EPFL-IC Report Number IC/2003/30, May 2003

A Location Based Routing Method for Irregular Mobile Ad Hoc Networks

Ljubica Blažević, Jean-Yves Le Boudec and Silvia Giordano

EPFL, CH-1015, Lausanne, Switzerland

Email: ljubica.blazevic@st.com, jean-yves.leboudec@epfl.ch, silvia.giordano@die.supsi.ch

Keywords: ad hoc network, scalable routing, location-based routing method, robustness to location inaccuracy

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. 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.

I. INTRODUCTION

Many existing routing protocols (DSDV [30], WRP [27], OLSR [22], FSR [21], LANDMAR [12],DSR [9], AODV [29], TORA [28],CBRP [23]), proposed within the MANET [26] working group of IETF, are designed to scale in networks of a few hundreds of 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. 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 [38]), to control packet flooding (DREAM [1]), to reduce intermediate system functions or to make simplified packet forwarding decisions (GPSR [24], GFG [8] and GRA [33]).

LAR is an on-demand routing protocol where location information is used to reduce the search space for a desired route, but it uses a DSR-like source routes for packet forwarding. The source uses the last known destination location in order to estimate the zone in which the destination is expected to be found. This zone 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 [24], GFG [8] and GRA [33] use only neighbour location information for forwarding data packets. Routing is done in a greedy way by forwarding the packet to a neighbour 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 neighbours is sufficient for a node to determine its local view of the planar graph. Perimeter mode can give a poor path in large networks when the source and destination are not well connected along the shortest geodesic path. 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 is large overhead for large mobile ad hoc networks. Thus the existing location-based routing protocols are not appropriate for large ad hoc networks of arbitrary node distribution, in particular when there are holes in the network topology.

Further, in location-based routing protocols, sources should know destination positions 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 positions change frequently. Existing location-based routing protocols do not address how to cope with location management inaccuracies.

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. Scalability is taken for now in an informal sense (see Section VI-D for a formal analysis). It means that the average total overhead, which includes control messages and the penalty paid for suboptimal routing, must not increase too severely, as the size of the network grows, or the mobility of nodes increases. An analysis done by Gupta and Kumar [16] estimates that the per node capacity 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 1000 nodes, we verified by simulations that we are not in the asymptotic regime proposed by Gupta and Kumar. We also found that our routing method does perform better than the existing MANET and location-based routing protocols we compared it to. Irregular topologies are likely to appear in metropolitan areas with mountains or lakes, like the Lake of Geneva area.

Terminode routing achieves its goal by combining a location based routing method with a link state based mechanisms for coping with location inaccuracies. Further, it introduces the concept of anchors, which are geographical points imagined by sources for routing to specific destinations. 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. An overview of terminode routing is given in Section II, and a detailed description in Sections III and IV in the form of protocol walkthrough. Section V develops the mathematical analysis supporting the computations made by a node in order to decide whether to use an anchored path or not.

We evaluated the performance of our protocol by detailed simulations and its scalability by analysis (Section VI). 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 regular networks that are uniformly 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 inaccuries. In all cases, terminode routing is characterized by low routing overhead, even when we include the overhead of location management.

Preliminary versions of some components of terminode routing are presented in [5], [6], [2], [3], [4], [7].

List of Acronyms.

EUI, End-system Unique Identifier (permanent address)
FAPD, Friend Assisted Path Discovery
GMPD, Geographical Map-based Path Discovery
LDA, Location Dependent Address (location)
RLF, Restricted Local Flooding
TLR, Terminode Local Routing
TRR, Terminode Local Routing

II. OVERVIEW OF TERMINODE ROUTING

A. A 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. Figure 1 (a) shows an example. It assumes that the source S knows an approximate location of the destination D (see Section II-E for details). S then sends the packet to a neighbour that brings the packet closer to the assumed location of D, and this is repeated by intermediate nodes. Then some intermediate node finds that D is one or two hops away, using TLR reachability information based on permanent addresses, and not location.

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

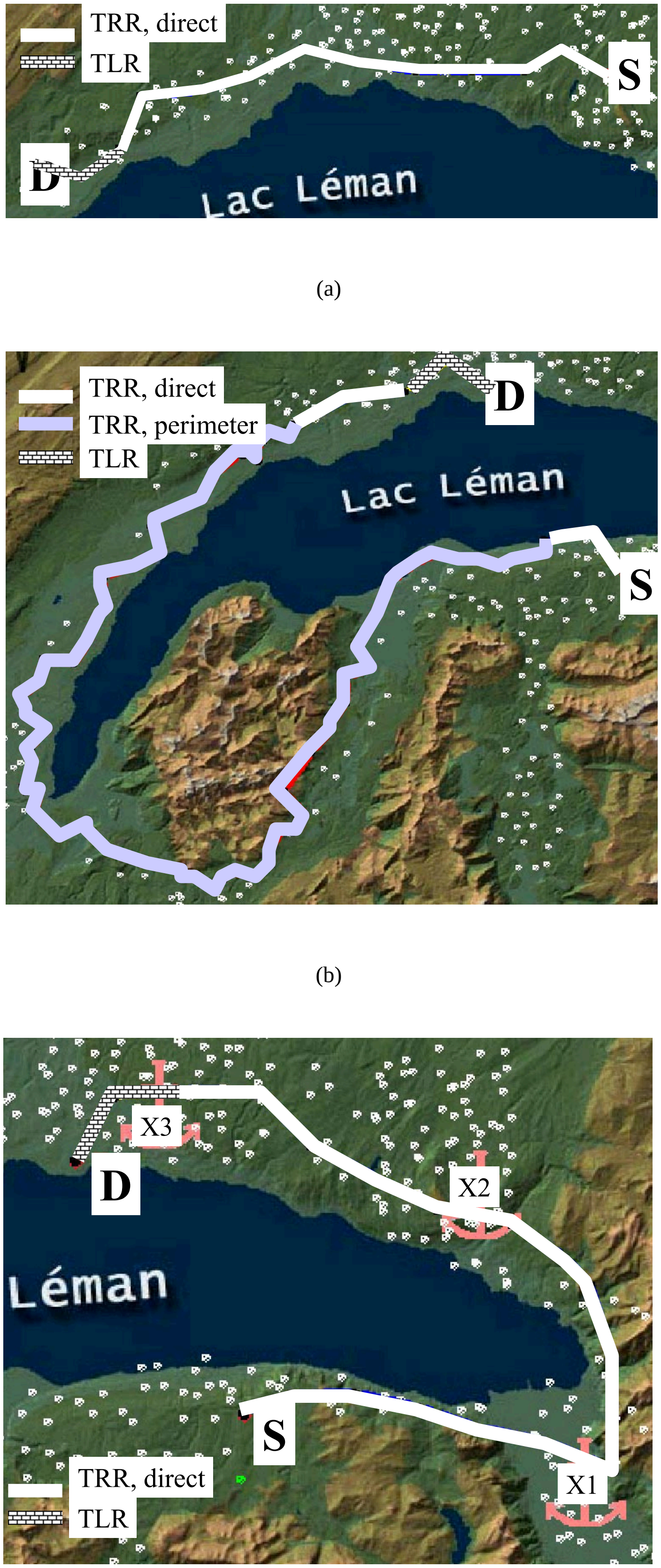
Figure 1 (b) shows a case where the direct path does not work well: the packet may be “stuck” at a node that does not have a neighbour that is closer to the destination than self. Here, TRR uses *perimeter* mode to circumvent the topology hole, similar to GFG [8] and GPSR [24]. A planar graph traversal is applied: the packet is routed around the perimeter of the problem area. This goes on until a node is found that reduces the distance to the destination, and thereon 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 [7]. Thus, we restrict the use of perimeter mode to discovery phases, when a better mode is not available to the source.

B. Anchored Paths in TRR

In order to avoid perimeter mode, we introduce the concept of anchors, which are imaginary positions used to assist in routing. On Figure 1 (c), source S uses three anchors to route the packet to D. The anchors are geographical positions, 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.

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,



(a)

(b)

(c)

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

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 (non-anchored) 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 needed if the packet path is significantly longer that estimated from the distribution of the number of hops along the greedy path.

C. Computing Anchors

Anchored paths however come at the price of computing good anchors. This is done by sources, using one of two low overhead protocols.

The former, Friend Assisted Path Discovery (FAPD) 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.

The latter, Geographical Map-based Path Discovery (GMPD), assumes that network density maps are available to a source node. This corresponds to the view of 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.

D. 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 a 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 III-E) or for a node type (FAPD responder, Section IV-A.2) and (2) establish long distance relations (Section IV-A.1).

E. 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 ([10], [32], [18]) 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. Firstly, a location tracking algorithm is assumed to exist between nodes when they have successfully established communication; this allows communicating nodes to continuously update their correspondant LDAs. Secondly, 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 VI, we present the location management scheme that we used to in simulations to evaluate the performance of terminode routing. Some other proposals are presented in [25], [35].

A Java applet implementation of terminode routing is available on the web for demonstration at `http://ica1www.epfl.ch/TNRouting`. The interested reader may find it a useful complement to this section.

III. 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 IV.

A. 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 neighbours.

The node keeps in its routing table (called "TLR table") the addresses and positions of its immediate neighbours, as well as the addresses of its two-hop neighbours. 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 neighbour for some amount of time, it removes from the routing table the entry that corresponds to the lost neighbour, as well as all two-hop distant terminodes that were reachable via the lost neighbour.

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 [24].

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 IV-B)
- it is a FAPD responder. If so, it will provide friend assistance to other nodes. The operation for that case is described in Section IV-A

B. 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 neighbour 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 neighbours. If this is not possible, the packet is dropped. This ensures that TLR is loop-free.

C. Source Node has Packet to Send, Destination Not in TLR Table

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 II 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 II). 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 use when the packet comes close to the assumed location of D (Section III-E). When the latter timer expires, the cache entry is considered stale and is removed.

2) ***Send Packet Using TRR Without anchors****:* The source sends the packet to an immediate neighbour that best improves the distance to D. The information about such a neighbour is obtained from the TLR table. If no such neighbour exists according to the table, perimeter mode is used instead; the packet is sent to the immediate neighbour computed by the Gabriel graph algorithm, as in [24].

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 Section V the source estimates if the path without anchors works well, or if an anchored path should be tried.

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 IV-B). Otherwise, S starts FAPD (Section IV-A).

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).

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.

D. 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.

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 III-B).

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.

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
- and the RLF bit is not equal to 1 (see Section III-E).

If the condition is met, the node terminates TRR as described in III-E.

*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, read from the packet header. The operation is the same as in Section III-C.2.

*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 III-C.2, but with the destination's location replaced by "Next location".

See also Figure 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.

E. Intermediate Node Expedites Termination of TRR

This section is applied once the condition in Section III-D.2 is satisfied. The action depend 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 III-C); it may also be set to 1 when a node is in search of a FAPD responder (Section IV-A.2).

*1) **"Use RLF" is Not Set in Packet Header**:* Before forwarding the packet, this node sets the TTL to $\min(term\_trr, TTL)$. This has the effect of limiting a loop due to destination location inaccuracy to $term\_trr$ hops ($term\_trr = 3$ in our current implementation).

