifier (EUI) and a temporary, location information called Location Dependent Address (LDA). The LDA is a triplet of geographic coordinates (longitude, latitude, altitude) obtained, for example, by means of the Global Positioning System (GPS) or, if GPS is not available (e.g., indoors), the GPS-free positioning methods ([9], [32], [17]) can be used. We assume that there exists a location management that enables nodes in the network to determine approximate locations of other nodes.

We envision that location management in a large ad hoc network is performed by a combination of the following functions. First, a location tracking algorithm is assumed to exist between nodes when they have successfully established communication; this allows communicating nodes to continuously update their correspondent LDAs. Second, a location discovery service is used at the source to obtain a probable location of the destination  $D$  ( $LDA_D$ ) that  $S$  is not tracking by the previous method. In Section 5, we present the location management scheme that we used in simulations to evaluate the performance of terminode routing. Some other proposals are presented in [24], [35].

A Java applet implementation of terminode routing, as well as the simulation code used in this paper, are available in [1].

## 3 PROTOCOL WALK THROUGH (without ANCHOR PATH DISCOVERY)

In this section, we present terminode routing by describing, in detail, the operations invoked in typical phases at source

and intermediate nodes. For readability, the actions related to anchor path discovery are gathered in Section 4.

### 3.1 Bootstrapping

When a node boots, it initiates the local routing method TLR by sending a broadcast HELLO message with its address (EUI) and location (LDA). It also starts listening for other nodes' HELLO messages, which it uses to build its local routing table. The node then periodically broadcasts HELLO messages, which, in addition to this node's address and location, contain the addresses (not the location) of this node's neighbors.

The node keeps in its routing table (called "TLR table") the addresses and locations of its immediate neighbors, as well as the addresses of its two-hop neighbors. The former is used by TLR packet forwarding, and the latter by TRR packet forwarding.

Each entry in the routing table has an associated holding time. If a node does not hear from an immediate neighbor for some amount of time, it removes from the routing table the entry that corresponds to the lost neighbor, as well as all two-hop distant terminodes that were reachable via the lost neighbor.

Thus, TLR contains a link-state routing component, limited in scope to nodes that are two hops away. A similar approach is used by the intrazone routing protocol (IARP) in ZRP [31]. The TLR table is used to compute the local view of the Gabriel graph, using the algorithm in [23].

In addition, the node determines by its local configuration whether

- it possesses density maps. If so, it will be able to use them to compute anchors with GMPD (Section 4.2)
- it is a FAPD responder. If so, it will provide friend assistance to other nodes. The operation for that case is described in Section 4.1.

### 3.2 Source Node has Packet to Send, Destination in TLR Table

When source  $S$  gets a packet from an application to forward to destination  $D$ , it first checks whether destination  $D$  is in its TLR table. If so, the "Use TLR" bit in the packet header is set to 1. From now on, the only mechanism used to forward this packet is TLR.

The next-hop is determined from the TLR routing table. If the table says that  $D$  is two-hops away and several next hops are possible, we choose the one-hop neighbor whose entry is updated most recently.

If the "Use TLR" in the packet to forward was already equal to 1, the packet should be sent directly to the destination, which should be a one-hop neighbors. If this is not possible, the packet is dropped. This ensures that TLR is loop-free.

### 3.3 Source Node Has Packet to Send, Destination Not in TLR Table

#### 3.3.1 Obtain Location of Destination

Node  $S$  determines whether a valid location of  $D$  is known. Immediately after booting, it is likely that this answer is no.  $S$  then uses a location discovery service, as mentioned in Section 2 to obtain  $LDA_D$ . The packet is buffered until  $S$  obtains this information, at which point  $S$  puts the location information  $LDA_D$  in the packet header and sends the packet using TRR without anchors.

The location of  $D$  is then kept in a cache and it is updated by a tracking protocol (see Section 2). A cache entry has two timers: When the former expires, the entry is valid but old; if so,  $S$  sets the "Use RLF" bit in the packet, thus telling intermediate nodes that the more sophisticated RLF method should be used when the packet comes close to the assumed location of  $D$  (Section 3.5). When the latter timer expires, the cache entry is considered stale and is removed.

#### 3.3.2 Send Packet Using TRR without Anchors

The source sends the packet to an immediate neighbor that best improves the distance to  $D$ . The information about such a neighbor is obtained from the TLR table. If no such neighbor exists according to the table, perimeter mode is used instead; the packet is sent to the immediate neighbor computed by the Gabriel graph algorithm, as in [23].

#### 3.3.3 Start Path Evaluation

The source obtains feedback from the destination about the number of hops it takes to reach the destination along the path without anchors. Based on this information and the method presented in the Appendix of [6], the source estimates if the path without anchors works well, or if an anchored path should be tried.

#### 3.3.4 Look for Anchored Path

Assume the source  $S$  estimates that an anchored path should be tried. If it owns density maps, the path without anchors does not perform well, it uses GMPD to compute an anchored path (Section 4.2). Otherwise,  $S$  starts FAPD (Section 4.1).

The anchored path is put in a cache by the source. Similar to the location cache, when an anchored path becomes old, a new anchored path is searched for, as above. When it becomes stale, it is removed. A source may prematurely age out an anchored path if it evaluates that it performs badly (e.g., the destination reports low packet delivery).

#### 3.3.5 Send Packet Using TRR with Anchors

When an anchored path is available in the cache, the source node appends to the packet header the anchored path, sets the "Next location" pointer in the packet header to the first anchor, and sends the packet in direction of the first anchor of the anchored path.

### 3.4 Intermediate Node Has Packet to Forward

When a node receives a packet, it checks if the destination address (EUI) is self; if yes, the packet is delivered internally, else, it is further forwarded.

#### 3.4.1 Forward Using TLR

Then, this node first checks whether the destination is in the TLR table or the packet's "Use TLR" bit is set. If either is true, the operation is the same as if this node were the source (Section 3.2).

In particular, once this node determines that TLR can be used, it sets the "Use TLR" bit. This has the effect that a packet can never revert from TLR forwarding to TRR; this is to avoid loops due to mobility.

#### 3.4.2 Decide Whether to Expedite TRR Termination

Else, the packet necessarily was always forwarded so far with TRR, and this node determines whether TRR termination should be expedited. Indeed, if the accuracy

of location management is not sufficient, or if the packet has been delayed (due to congestion or bad paths), the “Use TLR” bit may never be set. Then, the packet may start circulating around  $LDA_D$ : It is forwarded via nodes that are close to  $LDA_D$ , but the packet does not reach the destination because  $D$  has moved considerably from  $LDA_D$  and no node in vicinity of  $LDA_D$  contains anymore  $D$  in their TLR-reachable area. Finally, the packet is dropped due to expiration of the time-to-live field (TTL).

Our approach is to avoid such cases, by expediting the termination of TRR. The condition for this is

- The distance between this node’s location and the destination location ( $LDA_D$ ) written by the source in the packet is less than the transmission range. Since the destination is not in the TLR table, this is a sign that  $LDA_D$  is not accurate.
- The RLF bit is not equal to 1 (see Section 3.5).

If the condition is met, the node terminates TRR as described in Section 3.5.

### 3.4.3 Forward with TRR, No Anchored Path in Packet Header

Else, if the condition to expedite termination of TRR is not satisfied, the intermediate node proceeds with TRR. If no anchored path is present in the packet header, the packet is sent in the direction of the destination’s location and read from the packet header. The operation is the same as in Section 3.3.2.

### 3.4.4 Forward with TRR, Anchored Path Present in Packet Header

This node updates the “Next location” pointer in the packet header, by finding out whether the “Next location” in the packet header falls within its transmission range. If so, it sets the “Next location” pointer to the following anchor, or it was the last, to the location of the final destination.

Then, this nodes sends the packet towards the updated next location, as in Section 3.3.2, but with the destination’s location replaced by “Next location.”

See also Fig. 1. If the anchors are correctly set, then there is a high probability that the packet will arrive at the destination. We can also imagine situations when an anchored path is not correctly set. Then, it may happen that there is no direct greedy path from one anchor to the next, in which case the packet may be forwarded in perimeter mode.

