Self Organized Terminode Routing

Ljubica Blazević   Silvia Giordano   Jean-Yves Le Boudec

EPFL - Communication Systems Department (DSC)
Ecole Polytechnique Federale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland
{ ljubica.blazevic, silvia.giordano, jean-yves.leboudec}@epfl.ch

Abstract

We consider the problem of routing in a wide area mobile ad-hoc network called Terminode Network. Routing in such a network is designed with the following objectives. Firstly, it should scale well in terms of the number of nodes and geographical coverage, secondly, routing should have scalable mechanisms that cope with the dynamicity in the network due to mobility, and thirdly nodes need to be highly collaborative and redundant, but, most of all, cannot use complex algorithms or protocols. Our routing scheme is a combination of two protocols called Terminode Local Routing (TLR) and Terminode Remote Routing (TRR). TLR is used to route packets to close destinations. TRR is used to route to remote destinations and is composed of the following elements: Anchored Geodesic Packet Forwarding (AGPF), Friend Assisted Path Discovery (FAPD), multipath routing and path maintenance. The combination of TLR and TRR has the following features: (1) it is highly scalable because every node relies only on itself and a small number of other nodes for packet forwarding; (2) it acts and reacts well to the dynamicity of the network because multipath routing is considered as a rule; and (3) it can be implemented and run in very simple devices because the algorithms and protocols are very simple and based on high collaboration. We have performed simulations of the TLR and TRR protocols in GloMoSim. The simulation results demonstrate that the routing protocol is able to deliver over 80% of user data in a large, highly mobile simulation environment whereas Dynamic Source Routing(DSR) achieves less than 10%.
Keywords: mobile ad-hoc wide-area network, routing, terminodes

1 Introduction

We focus on the problem of routing in a large mobile ad-hoc network that we call terminode network. We call nodes in a terminode network, *terminodes*, because they act as network nodes and terminals at the same time. Our routing solution is designed with three requirements in mind: firstly, it should scale well in a very large mobile ad-hoc network; secondly, it should cope with dynamically changing network connectivity owing to mobility; and thirdly terminodes need to be highly collaborative and redundant, but, most of all, cannot use complex algorithms or protocols. For the first requirement, our solution is designed such that a terminode relies only on itself and a small number of other terminodes for packet forwarding.

The second requirement, uncertainty in the network due to mobility, is addressed in our work by considering multipath routing as a rule, and not as an exception.

We note that the target of our work is different from MANET[10] proposals that focus on networks consisting of up to several hundreds of nodes.

Each terminode has a permanent End-system Unique Identifier (EUI), and a temporary, location-dependent address (LDA). The LDA is simply a triplet of geographic coordinates (longitude, latitude, altitude) obtained, for example, by means of the Global Positioning System (GPS) or the GPS-free positioning method[5]).

In this paper, we focus on the problem of unicast packet forwarding, assuming that the source terminode knows or can obtain the LDA of the destination. A packet sent by a terminode contains, among other fields, the destination LDA and EUI, and possibly some source routing information, as mentioned later. Mobility management in a terminode network may be performed by a combination of the following functions. Firstly, a location tracking algorithm is assumed to exist between communicating terminodes; this allows a terminode to predict the location (LDA) of corresponding terminodes. Secondly, LDA management, which is based on the distributed location database, allows a terminode A to obtain a probable location of terminode B (LDA_B) that A is not tracking by the previous method. Mobility management is out of scope of this paper. (see for example [7],[9]).

Our assumption is that we can get with the mobility management the destination LDA with precision of approximately one transmission range and validity of about ten seconds.

We also assume that multipath routing is acceptable for the transport protocol. However, with the current TCP this is not acceptable since there are problems with managing a large number of timers due to many paths. We envision either to bring enhancements to the current TCP, or to use of multiple description coding techniques. In this latter case, the source data is encoded and sent over multiple paths in order to provide better load balancing and path failure protection.

We use a combination of two routing protocols: Terminode Local Routing (TLR) and Terminode Remote Routing (TRR). TLR is a mechanism that allows to reach destinations in the vicinity of a terminode and does not use location information for making packet forwarding decisions. In contrast, TRR is used to send data to remote destinations and uses geographic information; it is the key element for achieving scalability and reduced dependence on intermediate systems.

TRR consists of the following elements:

- *Anchored Geodesic Packet Forwarding (AGPF)* is a method that allows for data to be sent to remote terminodes. AGPF is solely based on locations. AGPF sends data along the *anchored path.* An anchored path defines a rough shape of the path from the source to the destination and is given with a list of anchors. Anchors are points described by geographical coordinates and do not, in general, correspond to any terminode location. A good anchored path should avoid obstacles and terminode "deserts" from the source to the destination. Between anchors geodesic packet forwarding is performed; this is a greedy method that follows successively closer geographic hops to an anchor or the final destination.

- *Friend Assisted Path Discovery (FAPD)* is the path discovery method used to obtain anchored paths. A terminode keeps a list of other terminodes, that it calls friends, to which it maintains one or several good path(s). In FAPD, a terminode may contact its friends in order to find an anchored path to the destination of interest. FAPD is based on the concept of small world graphs[18].

- *Path Maintenance* is a method that allows a terminode to improve acquired paths, and delete obsolete or mal-functioning paths.
- *Multipath Routing.* A terminode normally attempts to maintain several anchored paths to any single destination of interest. In a highly mobile environment, anchored paths can be broken or become congested. A path that worked well suddenly can deteriorate. As a response to such uncertainty in the network, TRR uses multipath routing.

TRR is used to send data to a remote destination. However, when a packet gets close to the destination, if locations are used for making packet forwarding decisions, positional errors and inconsistent location information can result in routing errors and loops. Therefore, when close to destination, the packet forwarding method becomes TLR, which does not use location information. Once a packet has been forwarded with TLR, the "use TLR" bit is set within the packet header, thus preventing downstream terminodes from using TRR for this packet. This avoids loops due to the combination of the two routing methods.

The rest of this paper is organized as follows. The following section gives a short overview of some existing mobile ad-hoc routing protocols that are relevant for our work. In Section 3 we give a complete description of TLR and TRR . This is followed by the description of the simulation results in Section 4. Finally, we give some conclusions and directions for future work.