*2) **"Use RLF" is Set in Packet Header**:* In this case, "Restricted Local Flooding (RLF)"is used. It consists in sending six duplicates of the packet in different directions around the sending node (X). In this way, packets with 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 were 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.

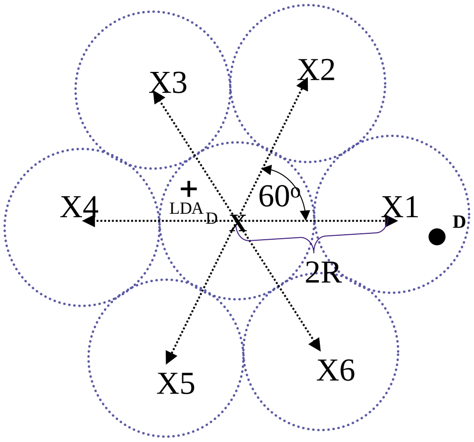


Fig. 2. Node X has a packet for D and finds LDA_D falls within its transmission range, but D is not TLR-reachable. X performs Restricted Local Flooding (RLF) by sending six duplicates of the packet towards six different geographic points around X.

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. In Figure 2 these geographic positions are denoted as Xi, $i \in 1..6$. Within the ith packet, the destination LDA in the packet header is set to Xi. However, the destination EUI field is not changed (i.e., EUI_D). X1 through X6 thus present virtual destination positions. All points $X1$ to $X6$ are at the same distance from X, which is equal to twice the transmission range of X. It can be seen from Figure 2, 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 $term\_rlf$ (equal to 4 in our implementation). In this way, we constrain the lifetime of a duplicate to $term\_rlf$ hops. Packets with the RLF bit equal to 1 forwarded towards one of geographic positions Xi. There are three possible situations for each of the 6 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 $term_rlf = 4$th hop).
3) Some intermediate node N, finds X_i (Xi is in the destination LDA field in the packet header) in its transmission range, but the destination is not the TLR table. Then N drops the packet. In this way we prevent further duplications.

IV. PROTOCOL WALKTHROUGH: ANCHORED PATH DISCOVERY

Anchored path discovery is triggered by a source node when it estimates that a non-anchored 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 Maps-based Path Discovery (GMPD).

A. *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.

*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 III-E and Figure 2, but with four duplicates instead of 6. 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 III-E, the goal is different: we want to establish some long distance friendships, whereas in Section III-E 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 FPi 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 [4] 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 positions 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 positions 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 [7].

*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 IV-A.4). On receiving the packet, $F1$ performs the actions in Section IV-A.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 IV-A.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 IV-A.1. Each of the 4 packet duplicates is thus sent in anycast mode (EUI is set to a predefined value meaning “Any FAPD Responder”), in a region up to 4 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 Section IV-A.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 reaches a FAPD responder, S does not get any anchored path to D.

*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 IV-A.2). If $tabu\_index$ field is non 0, it also acts as if it were the source of the anchored path request (Section IV-A.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) **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.

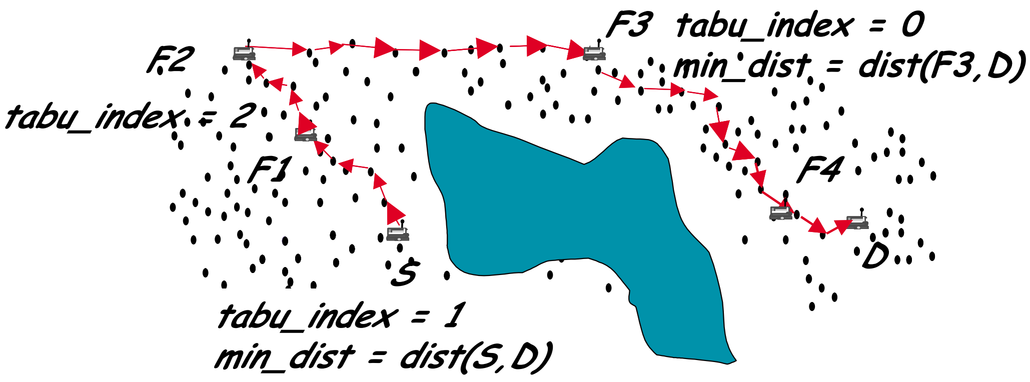


Fig. 3. Illustration of the FAPD search in tabu mode. Source S does not have a friend closer to D than self. S sends anchored path request to its friend $F1$ that is farther from D in geometrical distance than S, but such that $dist(S, F1) < max\_dist$. S sets the $tabu\_index$ field to 1 in the packet 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 in the received anchored path discovery message is 0, the min_dist field is set to the distance from self to destination.
- 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 (5 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 (Figure 3) for an example. This mode is inspired by the Tabu Search heuristic ([15], [17]), 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 IV-A.2, whenever a FAPD responder does have a friend closer to destination, $tabu\_index$ is reset. Thus any tabu phase is limited to 2 friends, but there can be several tabu phases in a path.

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. 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 Figure 3).

*6) **Path Simplification**:* Path simplification consists in approximating an existing anchored path by a path with fewer anchors. Anchors (which correspond to positions 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.

B. 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 Figure 8. 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 Figure 8. 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 VI-B, 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 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.

V. ESTIMATION OF NECESSITY OF USING ANCHORS

In this section we address the problem of how the source can estimate whether anchors are necessary in order to forward packets to the destination. If the source and the destination are well connected along the shortest geodesic line, the basic greedy mode of TRR without anchors works well: in this case the source and intermediate nodes have neighbours that are closer to the destination. Otherwise, if the distribution of nodes from the source to the destination is such that greedy forwarding is not possible, packets may travel along long paths in the perimeter mode. In this case, it is beneficial for the source to consume its resources to discover the anchored path to the destination.

A greedy path exists when the source and all the intermediate nodes find, among the neighbours, the next hop node that is closer to the destination. Given a transmitting node S and receiver X, the progress is defined as the projection of the line connecting S and X onto the line connecting S and the final destination. The notion of progress is illustrated in Figure 4. If there exists a greedy path from S to D then, source S and all intermediate nodes make forward progress towards D. In Section V-A we present the method for the estimation of the distribution number of hops along a greedy path; in Section V-B we describe how this result can be used in real networks.

A. Estimation of distribution of number of hops from the source to the destination

We introduce the following assumptions. First, we assume that nodes in the network are distributed as a two-dimensional Poisson point process with density λ, i.e., the probability of finding i nodes in an area of size A is equal to: $(\lambda A)^i exp(-\lambda A)/i!$, i = 0,1,2,3 ... Second, we assume that all nodes have an equal transmission range (R). And third, we assume that progress made at different hops is independent. This assumption is reasonable in the case of ad hoc networks where the network topology is changing either because of node mobility or because nodes are going up and down (e.g., in sensor networks). In this case, we assume that the network topology is redrawn at every hop that receives the packet.

Given the first assumption, the average number of nodes (N) within transmission range R is then $N = \lambda\pi R^2$.

Let's the random variable N_{SD} represents the number of hops between S and D along the greedy path that connects them. The distance between S and D is equal to d. We are interested in the conditional distribution of N_{SD}, given that it is finite. Let's denote this distribution as $P_r(N_{SD} > k \mid N_{SD} < \infty)$.

Here is how we find this distribution.

In order to reach D, S sends the packet to X because X is closest to D among all neighbours of S (for illustration see Figure 4). In this way the number of hops from S to D is equal to one plus the average number of hops from X to D. This can be expressed by the following equation:

$$N_{SD} 1_{N_{SD}<\infty} = 1_{N_{SD}<\infty} + N_{XD} 1_{N_{XD}<\infty} \tag{1}$$

Let's random variable Z be the progress for the transmission from a node to its neighbour that reduces the distance to the destination the most. Let's $G_d(z)$ denote the conditional distribution of the progress z at the node where the distance to the destination D is equal to d assuming that N_{SD} is finite. Then $G_d(z)$ is given by the following equation.

$$G_d(z) = P_r(Z \leq z \mid N_{SD} < \infty) = \frac{P_r(Z \leq z, N_{SD} < \infty)}{P_r(N_{SD} < \infty)} \tag{2}$$

where,

$$P_r(Z \leq z, N_{SD} < \infty) = \int_0^z f_d(u) P_r(N_{XD} < \infty) du \tag{3}$$

In Equation (3) we denoted with $f_d(u)$ the density function of the progress u made in one hop when the distance to the destination is equal to d.

From (2) and (3),

$$\frac{dG_d(z)}{dz} = dP_r(Z \leq z \mid N_{SD} < \infty)/dz = \frac{f_d(z) P_r(N_{XD} < \infty)}{P_r(N_{SD} < \infty)} \tag{4}$$

We can obtain the probability of existence of the greedy path from S to D by using the following recursive equation.

$$P_r(N_{SD} < \infty) = \int_0^R P_r(N_{XD} < \infty) f_d(z) dz \tag{5}$$

In the following we present how the density function $f_d(z)$ of the progress made in one transmission is determined.

Since, our assumption is that the progress performed at two hops is independent, i.e., the distribution of the progress at the current node does not depend on the progress made in the previous hops, then the probability distribution function of Z is determined as follows (for the illustration see Figure 4):

$$F_d(z) = P_r(Z \leq z) = P_r(\text{no nodes in } A_z) = e^{-\lambda A_z} \quad , 0 < z \leq R \tag{6}$$

In (6) we denoted with A_z the excluded region without nodes and the surface of this region is equal to the sum of two surfaces P_1 and P_2, given by (8) and (9). Under the assumption that progress made at different

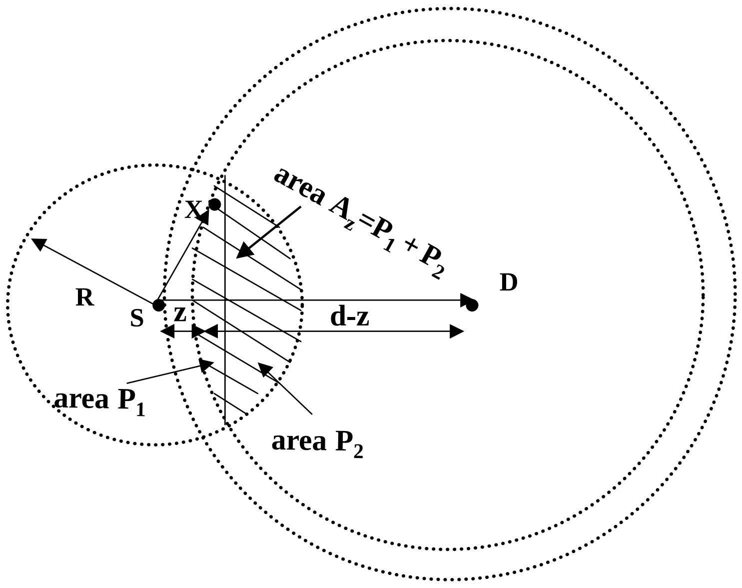


Fig. 4. Progress in distance made in transmission from S to X is equal to z. Three circles are presented: the first centered at S and radius R, the second centered at D and radius d, and the third centered at D and radius $d-z$. The progress from S to D is less than z when there is no nodes in shaded area A_z.

