A Scalable 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, jean-yves.leboudec, silvia.giordano}@epfl.ch

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

Abstract

We designed the terminode routing protocol with the objective to scale in large mobile ad hoc networks where the topology, or node distribution, is irregular. Our routing protocol is a combination of two protocols: Terminode Local Routing (TLR - to reach a close destination) and Terminode Remote Routing (TRR - to send data to remote destinations). TRR is the key element to achieve scalability and reduce dependence on intermediate systems. Terminode routing uses anchored paths, a list of geographic points - that are not affected by nodes mobility -, rather than conventional paths of nodes. Terminode routing is completed by a low-overhead distributed method for discovering of anchored paths, and by a method for handling the inaccuracy of the location information. The presented simulation results confirm that terminode routing performs well in different sized networks. In smaller ad hoc networks performance of terminode routing is comparable to MANET routing protocols. In larger networks, where MANET-like routing protocols break, terminode routing performs well; moreover, in larger networks that are not uniformly populated with nodes, terminode routing outperforms the existing location-based routing protocols.

I. Introduction

In this paper, we present terminode routing, a scalable routing protocol for large mobile ad hoc networks. Many existing routing protocols (DSDV [27], WRP [24], OLSR [19], FSR [18], LANDMAR [13],DSR [9], AODV [26], TORA [25],CBRP [20]) are proposed within MANET [23] working group of IETF. These protocols are designed to scale in networks of a few hundreds of nodes. MANET protocols rely on state concerning all links in the network or links on a route between a source and a destination. This results 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. The objective of these algorithms is to 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 [35]), to control packet flooding (DREAM [1]), to reduce intermediate system functions or to make simplified packet forwarding decisions (GPSR [21], GFG [8] and GRA [30]). In GPSR [21], GFG [8] and GRA [30] routing is done in a greedy way by forwarding the packet each time closer to the physical location of the destination. When this greedy process fails, GPSR and GFG route the packet around the problem region in so-called $perimeter$ mode packet forwarding. Perimeter mode of packet forwarding can give a very suboptimal path in large networks when the source and destination are not well connected along the shortest geodesic path. In the case of GRA, when 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 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 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.

Our routing method, called terminode routing, has the objective to achieve scalability in a “large” ad hoc network with irregular node distribution. Scalability is taken in an informal sense. 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 [15] 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 node distributions are likely to appear in metropolitan areas with mountains or lakes, like the Lake of Geneva area.

The main four elements of our method are:

1) **A combination of two protocols:** Terminode Local Routing (TLR) and Terminode Remote Routing (TRR). TRR is used to send data to remote destinations and utilizes geographic information; it is the key element to achieve scalability and reduce dependence on intermediate systems. TLR is the mechanism that allows destinations to be reached in the vicinity of a node. TLR does not use location information for making packet forwarding decisions, instead TLR uses local routing tables that each node proactively maintains for its nearby nodes. TLR is a strategy for handling destination position

deviation due to mobility. In order to avoid loops, once TLR is started, packet forwarding cannot revert to TRR.

2) **Anchored paths in TRR.** TRR is based on GPSR and GFG, to which we add the concept of "anchored path". An anchored path is a list of geographic points, which we call anchors, used as loose source routing information. Anchors help to circumvent large holes in nodes distribution by directing packets around the problem region. When no anchor is used, TRR is a combination of the existing routing protocols GPSR and GFG. TRR without anchors works well if, at any intermediate location, greedy forwarding brings the packet closer to destination. This may fail in irregular networks, in which case TRR uses anchors. 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.
3) **Methods for Computing Anchors.** We provide two methods for computing anchors. Friend Assisted Path Discovery (FAPD) enables the source to learn the anchored path(s) to the destination using, so-called, *friends*, terminodes where the source already knows how to route packets. FAPD never results in a network-wide flooding. GMPD (Geographic Map-based Path Discovery) is another method for anchored path discovery, which assumes that the some knowledge of the node distribution in the network is known. We find that GMPD performs better, but requires the overhead of map distribution (methods for map distribution are left outside the scope of this paper). FAPD always has low overhead.
4) We account for situations where the accuracy of location management is very low and TLR alone is not sufficient to cope with it. Our novel method called **Restricted Local Flooding** (RLF), does the control flooding in the region where the destination is expected to be, thus increasing the probability of reaching the destination.

We show by simulations that the combination of TLR and TRR performs well in networks of different sizes. In smaller ad hoc networks (up to 100 nodes) we compared terminode routing to some existing MANET-like routing protocols (AODV and LAR1 protocols). In order for the comparison to be fair, we implemented an ad-hoc location management scheme for smaller networks. Our simulation results show that in ad hoc networks of 100 nodes terminode routing performs comparable to MANET-like routing protocols. We found by simulations that in larger ad hoc networks (500 nodes), MANET-like routing protocols break, 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 than GPSR when the location information accuracy is low. We also consider irregular networks with holes in node distribution. Here we show the benefits of using TRR with anchors compared to the existing location-based routing protocols where anchors are not used (such are TRR without anchors, GPSR, GFG).

Preliminary versions of some components of terminode routing are presented in [5], [6], [2], [3], [4], [7]. In this paper we present the modified or improved versions of these components. In addition we present: how TLR and TRR interwork together; how to estimate whether the use of anchors is a necessity because directing packets in the direction of the destination does not give a good path; we explain the methods for the anchored path discovery; and we present detailed performance evaluation results.

The remainder of this paper is as follows. Section II presents TLR and TRR and explain how the two protocols interwork. In Section III we present how to find the distribution of the number of hops from the source to the destination when TRR without anchors is used. This result is used to estimate whether an anchored path from the source to the destination is needed. In Section IV we present how anchored paths are discovered. Sections V-A and V-B present performance evaluation of terminode routing in both smaller and larger ad hoc networks.

II. Overview of the Terminode Routing Protocol

Terminode routing assumes that each node has a permanent End-system Unique Identifier (EUI), and a temporary, 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 ([11], [29], [17]) can be used. We assume that there exists a location management that enables nodes in the network to determine approximate locations of other nodes. We envision that a 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 the corresponding LDA information. 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. Location management is out of the scope of this paper. Different proposals can be found at [22], [32].

In this section we overview the two building elements of terminode routing: TLR (Section II-A), TRR (Section II-B), and how the two interwork together (Section II-C). With TRR packets are forwarded as close as possible to the destination location (LDA_D). Once the packet arrives at some node that is close to LDA_D, the routing method is switched to TLR. TLR does not use location information and thus this helps to overcome problems due to location information inaccuracy.

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.

A. Terminode Local Routing (TLR)

Terminode Local Routing (TLR) is used by nodes to proactively learn about nodes in their vicinity, and for packet forwarding to these nodes.

The TLR-reachable area of S includes the nodes whose minimum distances in hops from S are at most equal to the *local radius*. TLR is a link-state routing protocol limited within a scope of a TLR-reachable area. A similar approach is used by the intrazone routing protocol (IARP) in ZRP[28]. In the current implementation of TLR, all nodes have the same *local radius* equal to two hops.