2 Related Work

Many routing protocols have been proposed for mobile ad hoc networks. A recent overview can be found for instance in [15]. The existing routing protocols can be classified either as pro-active or reactive.

Proactive protocols attempt to maintain routes continuously, so that the route is already available when it is needed for a packet to be forwarded. In this protocols, routing tables are exchanged among neighbouring nodes each time a change occurs in the network topology. As a consequence that pro-active protocols are not suitable when the mobility rate in the network is high.

An attempt to overcome these limitations is to look for a route only on demand. This is the basic idea of reactive protocols such is DSR[4], TORA[11] and AODV[12]. In reactive protocols a control message is sent to discover a route to a given destination.

In DSR, when a source S needs a route to a destination node D, S first checks if some of its neighbours possess the route in question. If this is not the case, S floods the network with a *route request* for the destination node. When the request reaches the destination, the destination returns a *route reply* to the request's originator. The reply message contains the list of all intermediate nodes from S to D. Then S uses source routing with the acquired source route to send packets to D. Several methods are proposed for limiting the propagation of requests. One of these is that nodes cache the route that they learn or overhear, so that intermediate nodes can reply on behalf of the destination if the route to the destination is known.

Reactive protocols have smaller control traffic overhead than pro-active protocols. However, since a route has to be discovered before the actual transmission of the data, these protocols can have a longer delay. Further more, due to mobility, the discovered route may be unusable since some links of the route may be broken.

ZRP[13] is a protocol that combines both a pro-active and reactive approach. Every node proactively maintains routes to other nodes whose distance is less than a certain number of hops (its zone). Within a zone, an arbitrary pro-active routing scheme can be applied. For inter-zone routing, on demand routing is used. ZRP avoids flooding the network in order to find routes and routing protocol overhead is limited. However, the on-demand solution for inter-zone routing poses the latency problem typical to on-demand routing schemes.

There are a number of proposed geographical routing protocols that use location information to reduce propagation of control messages, to reduce intermediate system functions or for making packet forwarding decisions.

Geographical routing allows nodes in the network to be nearly stateless; the information that nodes in the network have to maintain is about their one-hop neighbours.

Location Aided Routing (LAR)[19] in an optimization of DSR where the knowledge of the destination location is used to limit the propagation of route request control packets. Those packets are propagated to the geographical region around the last known destination position. LAR does not use location for data packet forwarding. With LAR, end-to-end routes are still DSR's source routes.

Location Distance Routing Effect Algorithm for Mobility (DREAM)[2] is a routing protocol in which the information about location and speed of the destination is used to obtain the direction of the destination. A node that has a packet to send determines the direction of the destination. Then it forwards data to all one hop neighbours in the calculated direction of the destination. DREAM proposes how to disseminate location information in the network in the scalable way.

GEDIR[16] and GPSR[8] routing protocols are very important to this paper, and we present them in more detail. GEDIR[16] and GPSR[8] propose to use a greedy method for making packet forwarding decisions. Packet forwarding decisions are made using only information about a node's immediate neighbours and the location of destination; packet is forwarded to the neighbour that is closest to destination. The GEDIR paper proves that provided the accuracy of location information, packet forwarding is loop-free.

However, a packet may reach a node that does not have any neighbours closer than itself to the ultimate destination. This situation indicates that there is a hole in the geographic distribution of nodes. As a solution to this situation, GPSR uses a planar subgraph of the wireless network's graph to route around the perimeter of a hole. Packet forwarding for such a packet is switched from greedy to perimeter mode. The knowledge of identities and location of its one-hop neighbours is sufficient for a node to determine the edges of the planar subgraph. GPSR forwards perimeter-mode packets using a simple planar graph traversal.A perimeter-mode packet is forwarded on progressively closer to destination faces of the planar graph. As soon as a packet reaches the node that is closer to destination than the node that initiated perimeter mode forwarding, a packet is then forwarded in a greedy way. In a dense network, packets are normally forwarded in greedy way, and the perimeter mode is used occasionally when a packet is stuck when a node does not have a one-hop neighbour that is closer to the destination. Then a packet is forwared in perimeter mode for only a very few (2-3) hops, before a node closer than the point of entry into perimeter mode is reached, then and greedy forwarding resumes. On sparser networks, perimeter mode tends to be used for longer sequences of hops.

Terminode routing does not maintain strict source routes. Differently from DSR, in terminode routing, there is no need for flooding of the whole network to discover the route or react when some link is broken. FAPD is a way to discover loose source paths without flooding of the network. When a path with anchors is known, AGPF is used. AGPF is a greedy method that uses locations for packet forwarding, and recovers from a link failure relaying only on terminodes' local information. If there is a hole in nodes's distribution, GPSR uses routing around the perimeter of a hole. In our approach, if anchors are correctly set, AGPF avoids holes in terminodes distribution and uses perimeter method only occasionally. As it is discussed in the introduction, use of location for packet forwarding may result in looping problems due to positional errors and inconsistent location information. In our design we use TLR in order to alleviate looping problems.

3 Terminode Routing

3.1 The Global Picture

As mentioned in the introduction our routing scheme is a combination of two protocols, Terminode Local Routing (TLR) and Terminode Remote Routing (TRR), which, roughly speaking, act for close and for remote destinations respectively.

TRR is used to send data to remote destinations and is based on geographical information. If the source S does not know the recent destination D's location, it must acquire it. Approximate value of D's location (LDA_D) is obtained either using the LDA management scheme (described in [7]) or by the location tracking. TRR consists of the following elements: *Anchored Geodesic Packet Forwarding (AGPF)*, *Friend Assisted Path Discovery (FAPD)* and the path maintenance method. As said in the introduction, AGPF sends data along the anchored path. Anchored path is discovered by FAPD and is given with the list of geographical points that are called anchors. Anchors define a loose source route from source to the destination and between anchors location-based packet forwarding is performed. All acquired paths are maintained by a terminode by the path maintenance method. The global routing method is presented in pseudocode in Figure 1.