hops is independent, the excluded region depends only on the current node distance to D, but not on other excluded areas.

$$A_z = P_1(z) + P_2(z) \tag{7}$$

$$P_1(z) = R^2(\arccos(a) - a\sqrt{1-a^2}),\, a = \tfrac{R^2+d^2-(d-z)^2}{2dR} \tag{8}$$

$$P_2(z) = (d-z)^2(\arccos(b) - b\sqrt{1-b^2}),\, b = \tfrac{d^2+(d-z)^2-R^2}{2d(d-z)} \tag{9}$$

Then we can write the probability density function of Z as,

$$f_d(z) = \tfrac{dF_d(z)}{dz} = -\lambda e^{(P_1(z)+P_2(z))}\left(\tfrac{\partial P_1(z)}{\partial z} + \tfrac{\partial P_2(z)}{\partial z}\right) \tag{10}$$

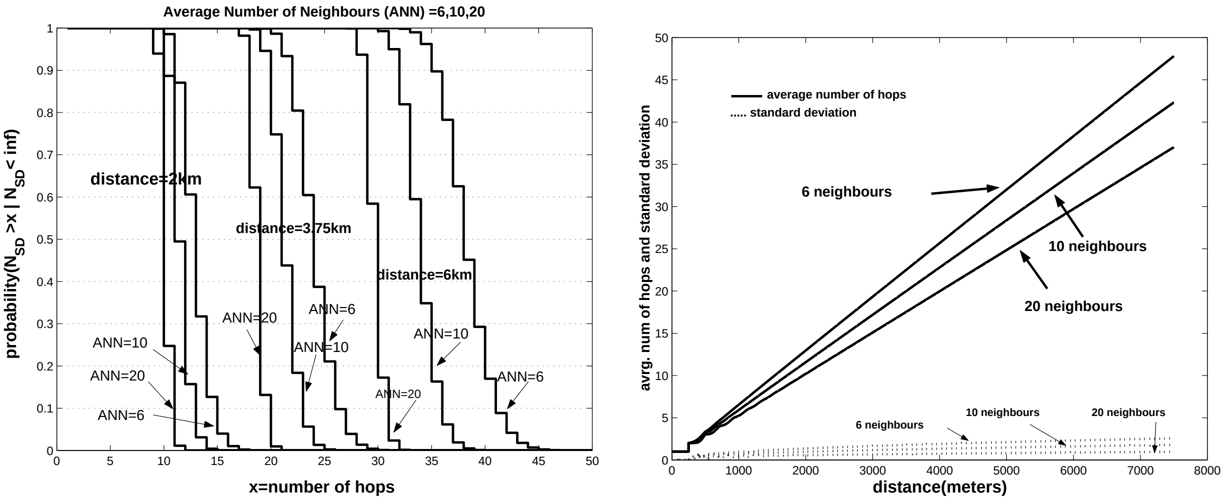
where,

$$\begin{aligned} \tfrac{\partial P_1}{\partial z} &= \tfrac{2(z-d)R}{d\sqrt{1-a^2}} \\ \tfrac{\partial P_2}{\partial z} &= 2(d-z)b\sqrt{1-b^2} - 2(d-z)\arccos(b) - \tfrac{\sqrt{1-b^2}}{d}(2dz - z^2 - R^2) \end{aligned} \tag{11}$$

Now, as we have presented all necessary elements to calculate $G_d(z)$, we can obtain the distribution of number of hops. To do so, we use the following recursive equation:

$$P_r(N_{SD} > k \mid N_{SD} < \infty) = \textstyle\int_0^R dG_d(z) P_r(N_{XD} > (k-1) \mid N_{XD} < \infty) \tag{12}$$

We solved numerically Equation (12) using the method that we provided. At first, we numerically find probabilities of existence of the greedy path, between the source and the destination for different distances between them, by using Equation (5). The initial values for probabilities used in this recursive equation are: $P_r(N_{SD}(d) < \infty) = 1$, for $d \leq R$. For $d \geq R$, we numerically solve the integral in (5) recursively using the values for probabilities that are already obtained for distances in range $(d - R, d)$. Once the probability in (5) is known, we can enter these values in (4) to obtain $dG_d(z)$. Then, the distribution of number of nodes is numerically obtained from (12) where the initial values are $P_r(N_{SD}(d) > 0) = 1$, for all values of d. The result for hop number distribution for different values of node density is presented in Figure 5 (a). These results are obtained assuming that the transmission range is the same for all nodes and is equal to 250 meters.



(a) (b)

Fig. 5. (a) Distributions of number of hops for various node densities (average number of neighbours is 6, 10 and 20) and distances (2km, 3.75km, and 6km), (b) Average number of hops and standard deviation obtained numerically for different node densities

We are now interested in how the obtained distribution is far from the normal distribution. To check that, we found the average number hops and standard deviation of number of hops from S to D along the greedy path where the distance between S and D is equal to d.

The average value of this random variable is denoted as $m(d) = E[N_{SD} \mid N_{SD} < \infty]$. Obviously,

$$m(d) = 1, \text{ for } d \leq R$$

For $d \geq R$, $m(d)$ is given with the following recursive equation:

$$m(d) = 1 + \textstyle\int_0^R m(d-z)dG_d(z) \tag{13}$$

Equation 13 is derived as follows. The progress that is made by forwarding the packet from S to X is equal to z, that is, at X the distance to D is reduced by z. At X, it remains the distance $(d - z)$ to reach

D. Since the progress that can be made in one transmission is between 0 to R, the integral in Equation (13) calculates the average number of hops from X to D, averaged on the progress that is made from S to X.

Standard deviation (σ) of N_{SD} can be obtained as follows:

$$\sigma^2 = E[N_{SD}^2 \mid N_{SD} < \infty] - E^2[N_{SD} \mid N_{SD} < \infty] \tag{14}$$

Consider events such that $N_{SD} < \infty$. Then, from Equation 1, on this set we can derive: $N_{SD}^2 = 1 + 2N_{XD} + N_{XD}^2$.

Then,

$$E[N_{SD}^2 \mid N_{SD} < \infty] = 1 + 2\int_0^R E[N_{XD} \mid N_{XD} < \infty] dG_d(z) + \int_0^R E[N_{XD}^2 \mid N_{XD} < \infty] dG_d(z) \tag{15}$$

Introducing (13) in (14) we obtain:

$$E[N_{SD}^2 \mid N_{SD} < \infty] = -1 + 2E[N_{SD} \mid N_{SD} < \infty] + \int_0^R E[N_{XD}^2 \mid N_{XD} < \infty] dG_d(z) \tag{16}$$

And finally standard deviation σ is given as,

$$\sigma^2 = -1 + \int_0^R E[N_{XD}^2 \mid N_{XD} < \infty] dG_d(z) + 2E[N_{SD} \mid N_{SD} < \infty] - E^2[N_{SD} \mid N_{SD} < \infty] \tag{17}$$

We have numerically solved Equation (13) to obtain the average hop count for different values of the distance from source to destination. The initial conditions used in (13) are: $E[N_{SD}(d) \mid N_{SD} < \infty] = 1$, for $d \leq R$.

In a similar way, we numerically get values of the standard deviation of the number of hops, given the distance between the nodes. The initial conditions used in (17) are: $E[N_{SD}^2(d) \mid N_{SD}(d) < \infty] = 1$, for $d \leq R$.

Figure 5 (b) presents the average hop number and standard deviation as a function of distance for different values of node densities.

The distribution of the number of hops for node density equal to 10 neighbours is presented in Figure 6 (a). In the same figure, normal distribution is presented with the mean value and variance equal to the values obtained from Equations (13) and (17). We see from Figure 6 (a) that the distribution of the hop number is close to the normal distribution for various values of distances between the source and the destination. Thus, the distribution of the number of hops can be modeled by the normal distribution.

We also verified our theoretical results by simulations. We performed a number of experiments in the fixed network. Nodes in the network are randomly placed according to the Poisson distribution with the given density. In such a network, since it is fixed, the progress made in different nodes is not independent; excluded areas in Figure 4 are not independent. For every two nodes in the network, we found the number of hops of the greedy path that connects them, if such a path exists. The crosses in Figure 6 (b) present the obtained length of the greedy path as a function of the distance between two nodes. We can see from Figure 6 (b) that number of hops obtained in experiments fall closely into a 95% confidence interval obtained theoretically. Our simulations verified that the number of hops obtained theoretically, where it is assumed that progress in different hops are independent, are close to experimental results within the fixed network where this assumption is not valid. Therefore, we conclude that obtained theoretical results will also be valid in real ad hoc networks in spite our simplifying assumptions.

B. How the obtained results can be applied in a real ad hoc network

In order to estimate the number of hops to the destination D along a greedy path, source S should first estimate the density of the nodes in a network. As the first attempt, we propose that S determines the density of nodes in its transmission range from the information in its local routing table, and we assume that the same density applies to the whole network. Knowing the geographic distance to D, S finds the distribution of the number of hops to D by applying the results that we have developed in Section V-A.

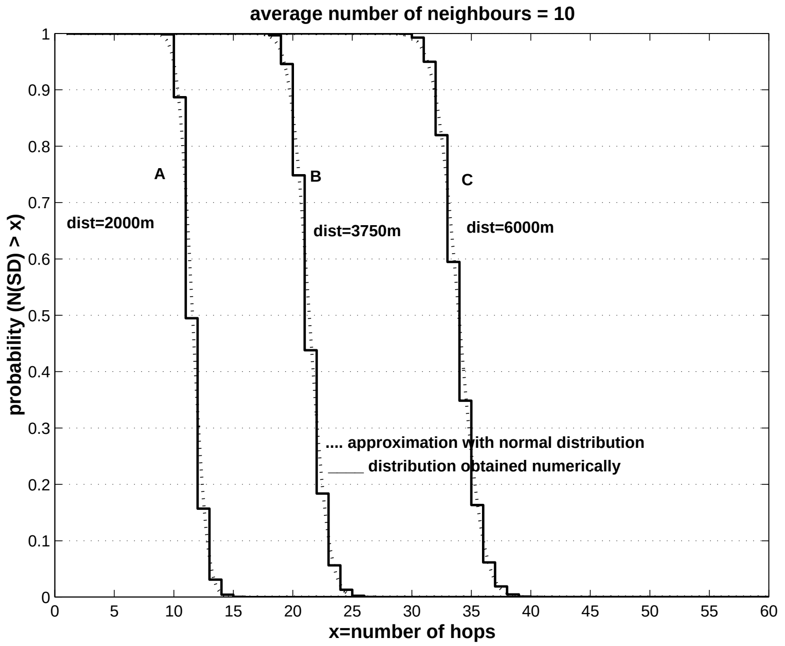
Then, S sends $explorer$ packets to D that are routed using TRR without anchors. D is supposed to send back to S the response of how many hops it took the explorer packet to reach from S to D. Then, S can make conclusions about the existence of a greedy path from S to D.