Now we describe the two methods of TLR: the building of local routing tables, and TLR packet forwarding presented in Figure 1.

Each node keeps in its routing table the EUI and LDA information of its immediate neighbours, as well as the EUI information about its two-hop distant nodes. The EUI information of immediate and two-hop distant nodes is used for TLR packet forwarding. The LDA information of immediate neighbours is used in TRR for sending packets to nodes out of the TLR-reachable area. In order to build TLR routing tables, each node periodically advertises by means of HELLO messages its current set of immediate neighbours. HELLO messages are periodically broadcasted at the MAC layer. We assume the existence of bidirectional links in the network, then, when node A receives a HELLO message from node B, A can reach B. A node announces in a HELLO message its own EUI and LDA, as well as EUIs of its immediate neighbours. When a HELLO message is received, a node updates its local routing table. This operation is presented in Figure 1(a) in pseudocode.

When the source, or an intermediate node finds that the destination is TLR-reachable, the “use TLR” bit in the packet header is set to one, if not already set. This is the sign that from now on, the only mechanism used to forward the packet is TLR.

If the destination is two-hops away, the next-hop to send the packet to is determined from the TLR routing table. This is the one-hop neighbour via which a two-hop distant node can be reached. If a two-hop distant node can be reached via several one-hop neighbours, we choose the one-hop neighbour whose entry is updated most recently. Otherwise, if the intermediate node receives the packet whose “use TLR” bit is already set to one, the packet should be sent directly to the destination; if the intermediate node does not find the destination among its one-hop neighbours, the packet is dropped. This ensures that TLR is loop-free. Figure 1(b) presents the TLR packet forwarding in pseudocode.

X receives HELLO from Y

if ($EUI_Y \in X.TLR\_routing\_table$)
  $X.TLR\_routing\_table.update(LDA_Y, EUIs$ reachable via Y)
else
$X.TLR\_routing\_table.add(EUI_Y, EUIs$ reachable via Y)

entry EUI_Y in $X.TLR\_routing\_table$ expires:

  $X.TLR\_routing\_table.remove(EUI_Y, EUIs$ reachable via Y)

(a)

X has packet p to forward to D with TLR:

if ($p.use\_tlr\_bit = 0$)
 $p.use\_tlr\_bit := 1$
 choose Y:
($EUI_D \in Y.neighbours$,$Y.update\_time$ most recent);
 X transmits p to EUI_Y
else//p already sent by TLR
  if($EUI_D \in X.neighbours$)
  X transmits p to EUI_D
 else drop p

(b)

Fig. 1. (a) Building and updating TLR routing tables, (b) The Terminode Local Routing (TLR) packet forwarding algorithm (pseudocode)

B. Overview of Terminode Remote Routing (TRR)

TRR is the method used to reach destinations that cannot be reached by means of TLR. TRR is composed of two routing methods: TRR without anchors, and anchors.

1) TRR Without Anchors: TRR's default packet forwarding method is not to use anchors. TRR is a combination of "greedy forwarding" and "perimeter mode", as explained now.

Greedy Forwarding forwards the packet closer to the destination location LDA_D until the destination D is reached. This is similar to GFG [8] and GPSR [21]: the source and intermediate nodes send the packet to an immediate neighbour (X) where the distance to D is the most reduced. With GFG and GPSR, in this way the packet is forwarded until the destination is reached. TRR without anchors is different: intermediate node X checks whether D is TLR-reachable: if not, X sends the packet to its neighbour that is closest to the destination. Otherwise, X uses TLR to forward the packet.

Perimeter Mode is used when a packet gets "stuck" at some node that does not have a neighbour that is closer to the destination. The solution that we adopt to solve this problem is proposed in GFG[8] and GPSR [21]). Here a *planar graph traversal* is applied, where a packet is routed around the perimeter of the problem region (where there are no nodes closer to the destination). The packet is forwarded in perimeter mode until it arrives at the node that reduces the distance to the destination, and thereon the packet is forwarded in a greedy manner, as described above. We verified that perimeter mode packet forwarding can give very long suboptimal path when there is not a good connectivity from the source to the destination along the shortest geodesic line. Furthermore, we presented in [7] that in mobile ad hoc networks where the topology is frequently changing, perimeter mode of packet forwarding can cause routing loops. Thus the use of perimeter mode should be as minimal as possible. Figure 2 (a) illustrates the operation of TRR without anchors.

(a) (b)

Fig. 2. (a) TRR without anchors. First, the packet is forwarded in greedy mode towards LDA_D (white line). The packet arrives at a node that does not have a neighbour that is closer to LDA_D. Then the packet is forwarded in perimeter mode (grey, thicker, line) until it arrives at a node that resumes greedy forwarding (white line again); close to destination, TLR is used (grey, thicker, line again) (b) TRR with anchors. The packet is sent first (white line) in greedy mode in the direction of $AP1$; when close to AP1, it it sent towards $AP2$, and so on; close to destination, TLR is used (grey, thicker, again).

2) TRR with anchors: We propose a method (described in Section III) that enables the source to estimate whether the greedy mode of TRR without anchors is successful in forwarding data to the destination. If the estimation is that TRR without anchors would not be successful, TRR primarily forwards packets on *anchored paths*. An anchored path is a list of fixed geographic points, called $anchors$. In traditional paths made of lists of nodes, if nodes move far from where they were at the time when the path was computed, the path cannot be used to reach the destination. Given that geographic points do not move, the advantage of anchored paths is that an anchored path is always “valid”.

Anchors are computed by source nodes, using the path discovery methods that are presented in Section IV. A source node adds to the packet the anchored path that is used as loose source routing information. The packet is forwarded so that it loosely follows an anchored path. The sequence of intermediate nodes on the way to the destination depends on the actual physical nodes distribution in the plane. Figure 2 (b) is an example of how TRR with anchors works.

If the anchors are correctly set, then there is a high probability that the packet will arrive at the destination. A good anchored path directs packets along regions with good node connectivity. Occasionally, when there is a hole in the node distribution between two anchors, perimeter mode packet forwarding is used. We can also imagine situations when anchored path is not correctly set. Then, it may happen that there is not a greedy path from one anchor to the next another. Then, the packet may be forwarded in perimeter mode in order to come close to the anchor to be reached. During this operation, if there is a large region without nodes in between two anchors, the packet may be lost due to the time-to-live (TTL) field expiration.

C. How to expedite termination of TRR

As it is described in the previous sections, TRR is the method that uses the location information in order to forward the packet as close as possible to the destination location (LDA_D), which is stamped in the packet by the source. TRR is used until some intermediate node finds that the destination can be reached by means of TLR. In this case the "use TLR" bit in the packet header is set to one. Thereon, only TLR will be only used for packet forwarding.

However, 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 the time-to-live field (TTL) expiration.

Our approach is to discover such situations and to prevent a long lifetime of circulating packets.