## 3.5 Intermediate Node Expedites Termination of TRR

This section is applied once the condition in Section 3.4.2 is satisfied. The action depends on the “Use RLF” bit in the packet header. This bit is normally set to 0; it is set to 1 by a source that suspects that its destination’s location is not accurate (Section 3.3); it may also be set to 1 when a node is in search of a FAPD responder (Section 4.1.2).

### 3.5.1 “Use RLF” Is Not Set in Packet Header

Before forwarding the packet, this node sets the TTL to  $\min(\text{term\_trr}, \text{TTL})$ . This has the effect of limiting a loop due to destination location inaccuracy to  $\text{term\_trr}$  hops ( $\text{term\_trr} = 3$  in our current implementation).

### 3.5.2 “Use RLF” Is Set in Packet Header

In this case, “Restricted Local Flooding (RLF)” is used. It consists of sending six duplicates of the packet in different directions around the sending node ( $X$ ). In this way, packets are sent in the area around  $X$ , where the destination is expected to be. All packets have the same destination address equal to the one of  $D$ , however, they have different destination location information.

This form of local flooding is restricted in that it does not use broadcasting like common flooding and because duplicate packets are dropped after a certain number of hops if not arrived at the destination. If instead of RLF the common flooding was used, then it would be necessary to control the flooding on a per packet basis. In order to avoid the redundant transmissions of the same packet, it would be necessary that intermediate nodes keep track of the packets that they have already seen. All this is not needed in the case of RLF because packet duplicates are forwarded in the same way as all other packets.

Within each duplicate,  $X$  sets the “RLF” bit in the packet header to 1 (this is not the same as the “Use RLF” bit). This will prevent cascaded uses of RLF.  $X$  sends each duplicate in the direction of one of the six geographic points around  $X$ . Let  $X_i, i \in 1..6$  be these geographic locations. Within the  $i$ th packet, the destination LDA in the packet header is set to  $X_i$ . However, the destination EUI field is not changed (i.e.,  $EUI_D$ ).  $X_1$  through  $X_6$  thus present virtual destination locations. All points  $X_1$  to  $X_6$  are at the same distance from  $X$ , which is equal to twice the transmission range of  $X$ . It follows that we roughly cover the region equal to twice the transmission range. If the destination is within this region, it is very probable that it receives at least one duplicate of the packet.

The TTL field in each duplicate is set to  $\text{term\_rlf}$  (equal to 4 in our implementation). In this way, we constrain the lifetime of a duplicate to  $\text{term\_rlf}$  hops. Packets with the RLF bit equal to 1 forwarded towards one of geographic locations  $X_i$ . There are three possible situations for each of the six duplicates:

1. The packet is delivered to the destination by some intermediate node that finds  $D$  in its TLR routing table.
2. The packet is dropped due to TTL expiration (this happens at most at the  $\text{term\_rlf} = 4$ th hop).
3. Some intermediate node  $N$ , finds  $X_i$  ( $X_i$  is in the destination LDA field in the packet header) in its transmission range, but the destination is not in the TLR table. Then,  $N$  drops the packet. In this way, we prevent further duplications.

## 4 PROTOCOL WALK THROUGH: ANCHORED PATH DISCOVERY

Anchored path discovery is triggered by a source node when it estimates that a nonanchored path does not perform well or the current anchored path becomes stale. There are two methods for anchored path discovery: Friend Assisted Path Discovery (FAPD) and Geographic Map-based Path Discovery (GMPD).

## 4.1 Friend Assisted Path Discovery (FAPD)

FAPD uses nodes, called *FAPD responders*, which provide assistance to other nodes to discover anchored paths. We assume that some percentage of nodes in the network are configured to act as FAPD responders. FAPD responders maintain “friendship” connections to a number of other FAPD responders in the network. When a responder receives from some source node a request to assist in anchored path discovery, and it does not know a path to the destination, it contacts its friend FAPD responders. Several FAPD responders can participate in an anchored path discovery. We present the main FAPD operations invoked in typical phases at source and FAPD responders.

### 4.1.1 FAPD Responders Discover Friend FAPD Responders

This operation is launched at boot time by a node, say  $T$ , that is configured to be a FAPD responder, and periodically later. Node  $T$  uses RLF, as described in Section 3.5, but with four duplicates instead of six. Each of the four packets contains a *get\_friends\_request* message, has the RLF bit set to 1, destination *EUI* to *any* (as  $T$  does not know the identity of FAPD responders), the *TTL* field set to 6, and destination *LDA* to one of four geographic points (FP1, FP2, FP3, and FP4). No anchors are used. Although we use here RLF as in Section 3.5, the goal is different: We want to establish some long distance friendships, whereas in Section 3.5 we wanted to search a limited area for a given node. The four points FP1 to FP4 are selected in orthogonal directions at four times the transmission range of  $T$ . Once some FAPD responder, let's say  $Y$ , on the way towards a point  $FP_i$  receives the *friends\_request* message, it does not forward it. Then,  $Y$  sends back a *friends\_reply* message to  $T$ , which contains a list a friends, selected from  $Y$ 's own list of friends plus  $Y$  itself. When node  $T$  eventually receives the *friends\_reply* message from the node  $Y$ , it combines the received information with the current one in its list of friends.

In [3], we presented how a node selects a number of friends from a list of potential friends. The key to generate the small-world phenomenon is the presence of a *small fraction* of long-range edges, which connect otherwise distant parts of the graph, while most edges remain *local*, thus contributing to the high clustering property of the graph. Our strategy is to consider geographic locations of nodes when building friends connections.

Once a FAPD responder has selected its friends, each friend is associated with the following information: friend's identity, location, and path to a friend. Friends locations are tracked and path to friends are evaluated. A friend is declared stale if a node does not hear from a friend for some time. If the number of friends is considered small, a FAPD responder may start friends discovery procedure again. The interested reader may find more information on friends maintenance in [5].

### 4.1.2 Source Starts Anchored Path Discovery

Assume a node  $S$  looks for an anchored path to destination  $D$ . We assume that  $S$  obtained location of  $D$  ( $LDA_D$ ).

- If  $S$  is itself a FAPD responder and has a list of friends, it requests assistance from some friend in providing an anchored path to destination  $D$ :  $S$  selects a friend, say  $F1$ , that brings the packet closer to  $D$ , if any exists

(else see next item).  $S$  then sends a control packet called *anchored path request* to  $F1$ .  $S$  uses the existing path that  $S$  maintains to  $F1$ . The control packet contains a *fapd\_anchored\_path* field, which will accumulate a path to  $D$ . If  $S$  has an anchored path to  $F1$ ,  $S$  simply initializes *fapd\_anchored\_path* to this path (and  $S$  sends the packet to  $F1$  using TRR with anchors).  $S$  stamps the anchored path request with a sequence number. Also, it sets *tabu\_index* to 0 in the anchored path request (see Section 4.1.4). On receiving the packet,  $F1$  performs the actions in Section 4.1.3.

- If  $S$  is itself a FAPD responder and has a list of friends, but none is closer to  $D$ ,  $S$  starts a FAPD search in *tabu* mode, as described in Section 4.1.4.
- else (i.e.  $S$  is not a FAPD responder)  $S$  sends several *anchored path request* packets in the geographical region around self. For this purpose,  $S$  uses the RLF method, as described in Section 4.1.1. Each of the four packet duplicates is thus sent in anycast mode (*EUI* is set to a predefined value meaning “Any FAPD Responder”), in a region up to four transmission ranges around  $S$ , and has *tabu\_index* = 0. Any node, say  $F1$ , that receives the request packet from  $S$ , and that itself maintains a list of friends, performs the actions in Sections 4.1.3. If several FAPD responders receive a path request packet from  $S$ ,  $S$  may learn several anchored paths to  $D$ . On the contrary, if no path request packets reach a FAPD responder,  $S$  does not get any anchored path to  $D$ .