For example, let's assume that S estimates that the average number of neighbours is ten, and the distance to D is equal to 3750m. Then from Figure 6 (a) the probability that the number of hops from S to D is higher than 23 is equal to 5%. Therefore, if S learns that the explorer packets have taken more than 23 hops to reach D, S may conclude with a high probability that the greedy path from S to D does not exist.

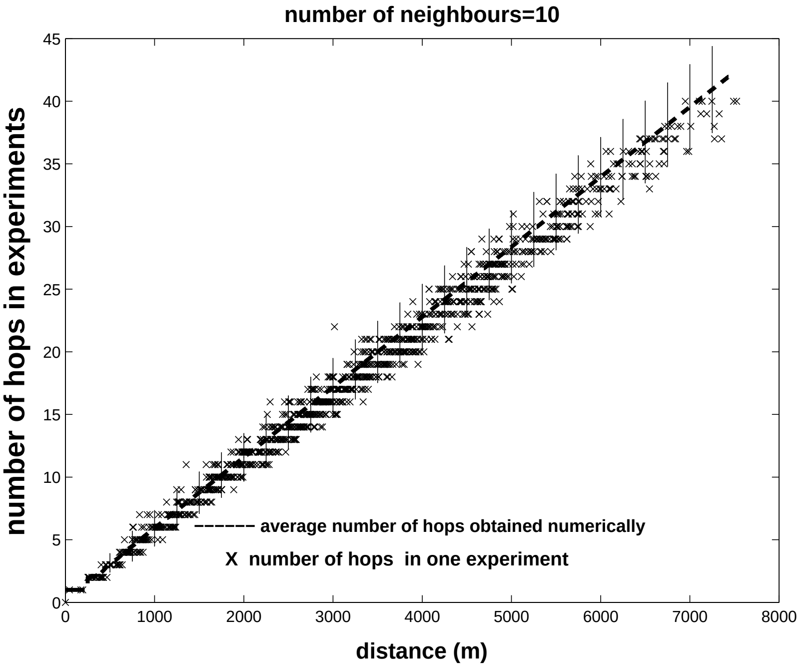
The assumption that the node density in the network is uniform may not be true. If this is the case, the distributions of number of hops for different node densities are taken into account (when evaluated whether explorer packets have taken a greedy path or a perimeter-mode packet forwarding has taken place). Figure 5 (a) presents distribution of number of hops for three values of the node density (average number of neighbours is 6, 10 or 20) and three different distances are taken (2km, 3.75km and 6km). We can see the larger distances are, the bigger difference is in hops distributions for different nodes densities. For example, when the distance between the source and the destination is 6km, if the explorer packets do not take more than 45 hops to reach the destination (that corresponds to the lowest density of the average of 6 neighbours), then the source may consider that with a high probability there is the greedy path to the destination.

VI. PERFORMANCE EVALUATION OF TERMINODE ROUTING

In this section we validate the operations of terminode routing by using simulations. We implemented and simulated the terminode routing protocol in GloMoSim[37]. The IEEE 802.11 Medium Access Con-



(a)



(b)

Fig. 6. (a) Distribution of number of hops for average number of neighbours equal to 10, (b) Number of hops obtained in experiments

trol(MAC) protocol is used; it implements the Distributed Coordination Function (DCF)[11]. In all simulations, radio range is the same for every node, and is equal to 250 meters. The channel capacity is 2Mbits/sec. The propagation model, included in the GloMoSim simulation package, is the two-ray model. It uses free space path loss for near sight and plane earth path loss for far sight.

In our implementation of terminode routing, we made several design choices concerning sending of HELLO messages and the location management.

Nodes periodically broadcast HELLO messages in order for nodes to build their TLR routing tables. Every node maintains a HELLO timer that is set to one second. When a HELLO message is sent, a HELLO timer is reset. Each entry in the routing table expires after two seconds, if it is not updated. In order to reduce the routing overhead, caused by sending HELLO messages, we make promiscuous use of the network interface. Then, by disabling MAC address filtering, nodes receive all packets from all nodes in their radio range. Nodes that have data or control packets to send should defer sending HELLO messages, because data packets piggyback the HELLO message information.

However, in our implementation, not all data packets piggyback the HELLO message information. Because a node does not considerably change its location and its neighbours list, between two data packets, it is not meaningful to increase the size of all data packets with the HELLO message control information. We do as follows. A HELLO timer is set to 0.5 seconds. When a HELLO timer expires, if a node has a data/control packet to send within 0.5 seconds from the time when a HELLO timer has expired, the packet piggybacks the HELLO message information, and a HELLO message is not sent, otherwise, a HELLO message is sent. In both cases, the node resets its HELLO timer. To avoid synchronization of HELLO messages, a terminode jitters each HELLO message transmission by a random interval between 0 and 1 second.

In our implementation every node knows accurately its current location all the time. Location management is used at sources to learn the destination location. Our simulations verify the performance of terminode routing in two cases. In the first case location management is implemented, and performance results are influenced by location management overhead; in the second case evaluation is performed with idealized no-overhead location management.

Location management

We implement an on-demand location management scheme. Our aim is to give a fair evaluation of terminode routing taking into account location management overhead. Our location management scheme is a simple one. Some other proposed location management schemes can be reviewed at [25], [35].

Our location management is as follows: if the source has some data to send to the destination whose location is not known, a flooding based approach is used for destination location discovery. The method is similar to DSR source route discovery [9]. Once the communication between source and destination is started, location tracking is used for updating destination location.

Location Discovery. 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 neighbours. Inside the packet, S stamps its own location. Node X, which receives a location request packet and is not the destination, broadcasts the request to its neighbours. In order to avoid a redundant transmission of the request, X should broadcast a particular location request packet only once. A source of a location request control packet stamps the packet with a sequence number. 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 by TRR 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 by using TRR 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. In our simulations location reply wait timeout is set to 5 seconds.

Location Tracking. Once two nodes begin to communicate, the 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 the given time (in simulations of small networks this parameter is set to 10 seconds, while in big networks is 20 seconds). The source then re-initiates learning of the destination location. Inspired by LAR protocol, 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.

In the case of TRR with anchors, 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

In the simulations of larger mobile ad hoc networks (500 nodes) we evaluate terminode routing also in case of idealized location management. We assume an idealized location database where all nodes can know all other nodes' locations at all times with no control overhead. However, a source does not stamp data packets with the true location of the destination at all times. We examine the terminode routing performance when there are inaccuracies in location information. We assume that the source cannot know an exact destination location all the time: the destination has moved from the location retrieved by the source. Thus, it could happen that the source stamps the packet with a destination location that is no more exact. In our simulations, the source learns a destination location and uses this information for the time that we call *location information lifetime*. After this time, the source again acquires an exact destination location and uses it for another location information lifetime interval. Location information lifetime is a parameter can be set at the beginning of the simulation. In our simulation results location information lifetime is set to 5 seconds.

A. Evaluation in a small network with uniform node distribution

The goal of this section is to compare by means of simulations the performance of terminode routing versus two other routing protocols, AODV and LAR1 (LAR scheme 1), in a small ad hoc network uniformly populated with nodes. Because the simulation area is small and unobstructed, terminode routing uses TRR without anchors. Terminode routing is evaluated with location management overhead taken into account. AODV and LAR1 are chosen because they perform very well for a small ad hoc network, and they are based on different routing strategies. AODV does not use geographic locations. The control part of LAR1 uses geographic locations, while packet forwarding in LAR1 uses source routes as in DSR. Simulations of AODV and LAR1 are performed using the implementations that are included with the GloMoSim simulation package.

In our simulations we use the rectangular unobstructed simulation area of the size 2200 m X 600 m with 100 nodes. Nodes in the network are uniformly distributed; nodes are free to move in the whole simulation area according to the mobility model presented below. The simulated network is densely populated. The density of the network (75 nodes per square kilometer) ensures that TRR forwards most of the packets in a greedy mode. Nodes locations that are used in TRR are obtained using the location management method that we described in Section VI. The mobility model is the “random waypoint” mobility model[9]. In this model a node chooses one random destination in the simulation area. Then it moves to that destination at a random speed (uniformly distributed between 1-20 m/sec). Upon reaching its destination, the node pauses for the *pause time*, selects another random destination inside the simulation area, and proceeds as previously described. In our simulations we vary the pause time, which affects the relative speed of mobile nodes. Traffic sources are CBR (constant bit-rate). The source-destination pairs are randomly spread over the network. All data packets are 64 bytes long. We performed simulations with 40 source-destination pairs. The packet rate is fixed at 2 packets/sec. The flows are low-bitrate, and the network is not congested, because these simulations are meant to measure routing protocols behaviour, not the limitation of the IEEE 802.11 MAC for data packet capacity.

Simulations are run for 900 simulated seconds. Each data point represents an average of six runs with identical traffic models, but different randomly generated mobility scenarios. For all simulated protocols we use identical mobility and traffic scenarios.

We looked at three performance metrics that are used also 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* of data packets, which 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; and *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.

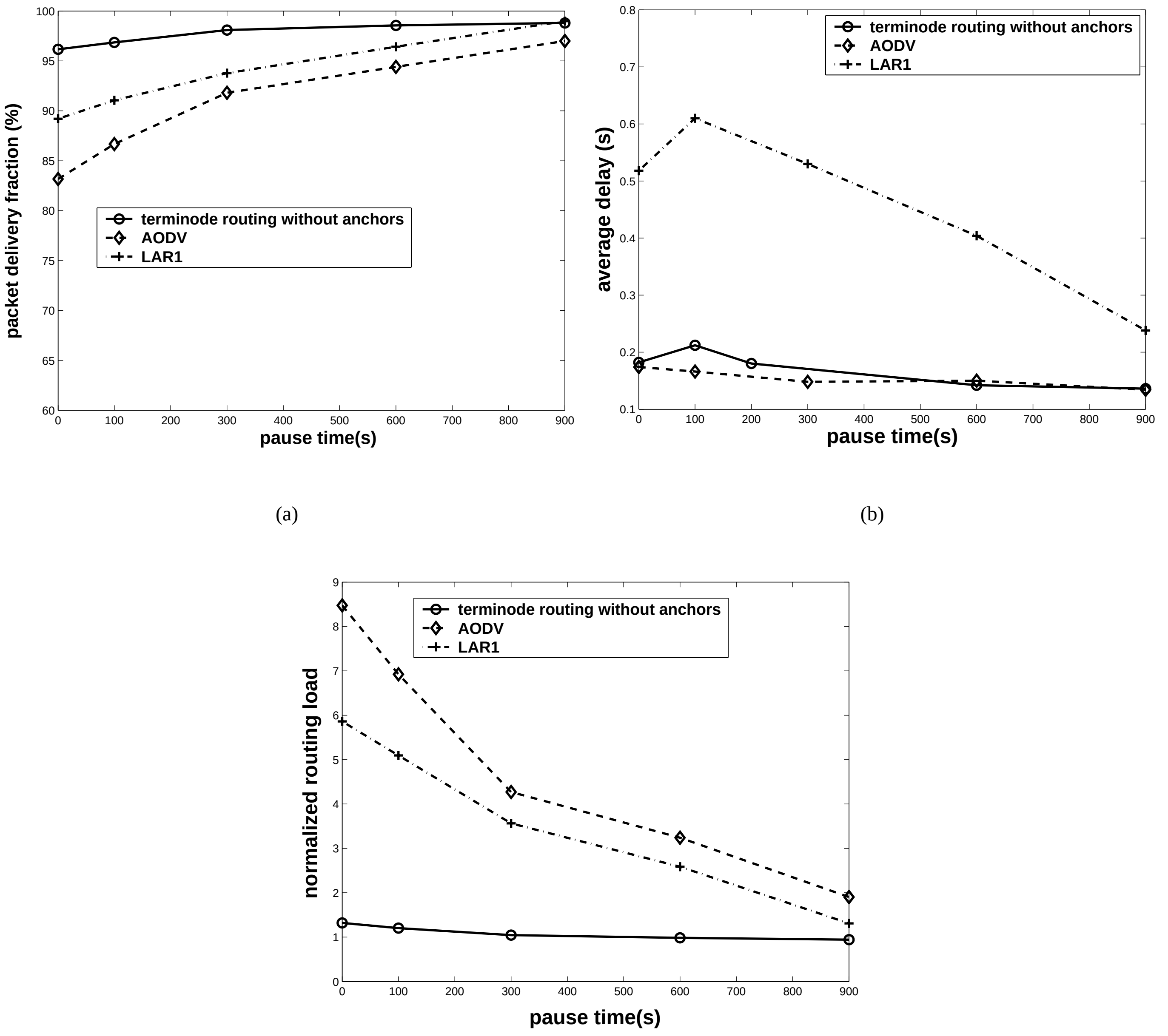
Simulation Results

Our simulation results presented in Figure 7, show the following.