A node X detects the case of packet circulation if X finds that LDA_D is within its transmission range (distance(LDA_D, LDA_X) $<$transmission_range$_X$), and the destination is not TLR-reachable.

We propose two possible actions to solve the problem of packets that continue to circulate due to location inaccuracy. The first approach is to limit the lifetime of circulating packets. The second approach is to control flooding in the region where the destination is expected to be. Below we present the two approaches in more details.

a) Limited lifetime of circulating packets: If X detects a circulating packet, X limits the lifetime of such a packet. In order to do so, X sets inside the packet the new value of TTL equal to $min(term\_trr, TTL)$. $term\_trr$ is a fixed value, which indicates that a loop due to destination location inaccuracy is always limited to $term\_trr$ hops. In our current implementation of TRR $term\_trr$ is equal to 3. After the packet has lived for $term\_trr$ hops without being delivered to D, it is dropped.

b) Restricted Local Flooding (RLF) helps in the case of location inaccuracy: Restricted Local Flooding (RLF) controls the flooding of packets, and works as follows.

Again, let's say that the packet is received at node X, which finds that LDA_D, as given in the packet, falls in its transmission range, but the destination is not TLR-reachable. Then, X moves to the RLF mode.

RLF consists in sending six copies of the packet in different directions around the sending node (X). In this way, these copies are sent in the area around X, where the destination is expected to be.

Local flooding is restricted because 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

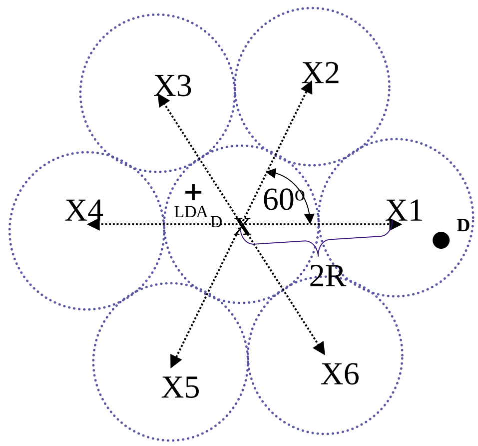


Fig. 3. Node X has a packet for D and finds LDA_D falls within its transmission range, but D is not TLR-reachable. X performs Restricted Local Flooding (RLF) by sending six copies of the packet towards six different geographic points around X.

nodes keep track of the packets that they have already seen. All this is not needed in the case of RLF because packet duplicates are forwarded in the same way as all other packets.

Within each copy, X sets the rlf bit in the packet header to one, thus denoting that the packet is in the RLF mode. X sends each copy in the direction of one of the six geographic points around X. In Figure 3 these geographic positions are denoted as Xi, $i \in 1..6$. Within the i^{th} packet, the destination LDA field in the packet header is set to Xi. However, the EUI field is not changed (that is 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 3, that with circles around each of six points Xi (whose radii is equal to the node transmission range), we 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 copy of the packet.

The TTL field in each copy is set to $term\_rlf$, which is a small number. In our current implementation of TRR $term\_rlf$ is equal to 4. In this way, we constrain the lifetime of a copy to $term\_rlf$ hops. Packets where RLF is started (rlf bit is set to one) are forwarded towards one of geographic positions Xi. There are three possible situations with packets whose rlf bit is set to one.

1) In the first case, the packet is delivered to the destination by some intermediate node that finds D in its TLR-reachable area.
2) In the second case, the packet has been flooded but D is not reached and therefore the packet is dropped due to TTL expiration.
3) The third case occurs when some intermediate node N, finds X_i (Xi is written in the destination LDA field inside the packet header) in its transmission range, but the destination is not TLR-reachable, and therefore N should expedite a termination of TRR. In this case, because the packet has rlf bit set to

one, N drops the packet. In this way we avoid restricted local flooding of the packet, which is itself created after the action of RLF.

RLF is valuable in the case when the accuracy of location information is low because it increases the geographic area where it is expected to find the destination. It is also possible to increase the region where RLF is applied. This can be done by taking points Xi further away from node X, and sending more packet copies. However, the drawback is the increased overhead due to packets that are duplicated and forwarded in the network when none of them reaches the destination.

III. 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 III-A we present the method for the estimation of the distribution number of hops along a greedy path; in Section III-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 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 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 the 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 $G_d(z)$ 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):

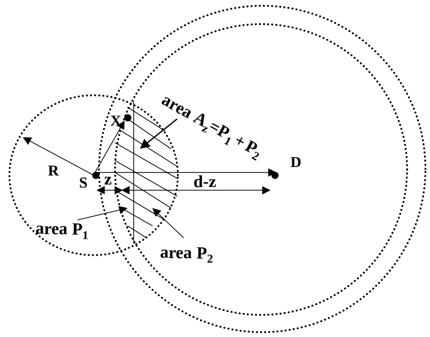


Fig. 4. Progress in distance made in transmission from S to X is equal to z. Three circles are presented: the first centered at S and radius R, the second centered at D and radius d, and the third centered at D and radius $d - z$. The progress from S to D is less than z when there is no nodes in shaded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

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

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

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

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

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

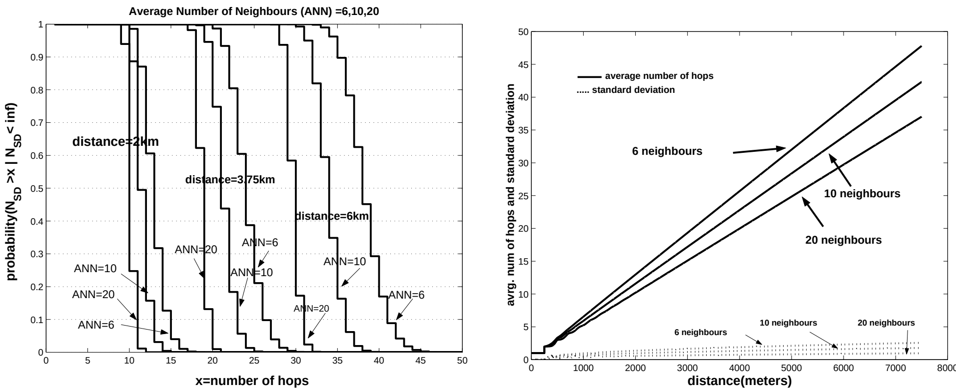
Then we can write the probability density function of Z as,

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

where,

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

Now, as we have presented all necessary elements to calculate $G_d(z)$, we can obtain the distribution of



(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

number of hops. To do so, we use the following recursive equation:

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

We solved numerically Equation (12) using a method that we provide, as follows. 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.

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:

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

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 the 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.