"use TLR" bit When close to the destination, if only the location information is used for packet forwarding, positional errors and inconsistent location information may result in routing errors and loops. This happens if the destination has considerably moved from the location that is known at the source. In order to cope with this problem, in our approach when close to destination, the packet forwarding method becomes TLR. TLR does not use location information. A terminode applies TLR for destinations that are at most two-hop away. Once a packet has been forwarded with TLR, the "use TLR" bit is set within the packet header, and downstream terminodes should not use TRR again.

How to expedite termination of TRR As we said in the introduction, we require that the mobility management gives the location information with precision of approximately one transmission range and validity of about ten seconds. Taking into account that the scope of TLR is around 200 meters and that terminodes move with the the maximum speed of 20 meters per second, in most cases the required location accuracy would be enough to insure termination of TRR.

But, if accuracy of location management is low or if the packet has been delayed due to congestion or bad paths, it may happen that the condition to set "use TLR" bit is never met and a packet may start looping. Our design point is to discover such loops and to drop looping packets. A sign of a loop happens when a terminode finds that the destination location written in the packet header is within its transmission range, but the destination is not TLR-reachable. In order to address this case, a terminode X performs the following action: if $distance(LDA_D, LDA_X)$ is less than the transmission range of X, and D is not TLR-reachable for X, X sets the TTL field within a packet header to the value equal to $min(term\_loop, TTL)$. $term\_loop$ is equal to 3. This effects that a loop due to destination location inaccuracy is always limited to $term\_loop$ hops.

3.2 Terminode Local Routing (TLR)

Terminode Local Routing (TLR) is inspired by the intrazone routing protocol (IARP) in ZRP[13]. This protocol allows to reach terminodes that are several wireless hops away, but is limited in distance and number of hops. Figure 2(a) describes TLR in pseudocode.

We say that terminode D is a *TLR-reachable* for terminode S if S has a means to reach D with the TLR protocol. The TLR-reachable area of S includes the terminodes whose minimum distance in hops from S is at most equal to *local radius*. The local radius is a measure, in number of hops, of the TLR-reachable area.

The only addressing information used by TLR is the EUI of the destination. Every terminode discovers the information (EUI, LDA) of the terminodes that are in its TLR-reachable area: EUI and LDA information is proactively maintained by the means of a HELLO message that every terminode periodically broadcasts at the MAC layer. LDA is not used for TLR, but it is added because it is necessary for TRR, as explained in the next section.

In the current implementation of TLR, the *local radius* is set to two hop. That is, TLR allows a terminode to discover identity and location information (EUI and LDA) of its one and two hop distant neighbours and to route packets to them. A terminode announces in a HELLO message its own EUI and LDA, as well as EUI and LDA of its immediate neighbours. TLR uses a simple two hop distance vector routing protocol to send data to TLR-reachable destinations.

It is also possible to use local radius greater than two. However, this would increase the TLR overhead due to the update traffic required for every node to maintain its TLR-reachable area. In addition, for greater local radius problems as is slow convergence of distance vector routing protocol would affect TLR in a highly mobile ad-hoc network.

3.3 Terminode Remote Routing (TRR)

Terminode Remote Routing (TRR) allows data to be sent to *non TLR-reachable* destinations. TRR consists of the following elements: *Geodesic Packet Forwarding*, *Anchored Geodesic Packet Forwarding (AGPF)*, *Friend Assisted Path Discovery (FAPD)* and the path maintenance method.

3.3.1 Geodesic Packet Forwarding

Geodesic Packet Forwarding is a simple method to send data to remote destinations. Unlike TLR, geodesic packet forwarding is based solely on locations. Similar method is used in GEDIR[16] and in the greedy mode of GPSR [8].

S sends packets by geodesic packet forwarding in the greedy manner: the packet is sent to some neighbour X within a transmission range of S where the distance to D is the most

```
Source  S has a packet for destination D:

 if (D  is in TLR-reachable region of S)
    S applies TLR; //fig.2(a)
else if (S has  anchored path(s) to D)
  S chooses one path and applies AGPF; //fig. 2(b)
else if (S has appropriate friend F1 in a friends list  to start FAPD)
 S sets "F" bit in the packet and sends a packet to F1; //fig.3
 else
       S  uses geodesic pkt. forwarding to D
```

(a)

```
Terminode X  receives a packet for destination D:
if (("use-TLR" bit is set) or (D  is in TLR-reachable region  of X))
   X applies TLR; //fig. 2 (a)
else if (dist(X,D) < transmission range of X)
  {TTL =min(term_loop, TTL);
   X uses geodesic pkt. forwarding to D;}
else if (packet contains anchored path )
     X performs  AGPF; //fig. 2 (b)
else X uses geodesic pkt. forwarding to D
```

(b)

Figure 1: (a)Packet forwarding algorithm at the source , (b)Packet forwarding algorithm at an intermediate terminode

```
Terminode X receives a packet for D:

if ("use-TLR" bit is not set)
  set  "use-TLR" bit;
if (D is immediate neighbour of X)
 send a packet directly to D;
else {
  Y= next-hop  to D;
  send the packet to Y; //fig. 1(b)
}
```

(a)

```
 X receives a packet for D with anchored path:
AP1=get anchor from a path;
if (AP1 is in not  within a transmission range of X)
  X uses geodesic packet forwarding to AP1;
else {
 delete AP1 from the path;
 AP2= get next anchor from the anchored path;
if (AP2 == NULL)
  use geodesic pkt. forwaring to D;
else use geodesic pkt.forwarding to AP2
}
```

(b)

Figure 2: (a) Terminode Local Routing (TLR), (b) Anchored Geodesic Packet Forwarding (AGPF)

reduced. In turn, X checks whether D is TLR-reachable. If this is not the case, X sends the packet to its neighbour that is closest to the destination. Otherwise, X uses TLR to forward the packet. X sets the "use TLR bit" bit within the packet header, thus preventing downstream terminodes from using TRR for this packet. In this simplest form, geodesic packet forwarding will often not work. If there is no connectivity along the shortest line due to obstacles or a terminodes desert, then the method fails. The packet may be "stuck" at some terminode that does not have a neighbour that is closer to the destination. One possible solution to this problem is to use the method of a planar graph traversal, where a packet is routed around the perimeter of the region where there are no terminodes closer to the destination. This solution is proposed in GPSR[8]. In this way a packet is routed until it arrives at the terminode that reduces the distance to the destination, and thereon the packet is forwarded in a greedy manner, as described above.