Terminode routing is comparable to LAR1 in packet delivery fraction and both outperform AODV (Figure 7 (a)).

The delay experienced by LAR1 is higher than the one of AODV and of terminode routing (Figure 7 (b)). 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 favouring the least congested route.

When terminode routing is used, the delay due to buffering during the destination location discovery is critical in the initial phase when the source learns about the destination location. If the location management works well, and the source regularly receives destination location updates, packets are not buffered at the source waiting for the destination location. In the case of AODV and LAR1, delays caused by packet buffering during the route discovery are present every time a source has to (re)discover a route to the destination.



(a)

(b)

(c)

Fig. 7. (a) Packet delivery fraction, (b) Average data packet delay, (c) Normalized routing loads for the 100 node model and 40 sources

Figure 7 (c) shows nnormalized routinh load. Terminode routing has the smallest normalized load. Moreover, terminode routing has a stable normalized routing load for different pause intervals. The explanation is the following: first, every node proactively generates HELLO messages unless a packet (data/control) is sent, and these messages are received, but not forwarded by neighbours. The overhead due to HELLO messages is independent of the mobility rate of nodes. Second, routing overhead due to location management does not change very much with the increase of mobility Location update control packets are generated proactively by destinations and independently of mobility. We verified by simulations that in most cases location request packets flood the network only at the beginning when the communication between source and destination starts. After that, the mobility tracking (with sending of location reply packets) ensures that the source receives periodic updates of the destination location, without need to often flood the network.

LAR1 builds source routes while AODV relies on routing tables at each node in order to reach the destination. In both protocols, when a single link in the built route is broken, a new route should be built. Both AODV and LAR1 include flooding in order to build new or repair broken routes. LAR1 floods to the zone where the destination is supposed to be found (expected zone), while AODV does an expanded ring search type of flooding. For small values of pause time (higher mobility) more routes are broken and in order to repair them, AODV and LAR1 generate more routing overhead for higher mobility. The use of route caching and reduced search space for the desired route, helps LAR1 to keep routing load lower than in case of AODV.

Both AODV and LAR1 maintain routes to destinations. We verified, by simulations, that maintaining routes with many hops in mobile ad hoc networks is a difficult challenge. Terminode routing does not build the route to the destination; routing decisions are made locally at each node. We showed by simulations that this strategy is better than the strategy of building routes - provided that node density is high and sources can acquire accurate destinations location.

B. Evaluation of TRR with anchors in a large ad hoc network with non-uniform node distribution

TRR with anchors is evaluated within a relatively large simulation area where nodes are not uniformly distributed. In such networks there are regions within a simulation area where nodes cannot move to, and thus there are holes in nodes distribution. We show that it is beneficial to use anchors compared when anchors are not used, as in TRR without anchors, GPSR [24] and GFG [8].

In large ad-hoc networks with non-uniform node distribution we use the mobility model called “restricted random waypoint”. This model is introduced in [5] for large mobile ad hoc networks. The model reflects that in a large network, it is less probable that, for each movement, node selects a random destination within a very large geographic area. On the contrary, the random destination is selected within a small area for a number of movements, and then a movement is made over a long distance.

For the restricted random waypoint mobility model, we use a topology based on towns and highways. The model of the simulated area that consists of four towns is presented in Figure 8. Nodes’ movement inside a town is the random waypoint mobility model. It repeats such movements for a number of times set by the *stay_in_town* parameter. Then a node selects at random a destination within a new town and moves there (the new town is randomly chosen from a list of towns that are connected with the current town by a highway).

Once it reaches the new location, a node applies inside the new town the random waypoint mobility model for another *stay_in_town* time. This is a mobility model for so-called “ordinary” nodes. There are also a number of nodes that frequently commute from one town to another. Those nodes are called “commuters” and they ensure the connectivity between towns. The commuter’s movement model is the restricted random waypoint where *stay_in_town* parameter is equal to one.

We evaluated TRR with anchors in two cases: when GMPD (described in Section IV-B) or FAPD (described in Section IV-A) is used for anchored path discovery.

When GMPD is used, we assume that all nodes have a knowledge of a map of towns. A map defines town areas and the existence of highways between towns. Map distribution is out of the scope of this paper. TRR without anchors is used for packet forwarding when the source and the destination town are the same, or directly connected with a highway. Otherwise, for example, in Figure 8, when S is in the area of town 0 and D is the area of town 3, S sets the anchored path to consist of one anchor: center of town 1. TRR with anchors forwards the packet along the path that goes to town 1. Once the packet is close to the center of town 1, the packet is forwarded towards D. In the case when no anchors are used, the resulting path is much longer. Figure 8 illustrates that the packet is first forwarded in the greedy mode toward D until it reaches node $P1$, where perimeter mode starts. The packet is thus forwarded in perimeter mode until greedy mode resumes at node $G1$ ($G1$ that is closer to D than $P1$). Through the combination of greedy and perimeter mode the packet arrives to D. Figure 8 clearly illustrates the case where the usage of anchors give shorter paths.

With FAPD, the source triggers anchored path discovery if the destination is not TLR-reachable and the distance to the destination does not exceed three times the transmission range of this node. While waiting to receive an anchored path, the source uses TRR without anchors to send packets to the destination. Figure 8 illustrates FAPD with an example. Once the source issued the path request, and it does learn an anchored path within 10 seconds, the source issues the new request. The source keeps every acquired anchored path for 20 seconds. After this time the anchored path is considered invalid. FAPD responders are situated in town areas and do not move. Ten percent of nodes are configured as FAPD responders. Our simulation results take into account the FAPD control overhead. This overhead includes control messages for FAPD responders to discover friends, and well the overhead in delivery of path request and reply control packets. The following is an example of how FAPD responders learn about their friends. Nodes A, B and C are FAPD responders situated in towns 2,0 and 1 correspondingly. Assume that B discovers C as its friend by using the friends discovery method presented in Section IV-A. Then, A discovers B as a friend and B transmits to A its lists of friends. As the result, A has C as friend. At A, the anchored path to C is given with one anchor that corresponds to B’s location. FAPD responders refresh knowledge about their friends

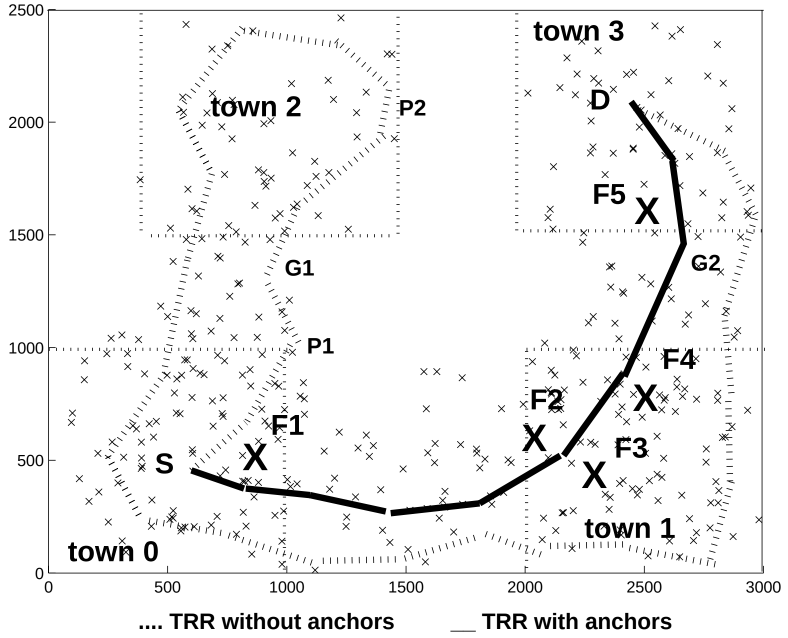


Fig. 8. Path of packet from source S to destination D without anchors (dotted) and with anchors (solid). When anchors are used, TRR gives a shorter path than TRR without anchors. An example of FAPD is also presented: 1) S starts FAPD by sending the request for anchored path; this request is received 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}, 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 learnt anchored path. Notice that data packets do not go through FAPD responders.

periodically with the time interval equal to 50 seconds.

Simulation Parameters

We conducted simulations of 500 nodes forming an ad hoc network presented in Figure 8. The size of the simulated area is 3000m x 2500m.

The mobility model is the restricted waypoint mobility model. For each movement, a node takes a random speed that is uniformly distributed between 1-20m/s; before each movement, a node pauses for some pause time. There are 200 ordinary nodes and 300 commuters. Recall that ordinary nodes stay within a boundary of a single town for a *stay_in_town* number of movements. Commuters are fast moving nodes, which are introduced in simulations in order to ensure a connected network. Their role is to relay packet on behalf of ordinary nodes. The *stay_in_town* parameter is set to 2 for ordinary nodes, and it equal to 1 for commuters.

We ran simulations with different pause times of ordinary nodes. This parameter define different degrees of ordinary node mobility. A longer pause time means that ordinary nodes are less mobile. We consider different mobility rates of ordinary nodes because this is the set of nodes where all traffic sources and destinations come from. In our simulations, commuters have higher mobility than ordinary nodes. For their movements they take a random speed that is uniformly distributed between 1-20m/s, and pause time equal to 0 seconds.

Traffic sources are continuous bit rate (CBR). The source-destination pairs are spread randomly over the network. All CBR sources send two packets per second, and uses 64-byte packets. All communication patterns are peer-to-peer. CBR connections are started at times uniformly distributed between 400 and 500 seconds (starting from initial locations at the beginning of the simulation this time is enough for nodes to establish the network of towns and highways), and they last until the end of simulation. All simulations last for 1200 seconds.

