Self Organized Terminode Routing

Ljubica Blažević, Silvia Giordano, Jean-Yves Le Boudec
Institute for computer Communication and Applications (ICA)
Communication Systems Department (DSC)
Swiss Federal Institute of Technology Lausanne (EPFL)
CH-1015 Lausanne, Switzerland
(ljubica.blazevic, silvia.giordano, jean-yves.leboudec)@epfl.ch

We consider the problem of routing in a wide area mobile ad-hoc network called Terminode Network. Routing in this network is designed with the following objectives. First, it should scale well in terms of the number of nodes and geographical coverage; second, routing should have scalable mechanisms that cope with the dynamicity in the network due to mobility; and third, 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. 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 as a rule multipath routing is considered; 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 performed simulations of the TLR and TRR protocols using the GloMoSim simulator. The simulation results for a large, highly mobile ad-hoc environment demonstrate benefits of the combination of TLR and TRR over an existing protocol that uses geographical information for packet forwarding.

Keywords: ad-hoc, mobile, wide-area network, self organized routing

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 concentrate 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 source 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]).

Here we assume that nodes move with the speed that corresponds to pedestrian or car speed, so that we can obtain (with the mobility management) the destination LDA with the precision of approximately one transmission range and validity of about ten seconds. We also assume that terminode network is mainly connected, although temporary partitions can occur. Even if our protocol uses geographical locations, it is independent from the physical infrastructure (i.e, it does not assume directional antennae) and from the physical underlying layer. We performed simulations with IEEE 802.11 MAC protocol, because it is a commonly used MAC protocol. We anticipate that our routing protocol would have the similar results with other MAC protocols.

We further 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 only locations are used for making packet forwarding decisions, positional errors and inconsistent location information can 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. 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. 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.

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 proactive 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 those protocols, routing tables are exchanged among neighbouring nodes each time a change occurs in the network topology. As a consequence, proactive 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 as 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 possesses 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 proactive 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 proactive 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 proactive 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 location. 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.

In Geographical Routing Algorithm (GRA)[22], every node has only a partial knowledge of a network. It knows about its immediate neighbors and a small number of remote nodes to which it has discovered a path. When an intermediate node receives a packet to forward, it checks which of the nodes that it knows is closest to the destination. Then the packet is forwarded to the neighbour that is next hop towards the node that is closest to the destination. Each node thus forwards the packet in the similar way till the packet reaches the destination. If it happens that some node S does not know about any node that is closer to the destination D than itself, a route discovery method is invoked. This method finds an acyclic path from S to D and all intermediate nodes on update their routing tables with the next hop information in order to reach D.

GEDIR[16], GPSR[8] and GFG[21] routing protocols are very important to this paper, and we present them in more detail. These protocols 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 packet forwarding is loop-free provided that location information is accurate.

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 and GFG use a planar subgraph of the wireless network's graph to route around the perimeter of a hole. This method is first proposed by Bose *et al.* in [20]. Packet forwarding for such a packet is switched from greedy to perimeter mode. The knowledge of identities and locations of its one-hop neighbours is sufficient for a node to determine the edges of the planar subgraph. Packets that are in perimeter mode are forwared 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, and then 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, if only geographical locations are used for making packet forwarding decisions, this 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

Terminode routing uses geographical information in making routing decisions. However, when the packet gets close to the destination, packet forwarding uses local routing tables that every terminode proactively maintains for its close terminodes.

Our routing scheme is a combination of two protocols, Terminode Local Routing (TLR) and Terminode Remote Routing (TRR).

TLR determines a route to destination D when D is in the local routing table (of a source or of an intermediate terminode). Otherwise, TRR uses mainly the locations of S and D to discover a path from S to D.

Below we describe elements of terminode routing in more details.

3.1. Terminode Local Routing (TLR)

Terminode Local Routing (TLR) is used by terminodes to build their local routing tables, and for forwarding packets to destinations in their vicinity.

The idea of TLR was inspired by the intrazone routing protocol (IARP) in ZRP[13].

We say that terminode D is *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 distances in hops from S are at most equal to *local radius*. The local radius is a measure, in number of hops, of the TLR-reachable area. In the current implementation of TLR, the *local radius* is set to two hops.

Building of local routing tables In order to build its local routing table, every terminode proactively maintains the identity (EUI) and location[1] (LDA) of the terminodes that are no more than two hops away. This is done by means of HELLO messages that every terminode periodically broadcasts at the MAC layer.