### 4.1.3 FAPD Responder Receives Anchored Path Request

When a FAPD responder, say  $T$ , receives an anchored path request packet, it appends its geographic location to *fapd\_anchored\_path*.

- If  $T$  has an anchored path to  $D$ , or to some location close to  $LDA_D$  (in our implementation, the distance to  $LDA_D$  should be less than two times the transmission range)  $T$  appends this path to *fapd\_anchored\_path* and sends the packet to  $D$  using TRR with anchors.
- Else, if  $T$  does not have its list of friends,  $T$  sends the packet directly to the destination, using TRR without anchors.
- Else, if the *tabu\_index* field is 0,  $T$  acts as if it were the source of the anchored path request (Section 4.1.2). If *tabu\_index* field is non 0, it also acts as if it were the source of the anchored path request (Section 4.1.2), with the following difference. For determining whether some further friend, say  $F1$ , brings the packet closer to destination, the remaining distance from  $F1$  to  $D$  is compared to the *min\_dist* field in the request, instead of the distance from  $T$  to  $D$ .

### 4.1.4 FAPD Search in Tabu Mode

This is triggered when a source or a responder node  $T$  has an anchored path discovery to process and knows of no friend that brings the request closer than self or than *min\_dist* field in the request.

- If the *tabu\_index* field in the received anchored path discovery message is 0, the *min\_dist* field is set to the distance from self to destination.

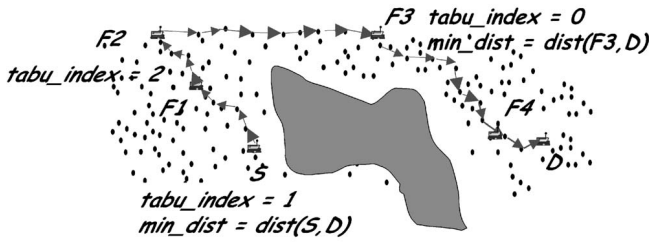


Fig. 2. Illustration of FAPD search in tabu mode. Source  $S$  does not have a friend closer to  $D$  than self.  $S$  sends anchored path request to its friend  $F1$  that is farther from  $D$  in geometrical distance than  $S$ , but such that  $dist(S, F1) < max\_dist$ .  $S$  sets the  $tabu\_index$  field to 1 in the packet and, thus, starts the tabu mode of FAPD.  $S$  puts  $dist(S, D)$  within  $min\_dist$  field. Neither  $F1$  has a friend whose distance to  $D$  is smaller than  $min\_dist$ .  $F1$  forwards the packet to its friend  $F2$  (that is in the opposite direction from  $D$ ), where  $dist(F1, F2) < max\_dist$ , and sets  $tabu\_index$  to 2.  $F2$  checks that  $tabu\_index$  is equal to its maximum value, and  $F2$  cannot forward the packet to its friend that does not reduce the distance  $min\_dist$ . In our example,  $F2$  has a friend  $F3$  whose distance to  $D$  is smaller than  $min\_dist$  and forwards the packet to it. At  $F3$ ,  $tabu\_index$  is reset to 0. From  $F3$  packet is forwarded to its friend  $F4$  and from there to  $D$  by using the TLR protocol. Once  $D$  receives the path request packet, it sends back to  $S$  the anchored path from  $S$  to  $D$  given with the list of anchors ( $LDA_{F1}, LDA_{F2}, LDA_{F3}, LDA_{F4}$ ).

- If the  $tabu\_index$  field does not exceed  $max\_tabu\_index$  ( $= 2$  in our implementation),  $T$  selects a friend  $F1$  at a distance not exceeding  $max\_dist$  (five times the transmission range of this node in our implementation), if any is available, else the packet is silently discarded. Then,  $T$  appends its path to  $F1$  to  $fapd\_anchored\_path$  and forwards the request to  $F1$ .
- Else  $T$  appends its LDA to  $fapd\_anchored\_path$  and sends the packet directly to the destination, using TRR without anchors.

See Fig. 2 for an example. This mode is inspired by the Tabu Search heuristic ([14], [16]), a local search procedure used in iterative optimization methods to get out of a local optimum. Indeed, in some topologies with obstacles, going in the opposite direction may sometimes be needed. As described in Section 4.1.2, whenever a FAPD responder does have a friend closer to destination,  $tabu\_index$  is reset. Thus, any tabu phase is limited to two friends, but there can be several tabu phases in a path.

#### 4.1.5 Destination Receives Anchored Path Request

The path request packet contained an accumulated list of anchors from  $S$  to  $D$ .  $D$  runs the path simplification method we present below. Then,  $D$  returns back to  $S$  a “path reply” control packet which contains the acquired anchored path from  $S$  to  $D$ . A path reply control packet is stamped with the same sequence number as the original path request. If  $S$  received several path replies, the sequence number is used to determine the freshness of the received anchored path. To send the path reply control packet,  $D$  reverts the anchored path and applies TRR with anchors. Once  $S$  receives from  $D$  a packet with the anchored path,  $S$  stores this path in its route cache (see Fig. 2).

#### 4.1.6 Path Simplification and Management

Path simplification consists of approximating an existing anchored path by a path with fewer anchors. Anchors (which correspond to locations of FAPD responders that

assist in path discovery) are accumulated from the source to the destination during the processing of the anchored path request by FAPD responders. For example, it is possible that many geographically close friends are consecutively contacted, and the resulting anchored path contains many close anchored points. The first goal of path simplification is to keep the number of anchors as small as possible. The destination simplifies the path by skipping a number of close anchors from an initial list of anchors.

Path management is used to decide when to trigger requests for new paths and how long existing paths should be kept. After issuing a path request, a source waits for  $wait\_for\_path$  seconds before issuing a new request. Paths are cached with a timeout of  $path\_validity$  seconds. FAPD responders periodically refresh their friends with time interval  $refresh\_friends$ .

## 4.2 Source Discovers Anchored Paths Using Geographic Maps-based Path Discovery (GMPD)

GMPD is another method for anchored path discovery, which assumes that maps of the network density are known to all nodes in the network. GMPD is performed at source nodes, and unlike FAPD, does not need assistance from other nodes.

Areas with a higher node density, are called “towns.” Two towns are interconnected by all the nodes in between them (we call it a “highway”). If two towns are interconnected with a highway, there is a high probability that there are nodes to ensure connectivity from one town to another. One example of a network modeled with towns and highways is presented in Fig. 3. GMPD assumes that each node has a summarized geographic view of the network. Each node has a knowledge of a “map” of towns. A map defines the network topology: It defines town areas and reports the existence of highways between towns. As a first attempt, we model a town area as a square centered in a geographic center. For each town, a map gives the location of its center and the size of the square area. One example of a map of a network is presented in Fig. 3. A map of the network can be presented as a graph with nodes corresponding to towns and edges corresponding to highways. Macroscopically, the graph of towns does not change frequently.

GMPD with a given map of towns works as follows:

- Source  $S$  determines from its own location  $LDA_S$  the town area ( $ST$ ) in which  $S$  is situated (or, the nearest town to  $LDA_S$  if it is not in the town area). In addition, since  $S$  knows the location of destination  $D$  ( $LDA_D$ ), it can determine from the  $LDA_D$  the town area  $DT$ , where  $D$  is situated (or, the nearest town to  $LDA_D$  if it is not in the town area).
- Then,  $S$  accesses the network map in order to find the anchored path from  $S$  to  $D$ . We call this operation a *map lookup*. An anchored path is the list of the geographical points: The points correspond to centers of the towns that the packet has to visit from  $ST$  in order to reach  $DT$ . One possible realization of the map lookup operation, which is used in our simulation in Section 5.3, is to find a list of towns that are on the shortest path from  $ST$  to  $DT$  in the graph of towns; the length of a path can be given