When FAPD is used for path discovery, we set that there are 50 FAPD responders distributed with four town areas. FAPD responders do not move.

Simulation Results

In all figures illustrating our simulation results each data point presents an average of at least six simulations with identical traffic models, but different randomly generated movement patterns.

Terminode routing is evaluated in two cases: with included location management overhead and with idealized no-overhead location management where the source learns the accurate destination location every 5 seconds (described in Section VI). We compare terminode routing performance to GPSR, AODV and LAR1.

Packet delivery fraction is shown in Figure 9. We observe that TRR with GMPD outperforms all protocols. As expected, FAPD delivers less packets than GMPD, but still performs better than GPSR. As pause time increases the difference between packet delivery fraction in FAPD and GPSR increases. GPSR does not use anchors, which gives complex and long packet paths when source-destination pairs are in towns not connected with a highway (Figure 8). For those packets there is a higher probability that they will be dropped. Moreover, GPSR often uses perimeter-mode to forward these packets. Simulations have shown that this mode suffers from the looping problem in mobile ad hoc environment and it frequently happens that the packets that are trapped in a loop are dropped due to TTL expiry (this problem is described in [7]). With smaller pause times (degree of nodes mobility is higher), those source-destination pairs where GPSR gives a small fraction delivery can move to towns where GPSR performs better. This explains why GPSR delivers less packets for higher pause times.

We see that location based routing protocols perform better than AODV and LAR1. LAR1 delivers less packets than AODV. While LAR1 has more success in smaller networks (Figure 7), in the large network topology with towns, LAR1 poorly adjusts its search region. Since connections take place between nodes that can be in different towns, the LAR1 search zone is often comparable to the entire network.

Figure 9 (b) illustrates the packet delivery fraction of the location-based routing protocols when idealized location management is used. We observe that TRR with GMPD, FAPD and GPSR deliver around 10% more packets than in the case when location management is included.

The normalized routing load is shown in Figure 10. We can observe that location-based routing protocols greatly reduce routing overhead compared to AODV and LAR1. AODV and LAR1 frequently use flooding as a means to build or repair broken routes, which causes a large overhead.

Location based routing protocols never use flooding for route discovery or repairing. The only case where flooding can occur is during location discovery. But, once the destination location is known, location tracking is used to update location information. As long as location tracking is successful, flooding is not used. Note that the success of location tracking depends on the quality of paths between the communicating nodes. When the path (with/without anchors) does not work well, the destination has less success in sending its location update to the source, and the source floods the location request packet more often, thus increasing the routing load. As is illustrated in Figure 9 (a) GPSR uses paths without anchors that perform worse than anchored paths. This causes more location management overhead than in case of terminode routing with anchors (FAPD and GMPD) (Figure 10 (a)).

Comparing Figures 10 (a) and (b) we see that in the case with location management included, routing overhead in location-based routing protocols is doubled compared to no-overhead location management. In both cases, FAPD discovers anchored paths with low overhead. We observe that FAPD has only a small additional overhead compared to GMPD.

With idealized location management, TRR with FAPD or GMPD has stable routing load for different pause times. The routing overhead is due to HELLO messages and the control packets in path discovery protocol. Since every node periodically generates HELLO messages, the overhead due to HELLO messages is independent of mobility rate. When GMPD is used for anchored paths discovery it is assumed that the network map is known to all nodes with no control message overhead. When FAPD is used for anchored path discovery, its amount of routing overhead is also independent of the mobility rate. This overhead is due to FAPD responders' friends management and the FAPD protocol. In our implementation FAPD responders periodically maintain their friendship connections. They run FAPD only when a new path is demanded, thus independently of mobility. Therefore, the pause time parameter that influence the mobility rate does not have a big impact on the routing overhead.

Figure 11 illustrates average end-to-end delays. We see that in case of idealized location management (Figure 11 (b)) GPSR has smaller delay than TRR with anchors. The reason is that GPSR has a lower packet delivery fraction than when anchors are used and the average delay counts only for delivered packets. We observed with GPSR a large number of the packets that take long paths are dropped, and that most of the packets that are received at the destination experienced short paths, with short delays. We observe that when location management is included, the delay is increased for all protocols (Figure 11 (a)). The delay then includes packet buffering time before the source learns the destination location. With GPSR, the buffering time tends to be longer than in case of FAPD since location tracking is less successful.

C. Evaluation of usefulness of Terminode Local Routing (TLR) and of Restricted Local Flooding (RLF) in the case when accuracy of location information is low

We use simulations to evaluate how TLR helps to cope with location information inaccuracies.

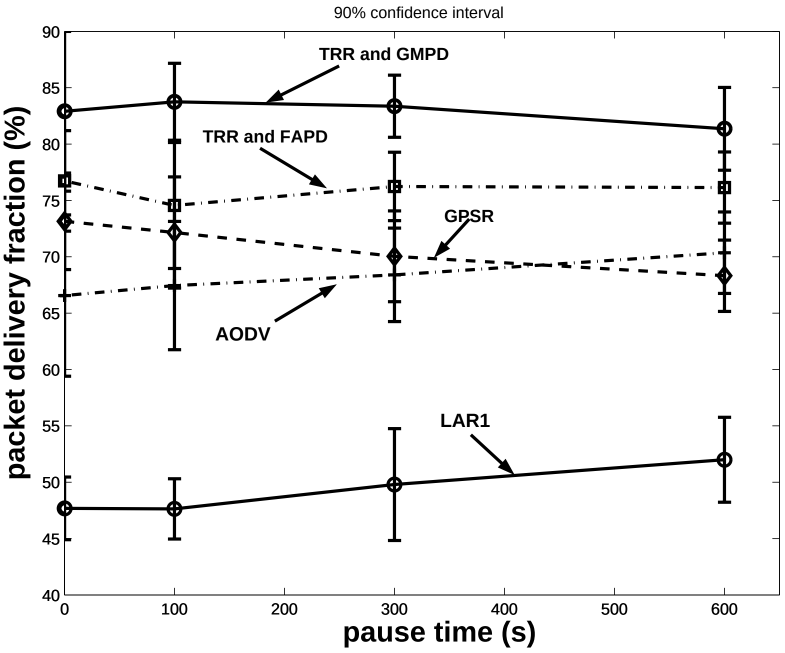
Remind that TRR without anchors is similar to GPSR. Note that the only difference between the two protocols is the following. GPSR uses the destination location for making packet forwarding decisions for the whole way until the packet arrives at the destination. TRR without anchors does the same until some intermediate node finds the destination is TLR-reachable and it switches to TLR. In the case when TRR without anchors cannot be terminated because the destination has moved from its reference location and no node finds the destination to be TLR-reachable, TRR termination is done by limitation of lifetime of a packet (“Use RLF” bit is not set).

Note that TLR is used in a two-hop neighborhood and does not need additional routing overhead compared to GPSR. The only additional requirement when TLR is used, is that all nodes keep in their routing tables information not only about immediate neighbours, but also about their two-hop neighbours.

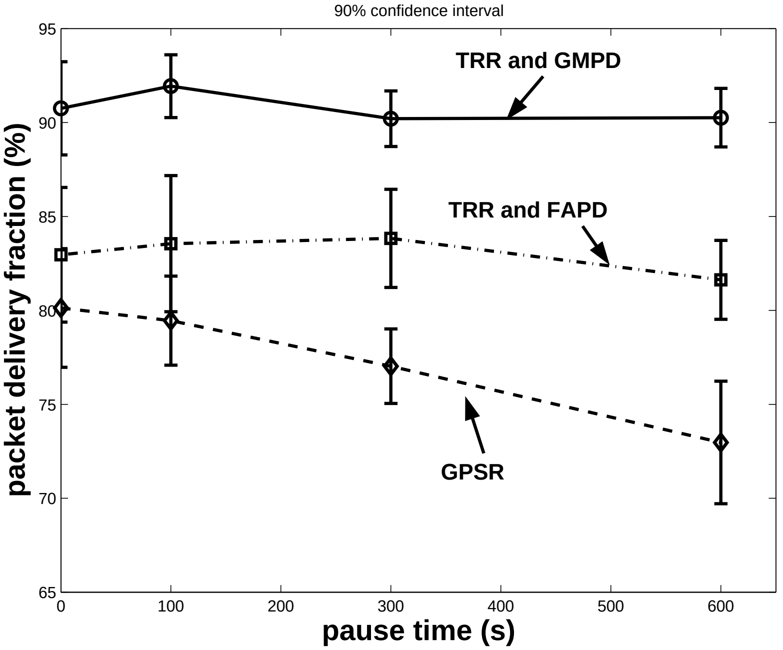
In our simulations we use a large network of 600 nodes, with the uniform node distribution. The simulation area is a square of the size 2900m X 2900m. The simulated network is quite dense; in this case we verified that TRR mostly forwards packets in the greedy mode. We simulated 20 CBR traffic flows. Each CBR flow sends two packets per second. Only 64-byte packets are used. Nodes move according to the “random waypoint” mobility model. In our simulations, a speed is uniformly distributed between 0-20m/s and the pause time is 10s.

In these simulations we do not include a distributed location database. However, we use the *location information lifetime* parameter, which is defined in Section VI, the time interval as which the source learns the exact destination location.

The two protocols are evaluated for different values of location information lifetime parameter. We simulated six different randomly generated motion patterns. Figure 12 (a) presents the average of packet delivery fraction for six simulation runs. This figure shows that for smaller location information lifetimes (less than 20 seconds), the packet delivery fraction is similar with terminode routing and GPSR. However, for higher location information lifetimes (lower precision of location information) terminode routing gives better delivery fraction than GPSR. Therefore, we conclude that when using TLR, routing is more robust in the case of positional errors and inconsistent location information. With the size of the TLR-area equal to two hops,

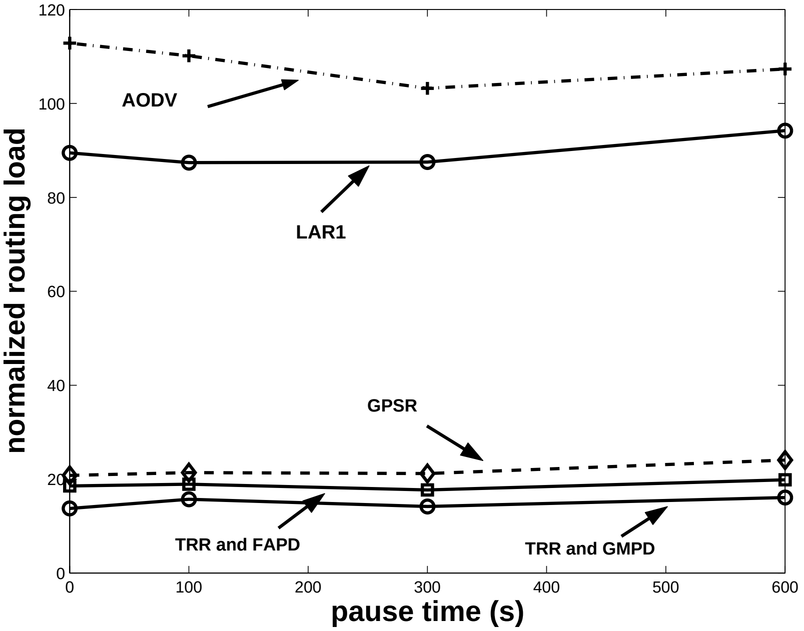


(a)