We propose a method called AGPF, to avoid holes in terminode distribution. It is completely based on routing towards a geographical point rather than towards a terminode, as explained in next section.

3.3.2 Anchored Geodesic Packet Forwarding (AGPF)

The key element of the *Anchored Geodesic Packet Forwarding (AGPF)*[3] are *anchors*. An anchor is a point, described by geographical coordinates; it does not, in general, correspond to any terminode location. In this scheme, a path is described by a list of anchors. Anchors are computed by source nodes, using the path discovery methods as presented below. Figure 2(b) presents AGPF in pseudocode.

A source terminode adds to the packet a route vector (anchored path) made of a list of anchors, which is used as loose source routing information. Between anchors, geodesic packet forwarding is employed. The source sends data to its immediate neighbour that has the minor distance to the first anchor in the route vector. When a relaying terminode receives a packet with a route vector, it checks whether the first anchor falls within its transmission range. If so, it removes the first anchor and sends it towards the next anchor or the final destination using geodesic packet forwarding. If the anchors are correctly set, then the packet will arrive at the destination with a high probability. Occasionally, when there is a hole in terminode distribution between two anchors, routing around the perimeter of a hole is used[8]. Figure 3 presents an example of AGPF.

3.3.3 Path Discovery

Friend Assisted Path Discovery (FAPD) is a path discovery method that is based on the concept of small world graphs[18]. Small world graphs are very large graphs that tend to be sparse, clustered and have a small diameter. 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; small world phenomenon allows people to speak of the "six-degrees of separation".

We view a terminode network as a large graph, with edges representing the "friend relationship". B is a *friend* of A if (1) A thinks 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 because A managed to maintain one or several anchored paths to B that work

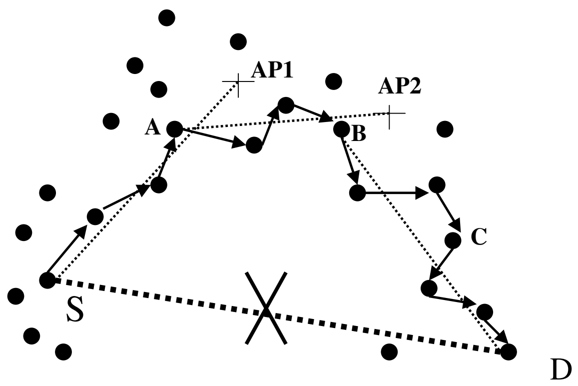

Figure 3: The figure presents how AGPF works when a terminode with EUI_S has some data to send to a terminode with EUI_D, and there is no connectivity along the shortest line from S to D. S has a path to D given by a list of geographical locations called anchors: $\{AP1,\ AP2\}$. First, geodesic packet forwarding in the direction of $AP1$ is used. After some hops the packet arrives at a terminode A which finds that $AP1$ falls within its transmission range. At A, the packet is forwarded by using geodesic packet forwarding in the direction of $AP2$. Second, when the packet comes to B, that is close to $AP2$, it starts sending the packet towards D. Last, when the packet comes to C it finds that D is TLR-reachable and forwards the packet to D by means of TLR.

well. Every terminode has a knowledge of a number of close terminodes in its TLR-reachable region; this makes a graph highly clustered. In addition, every terminode has a number of remote friends to which it maintains a good path(s). We conjecture that this 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 any two terminodes are likely to be connected with a small number of intermediate friends.

With FADP, each terminode keeps the list of its friends with the following information: location of friend, path(s) to friend and potentially some information about the quality of path(s).

FAPD is a distributed method to find an anchored path between the two terminodes in a terminode network. When a source A wants to discover a path to a destination C, it requests assistance from some friends, let's say B. If B is in condition to collaborate, it tries to provide A with some path to C (it can have it already or try to find it, perhaps with the help of its own friends). If the desired path it found, it is returned to A. Figure 4 presents FAPD in pseudocode.

To select which friend to contact, the source first chooses from a list of friends a set of friends that reduce the distance to the destination. All friends whose distance to destination are nearly equal are considered in this set. The first example illustrated in Figure 5 explains the case when this set is not empty . If this set is empty, the source may decide to contact a friend where the distance to the destination is not reduced. At the same time, it marks the occurrence of this exception by increasing the $taboo-index$, in order to prevent FADP from staying longer in taboo mode and assure the termination. This is illustrated in Figure 6.

The source may perform FAPD several times by contacting different friends. In this way the source can acquire multiple paths to the destination.

```
F1 is intended receiver of  a path discovery packet ("F" bit is set): S  needs a path to D
if (F1 == D) {send path reply with fapd_anchored_path  to S;}
else if (F1 has a path to D)
   append this path in fapd_anchored_path and send the packet to D;
else if  (packet in taboo-mode)
   {
    if ( F1 has a friend F2 where dist(F2, D) < min_dist)
      {exit  taboo-mode; taboo-index=0;  send the packet to F2}
    else if (taboo-index < 2 and F1 has a friend F3 such that dist(F1, F3) < max_dist )
         {  taboo-index++;  send a packet to F3}
    else // taboo-index reached the maximum value
        {send a packet to D by geodesic packet forwarding}
   }
else //packet not in taboo mode
 {
   if (F1 has a friend F2 where dist(F2,D)<dist(F1,D) )
    send a packet to F2;
  else if (F1 has a friend F3 such that dist(F1, F3) < max_dist )
       {start taboo mode; taboo_index=1; min_dist=dist(F1,D); send a packet to F3}
  else apply geodesic packet forwarding to D;
 }
```