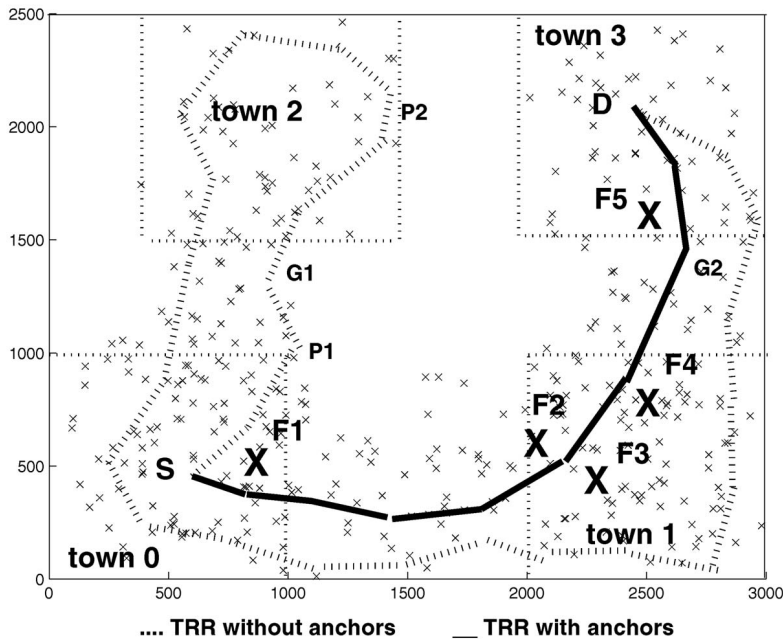


Fig. 3. Path of packet from source  $S$  to destination  $D$  without anchors (dotted) and with anchors (solid). When anchors are used, TRR gives a shorter path than TRR without anchors. An example of FAPD is also presented:  $S$  starts FAPD by sending the request for anchored path; this request is received by responder  $F1$ , which in its turn contacts its friend responder  $F2$ . Then, responders  $F3$ ,  $F4$ , and  $F5$  consequently participate in path discovery.  $F5$  forwards the anchored path request to  $D$ . The anchored path contains the following anchors:  $LDA_{F1}$ ,  $LDA_{F2}$ ,  $LDA_{F3}$ ,  $LDA_{F4}$ , and  $LDA_{F5}$ .  $D$  simplifies the anchored path: It removes  $LDA_{F3}$  from the anchored path since this anchor is close to  $LDA_{F2}$ .  $D$  sends back to  $S$  the anchored path with anchors:  $LDA_{F1}$ ,  $LDA_{F2}$ ,  $LDA_{F4}$ , and  $LDA_{F5}$ .  $S$  forwards data to  $D$  with TRR using the learned anchored path. Notice that data packets do not go through FAPD responders.

either as the number of towns between  $ST$  and  $DT$ , or the length of the topological (Euclidean) shortest path connecting  $ST$  and  $DT$  in a graph of towns.

## 5 PERFORMANCE EVALUATION OF TERMINODE ROUTING

In this section, we evaluate the performance of terminode routing by simulation (and by scalability analysis in Section 5.6). The simulation code is publicly available [1].

### 5.1 Simulation Setup

#### 5.1.1 Global Settings

We used GloMoSim [36] with the following settings. The IEEE 802.11 Medium Access Control (MAC) protocol is used with the Distributed Coordination Function (DCF) [11]. The radio range is 250 meters. The channel capacity is 2Mb/s. The propagation model is two-ray. It uses free space path loss for near sight and plane earth path loss for far sight.

#### 5.1.2 Protocol Constants

(Section 3.1) We used the following configuration for the Helloing protocol. The HELLO timer is 1 second. Each entry in the routing table expires after two seconds, if it is not updated. All nodes promiscuously listen to all HELLO messages within their radio range. Nodes that have data or control packets to send should defer sending HELLO messages (up to the timer value) and piggyback the HELLO message to the data or control packet.

For FAPD, the parameter values are: *refresh\_friends*=50s, *path\_validity*=20s, and *wait\_for\_path*=10s.

#### 5.1.3 Location Management

We implement a location management method, in order to account for its overhead. Our location management scheme is simple and was developed for the sole purpose of a fair comparison with routing protocols that do not require location management. Other, more sophisticated location management schemes can be found in [24], [35]. Location management consists of location discovery and tracking.

**Location Discovery.** Our method is similar to DSR source route discovery [8]. When source  $S$  has data to send to destination  $D$  that is not reachable by TLR,  $S$  needs to find the location of  $D$  ( $LDA_D$ ).  $S$  buffers all data packets until it learns  $LDA_D$ . To do so,  $S$  broadcasts a *location request* control packet to all its neighbors. Inside the packet,  $S$  stamps its own location and a sequence number. Node  $X$ , which receives a location request packet and is not the destination, broadcasts the request to its neighbors. In order to avoid a redundant transmission of the request,  $X$  should broadcast a particular location request packet only once. Intermediate nodes keep a cache of already seen location request packets. Entries in this cache are kept for 30 seconds. An already seen location request packet is discarded. On receiving the location request, destination  $D$  responds to  $S$  with the *location reply* control packet. The location reply carries  $LDA_D$ .  $D$  sends the location reply back to  $S$  without anchors. ( $D$  learns  $LDA_S$  from the location request packet). Upon reception of the location reply,  $S$  stores in its location cache  $LDA_D$ , as well the time this information is learnt.  $S$  then sends buffered data packets without anchors. But, if  $S$  does not receive a location reply from the destination after the timeout,  $S$  initiates again the flooding of the location request control packet with the new sequence number. The location reply wait timeout is 5 seconds.



**Location Tracking.** Once two nodes begin to communicate, location tracking is used: data packets periodically (every 5 seconds) piggyback the local location of the sending node. If no data packet is to be sent, a node periodically sends a location reply control message with its location information. The destination location is considered stale if not refreshed for more than 10 seconds in small networks, 20s in large networks. The source then reinitiates learning of the destination location. The source does not flood the network, but uses the last known destination location to reduce the search space for the destination. Similarly to LAR1, location request is flooded only in the expected rectangular region of the destination. If available, anchored paths are used to facilitate location management operation. In this case, if the source sends data to the destination using anchored paths, the destination sends back to the source its location updates using the reversed path.

**Idealized no-overhead location management.** For completeness, in simulations of large networks, we separate the cost of location management and location-based routing. We do this by also simulating a hypothetical no-overhead location management. In this idealized scenario, we assume that sources use location information with a lifetime of 5 seconds (which causes some location inaccuracies).

#### 5.1.4 Performance Metrics

We measured three commonly used performance metrics described in [34].

**Packet delivery fraction.** The ratio of the data packets delivered to the destinations to data packets generated by the CBR sources.

**Average end-to-end delay.** It includes all possible delays caused by queuing, retransmissions at the MAC, propagation, and transfer time. In the cases of AODV and LAR1, this also includes delays caused by buffering during route discovery. In the case of terminode routing, this includes delays caused by packets buffering during the destination location discovery.

**Normalized routing load.** The number of transmitted routing (control) packets per data packets delivered at destinations. In the case of AODV and LAR1, control packets are route request, reply, and error packets. Route request packets are generated by sources and flooded in the whole or a part of the network, route reply, and error packets are generated by destinations and forwarded to packet sources. Terminode routing generates four types of routing packets: HELLO messages that are generated periodically (unless data or control packets are sent) but not forwarded more than one hop; location request packets, generated by sources when the destination address is needed, and flooded to the network; location reply packets are generated by destinations and forwarded to sources upon reception of the location request; and location reply packets that are periodically generated by destinations and forwarded to packet sources. Each hop-wise transmission of a routing packet is counted as one transmission.

#### 5.1.5 Mobility Model and Mobility Index

**Random Waypoint.** It is described for example in [10]. We use it for unobstructed networks. A node chooses one random destination in the simulation area. Then, it moves to that destination at a random speed (uniformly chosen between 1-20 m/sec). Upon reaching its destination, the node pauses for *pause time*, selects another random

TABLE 1  
Parameters of Mobility Scenarios  
Used with Restricted Random Waypoint

Scenario number	1	2	3	4	5	6	7
# Stationary Nodes	50	50	50	50	250	250	250
# Commuter Nodes	300	300	300	300	0	0	0
Pause Time, Commuters (sec)	1	1	1	1	NA	NA	NA
# Ordinary Nodes	150	150	150	150	250	250	250
Pause Time, Ordinary (sec)	0	100	300	600	100	300	600
Mobility Index (m/s)	5.70	4.91	4.40	4.16	1.88	1.03	0.62

destination inside the simulation area, and proceeds as previously described. The area is an unobstructed rectangle of size 2,200m × 600m with 100 nodes. For compatibility, we use as **mobility index** for this model the parameter *pause time*. We use this type of model because it is common, and in order to be able compare our results with the existing literature.