A terminode announces in a HELLO message its own EUI and LDA, as well as EUIs of its immediate neighbours. Upon reception of a HELLO message, a node updates its local routing table with EUI and LDA of its one-hop and two-hop distant terminodes. For two-hop distant terminodes, a terminode also keeps in its table the next hop terminode via which a two-hop distant terminode can be reached. If a node does not hear from its neighbour for some amount of time, it removes from the routing table the entry that corresponds to the lost neighbour, as well as all entries that correspond to two-hop distant terminodes that were reachable via the lost neighbour.

TLR packet forwarding TLR uses a simple distance vector routing protocol to send data to TLR-reachable destinations. The only addressing information used by TLR is the EUI of the destination.

When the source, or an intermediate node, has to forward a packet to the destination that is TLR-reachable, the "use TLR" bit in the packet header is set, if not already set. If the destination is two-hop away, the packet is sent to the next hop terminode, as from the routing table. Otherwise, the packet is sent directly to the destination.

It is also possible to use a local radius larger than two. However, this would increase the TLR overhead because of the update traffic required for every node to maintain its TLR-reachable area. In

[1] The LDA information is not used for TLR, but it is added because it is used by TRR, as explained below.

addition, for a larger local radius, the slow convergence problem, typical to distance vector routing protocol, would affect TLR.

3.2. Terminode Remote Routing (TRR)

Terminode Remote Routing (TRR) allows data to be sent to *non TLR-reachable* destinations. TRR primarily forwards packets on *anchored paths.* In contrast with traditional routing algorithms, an anchored path does not consist of a list of nodes to be visited to reach the destination. An anchored path is a list of fixed geographical points, called *anchors*. In traditional paths made of lists of nodes, if nodes move far from the location they had at the time when the path was computed, the path cannot be used to reach the destination. Given that geographical points do not move, the advantage is that an anchored path is always "valid".

In order to forward packets along an anchored path, TRR uses the method called *Anchored Geodesic Packet Forwarding (AGPF).* With AGPF the packet is sent in the direction of an anchor, thus trying to reach some terminode in proximity of this anchor. Thereon, the packet is forwarded in the direction of the next anchor on the anchored path. Anchored paths are obtained by the source with the method called *Friend Assisted Path Discovery (FAPD).* If the source does not have an available anchored path to the destination, it uses a default method called *Geodesic Packet Forwarding (GPF).*

3.2.1. Geodesic Packet Forwarding (GPF)

GPF is a simple method for sending data in the direction of a geographical point. This point can be an anchor (see AGPF below) or the destination location. Unlike TLR, GPF is based solely on locations. A similar method is used in GFG[21] and in GPSR [8].

Source S can use GPF to send data to remote destination D in the following way. At first, S acquires some approximate value of the D's LDA. Then S sends packets by GPF 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 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.

In this simplest form, GPF will often not work. If there is no connectivity along the shortest line due to obstacles or a terminode 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 GFG and GPSR). 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.

GPF works well when there is a good connectivity along the shortest line from the source to the destination. However, this may not always work. In order to circumvent holes in terminodes distribution in a large area mobile ad-hoc network, we introduce the method called AGPF.

3.2.2. Anchored Geodesic Packet Forwarding (AGPF)

The key element of the *AGPF* is *anchored path.*

Anchors are computed by source nodes, using the path discovery method called FAPD, as presented below. A source terminode adds to the packet the anchored path that is used as loose source routing information. With AGPF the packet is forwarded such that it loosely follows an anchored path. The sequence of intermediate terminodes on the way to the destination depends on the actual terminode distribution.

AGPF works as follows. At the source, the packet is sent in the direction of the first anchor (AP1) on the anchored path by applying geodesic packet forwarding: the source sends data to its immediate neighbour that has the minor distance to AP1. When an intermediate terminode receives a packet with the anchored path, it checks whether AP1 geographically falls within its transmission range. If so, it deletes AP1 from the anchored path and sends the packet in the direction of the next anchor (AP2). This is repeated until all anchor points are deleted from the anchored path. Then the packet is sent in direction of the final destination by using GPF as described in the previous section.

If the anchors are correctly set, then there is a high probability that the packet will arrive at the destination. This is verified in our simulations (see Section 4.2). A good anchored path directs packets along regions with good terminode connectivity. Occasionally, when there is a hole in terminode distribution between two anchors, routing around the perimeter of a hole is used. Figure 1 presents an example of AGPF.