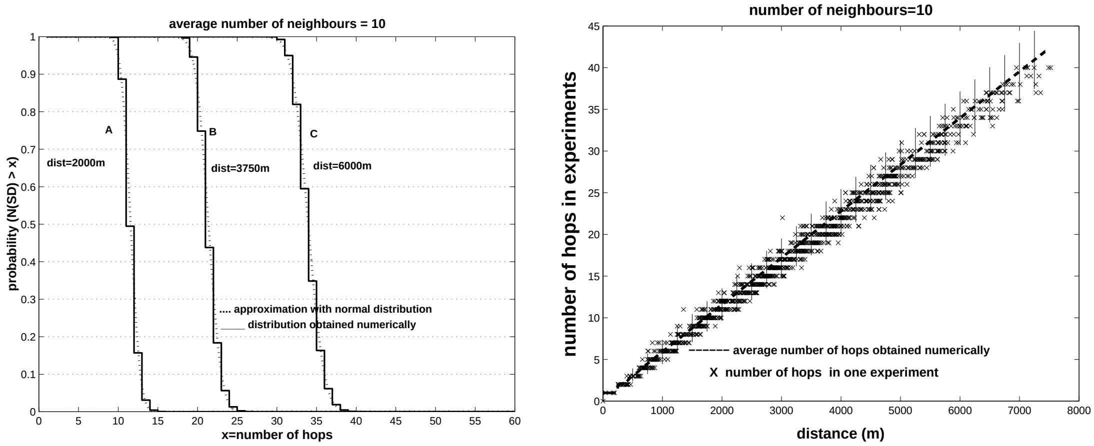
B. How the obtained results can be applied in a real ad hoc network

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 III-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,



(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

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.

IV. Anchored Path Discovery

Anchored path discovery is used at the source when it estimates that TRR without anchors does not perform well in packet delivering to the destination (see Section III), or for discovering anchored paths that can be used in conjunction with the direct geodesic path without anchors.

In this section we present two methods for path discovery, namely Friend Assisted Path Discovery (FAPD) and Geographic Maps-based Path Discovery (GMPD). The two schemes are complementary and can coexist. The first one, FAPD, assumes a common protocol in all nodes and a high degree of cooperation among nodes for providing paths. It is a social oriented path discovery scheme. The second one, GMPD, needs to have or to build a summarized view of the network topology, but does not require explicit cooperation of nodes for acquiring paths.

A. Friend Assisted Path Discovery (FAPD)

FAPD is inspired by the concept of small world graphs[34]. Small world graphs are very large graphs that tend to be sparse, clustered, and have a small diameter. The small-world phenomenon was inaugurated as an area of experimental study in social science through the work of Stanley Milgram in the 60's. These experiments have shown that the acquintanceship graph connecting the entire human population has a diameter of six or less; the small world phenomenon allows people to speak of the "six-degrees of separation".

FAPD is based on the notion of $friends$. B is a ***friend*** of A if (1) A evaluates that it has a good path to B and (2) A decides to keep B in its list of friends. A may have a good path to B because A can reach B by applying TLR, or by TRR without anchors, or because A managed to maintain one or several anchored paths to B that work well. The value of a path is given in terms of congestion feedback information such as packet loss and delay.

Every node has a knowledge of a number of nodes in its TLR-reachable region. These nodes can be considered as close friends. In addition, a node discovers a number of non-TLR reachable nodes to which it maintains a good path(s). A node can consider a number of such nodes as its remote friends. In the friendship graph vertices correspond to nodes, and there is an edge between nodes i and j if i keeps j in its list of friends. Close friends make the friendship graph highly clustered, while remote friends make diameter of the friendship graph small. We conjecture that the friendship graph has the properties of a small world graph. In a small world graph, roughly speaking, any two vertices are likely to be connected through a short sequence of intermediate vertices. This means that then any two nodes are likely to be connected with a small number of intermediate friends.

In general, not all nodes in the network maintain friends connections. We call nodes that maintain friendship connections *FAPD responders*. A FAPD responder runs the FAPD protocol in two cases: (1) when it is the source of data packets and needs a path to the destination; (2) when it acts on behalf of other nodes that need to discover anchored paths and that do not maintain friends.

FAPD is composed by two elements: *Friends Assisted Path Discovery Protocol (FAPDP)* and *Friends Management (FM)*.

Friends Assisted Path Discovery Protocol (FAPDP)

$FAPDP$ is a distributed method for finding an anchored path between two nodes. In the following we describe the operation of FAPDP.

We distinguish two possible cases:

- If source S maintains its list of friends, it requests assistance from some friend in providing an anchored path to destination D. S selects friend that is closest to D to starts FAPDP. If S does not have a friend that reduces distance to the destination, S starts "tabu" mode of FAPDP, as described below. S sends the control packet called *path discovery packet* to the selected friend $F1$ according to the existing path that S maintains to $F1$.
- If source S itself does not know about any friend, it sends path discovery packets in the geographical region around itself. The aim is to reach some FAPD responder that helps in path discovery. For this purpose, S uses the RLF method, described in Section II-C and illustrated in Figure 3. Recall that

with the RLF method S sends each of six path discovery packets by using TRR without anchors in the direction of one of the six geographic points around S. Since S does not know the identity of nodes that serve as FAPD responders the destination EUI in path discovery packets is set to any. In this case any node, let's say $F1$, that receives the path discovery packet from S, and that itself maintains a list of friends acts as FAPD responder to S. If several FAPD responders receive a path discovery packet from S, S may learn several anchored paths to D. On the contrary, if no path discovery packets reaches a FAPD responder, S does not get any anchored path to D.

The $fapd\_anchored\_path$ field inside the path discovery packet progressively contains anchor points from S to D. If S has an anchored path to $F1$, S simply adds anchors of this path in the $fapd\_anchored\_path$ field (S sends data to $F1$ using TRR with anchors). Otherwise, S leaves this field empty. Upon reception of the path discovery packet, $F1$ (an intended friend or a FAPD responder) puts its geographic location inside $fapd\_anchored\_path$ field as one anchor. If $F1$ has an anchored path to D, $F1$ appends this path into the $fapd\_anchored\_path$ field and sends the packet to D using TRR with anchors. If $F1$ does not have a path to D, it recursively uses FAPDP. In this case, $F1$ checks if it has a friend $F2$ closer to D, and then it performs the same steps as S. This is repeated until the packet is received by some friend that has a path to D, or the packet is forwarded to some node that is close to D and it forwards the packet to D by TLR.

However, there are situations where the source or an intermediate friend does not have a friend closer to the destination. In some topologies with obstacles, at some point, going in the direction opposite from the destination may be the only way to reach the destination. FAPDP permits that some node T (the source, a FAPD responder, or an intermediate friend) sends a path discovery packet to its friend even though the packet is not getting closer to the destination. However such a friend must not be distant from T more than distance max_dist. In our current implementation we use max_dist equal to five times the transmission range of a node. Here is where the "tabu" mode of FAPDP starts. With the tabu mode mechanism, intermediate friends can send the packet in a direction opposite to D for a limited number of times. Our method is inspired by the Tabu Search heuristic ([14], [16]). Tabu Search is a general heuristic in which a local search procedure is applied at each step of the general iterative process. It could be superimposed on other heuristics to prevent those being trapped in a local minimum. We use the tabu mechanism in order to get out of a local minimum that can happen at some node that does not have a friend closer to the destination. Then, with the tabu mechanism, we try the opposite direction (non-improving move) from the destination with the aim to finally get out of a local minimum and further approach towards the destination. In order to avoid cycling, in FAPDP we limit the number of consecutive non-improving moves.