Figure 4: Friend Assisted Path Discovery

Path Discovery Examples

In the first example, the source S, which has some data to send to D, has a friend $F1$ that is closer to D than any other friend of S. S sends data packet to $F1$ according to the existing path that S maintains to $F1$. In order that $F1$ can help S to forward the packet to D, S sets within a packet header "F" bit. When this bit is set, it denotes that the corresponding packet is a "path discovery packet". Inside the path discovery packet there is an $fapd\_anchored\_path$ field where anchor points are accumulated from S to D. If S has an anchored path to $F1$, S would append anchors of this path in $fapd\_anchored\_path$ field of the path discovery packet; S sends data to $F1$ by AGPF. Otherwise, S leaves this field empty and uses geodesic packet forwarding to $F1$. Upon reception of the path discovery packet, $F1$ puts its geographical location inside the $fapd\_anchored\_path$ field as one anchor. If $F1$ has an anchored path to D, $F1$ appends this path into $fapd\_anchored\_path$ field and send the packet to D by AGPF. Suppose now that $F1$ does not have a path to D, but has a friend $F2$ whose distance to D is smaller than the distance from $F1$ to D. If $F1$ has an anchored path to $F2$, $F1$ appends it in the $fapd\_achored\_path$ field of the path discovery packet, and sends the packet to $F2$. Once $F2$ receives the packet, it checks that D is TLR-reachable and $F2$ forwards the packet to D by TLR. When D receives the packet with set "F" bit, it should send back to S a "path reply" control packet with the acquired anchored path from S to D. Assuming that the path from S to $F1$ and from $F1$ to $F2$ does not contain any anchors, $fapd\_anchored\_path$ is thus a list of anchors (LDA_{F1}, LDA_{F2}).This list is sent from D to S by applying AGPF with the anchored path (LDA_{F2}, LDA_{F1}) (that is a reversed path from the one D received within the the path discovery packet).

Once, S receives from D a packet with the acquired anchored path, S stores this path in its route cache. S can send subsequent packets to D, by applying AGPF with the acquired anchored path. Otherwise, if S has not received the anchored path during some time, or if S

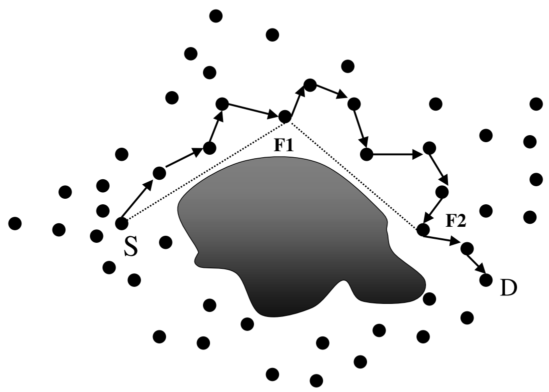

Figure 5: The figure presents how FAPD works when source S, has a friend $F1$ that is closer to D than S. S sends data packet to $F1$ and sets "F" bit in the packet header in order to denote that this is a "path discovery packet". Upon reception of the path discovery packet, $F1$ puts its geographical location inside the $fapd\_anchored\_path$ field of the path discovery packet as one anchor. In this example $F1$ does not have path to D, but has a friend $F2$ whose distance to D is smaller than the distance from $F1$ to D. $F1$ sends path discovery packet to $F2$. Once $F2$ receives the packet, it finds out that D is TLR-reachable and $F2$ forwards the packet to D by TLR. When D receives the packet with set "F" bit, it should send back to S a "path reply" control packet with the acquired anchored path from S to D. Assuming that the path from S to $F1$ and from $F1$ to $F2$ does not contain any anchors, the anchored path from S to D is thus a list of anchors (LDA_{F1}, LDA_{F2}).

needs more paths to D, S may start FAPD with some other friend.

With FAPD, a source or an intermediate friend normally attempts to send data to its friend that reduces the distance to the destination. However, there are situations when this is not possible because there is no friend closer to the destination. In some topologies with obstacles, at some point, going in the direction opposite from the destination may be the best way to get the path. This situation is presented in the second example illustrated with Figure 6. Here, S does not have a friend that is closer to D than itself. FAPD permits that the path discovery packet is sent to a friend even though the packet is not getting closer to the destination, and there "taboo-mode" of FAPD starts. However, such a friend should not be farther than max_dist from S. We use that max_dist is equal to five times the transmission range of a terminode. In addition, FAPD limits the number of times that the packet is forwarded to some friend that is farther from the position where the packet was closest to the destination. In Figure 6, S contacts its friend $F1$ that is farther from D, but such that $dist(S, F1) < max\_dist$. S sends the path discovery packet with "F" bit set to its friend $F1$. Inside the path discovery packet there is $taboo - index$ field that S sets to 1 and thus starts the "taboo mode" of FAPD. In addition, inside the packet header there is the field called min_dist. S puts in min_dist field the distance from S to D ($dist(S, D)$), that is the smallest distance to the destination that the packet achieved when "taboo-mode" is started. Upon reception of the path discovery packet, $F1$ finds out that it does not have a friend whose distance to D is smaller than min_dist. $F1$ forwards the path discovery packet to its friend $F2$ that is farther from D but such that $dist(F1, F2) < max\_dist$ and sets $taboo - index$ to 2. In our current implementation of FAPD we set the maximum value of the $taboo - index$ to two. This means, that the path discovery packet can be sent in sequence to, at most, two friends where the distance to the destination is not smaller than min_dist. Upon reception of