**Restricted Random Waypoint.** It is defined in [4]. We use it for networks with obstacles. It reflects that in a large network, it is less probable that, for each movement, a node selects a random destination within a very large geographic area. We use a topology based on towns and highways as described in (3). There are 500 nodes and the size of the simulated area is 3,000m × 2,500m.

The model works as follows: A node starts in a town, selects at random a destination within a new town, moves there with a speed uniformly chosen between 1 and 20 m/s, and stays at destination for *pause\_time* seconds. If the node's fixed parameter *stay\_in\_town* is larger than 1, it picks a new destination *inside* the town, goes there using the same method, stays at destination for another *pause\_time*, and repeats such that the total number of consecutive pauses inside the town is equal *stay\_in\_town*. The node then repeats the entire procedure from where it is now. We use three types of nodes: "commuter," "ordinary," and "stationary." Commuters represent mobiles that frequently commute from one town to another. They have *stay\_in\_town* = *pause\_time* = 1s. For "ordinary" nodes, *stay\_in\_town* is 2 and we vary *pause\_time*. Stationary nodes do not move. Their position is drawn from the stationary distribution of commuters (so they are more likely to be in town but have a small probability of being on highways).

As **mobility index** for restricted random waypoint, we cannot simply take *pause\_time* since different parameters influence the mobility. We take instead the *average node speed*. This is computed by taking the average speed for an arbitrary mobile, sampled at an arbitrary instant, and averaging over all mobiles in the simulation. It can be obtained by measurement in the simulation, but can also be computed using the stationary distribution of the random waypoint model [10]. Note that the average speed is much less than the arithmetic average of maximum and minimum speed; this is because the stationary distribution of speeds is not uniform, but is skewed to the left [10].

We use two families of scenarios, ranging from high to low mobility. The parameters and the corresponding mobility indices are given in Table 1.

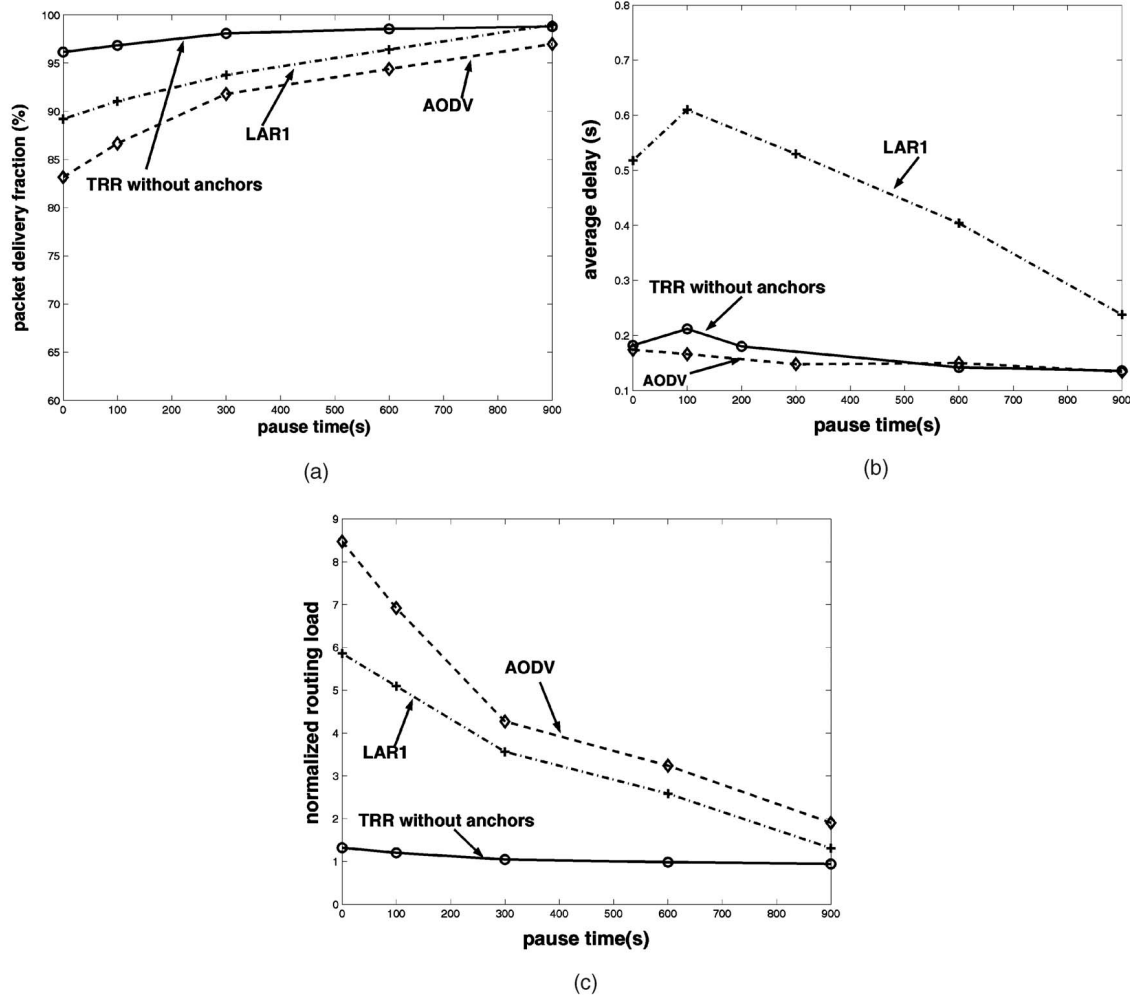


Fig. 4. (a) Packet delivery fraction. (b) Average data packet delay. (c) Normalized routing loads for the 100 node model and 40 sources.

### 5.1.6 Common Simulation Parameters

Sources are constant bit rate (CBR) and send two packets of 64 bytes per second. There are 40 source destination pairs, chosen randomly (uniformly) over the set of nodes. With these values, the network is not congested because we want to measure routing protocol behavior, not the limitation of the IEEE 802.11 MAC for data packet capacity. CBR connections are started at times uniformly distributed between 400 and 500 seconds.

Simulations run for 1,200 simulated seconds of simulated time. measurements start after 300s of simulated time (small networks), or 600s (large networks). Each data point represents an average of six runs with identical traffic models, but different randomly generated mobility scenarios.

## 5.2 Experiment 1: Small, Unobstructed Network

The goal of this experiment is to compare terminode routing versus two other routing protocols, AODV and LAR1 (LAR scheme 1), in a small unobstructed ad hoc network, where we expect AODV and LAR to perform optimally. Because the simulation area is small and unobstructed, terminode routing uses TRR without anchors. Terminode routing is evaluated with location management overhead included. Simulations of AODV and LAR1 are performed using the

latest implementations available in GloMoSim. We use random waypoint as described in Section 5.1.5, random waypoint. The results are shown in Fig. 4.

**Packet Delivery.** Terminode routing is comparable to LAR1 and both outperform AODV (Fig. 4a).

**Delay.** LAR1 is higher for LAR1 than AODV and terminode routing (Fig. 4b). The higher delay of LAR1 is mainly attributed to its use of route caching, and lack of any mechanism to expire stale routes or to determine the freshness of routes when multiple routes are available. On the other hand, caching of routes and control of route request control packets helps LAR1 to keep routing load lower than in case of AODV. AODV replies to the first arriving route request packet, thus favoring the least congested route. Terminode routing delay is due to buffering during location discovery.