Finally, when D receives the path discovery packet with the accumulated anchors from S to D, D tries to do the path simplification with a method we present below. Then D returns back to S a "path reply" control

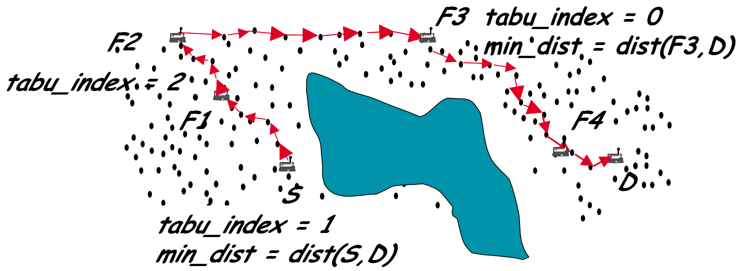


Fig. 7. Illustration of FAPDP operation. Source S does not have a friend closer to D than itself. S sends the path discovery packet to its friend $F1$ that is farther from D in geometrical distance than S is, but such that $dist(S, F1) < max\_dist$. S sets the $tabu\_index$ field to 1 in the packet and thus starts the tabu mode of FAPDP. 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 discovery packet, it sends back to S the anchored path from S to D given with the list of anchors $(LDA_{F1}, LDA_{F2}, LDA_{F3}, LDA_{F4})$

packet which contains the acquired anchored path from S to D. 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. The operation of FAPDP is presented in Figure 7.

Path Simplification

The path simplification operation consists in approximating an existing anchored path by a path with fewer anchors. As already described, anchors are accumulated from the source to the destination during the FAPDP operation. For example, it is possible that many geographically close friends are consecutively contacted, and thus resulting anchored path contains many close anchored points. The first goal of path simplification is to keep as minimal number of anchors as possible. So, in this case the destination simplifies the path by skipping a number of close anchors from an initial list of anchors.

Another situation that may happen is when there is two non consecutive anchors that are geographically close to each other. Path simplification in this case keeps only one anchor in the resulting anchored path. In this way we prevent that the packet revisits the geographical region already visited on the way to the destination.

Friends Management

Friends Management (FM) is a set of procedures for discovering, monitoring and evaluating friends. FM is performed by FAPD responders. Friends Management consists of the following components: *Friends Discovery* , *Friends Monitoring*, and *Friends Evaluation*.

Friends Discovery

Friends Discovery is performed by a FAPD responder with the aim to learn from other nodes information about some potential friends.

As it is already described in Section II-A, nodes periodically send HELLO messages, for the purpose of building the TLR routing tables. In this process, nodes can learn about EUIs and LDAs of the one-hop and the two-hop distant nodes. Given that this information is periodically maintained, a node always has information about close nodes which can be considered as *close friends*.

Friends that are further than two hops (i.e. the node does not maintain information about their EUIs and the LDAs by means of HELLO messages) are called *remote friends*. One way for a node T to learn about remote friends is to extract this information from its previous communications. We propose a method that enables a node to explicitly discover remote potential friends. In this scheme, each node T sends the $get\_friends\_request$ message towards four geographic points (GP1, GP2, GP3 and GP4). The TTL field inside these messages are set to a small number equal to 6. These points are randomly selected as four points in orthogonal directions at four times the transmission range of T. Once some FAPDP responder, let's say Y, on the way towards a point FPi receives the $get\_friends\_request$ message it stops its forwarding. Then Y sends back the $get\_friends\_reply$ message to T, which contains the list of friends of Y. If this table is empty, Y puts itself in the content field of the message. When node T eventually receives the $get\_friends\_reply$ 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.

Friends Monitoring and Friends Evaluation

FAPD responders periodically evaluate their friends in order to assure the consistency of the information on current friends and for testing the validity of these friends. Here we present the outline of how to perform friends monitoring and evaluation. This is the matter of our ongoing work.

We assume that some form of location tracking is active between friends. The *Friends Monitoring* component of FM keeps under control, for a node A, a set of parameters for each friend F_i of A.

A series of parameters are used to evaluate friends. These are:

1) *Value of path(s) to friend* F_i: A may evaluate that the path to its friend F_i, that worked well in the past, deteriorated.
2) *Location of friend* F_i *and the average distance to* F_i: F_i may have moved considerably from the location where is was at the time when it was included in $A's$ list of friends.
3) *The number of times friend* F_i *was contacted to provide a path and the number of paths that are found*

with the help of friend F_i: A may contact F_i in FAPDP to learn the path to the destination, but the path is never returned back to A.

A node evaluates a friend as bad if any of the following is true: path to the friend deteriorates, or, the friend has moved considerably from the location where it was when it has been selected to be a friend, or, a friend was contacted several times in FAPDP, but the path was never acquired.

Based on these parameters, the *Friends Evaluation* component periodically evaluates whether it is beneficial to keep a node in the list of friends, or it is better to discard it. Friends with bad evaluation results are discarded from a friends list.

B. Geographic Maps-based Path Discovery (GMPD)

GMPD is another method for anchored path discovery, which assumes that the network topology is known to all nodes in the network.

We believe that a good model of a large mobile network does not assume that nodes are uniformly distributed in the network. In order to model a large mobile ad hoc network, we identify the areas with a higher node density, which we call $towns$. 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 9. 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 position of its center and the size of the square area. One example of a map of a network is presented in Figure 9. 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 position 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 V-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. PERFORMANCE EVALUATION OF TERMINODE ROUTING

In this section we validate the operations of terminode routing by using simulations. Our objective is to test how terminode routing performs in several simulation environments: in different sized mobile ad hoc networks, under different node mobility or under different load in the network.

We implemented and simulated the terminode routing protocol in GloMoSim[33]. The IEEE 802.11 Medium Access Control(MAC) protocol is used; it implements the Distributed Coordination Function (DCF)[12]. In all simulations, radio range is the same for every node, and is equal to 250 meters. The channel capacity is 2Mbits/sec. This corresponds to characteristics of Lucent Technologies WaveLAN [10]. 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.

As explained in Section II-A, 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 packets to send should defer sending HELLO messages, because data packets piggyback the HELLO message information.

In our implementation every node knows accurately its current location all the time. Terminode routing uses the destination location for packet forwarding. Location management is used at sources to learn the destination location. In our simulations we distinguish location management in small and in large networks.