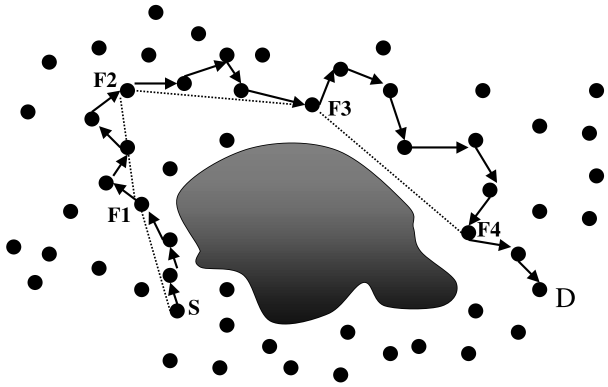

Figure 6: The figure presents how FAPD works when the source S does not have a friend that is closer to D than itself. S contacts its friend $F1$ that is farther from D in geometrical distance than S is, but such that $dist(S, F1) < max\_dist$. As in the previous example, S sends data packet to $F1$ with "F" bit set. In addition S sets the $taboo - index$ field to 1 and thus starts the "taboo mode" of FAPD. S puts $dist(S, D)$ within min_dist field. Upon reception of the path discovery packet, $F1$ finds out that it does not have a friend whose distance to D is smaller than min_dist. $F1$ forwards the path discovery packet to its friend $F2$ where $dist(F1, F2) < max\_dist$, and sets $taboo - index$ to 2. Upon reception of the packet, $F2$ checks that $taboo - index$ is equal to its maximum value equal to 2, 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 $F3$, $taboo - index$ is reset to 0. This means that FAPD is not longer in "taboo mode". From $F3$ packet is forwarded to its friend $F4$ and from there to D by using the TLR protocol.

the packet, $F2$ checks that $taboo - index$ is equal to its maximum value of 2, 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. At $F2$ the packet is no more in taboo mode, $taboo - index$ is reset to 0 and a packet is forwarded to $F3$. Otherwise, if $F2$ does not have a closer friend, $F2$ would forward the packet to D by geodesic packet forwarding.

3.3.4 Path Maintenance

Every terminode normally attempts to maintain multiple anchored paths to the destinations that it communicates with. Multipath routing is a way to cope with uncertainty in a terminode network; the paths that a source has acquired by FAPD can deteriorate due to mobility and packets can be lost. We advocate that the source data is encoded and sent over multiple independent paths in order to provide better load balancing and path failure protection. Diversity of paths is essential for taking advantage of multipath routing [14].

Path maintenance consists of three main functions: independent path selection, path simplification; path monitoring and deletion; congestion control.

independent path selection A terminode analyzes all acquired paths to a destination. Then its selects a set of independent paths. They are paths that are as diverse as possible in geographical points (anchors) that they consist of.

path simplification One method consists in approximating an existing path with a path with fewer anchors. Such an approximation yields a candidate path, which may be better or worse than the old one. We use a heuristic based on curve fitting.

path monitoring and deletion A terminode constantly monitors existing paths in order to collect necessary information to give the value to the path. The value of the path is given in terms of congestion feedback information such as packet loss and delay. Other factors like robustness, stability and security are also relevant to the value of a path.

This allows a terminode to improve paths, and delete mal-functioning paths or obsolete paths (e.g the path that corresponds to two terminodes that do not communicate any more).

congestion control The value of the path given in terms of congestion feedback information is used for a terminode to decide how to split the traffic among several paths that exist to the destination. A terminode gives more load to paths that give least congestion feedback information.

4 Performance Evaluation

We simulated the terminode routing protocol (the TLR and TRR protocols) in GloMoSim[17]. GloMoSim is a scalable simulation environment for wireless network systems. It is based on the parallel, discrete-event simulation language PARSEC[1]. The MAC layer is implemented using the distributed coordination function(DCF) of IEEE 802.11 standard[6]. In our experiments, radio range is the same for every terminode, and is equal to 250 meters. The channel capacity is 2Mbits/sec.

Movement Model

In the most recent papers about mobile ad-hoc networks simulations, nodes in the simulation move according to the "random waypoint"[4] model. In the "random waypoint" mobility model, a node chooses one random destination in the simulation area. Then it moves to that destination at a speed distributed uniformly between 0 and some maximum speed. Upon reaching the destination, the node pauses for *pause time*, selects another random destination inside the simulation area, and proceeds there as previously described.

We find this model unrealistic for a wide area mobile ad-hoc network such as a terminode network. In this network, terminodes are small personal devices that are distributed geographically within a very large area. It is less probable that a terminode for each movement selects a random destination within a very large geographical 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. We implemented a new mobility model that we call "restricted random waypoint". This model is closer to a real-life situation for a wide-area mobile ad-hoc network than the "random waypoint".

For the "restricted random waypoint" mobility model, we introduce the topology based on towns and highways. Towns are areas that are connected with highways. Inside town areas, terminodes are moving with the "random waypoint" mobility model. After a certain

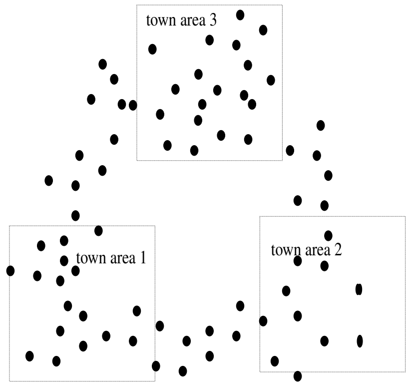

Figure 7: Model of the simulation area with three towns

number of movements in the same town, a terminode moves to another town. Terminodes that are moving between the town areas, simulate highways between towns.

In our simulations we use a simple topology that consists of three town areas with a highway between each town. A town is denoted by a town center and the town area around the town center. The model of the simulated area is presented in Figure 7.

We distinguish two types of restricted mobility that represent the "ordinary terminode" and the "commuter terminode".

An "ordinary terminode" begins the simulation by selecting at random one of three town areas in which it is going to stay. It moves to one random destination within a selected town area at the speed distributed uniformly between 0 and some maximum speed. Upon reaching the destination within a selected town, the ordinary terminode pauses for *pause time*, selects another destination within the same town, and proceeds there as previously described. Thus, the ordinary terminode's movement inside a town is based on the random waypoint mobility model. It repeats such movements for a number of times set by *stay_in_town* paremeter. Then a terminode selects a new random town to which it moves and once it is there, it applies the random waypoint mobility model for another *stay_in_town* time.

There are also a number of terminodes that frequently commute from one town to another. Those termnodes are called "commuters" and they insure the connectivity between towns. A commuter's movement model is the "restricted random waypoint" where *stay_in_town* parameter is equal to one. A commuter selects a random destination within one town area and moves to that destination with a speed distributed uniformly between some minimum speed and some maximum speed. Once this destination is reached, a commuter pauses for a some small pause time. Then it selects at random another town and the random destination inside the chosen town, and moves to this destination. It pauses in the second town for a small interval of time and repeats its behaviour.