**Routing Overhead.** Terminode routing has the smallest normalized routing load (Fig. 4c) and it is independent of mobility index. This is due for one part to HELLO messages being independent of the mobility rate of nodes. For another part, routing overhead due to location management does not change very much with the increase of mobility. We verified in the simulations that, in most cases, mobility tracking ensures that the source receives periodic updates of the destination location, without the need to often flood

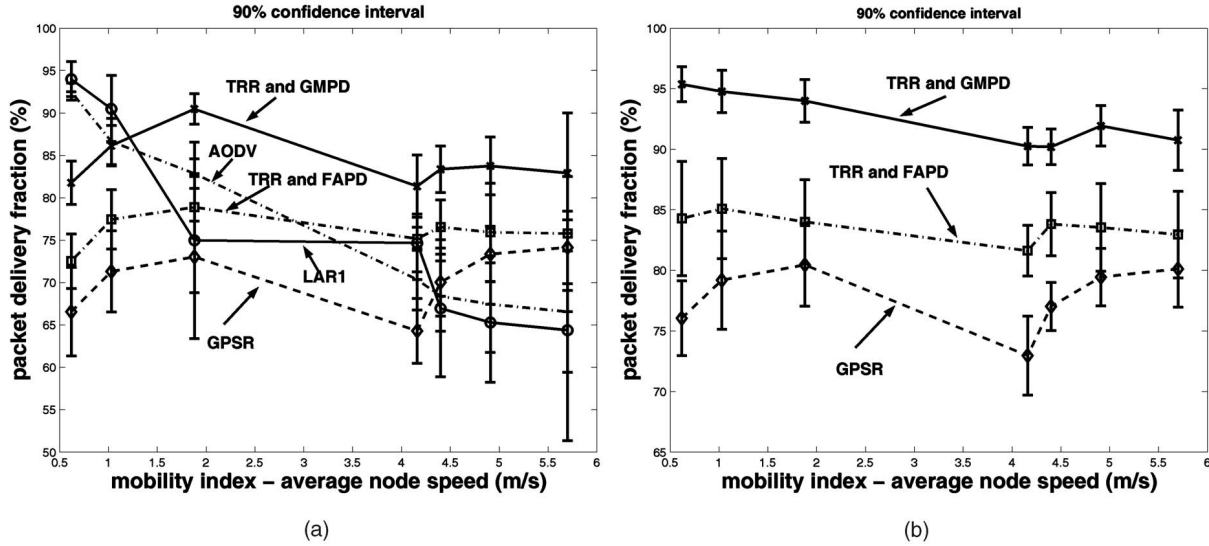


Fig. 5. Packet delivery fractions for 500 nodes network of Fig. 3 with restricted random waypoint mobility model, versus mobility index as defined in Table 1. (a) With location management and (b) with idealized no-overhead location management.

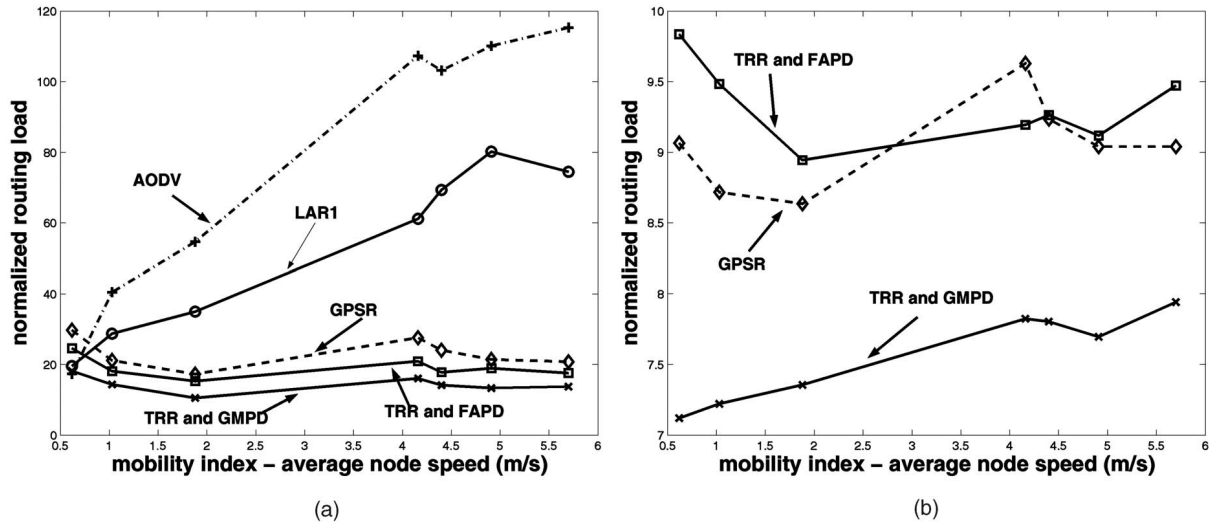


Fig. 6. Normalized routing load, same conditions as Fig. 5; (a) with location management, (b) with idealized no-overhead location management.

the network. In contrast, both AODV and LAR1 maintain routes per destination, which consumes more overhead with high mobility.

### 5.3 Experiment 2: Large Network with Obstacles

We use restricted random waypoint as described in Section 5.1.5, restricted random waypoint and Table 1. We separately evaluated terminode routing with GMPD (Section 4.2) and with FAPD (Section 4.1). With GMPD, the overhead of map distribution is neglected, as we expect this to occur only very infrequently. All other overhead of GMPD, FAPD and location management is included. When FAPD is used for path discovery, 50 of the stationary nodes are FAPD transponders.

Simulation results are shown on Figs. 5 and 6.

**High Mobility Scenarios.** Fig. 5a shows that, at high mobility, terminode routing outperforms all other methods. Moreover, GMPD outperforms FAPD, which is to be expected since GMPD is able to exploit the topology

information that FAPD has to discover. Both AODV and LAR1 suffer from broken paths. LAR1 uses a form of geographic discovery, however, in a network with obstacles, there are many cases where the form of the search region used by LAR1 is not efficient. Take, for example, a source and a destination that are in towns 2 and 3 of Fig. 3. The LAR1 request zone in this case is a rectangle that contains the void area. It fails in searching a feasible route to the destination and has to repeat with an expanded request zone up to the whole network. A similar observation is reported in [25].

The problems of AODV and LAR1 in these scenarios do not affect Terminode routing nor GPSR. However, the performance of GPSR is impeded by its frequent use of perimeter mode, which often causes loops [5]—a problem avoided by the use of anchors.

Fig. 5b shows that the location management scheme implemented in this simulation costs about 10 percent of the packet delivery fraction.

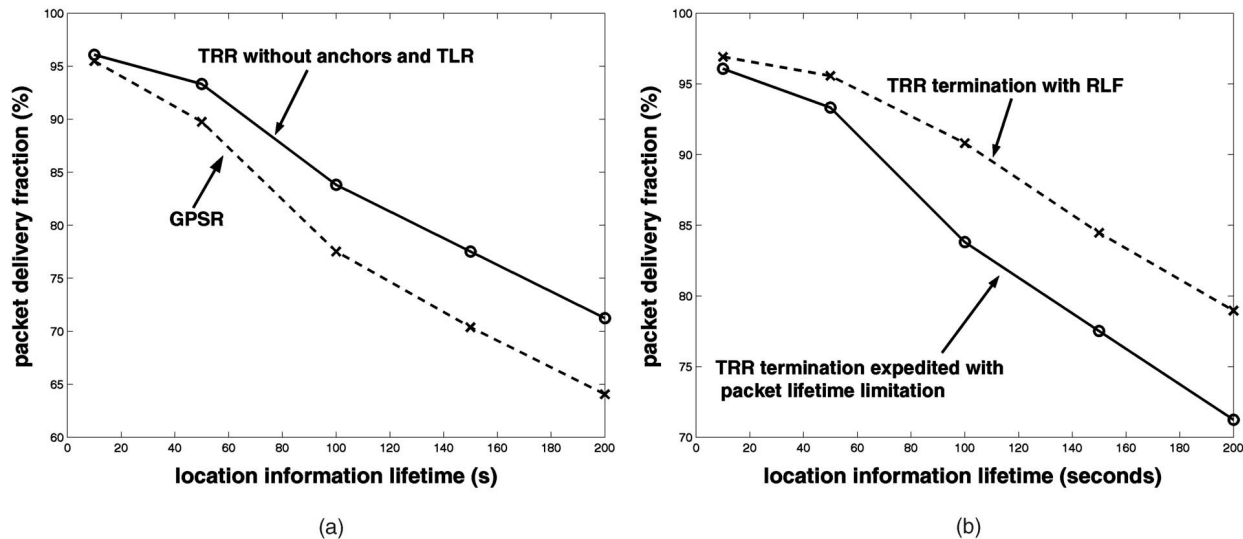


Fig. 7. (a) Using TLR results in higher packet delivery fraction than in the case when only location-based routing is used. (b) With Restricted Local Flooding, the fraction of received packets is higher than without RLF (when TRR is terminated by the packet lifetime limitation).