Location management in a small mobile ad hoc network simulation

In the simulations of small ad hoc networks (100 nodes), we include the simple location management that works as follows. Sources learn about the destination location on demand. When 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. S buffers all data packets until it learns D's location. 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 D's location. D sends the location reply back to S by TRR without anchors, using the S's location (that D learnt from the location request). Upon reception of the location reply, S stores D's location in its location cache and sends buffered data packets by using TRR without anchors. But, if S does not receive a location reply from the destination after 2 seconds, S initiates again the flooding of the location request control packet with the new sequence number.

location tracking

Once the two nodes begin to communicate, the location tracking is used: data packets periodically (every 5 seconds) piggyback the local position of the sending node.

If the destination location entry is not refreshed for more than 10 seconds, the entry is removed, and the source re-initiates learning of the destination location through flooding.

We observe that, as expected, the described location management scheme is not applicable within a larger mobile ad hoc network because it includes flooding of the network.

Idealized location management in a large mobile ad hoc network simulation

In the simulations of larger mobile ad hoc networks (500 nodes) we do not include a distributed location database for annotating packets with destinations' positions. We assume an idealized location database where all nodes can know all other nodes' positions 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 positions. The control part of LAR1 uses geographic positions, 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 positions that are used in TRR are obtained using the location management method that we described in Section V.

Mobility Model: 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 0-20 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.

Communication Model: 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 [31]:

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 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.

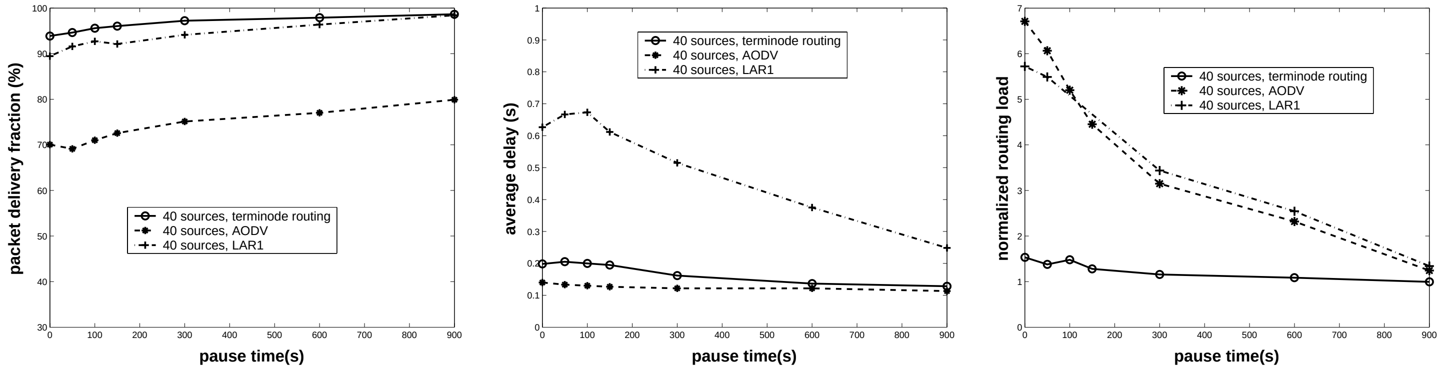
Simulation Results

Our simulation results show the following.

Terminode routing is comparable to LAR1 in packet delivery fraction and both outperform AODV.

The delay experienced by LAR1 is higher than the one of AODV and of terminode routing. 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.

In all experiments, terminode routing has the smallest normalized load. Moreover, terminode routing has a stable normalized routing load for different pause intervals. This happens for two reasons: First, every node proactively generates HELLO messages every second, unless a data packet 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 and the number of traffic flows. 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. As the number of sources increases, terminode routing generates more routing overhead due to location



(a) (b) (c)

Fig. 8. (a) Packet delivery fraction, (b) Average data packet delay, (c) Normalized routing loads for the 100 node model and 40 sources

management control packets, but we observe a slight increase of routing overhead with the increase of the number of data sources.

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.

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

In this section 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 verify by simulations that in this environment MANET protocols LAR and AODV break, whereas TRR works well. We show that it is beneficial to use anchors compared when anchors are not used, as in TRR without anchors, GPSR [21] 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. This better represents the fact that most people move for a certain period within one area, and then they move away to another distant area. Restricted random waypoint is closer to a real-life situation for a wide-area mobile ad hoc network than the random waypoint model [9] that is appropriate for smaller networks.

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 9. 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 a high level geographic view of the network is available at every node with no overhead. Each node has a knowledge of a map of towns. A map defines town areas and the existence of highways between towns. 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 9, 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 9 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 9 clearly illustrates the case where the usage of anchors give shorter paths.

When FAPD is used for path discovery, nodes in the network do not have maps in order to discover the anchored paths, but the FAPDP algorithm is used for this purpose. The source starts the FAPDP protocol whenever the destination is not TLR-reachable and it does not have a valid anchored path to the destination. In the meanwhile, until some anchored path is discovered, the source uses TRR without anchors to send