Communication Model

We simulated 30 CBR traffic flows originated by 15 sending sources. Each CBR flow sends at 2kbps, and uses 256-byte packets. All communication patterns were peer-to-peer, and CBR connections were started at times uniformly distributed between 500 and 700 seconds, and they lasted until the end of simulation. CBR connections were started after at least

500 seconds in order that terminodes come from their initial positions at time 0 to their town areas and from there start moving as ordinary or commuter terminodes. All source destination pairs were chosen from the group of ordinary terminodes.

Scenario Characteristics

A source terminode normally tries to acquire several anchored paths to the destination of interest and use them in multipath routing. This is done with a path discovery method. A source tries to analyze all anchored paths that it learns and only uses "diverse" paths for packet forwarding . In our simulations we use a simple method for choosing the friends: when S is a CBR source of data packets for F, S selects F as its friend, and F selects S as its friend. We note here that in our simulations, due to the simple topology of three towns, whenever a couple source-destination is in two different towns, we have only two diverse paths: one direct path along the "highway" between two different towns, and the second path via the third town. Therefore, in our simulations, FAPD selects which friend to contact by the following method: when S has some data to send to D that is in another town than itself, S contacts its friend F if F's location is in the third town. In order to avoid loop conditions, S uses geodesic packet forwarding to send packets to its friend F (direct path) even thought S may have also an anchored path to F. This is because the anchored path may go via the D's town area.

For example, suppose that a terminode S is in the area of town 1 and has some data packets to send to $D1$ that is located in the area town 2. S puts $D1$ in the list of its friends. S sends data to $D1$ by geodesic packet forwarding (along the direct path from S to $D1$). Then, after some time, S has some packets to send to $D2$ that is situated in the area of town 3. S starts sending packets by using the geodesic packet forwarding from town 1 to town 3. But, in order to learn more paths to $D2$, S contacts its friend $D1$. At first S acquires location of $D1$. S checks whether this location is close to the one where $D1$ was positioned at the time when S put $D1$ in its list of friends and to where S knows how to route[1] . If so, S sends some data packets to $D1$ with the "F" bit set in the packet header ("F" bit set means that the packet is a path discovery packet). When $D1$ receives a path discovery packet it puts its location in the $fapd\_anchored\_path$ field, and forwards the packet to the destination terminode $D2$ by geodesic packet forwarding. Upon reception of a data packet with a "F" bit set, $D2$ sends back to S a control packet with $fapd\_anchored\_path$. Now S has available two paths to $D2$: the anchored path via town 2 and the direct path to town 3. Then for every data packet that it has to send to $D2$, S uses at random one of the two available paths. S applies geodesic packet forwarding if the direct path is used; otherwise, if the anchored path is used, S applies AGPF.

We have not implemented, so far, any form of congestion feedback control that would help the source to know how to split data among several existing paths to the destination.

Some Implementation Details

Each terminode maintains a neighbour table of its immediate neighbours, as well as each neighbour's neighbour. Each entry in the neighbour table includes the terminode's identifier (EUI), its location (LDA) and a timestamp. A terminode periodically broadcasts a HELLO

[1] In our simulations, ordinary terminodes stay for most of a simulation duration within the same town area. This is achieved by setting of the parameter $stay\_in\_town$ to a large number equal to 20

message with the list of its neighbour. When a node receives a HELLO message, it updates its neighbour table with the information in the HELLO message. A terminode thus learns about its immediate and two-hop distant terminodes. When a terminode learns about its two-hop distant terminode it keeps the sender of a HELLO message as the next hop terminode via which a two-hop away terminode can be reached. TLR uses the next hop information when a packet is to be sent to the two-hop distant destination. In our implementation every terminode sends a HELLO message every 1 second. Each entry in the neighbour table expires after 2 seconds if it is not updated.

The 802.11 MAC layer notifies when a unicast packet exceeds the maximum number of retransmissions and the acknowledgement has not arrived. This means that the intended neighbour has left the sender's transmission range and that the entry that corresponds to that neighbour is invalid and can be removed from a sender's neighbour table. In our implementation such a packet is sent back to the routing layer where a new neighbour to send a packet is chosen.

Simulation Results

We evaluate the combination of terminodes routing protocols(TLR and TRR) according to the following two metrics:

- *packet delivery ratio* is the ratio between the total number of packets originated by CBR sources and the number of packets received by CBR sink at final destinations.

  Figures 8 and 9 show the fraction of originated data packets that the terminode routing protocol was able to deliver as a function of the ordinary terminode mobility rate. The first figure is obtained when there are 100 ordinary and 300 commuter terminodes. The simulation area is 3500 meter X 2500 meter. Centers of three towns are located at (550 m ,550 m), (1750 m, 2000 m) and (3000 m, 550 m) respectively. The town area is a square around the town center with the width of 600 m from the town center. The second figure corresponds to a larger network both in number of terminodes and the geographical coverage. There are 100 ordinary and 500 commuter terminodes. In this second case the simulation area is 4500 meter X 3500 meter. Centers of three towns are located at (550 m ,550 m), (2000 m, 2200 m) and (3500 m, 550 m) respectively.

  In both cases the maximum speed of an ordinary terminode is 10 meters per second, and *stay_in_town* parameter is set to 20. We simulate pause times of an ordinary terminode of 0, 50,100, 150 and 200 seconds. In our simulations, the pause time acts as a degree of ordinary terminode mobility. A longer pause time means that ordinary terminodes are less mobile. We consider different mobility rates of ordinary terminodes because this is set of nodes where all CBR sources and destinations come from. In our simulations, commuters are moving faster than ordinary terminodes. Their minimum speed is equal to 10 meters per second and the maximum speed equal to 20 meters per second. The pause time for commuter terminodes is equal to 1 second.

  We simulated at each pause time with five different motion patterns, and present the mean of each metric over these five runs.

  In both simulated networks terminodes routing succeeded to deliver over 80% of data packets.