Fig. 6a shows that the routing overhead of location-based routing (which includes location overhead) is far less than for AODV and LAR1, and is largely independent of mobility. Fig. 6b further shows that most of routing overhead actually comes from location management.

**Low Mobility Scenarios.** For low mobility scenarios (most nodes are stationary and always on), Fig. 5a shows that LAR1 and AODV outperform location-based methods like terminode routing and GPSR. AODV, and LAR1 routes are stable and need not be rediscovered and, thus, packet delivery approaches 100 percent success.

Fig. 5b shows that the cause is not location overhead. Instead, it is due to the periodic expiry of anchored paths. A possible optimization would be that sources monitor the performance of packet delivery over an anchored path, and issues a new path discovery only when the anchored path does not perform well.

#### 5.4 Experiment 3: Usefulness of Terminode Local Routing

The goal of this experiment is to assess the value of combining TLR to TRR when location accuracy is low. To that end, we compared terminode routing without anchors to GPSR (remember from Section 2.1 that terminode routing without anchors is essentially equivalent to combining GPSR with TLR). We use the random waypoint mobility model as in Section 5.1.5, random waypoint, however, in a larger network (600 nodes in an area of size 2,900m  $\times$  2,900m). We used a dense, large network so that direct paths are most often used (the use of perimeter mode would largely bias the simulation results). We disable RLF (see Experiment 4) in order to isolate the effect of TLR. We used the "idealized no overhead location management" (Section 5.1.3), since both GPSR and terminode routing have the same location management needs. We varied the parameter *location information lifetime*, which controls how old location information can be.

Fig. 7a shows that for small location information lifetimes (less than 20 seconds), the packet delivery fraction is similar with terminode routing and GPSR. However, for

higher location information lifetimes, the opposite is true. With location information older than one minute, the presence of TLR significantly improves packet delivery.

#### 5.5 Experiment 4: Usefulness of Restricted Local Flooding

RLF is a mechanism that we propose to use in addition to TLR in order to improve packet delivery in presence of location inaccuracy (Section 3.5). We use the same simulation settings as in Experiment 3, but we now compare terminode routing with and without RLF. The latter case is the same as one of the two cases in Experiment 3. Fig. 7b shows that RLF has a significant impact with location information older than one minute.

Combining the results of Experiments 3 and 4, we find that both TLR and RLF are able to significantly improve packet delivery with inaccurate positioning.

#### 5.6 Experiment 5: Scalability Analysis

We analyze the scalability of terminode routing based on the theoretical model described in [19], [18]. The analysis consists of comparing how the overhead cost of the protocol scales with respect to the total, theoretical capacity. The overhead considered in [19] is classified as

1. proactive (capacity consumed for propagating route information),
2. reactive (capacity consumed for building paths when necessary, and
3. suboptimal (capacity wasted due to suboptimal paths).

We add an additional point, specific to location-based routing:

4. addressing/location overhead (capacity consumed for propagating location information).

In order to use an existing comparison basis, we use the class of network models defined in [18], which has the following properties. The average in-degree  $d$  is constant

and there is an uniform node distribution. The traffic generated by a node is independent on the network size and all destination are equiprobable. The link status changes are due to mobility. The network model has three scaling factors:  $\lambda_{lc}$  (mobility rate),  $\lambda_t$  (total traffic), and  $\lambda_N$  (number of nodes). For a factor  $i$ , the analysis in [18] defines

$$\rho_i \stackrel{\text{def}}{=} \limsup_{\lambda_i \rightarrow +\infty} \frac{X_{ov}}{\lambda_i} \leq \Psi_i \stackrel{\text{def}}{=} \limsup_{\lambda_i \rightarrow +\infty} \frac{T}{\lambda_i},$$

where  $X_{ov}$  is the total capacity consumed by overhead and  $T$  the theoretical capacity. Thus,  $\rho_i$  [respectively,  $\Psi_i$ ] the scaling exponent for overhead of factor  $i$  [respectively, for the theoretical capacity]. The scaling analysis consists of comparing  $\rho_i$  to  $\Psi_i$ . We have [18]:  $\Psi_{lc} = 0$ ,  $\Psi_t = 1$ , and  $\Psi_N = 1.5$ .

The detailed analysis in the Appendix of [6] shows that scalability depends on the location management scheme. With the simple location management method used in the simulations, we obtain  $\rho_{\lambda_{lc}} = 0 = \Psi_{\lambda_{lc}}$ ,  $\rho_{\lambda_t} = 1 = \Psi_{\lambda_t}$ , and  $\rho_{\lambda_N} = 2 > \Psi_N$ . With a more advanced scheme such as [13], we have  $\rho_{\lambda_N} = 1.5 = \Psi_N$  instead. Thus, terminode routing is scalable with respect to the most relevant parameters provided that an advanced location management scheme is used.

## 6 CONCLUSION

Terminode routing aims to support location-based routing on irregular topologies with mobile nodes. It achieves its goal by combining a location-based routing method with a link state-based mechanism. Further, it introduces the concept of anchors, which are geographical points imagined by sources for routing to specific destinations, and proposes low overhead methods for computing anchors. Last, a special form of restricted search mode (Restricted Local Flooding, RLF), solves problems due to the inaccuracy of location information, in particular for control packets.

The performance analysis shows that, in large mobile ad hoc networks, terminode routing performs better than MANET-like, or existing location-based routing protocols. It does so by maintaining its routing overhead low and by efficiently solving location inaccuracies.