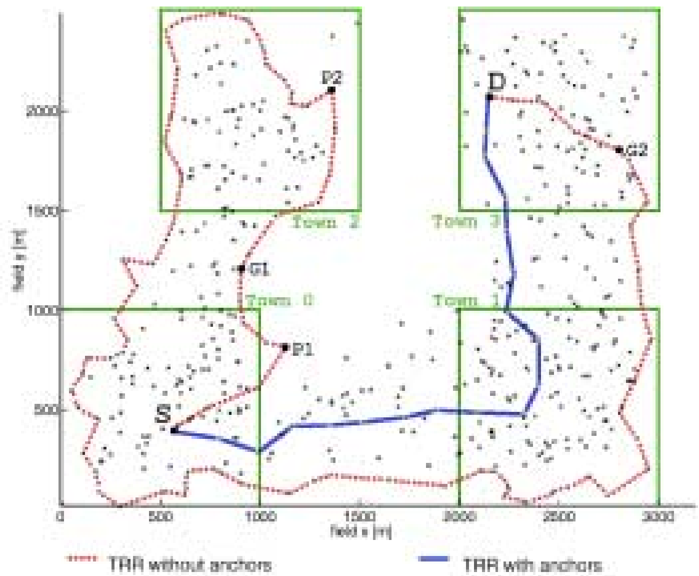


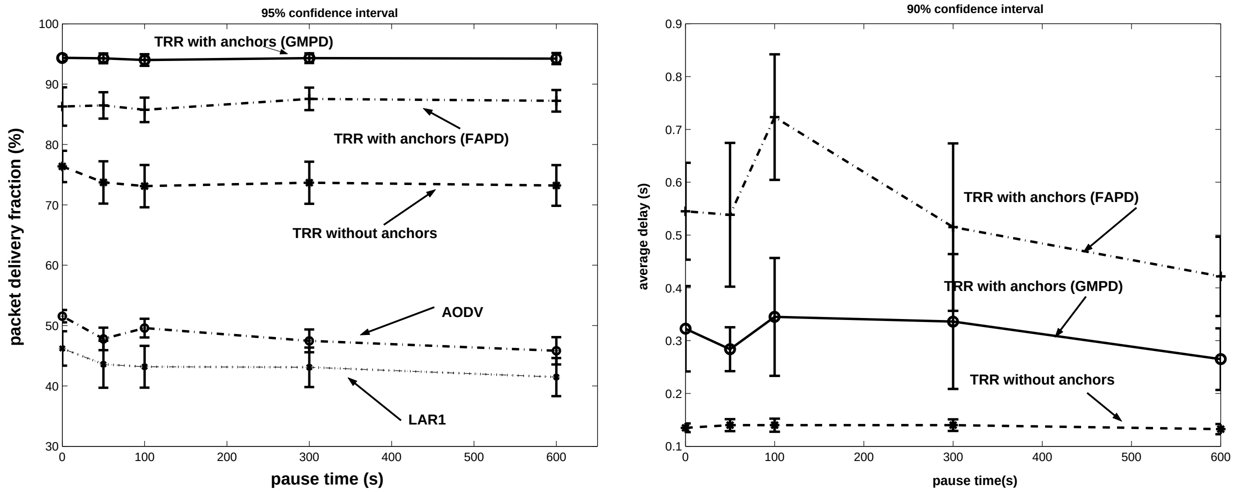
Fig. 9. Figure presents the path of the packet from source S to destination D in case of two routing protocols: TRR without/with anchors . When anchors are used, TRR gives a shorter path than TRR without anchors.

packets to the destination. Once the source issued the path discovery request, and it does not gets a path within 10 seconds, the source issues the new request. The source keeps every acquired anchored path for 5 seconds. After this time the anchored path is considered invalid and is not used. In our simulations we take that only a fraction of nodes (which we called FAPD responders in Section IV) in the network reply to FAPD requests. FAPD responders are situated in town areas and do not move. Our simulation results take into account the FAPD control overhead. This overhead includes control messages needed for FAPD responders to discover friends, and well the overhead in delivery of path request 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 periodically with the time interval equal to 50 seconds.

Simulation Results

We conducted simulations of 500 nodes forming an ad hoc network presented in Figure 9. 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 0-20m/s; before each movement, a node pauses for some pause time. There are 200 ordinary nodes and 300 commuters. Recall that ordinary nodes simulate small personal



(a) (b)

Fig. 10. (a) Packet Delivery Fraction with 40 sources; stay_in_town parameter is 10, (b) Average delay with 40 sources; stay_in_town parameter is 10

devices that stay within a boundary of a single town for a 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.We ran simulations with different pause times and different *stay_in_town* parameters of ordinary nodes. These parameters define different degrees of ordinary node mobility. A longer pause time means that ordinary nodes are less mobile. For a fixed pause time, a larger *stay_in_town* means that a node is staying longer within a geographic region that corresponds to a single town. 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 0-20m/s, and pause time equal to 0 seconds. The *stay_in_town* parameter for commuters is equal to 1.

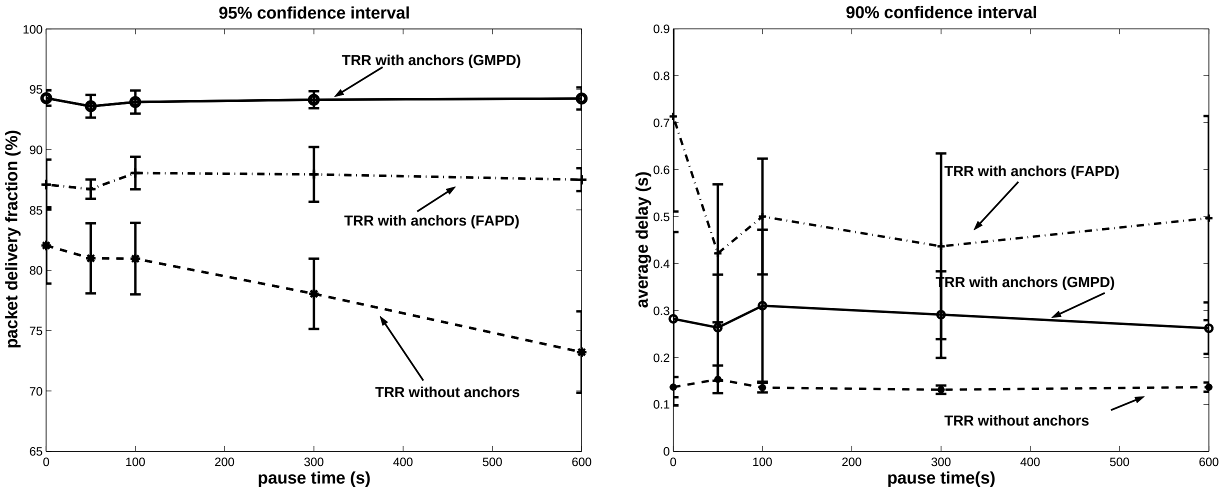
Traffic sources are continuous bit rate (CBR). The source-destination pairs are spread randomly over the network. All CBR sources send two packets per second, and uses 64-byte packets. All communication patterns are peer-to-peer. All source destination pairs are chosen from the group of ordinary nodes. CBR connections are started at times uniformly distributed between 400 and 500 seconds (starting from initial positions 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.

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.

In the first set of simulations, the *stay_in_town* parameter is set to 10 for ordinary nodes (CBR sources and sinks). Different degrees of mobility are obtained for different pause times of ordinary nodes. For higher pause times, because *stay_in_town* is high, ordinary nodes for most of the simulation time move inside the same town area. For smaller pause times they move to different town areas more frequently. Figure 10 (a) shows that TRR with anchors and GMPD deliver about 20 percent more packets compared to the case where TRR without anchors is used. This result is explained as follows. TRR without anchors can give complex and long paths for source-destination pairs that are situated in towns not connected with a highway (Figure 9). For those packets there is a higher probability that they will be dropped. Moreover, TRR without anchors often uses perimeter-mode packet forwarding. We have found using the simulations that this mode of packet forwarding can have 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]). Because in this simulation CBR sources and destinations do not frequently change town areas, there are several flows where TRR without anchors loses many packets, while TRR with anchors has more success. Figure 10 (a) also illustrates that when FAPD is used for path discovery, TRR with anchors still outperforms TRR without anchors by more than 10 percent. The same figure presents the performance of AODV and LAR. We see that these protocols succeed to deliver only half of the packets compared to TRR with anchors. So, although these protocols perform well in 100-node network, we see that they break in larger non-uniform networks.

We may observe that whether anchors are used or not, TRR has packet delivery fraction that does not change a lot for different pause time. There are two reasons for that: First, note that in our simulations, sources know destination positions accurately enough at all times. The number of packets that is dropped due to location inaccuracy is small even when the pause time is small. We found that the main reason for packet dropping is either buffer overflow due to congestion, or because packets that are forwarded in perimeter mode suffer from the loop problem. Second, the routing overhead is independent of mobility. This is explained as follows: 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 FAPDP 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 packet delivery success. Figure 12 compares the normalized routing overhead when FAPD or GMPD is used for path discovery. We see that FAPD does adds only a small fraction of overhead compared to GMPD.