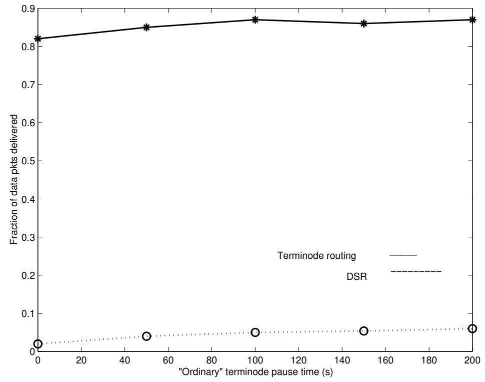

Figure 8: Packet delivery success rate of terminode routing and DSR with varying "ordinary" terminodes' pause time. 400 nodes (100 "ordinary" and 300 "commuter" terminodes)

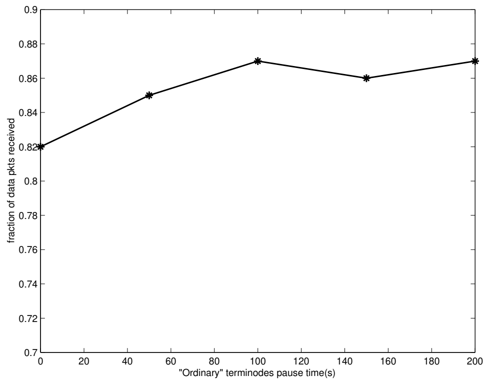

Figure 9: Packet delivery success rate of terminode routing with varying "ordinary" terminodes' pause time. 600 nodes (100 "ordinary" and 500 "commuter" terminodes)

For comparison, we present in the same figure the fraction of data packets delivered when the DSR[4] routing protocol is used in the same simulation scenario. For DSR, we used simulation parameters identical to those used by authors of DSR in [4].

DSR succeeded in delivering only a small fraction of data packets. With DSR less than 10% of the data packets were delivered. This can be explained by the fact that DSR builds strict source routes from source to destination. DSR is optimized for relatively smaller networks. In our simulation scenarios on a large network with highly mobile nodes, DSR source routes are frequently broken. The routes that contain links between commuters terminodes are especially susceptible to be broken, since these nodes are highly mobile. DSR uses a mechanism of route maintenance to repair the broken source routes. However, this causes that a large number of control messages floods the network, that can induce congestion in the network.

- *routing overhead* can be expressed as the total number of routing packets transmitted during the simulation. Terminodes routing generates two types of protocol packets. TLR uses HELLO messages, while TRR uses control messages that are needed for

FAPD. Every terminode proactively generates HELLO messages every second and those messages are received but not forwarded by its neighbours. Overhead due to HELLO messages is independent of mobility rate of terminodes and the number of traffic flows. As the size of the network increases, the network-wide count of HELLO messages increases. However, at a constant terminode density, size of the network does not have effect on local TLR overhead at each node, since HELLO messages are not propagated beyond a single hop.

In FAPD, source piggybacks "path request" packets in data packets that it sends to the destination. Those packets are sent for every new anchored path that the source wants to acquire. The "path request" packet is forwarded to the destination through a number of intermediate friends. Upon reception of this packet, the destination sends back to the source a "path reply" packet with the anchored path. If a destination has some data packets to send to the source, "path reply" can be piggybacked in the data packet. Thus, the overhead due to FAPD control messages is not big.

5 Future Work and Conclusions

We focused on the problem of routing in a wide area mobile ad-hoc network called Terminode Network. Routing in such a network is designed with the following objectives. Firstly, it should scale well in terms of the number of nodes and geographical coverage. Secondly, routing should have scalable mechanisms that cope with the dynamicity in the network due to mobility. Our routing scheme is a combination of two protocols called Terminode Local Routing (TLR) and Terminode Remote Routing (TRR). TRR is activated when the destination D is remote and uses location of the destination obtained either via location management or by location tracking. TLR acts when the packet gets close to the destination and uses routing tables built with hello messages. The use of TRR results in a scalable solution that reduces dependence on the intermediate systems, while TLR allows to reduce problems of loops due to location inaccuracy. The simulations that are performed illustrate that the combination of TRR and TLR was able to deliver over 80% of user data. In the same simulation environment DSR succeeded to deliver less than 10% of user data. In our simulations, we introduced the topology based on three towns and nodes are moving between towns according to the movement model that we called "restricted random waypoint".

In our design, multipath routing is intended for coping with the dynamicity in the network due to mobility. However, we note here that we were not able to exploit the benefits of multipath routing given the simplicity of our topology based of three towns. In the future we will perform simulations on a more complex topologies where we can benefit in more diverse paths. We anticipate that the multipath routing in that case would perform better than when only a single path is used.

References

[1] R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, and H.Y. Song. PARSEC: A Parallel Simulation Environment for Complex Systems. *IEEE Computer*, 31(10), October 1998.

[2] 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.

[3] L. Blazevic, S. Giordano, and J.-Y. Le Boudec. Self-Organizing Wide-Area Routing. *Proceedings of SCI 2000/ISAS 2000, Orlando*, July 2000.

[4] 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.

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

[6] 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.

[7] J.-P. Hubaux, J.-Y. Le Boudec, S. Giordano, M. Hamdi, L. Blazevic, L. Buttyan, and M. Vojnovic. Towards Mobile Ad-Hoc WANs: Terminodes. *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'00),*, Chicago, USA, September 2000.

[8] 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.

[9] 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.

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

[11] 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.

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

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

[14] M.R. Perlman, Z.J. Haas, P. Sholander, and S.S. Tabrizi. On the impact of alternate path routing for load balancing in mobile ad hoc networks. In *Proceedings of the First IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC)*, August 2000.

[15] Elizabeth M. Royer and C.-K. Toh. A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks. *IEEE Personal Communications Magazine*, pages 46–55, April 1999.

[16] I. Stojmenovic and X. Lin. GEDIR: Loop-free location based routing in wireless networks. *Proceedings of the IASTED Int. Conf. on Parallel and Distributed Computing and Systems*, Boston, USA, November 1999.

[17] 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.

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

[19] 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.