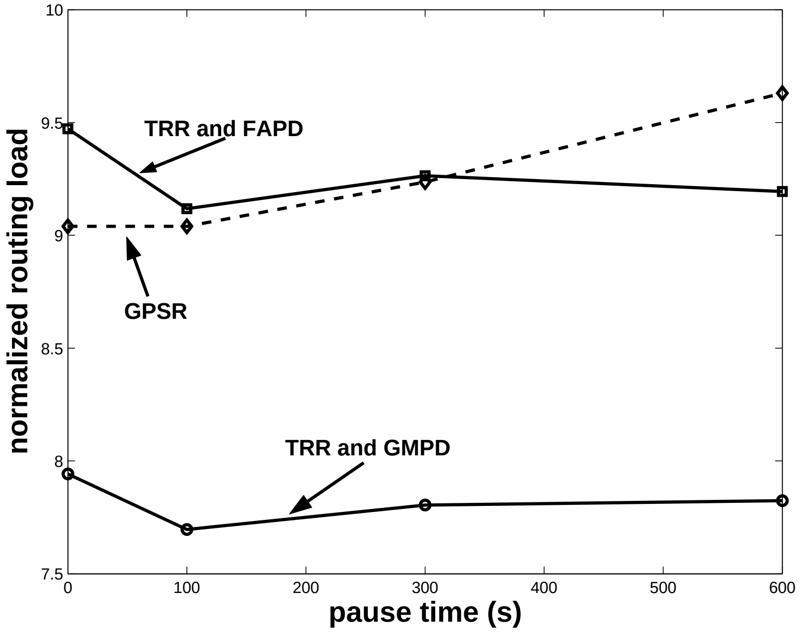


(b)

Fig. 9. Packet delivery fractions for 500 nodes network; (a) with location management, (b)with idealized no-overhead location management



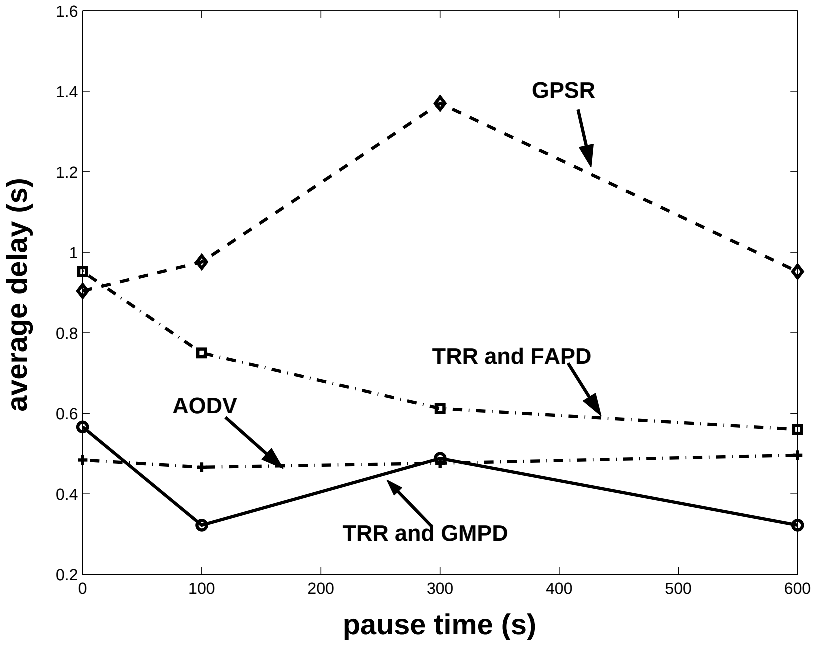
(a)



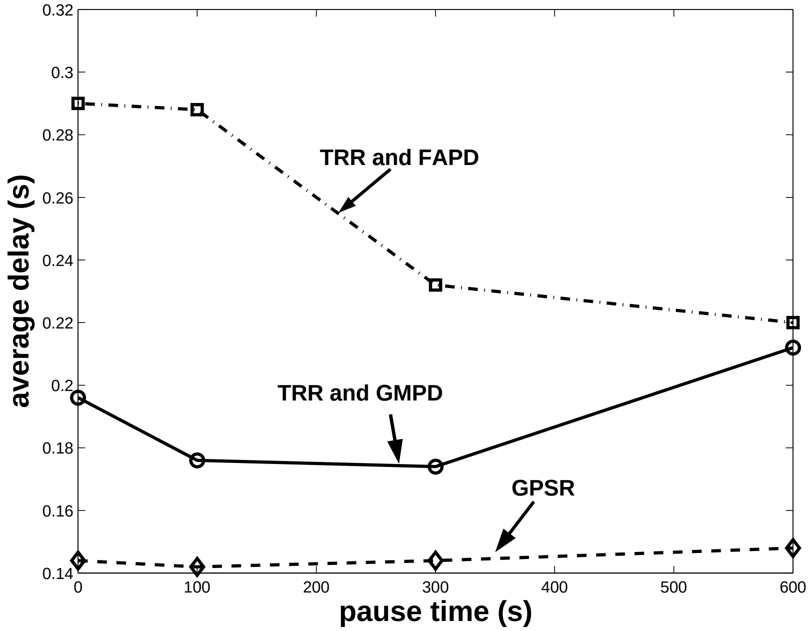
(b)

Fig. 10. Normalized routing load for 500 nodes network (a) terminode routing with location management (b)terminode routing with no location management overhead

routing continues to successfully deliver packets to destinations even if the location management is not able to provide the locations updates more frequently than one minute.



(a)

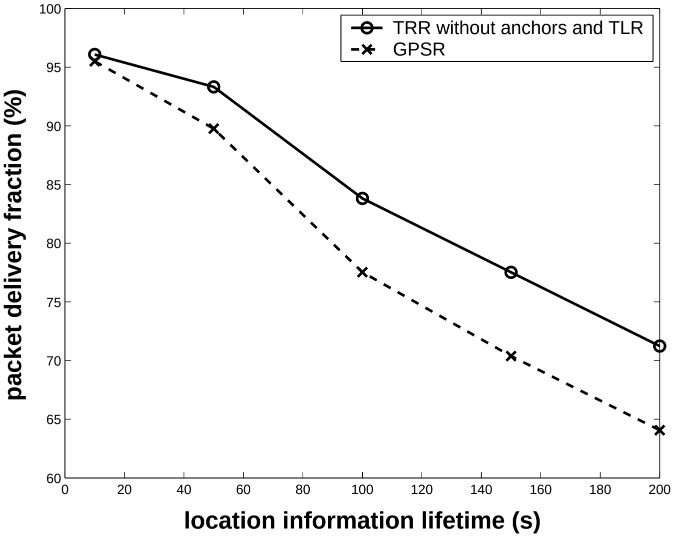


(b)

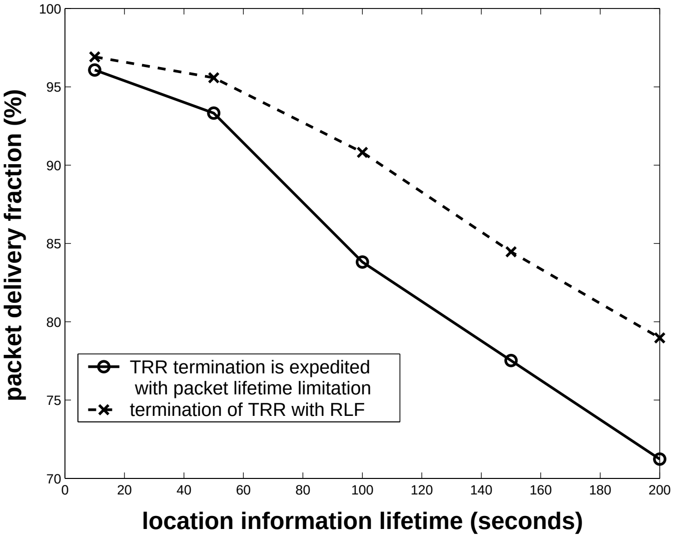
Fig. 11. Average data packet delays for 500 nodes network; (a) with location management, (b)with idealized no-overhead location management

Example of usefulness of RLF

RLF is used in situations where the accuracy of location management is low and TLR alone is not sufficient (Section III-E). Recall that the source sets "Use RLF" bit in packets to destinations when the source suspect



(a)



(b)

Fig. 12. (a) Figure shows that using TLR results in higher packet delivery fraction than in the case when only location based routing is used, (b)When Restricted Local Flooding is performed, fraction of received packets is higher than without RLF (when TRR is terminated by the packet lifetime limitation)

destination locations are not accurate.

We use the same simulation settings as previously where we evaluate TLR. Node X that initiates RLF, sends six copies of the packet towards six different geographic points around X. In this way, we increase the expected region where destination can be found. Therefore, we increase the probability that some of the flooded packets will arrive at the destination. Figure 12 (b) compares packet delivery fraction when TRR termination is performed in two different ways: in the first case termination of TRR is done by limitation of lifetime of a packet (TTL field is set to 3), and the second case corresponds when RLF is used. Figures 12 (b) illustrate the improvements in the fraction of delivered packets when RLF is used compared to the case when RLF is not used. We see that RLF is especially beneficial for larger values of location information lifetime (e.i., the destination location is less accurate because the source gets the destination location updates less frequently).

D. Scalability Analysis

We analyze the scalability of terminode routing based on the theoretical model described in [19], [20]. The analysis consists in 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 (3) sub-optimal (capacity wasted due to sub-optimal 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 [20], 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 3 scaling factors: λ_{lc} (mobility rate), λ_t (total traffic) λ_N (number of nodes). For a factor i, the analysis in [20] defines

$$\rho_i \overset{\text{def}}{=} \limsup_{\lambda_i \to +\infty} \frac{X_{ov}}{\lambda_i} \leq \Psi_i \overset{\text{def}}{=} \limsup_{\lambda_i \to +\infty} \frac{T}{\lambda_i}$$

where X_{ov} is the total capacity consumed by overhead and T the theoretical capacity. Thus ρ_i [resp. Ψ_i] the scaling exponent for overhead of factor i [resp. for the theoretical capacity]. The scaling analysis consists in comparing ρ_i to Ψ_i. We have [20]: $\Psi_{lc} = 0$, $\Psi_t = 1$, $\Psi_N = 1.5$.

Proactive overhead. There are two types of routing information propagated by terminode routing: local routing and friends position. Each node has full knowledge of its 2-hop neighbours via periodic HELLO messages dissemination. Every second, each node generates an average number H of HELLO message per neighbour (length = l_H). Let n be the average number of neighbours; each message is sent to $O(n^2)$ nodes, and we can say that $n^2 \ll N$. Thus, the total proactive cost generated by TLR is $O(l_H n^2 H \lambda_N)$ every second. Now, as l_H and n both depend only on node density, which is bounded in average, and H is fixed, we obtain that it is $O(\lambda_N)$ every second.