(a) (b)

Fig. 11. (a) Packet Delivery Fraction with 40 sources; stay_in_town parameter is 2, (b) Average delay with 40 sources; stay_in_town parameter is 2

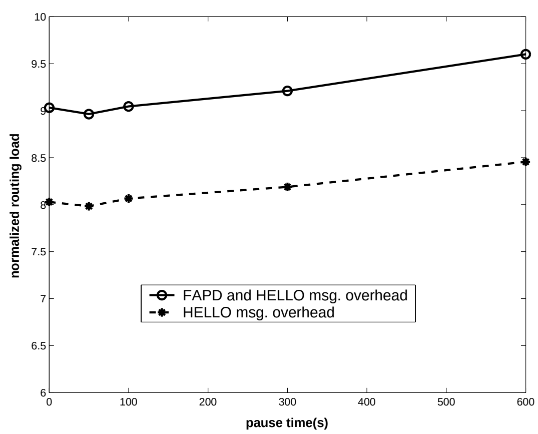


Fig. 12. Figure presents the normalized routing overhead when TRR is used with GMPD (overhead due to HELLO messages) and when TRR is used with FAPD (overhead due to HELLO messages and FAPD messages)

Figure 10 (b) illustrates average end-to-end delays for the first set of simulations. TRR without anchors has smaller delay than TRR with anchors. The reason is that TRR without anchors has a lower packet delivery fraction than when anchors are used and the average delay counts only for delivered packets. We observed that with TRR without anchors 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.

The results of the second set simulations are presented in Figures 11. All simulation parameters are the same as in the previous simulation, except that ordinary nodes move more frequently from one town area to another. Here the *stay_in_town* parameter is set to 2. We observe that TRR without anchors delivers more packets than in the previous simulations (where *stay_in_town* is set to 10, Figure 10). This can be explained: with increased mobility, those source-destination pairs for which TRR without anchors gives a small fraction delivery in the previous simulations can move to towns where TRR without anchors gives a better path. In this way bad situations, where TRR without anchors gives long complex paths, can last shorter. This is especially true for lower pause times. For higher pause times, again we observe that TRR without anchors has a lower packet delivery fraction.

We conclude that in all our simulations that TRR with anchors results in a higher packet delivery fraction than TRR without anchors is used. We observe that the use of anchors is more important for a higher $stay\_in\_town$ parameter (nodes stay in single town areas for longer time).

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 evaluated the benefits of TLR by simulations. We compare the combination of TRR without anchors and TLR against GPSR [21].

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 position and no node finds the destination to be TLR-reachable, TRR termination is done by limitation of lifetime of a packet.

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 V, 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 13 (a) presents the average of packet delivery fraction for six simulation runs. This figure shows that for smaller location information lifetimes (less than

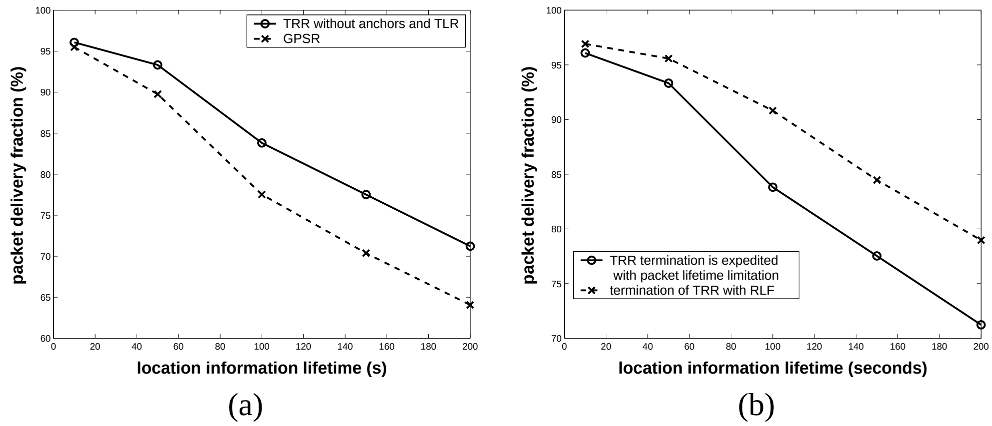


Fig. 13. (a) Figure shows that using TLR results in higher packet delivery fraction than in the case when only position based routing is used, (b)When Restricted Local Flooding is performed, fraction of received packets is higher than without RLF (when TRR is terminated by the packet lifetime limitation)

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, 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.

Example of usefulness of RLF

In all the simulations presented so far, when the packet arrives at some node where conditions for expedited termination of TRR are met (because the packet arrives close to the destination's location stamped in the packet, but the destination is not TLR-reachable), the lifetime of the packets is limited to $term\_trr$ hops ($term\_trr$ is equal to 3 hops). Then if the packet does not arrive at the destination, the packet is dropped.

Now, we evaluate by simulations the Restricted Local Flooding (RLF) method, which is the second method that is used to terminate TRR (see Section II-C). We use the same simulation settings as previously where we evaluate TLR.

Recall from Section II-C, that 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 13 (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, and the second case corresponds when RLF is used. Figures 13 (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).

VI. Conclusion

We presented a scalable routing method for large ad hoc networks, called terminode routing. It is a combination of two routing protocols, TLR and TRR. TRR is used to send data to remote destinations and utilizes geographic information; it is the key element to achieve scalability and reduce dependence on intermediate systems. TLR is the mechanism that allows destinations to be reached in the vicinity of a node. TLR does not use location information; instead TLR uses local routing tables that each node proactively maintains for its nearby nodes. With TLR routing is more robust against location information inaccuracy. We introduced the concept of anchored paths in TRR. Anchored paths help to circumvent holes in node distribution. By a method that we propose the source estimates whether anchors are needed. If this is the case, we propose a low overhead distributed protocol for anchored path discovery.

We evaluated terminode routing in both small and larger networks. Our simulation results show that in smaller ad hoc networks of 100 nodes terminode routing performs comparable to MANET-like routing protocols. We found by simulations that in larger ad hoc networks of 500 nodes, MANET-like routing protocols break, while terminode routing still performs well. We also considered irregular networks with holes in node distribution. Here, our results illustrate the benefits of TRR with anchors over the existing location-based routing protocols that do not use anchors.

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] Tuch Bruce. Development of WaveLAN, an ISM Band Wireless LAN. *AT&T Technical Journal*, July/August 1993.

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

[12] 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.

[13] 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.

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

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

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

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

[18] 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.

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

[20] 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*.

[21] 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.

[22] 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.

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

[24] 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.

[25] 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.

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

[27] 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.

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

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

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

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

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

[33] 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.

[34] D. J. Watts. In *Small Worlds, The dynamics of networks between order and randomness*. Princeton University Press, 1999.

[35] 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.