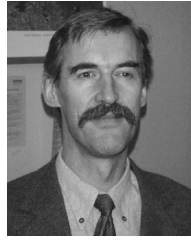
## REFERENCES

- [1] <http://ica1www.epfl.ch/TNRouting>, Simulation Source Code of Terminode Routing (in Glomosim) and Interactive Java applet, 2004.
- [2] S. Basagni, I. Chlamtac, V. Syrotiuk, and B. Woodward, "A Distance Routing Effect Algorithm for Mobility (DREAM)," *Proc. Fourth Ann. ACM/IEEE Int'l Conf. Mobile Computing and Networking (MobiCom '98)*, Aug. 1998.
- [3] L. Blazevic, S. Giordano, and J.-Y. Le Boudec, "Anchored Path Discovery in Terminode Routing," *Proc. Second IFIP-TC6 Networking Conf. (Networking 2002)*, May 2002.
- [4] L. Blazevic, S. Giordano, and J.-Y. Le Boudec, "Self Organized Terminode Routing," *Cluster Computing J.*, Apr. 2002.
- [5] L. Blazevic, "Scalable Routing Protocols with Applications to Mobility," EPFL, PhD thesis no. 2517, 2002.
- [6] L. Blazevic, J.-Y. Le Boudec, and Silvia Giordano, "A Location-Based Routing Method for Irregular Mobile Ad Hoc Networks," Technical Report IC 200330, EPFL-DI-ICA, May 2003.
- [7] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," *Proc. Third Int'l Workshop Discrete Algorithms and Methods for Mobile Computing Comm. (DIAL M)*, Aug. 1999.
- [8] J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," *Proc. Fourth Ann. ACM/IEEE Int'l Conf. Mobile Computing and Networking (MobiCom '98)*, Aug. 1998.
- [9] S. Capkun, M. Hamdi, and J.-P. Hubaux, "GPS-Free Positioning in Mobile Ad-Hoc Networks," *Proc. 34th Hawaii Int'l Conf. System Sciences*, Jan. 2001.
- [10] H. Hartenstein, C. Bettstetter, and X. Prez-Costa, "Stochastic Properties of the Random Waypoint Mobility Model," *ACM/Kluwer Wireless Networks*, special issue on modeling and analysis of mobile networks, 2004.
- [11] IEEE Society LAN MAN Standards Committee, "Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification," *IEEE Std 802.11-1997*, IEEE, New York, 1997.
- [12] M. Gerla, G. Pei, and X. Hong, "Lanmar: Landmark Routing for Large Scale Wireless Ad Hoc Networks with Group Mobility," *Proc. First IEEE/ACM Workshop Mobile Ad Hoc Networking and Computing (MobiHOC)*, Aug. 2000.
- [13] S. Giordano and M. Hamdi, "Mobility Management: The Virtual Home Region," EPFL DSC technical report, [http://icwww.epfl.ch/publications/documents/IC\\_TECH\\_REPORT\\_199937.pdf](http://icwww.epfl.ch/publications/documents/IC_TECH_REPORT_199937.pdf), Mar. 2003.
- [14] F. Glover, "Future Paths for Integer Programming and Links to Artificial Intelligence," *Computers and Operational Research*, vol. 13, 1986.
- [15] P. Gupta and P.R. Kumar, "The Capacity of Wireless Networks," *IEEE Trans. Information Theory*, vol. 46, no. 2, Mar. 2000.
- [16] P. Hansen, "The Steepest Ascent Mildest Descent Heuristic for Combinatorial Programming," *Proc. Congress Numerical Method in Combinatorial Programming*, 1986.
- [17] A. Harter and A. Hopper, "A New Location Technique for the Active Office," *IEEE Personal Comm.*, pp. 42-47, Oct. 1997.
- [18] R. Ramanathan, I. Stavrakakis, B. McDonald, and C. Santivanez, "On the Scalability of Ad Hoc Routing Protocols. Part 2: Asymptotic Analysis with Respect to Traffic, Mobility, and Network Size," *ACM/IEEE Trans. Networking*, to appear.
- [19] R. Ramanathan, I. Stavrakakis, B. McDonald, and C. Santivanez, "On the Scalability of Ad Hoc Routing Protocols. Part 1: Making Link State Routing Scale," *ACM/IEEE Trans. Networking*, to appear.
- [20] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen, "Scalable Routing Strategies for Ad-Hoc Wireless Networks," *IEEE J. Selected Areas in Comm.*, vol. 17, no. 8, Aug. 1999.
- [21] P. Jacquet, P. Muhlenhaller, and A. Qayyum, "Optimized Link State Routing Protocol," Internet Draft, draft-ietf-manet-olsr-\*.txt, work in progress.
- [22] M. Jiang, J. Li, and Y. -C. Tay, "Cluster Based Routing Protocol Functional Specification," Internet Draft, draft-ietf-manet-cbrp-spec\*.txt, work in progress.
- [23] B. Karp and H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proc. ACM/IEEE Int'l Conf. Mobile Computing and Networking (MobiCom '00)*, Aug. 2000.
- [24] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," *Proc. Mobicom '00*, 2000.
- [25] J. Li and P. Mohapatra, "LAKER: Location Aided Knowledge Extraction Routing for Mobile Ad Hoc Networks," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '03)*, 2003.
- [26] Mobile Ad-Hoc Networks (MANET) Working Group, <http://www.ietf.org/html.charters/manet-charter.html>, 2004.
- [27] S. Murthy and J.J. Garcia-Luna-Aveces, "An Efficient Routing Protocol for Wireless Networks," *ACM Mobile Networks and Applications J.*, special issue on routing in mobile communication networks, Oct. 1996.
- [28] V.D. Park and M.S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," *Proc. IEEE INFOCOM '97*, Apr. 1997.
- [29] C.E. Perkins and E.M. Royer, "Ad-Hoc On-Demand Distance Vector Routing," *Proc. Second IEEE Workshop Mobile Computer Systems and Applications (WMCSA '99)*, Feb. 1999.
- [30] C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector (dsdv) Routing for Mobile Computers," *Proc. ACM SIGCOMM '94*, Aug. 1994.
- [31] M.R. Perlman and Z.J. Haas, "Determining the Optimal Configuration for the Zone Routing Protocol," *IEEE J. Selected Areas in Comm.*, vol. 17, no. 8, Aug. 1999.
- [32] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," *Proc. Mobicom '00*, 2000.

- [33] R. Jain, A. Puri, and R. Sengupta, "Geographical Routing Using Partial Information for Wireless Ad Hoc Networks," *Proc. IEEE INFOCOM*, 2001.
- [34] S. Das, C. Perkins, and E. Royer, "Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks," *Proc. IEEE INFOCOM*, 2000.
- [35] S. Giordano and M. Hamdi, "Mobility Management: The Virtual Home Region," Technical report no. ssc/1999/037, EPFL-ICA, 1999.
- [36] M. Takai, L. Bajaj, R. Ahuja, R. Bagrodia, and M. Gerla, "GloMoSim: A Scalable Network Simulation Environment," Technical Report 990027, Univ. of California at Los Angeles, Computer Science Dept., 1999.
- [37] Y. Ko and N. Vaidya, "Location-Aided Routing in Mobile Ad-Hoc Networks," *Proc. Fourth Ann. ACM/IEEE Int'l Conf. Mobile Computing and Networking (MobiCom '98)*, Aug. 1998.



**Ljubica Blazevic** graduated from Faculty of Electrical Engineering, Belgrade, Yugoslavia in 1993. She received the doctorate degree in 2002 from EPFL, Switzerland, on routing for mobile ad hoc networks. From 1993-1996, she worked as a R&D engineer at the Institute "Mihajlo Pupin" in Belgrade. In 1996-1997, she finished Doctoral School in Communication Systems, EPFL. She joined STM in 2002 in AST Geneva, in Broadband Wireless LAN group as a R&D system engineer. Her current R&D activities focus on MAC, networking and localization aspects of ultra-wideband communication systems. She is a member of the IEEE and the IEEE Computer Society.



**Jean-Yves Le Boudec** (Fellow, 2004) graduated from Ecole Normale Supérieure de Saint-Cloud, Paris, and received his doctorate in 1984 from the University of Rennes, France. In 1987, he joined Bell Northern Research, Ottawa, Canada, as a member of scientific staff in the Network and Product Traffic Design Department. In 1988, he joined the IBM Zurich Research Laboratory where he was a manager of the Customer Premises Network group. In 1994, he became professor at EPFL, where he is now full professor. He is coauthor of the book *Network Calculus*. His interests are in the architecture and performance of communication systems. He is a fellow of the IEEE.



**Silvia Giordano** received the PhD degree from the Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland. She is currently working as a senior researcher at the University of Applied Science—SUPSI in Ticino, Switzerland. She is teaching several courses in the areas of wireless and mobile networking, quality of services, and networks applications. Previously, she was on the faculty of the EPFL and the University of Pisa. Since October 2001, she is also an associate researcher at CNR, Pisa. She has published extensively in the areas of quality of services, traffic control, wireless, and mobile ad hoc networks. She has participated in several European ACTS/IST projects and European Science Foundation (ESF) activities. Since 1999, she has served as a technical editor of *IEEE Communications Magazine*. She has been coeditor of several special issues of *IEEE Communications Magazine* and *Baltzer MONET* and *Cluster Computing* on mobile ad hoc networking and QoS networking. She was on the executive committee and TCP of several international conferences, and has served as reviewer on transactions and journals, as well as for several important conferences. She is a member of the IEEE and the IEEE Computer Society and IFIP WG 6.8. Her current research interests include QoS and traffic control, wireless and mobile ad hoc networks.

► For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).