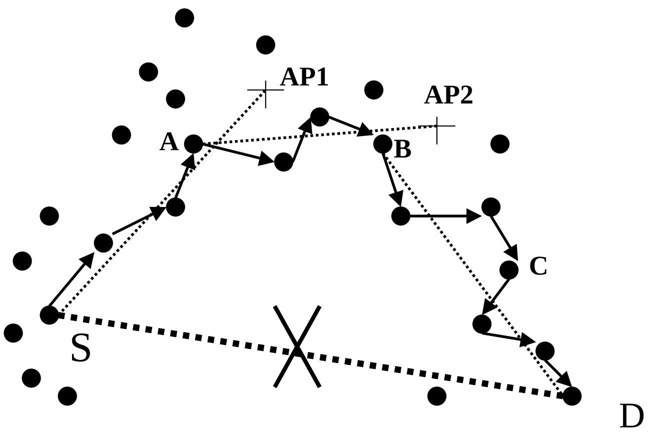

Figure 1. The figure presents how AGPF works when a terminode with EUI_S has some data to send to a terminode with EUI_D, and there is no connectivity along the shortest line from S to D. S has a path to D given by a list of geographical locations called anchors: $\{AP1, AP2\}$. First, geodesic packet forwarding in the direction of $AP1$ is used. After some hops the packet arrives at a terminode A that 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.

3.2.3. How to expedite termination of TRR

With TRR the packet gets closer to the destination. 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. Thereon, TLR can be only used for packet forwarding. Termination of TRR is important in order to avoid loops that may happen if packet forwarding from TLR reverts to TRR.

However, 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 approach is to discover such loops and to drop looping packets. We recognize a possible loop 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 a fixed value, which indicates that a loop due to destination location inaccuracy is always limited to $term\_loop$[2] hops.

3.2.4. Path Discovery

Friend Assisted Path Discovery (FAPD) is a method to obtain anchored paths. It 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. 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; small world phenomenon al-

---

[2] In our current implementation of TRR $term\_loop$ is equal to 3.

lows 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 by geodesic packet forwarding, or because A managed to maintain one or several anchored paths to B that work 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 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 to find an anchored path between two terminodes in a terminode network. When a source S wants to discover a path to destination D, it requests assistance from some friend, let’s say F. If F is in condition to collaborate, it tries to provide S with some path to D (it can have it already or try to find it, perhaps with the help of its own friends). Figures 2 and 3 present FAPDP in pseudocode at the source and at an intermediate friend.

```
if (S has a friend F1 where dist(F1,D)<dist(S,D) )
  {S sets “F” bit in the packet header; send a packet to F1;}
else if (S has a friend F3 such that dist(S, F3) < max_dist )
    {S sets “F” bit in the packet header;
     tabu_index=1; min_dist=dist(S,D); //start tabu mode
        send the packet to F3;}
else apply geodesic packet forwarding to D;
```

Figure 2. Friend Assisted Path Discovery Protocol at the source

```
F1 is intended receiver of  a path discovery packet (“F”bit = 1 ): 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  (tabu_index  > 0 ) //packet in tabu mode
   {
    if ( F1 has a friend F2 where dist(F2, D) < min_dist)
     {tabu_index=0;  send the packet to F2}
    else if (tabu_index < 2 and F1 has a friend F3 such that dist(F1, F3) < max_dist )
        {  tabu_index++;  send a packet to F3}
    else // tabu_index reached the maximum value
       {send a packet to D by geodesic packet forwarding}
   }
else //packet not in tabu 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 )
     {tabu_index=1; min_dist=dist(F1,D); send a packet to F3}
  else apply geodesic packet forwarding to D;
}
```

Figure 3. Friend Assisted Path Discovery Protocol at the intermediate friend and at the destination

When source S, which has some data to send to D, has some friends that are closer to D than S itself, it selects friend $F1$ that is closest to D, and starts FAPDP with $F1$. S sends the data packet to $F1$ according to the existing path that S maintains to $F1$ because $F1$ is a friend of S. S sets, within the data packet header, “F” bit[3]. This denotes that the corresponding packet is a *path discovery packet.*

When $F1$ receives this packet it recognizes the packet as a request of a 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 put anchors of this path in $fapd\_anchored\_path$ field (S sends data to $F1$ with AGPF). Otherwise, S leaves this field empty (in this case S sends to $F1$ with geodesic packet forwarding). Upon reception of the path discovery packet, $F1$ puts its geographical location inside $fapd\_anchored\_path$ field as one anchor. If $F1$ has an anchored path to D, $F1$ appends this path into $fapd\_anchored\_path$ field and