In addition, there is another proactive factor induced by the mechanisms for discovering and maintaining friends. If mobility tracking is used to track existing friends, we have to consider that each node has in average f friends. Assume that the tracking message has length l_{MT} and is transmitted with rate TM; as the average number of hops between two friends is $O(\sqrt{N})$, the total proactive cost generated by mobility tracking is $f * \sqrt{N}\lambda_N l_{MT} MT$ every second. Considering also that $f \ll N$, and that the tracking rate MT is $\frac{M}{\sqrt{\lambda_N}}$), we finally find $O(N)$. The number of transmissions of a request will be, at most (when no FAPDP responders are encountered, and no nodes receive twice the request), $\sum_{i=1}^{6} 4^i$ (for 4 requests up to 6 hops). However, we can note that this protocol is activated only when a node needs to add new friends to its list. So, the average generation rate of this message is very small. Moreover, not all the nodes will use it (only the FAPDP responders). Therefore, we can neglect this factor, and the total proactive cost generated by terminode routing is given by TRR and mobility tracking. We can conclude that the terminode routing proactive cost per second is $O(\lambda_N)$.

Reactive Overhead We consider the cost for a new path for both TRR default mode and TRR with anchors (with the support of FAPD or GMPD to discover an anchored path). If TRR without anchors is used, the request (of length l_R) is forwarded among neighbours until it reaches the destination. However, this is included in the message forwarding and there is no reactive overhead here. With FAPD, the request is sent to friends. In the worst case, all the f friends of a node need to be contacted each time until the path of average length $\sqrt{\lambda_N}$ is completed. As there are $s\lambda_N$ new messages generated per second, the reactive cost of FAPD is $s\lambda_N N h_f f$, which is $O(\lambda_N\sqrt{\lambda_N})$. With GMPD is employed, maps are used for discovering the anchor points and there is not reactive overhead. Thus, the reactive cost per second is at most $O(\lambda_N\sqrt{\lambda_N})$.

Sub-optimal Path Overhead. In [36] it is shown that the probability p of a bad next hop is independent of the traffic, the size and approximately constant for different distances to the destination. The same reasoning apply to terminode routing. Therefore, the cost for a single hop is $\frac{p}{1-p}l_d$, where l_d is the data packet length. p depends on the probability p_m that the destination moves out from the area where it can still be found (considering the method for dealing with location inaccuracy) and the probability p_{nr} that this information has not been corrected. The latter is proportional to M for TRR (and H for TLR). p_m is proportional to λ_{lc}. We can approximate the contribution of RLF mechanism with the fact that, if the destination has not moved further than four hops, it is still reachable via RLF. And this can be approximately distributed with λ_{lc}^4. Thus, for TRR, $p_m = M\lambda_{lc}^4$, and we have a cost of $\lambda_t\lambda_N\sqrt{\lambda_N}\frac{M\lambda_{lc}^4}{1-M\lambda_{lc}^4}$. Consecutive routing decisions are independent, as they are based on local information and consecutive remote geographical information (i.e. with anchor paths, the routing between two anchors is performed with local information). Thus, as we have $\lambda_t\lambda_N$ packets generated per second for paths of average length $O(\sqrt{\lambda_N})$, the sub-optimal paths cost grows as $O(\lambda_t\lambda_N\sqrt{\lambda_N})$.

Location Overhead In the simple, on-demand location management scheme that we used to evaluate terminode routing performance (see next section), the destination is located with a first phase of flooding, and after that the location of the destination is tracked. The cost of tracking is $O(\lambda_N)$, as seen above, and that of flooding is $O(\lambda_N^2)$. Thus the location overhead cost with this scheme is $O(\lambda_N^2)$.

A more efficient framework for location was proposed in [14], [13]. In this case, all nodes send they information toward a known region (Virtual Home Region or Virtual Hot Spot) and the nodes in it acts as database for the rest of the network. Thus, we obtain the same cost we have for mobility tracking plus the cost for retrieving the information from the location system, i.e. $O(\sqrt{\lambda_N})$. The latter, for the whole network, results in a cost per second of $\emptyset(\lambda_N\sqrt{\lambda_N})$. Note also that, with the termination strategy of terminode routing, which allows to reach the destination even in presence of inaccurate location information, the mobility management overhead can be substantially reduced without increasing the sub-optimal paths overhead.

Total Overhead The total overhead, with the simple location management scheme is $O(\lambda_t\lambda_N^2)$, and with the advanced location management scheme is $O(\lambda_t\lambda_N^{1.5})$. The routing protocol scalability terms are: $\rho_{\lambda_{lc}} = 0 = \Psi_{\lambda_{lc}}$, $\rho_{\lambda_t} = 1 = \Psi_{\lambda_t}$, and $\rho_{\lambda_N} = 2 > \Psi_N$ with the simple location management scheme, wherears $\rho_{\lambda_N} = 1.5 = \Psi_N$ with the advanced one. Thus terminode routing is scalable with respect to the most relevant parameters provided that the advanced location management scheme is used.

VII. 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.

Simulation and scalability analysis show 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] S. Basagni, I. Chlamtac, V. Syrotiuk, and B. Woodward. A Distance Routing Effect Algorithm for Mobility (DREAM). *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98), Dallas, Texas, USA,* August 1998.

[2] L. Blazevic, S. Giordano, and J.-Y. Le Boudec. Self Organized Routing in Wide Area Mobile Ad-Hoc Network. In *IEEE Symposium on Ad-Hoc Wireless Networks (Globecom 2001),* November 2001.

[3] L. Blazevic, S. Giordano, and J.-Y. Le Boudec. Self Organized Terminode Routing Simulation. In *Proceedings of ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile systems (MSWiM 2001),* July 2001.

[4] L. Blazevic, S. Giordano, and J.-Y. Le Boudec. Anchored Path Discovery in Terminode Routing. In *Proceedings of The Second IFIP-TC6 Networking Conference (Networking 2002),* May 2002.

[5] L. Blazevic, S. Giordano, and J.-Y. Le Boudec. Self Organized Terminode Routing. *Cluster Computing Journal*, April 2002.

[6] L. Blazevic and J.-Y. Le Boudec. Terminode Routing: Scalable Routing for Large Mobile Ad-Hoc Networks. In *Proceedings of Communication networks and distributed systems modeling and simulation conference (CNDS)*, January 2002.

[7] Ljubica Blazevic. Scalable Routing Protocols with Applications to Mobility, PhD Thesis. *EPFL, PhD Thesis no. 2517*, 2002.

[8] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless newtorks. *3rd Int. Workshop on Descrete Algorithms and methods for mobile computing communications (DIAL M)*, August 1999.

[9] 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. *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98), Dallas, Texas, USA*, August 1998.

[10] Srdjan Capkun, Maher Hamdi, and Jean-Pierre Hubaux. GPS-free positioning in mobile Ad-Hoc networks. *Proceedings of the 34th HICSS*, January 2001.

[11] IEEE Computer Society LAN MAN Standards Committee. Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification. *IEEE Std 802.11-1997, The Institute of Electrical and Electronics Engineers*, New York, 1997.

[12] M. Gerla G. Pei and X. Hong. Lanmar: Landmark routing for large scale wireless ad hoc networks with group mobility. In *Proceedings of the First IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC)*, August 2000.

[13] Silvia Giordano. Mobileman project proposal: Virtual hot spots. Project Proposal IST 32018, DIE-SUPSI, CH-6829, Manno, Switzerland, October 2002.

[14] Silvia Giordano and Maher Hamdi. Mobility management: The virtual home region. EPFL DSC Technical Report, http://icwww.epfl.ch/publications/documents/IC_TECH_REPORT_199937.pdf, March 2003.

[15] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operational Research*, 13, 1986.

[16] P. Gupta and P.R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, 46(2), March 2000.

[17] P. Hansen. The steepest ascent mildest descent heuristic for combinatorial programming. *Proc. Congr. on Numerical Method in Combinatorial Programming*, Academic Press, 1986.

[18] A. Harter and A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, pages 42–47, October 1997.

[19] Ram Ramanathan I. Stavrakakis B. McDonald C. Santivanez. On the scalability of ad hoc routing protocols. part1: Making link state routing scale. *ACM/IEEE Transactions on Networking*, to appear.

[20] Ram Ramanathan I. Stavrakakis B. McDonald C. Santivanez. On the scalability of ad hoc routing protocols. part 2: Asymptotic analysis with respect to traffic, mobility, and network size. *ACM/IEEE Transactions on Networking*, to appear.

[21] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and Chen T.-W. Scalable Routing Strategies for Ad-hoc Wireless Networks. *IEEE JSAC*, 17(8), August 1999.

[22] P. Jacquet, P. Muhlenthaler, and A. Qayyum. Optimized Link State Routing Protocol. *Internet Draft, draft-ietf-manet-olsr-*.txt. Work in progress*.

[23] 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*.

[24] Brad Karp and H.T. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. *Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00)*, August 2000.

[25] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris. A Scalable Location Service for Geographic Ad Hoc Routing. *Mobicom'00, Boston*, 2000.

[26] Mobile Ad-hoc Networks (MANET) Working Group. http://www.ietf.org/html.charters/manet-charter.html.

[27] S. Murthy and J.J. Garcia-Luna-Aveces. An efficient Routing Protocol for Wireless Networks. *ACM Mobile Networks and App. Journal, Special Issue on Routing in Mobile Communication Networks*, October 1996.

[28] V.D. Park and M.S. Corson. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. *Proceedings of IEEE INFOCOM'97, Kobe, Japan*, April 1997.

[29] C.E. Perkins and E.M. Royer. Ad-Hoc On-Demand Distance Vector Routing. *Proceedings of IEEE WMCSA'99, New Orleans*, Feb. 1999.

[30] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector (dsdv) routing for mobile computers. In *Proceedings of ACM SIGCOMM'94. London, UK.*, August 1994.

[31] Marc R. Perlman and Zygmunt J. Haas. Determining the Optimal Configuration for the Zone Routing Protocol. *IEEE JSAC*, 17(8), August 1999.

[32] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location-support system. *Mobicom'00, Boston*, 2000.

[33] R. Jain, A. Puri , R. Sengupta. Geographical Routing Using Partial Information for Wireless Ad Hoc Networks. *IEEE INFOCOM*, 2001.

[34] S. Das, C. Perkins , E. Royer. Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks. *IEEE INFOCOM*, 2000.

[35] S. Giordano, M. Hamdi. Mobility Management: The Virtual Home Region. Technical report no. ssc/1999/037, EPFL-ICA, 1999.

[36] R. Ramanathan C. Santivanez. Hazy sighted link state (hsls) routing: A scalable link state algorithm. BBN technical memo BBN-TM-I30I, BBN technologies, Cambridge, MA, August 2001.

[37] M. Takai, L. Bajaj, R. Ahuja, R. Bagrodia, and M. Gerla. GloMoSim:A Scalable Network Simulation Environment. *Technical Report 990027, UCLA, Computer Science Department*, 1999.

[38] Ko Y. and Vaidya N. Location-aided routing in mobile ad-hoc networks. *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98), Dallas, Texas, USA*, August 1998.