---

[3] “F” bit is not reset before reaching D

sends the packet to D by AGPF. 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 intermediate node that finds D to be TLR-reachable 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 terminode T (the source or an intermediate friend) sends a path discovery packet to a 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*[4]. Here is where "tabu mode" mechanism of FAPDP starts. With the tabu mode mechanism, the packet can be sent in direction opposite to D for a limited number of times. Tabu mode is denoted at T by setting the *tabu_index* field inside the packet to 1 (default value of *tabu_index* is 0).

Tabu mode mechanism uses a field called *min_dist*, where the terminode that started tabu mode puts its distance to the destination. When an intermediate friend $F1$ receives the path discovery packet, which is in tabu mode, it first checks if it has a friend whose distance to D is smaller than *min_dist*. If this is the case, the packet is sent to such a friend, and *tabu_index* is reset to 0. Otherwise, $F1$ may forward the packet to its friend $F2$ whose distance to D is more than *min_dist* and $F2$ increments *tabu_index*. In FAPDP, the number of times that the packet is forwarded to a friend that is further from D than *min_dist* is limited to two (i.e, the value of *tabu_index* must not be larger than two). Tabu mode mechanism stops either because a friend that is distant from D less than *min_dist* is found, or because *tabu_index* is equal to 2. In the second case the packet is forwarded directly to D by geodesic packet forwarding.

Finally, when D receives the packet with "F" bit equal to one, D must send back to S a "path reply" control packet with the acquired anchored path from S to D. This packet is sent to S by reverting the anchored path and applying AGPF. Once S receives from D a packet with the anchored path, S stores this path in its route cache.

If S does not receive a anchored path within some time, or if S wants more paths to D, S starts FAPDP with some other friend.

FAPDP is illustrated with two examples. The first example, presented in Figure 4 shows the case where the path from S to D is found by using two intermediate friends. The second example, in Figure 5 illustrates the tabu mode of FAPDP.

Friends Management

Friends Management (FM) is a set of procedures for selecting, monitoring and evaluating friends. For each node A, FM maintains a (fixed-size) set of nodes: the *set of friends*. The set of friends contains the nodes that are contacted with $FAPDP$ for discovering paths. These nodes are periodically evaluated in order to assure their availability and their validity as friends. For that reason, the *Friends Monitoring* component of FM keeps under control, for a node A, a set of parameters for each friend of A. These parameters ranges from technical characteristics (as, for example, the active time, the average distance, etc..) to more "social" information (as being at disposal for supporting $FAPDP$ or forwarding packets). Based on these parameters, the *Friends Evaluation* component periodically evaluates whether it is beneficial to keep a node in the friend set or it is better to discard it.

It is the responsibility of *Friend Selection* to identify a set of nodes that are suitable to act as friends. A node B can be selected to become a friend of a node A because there are frequent communications between A and B, because B is often in a strategic position for A's communications, be-

---

[4] we use *max_dist* equal to five times the transmission range of a terminode

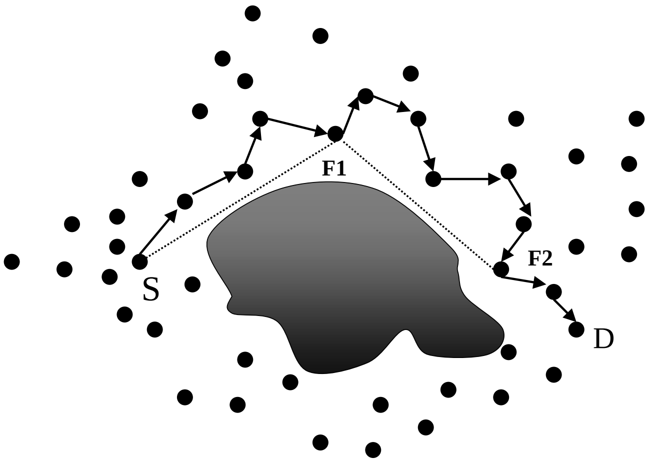

Figure 4. The figure presents how FAPDP works when source S, has a friend $F1$ that is closer to D than S. S sends data packet to $F1$ and sets "F" bit in the packet header in order to denote that this is a "path discovery packet". Upon reception of the path discovery packet, $F1$ puts its geographical 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}).

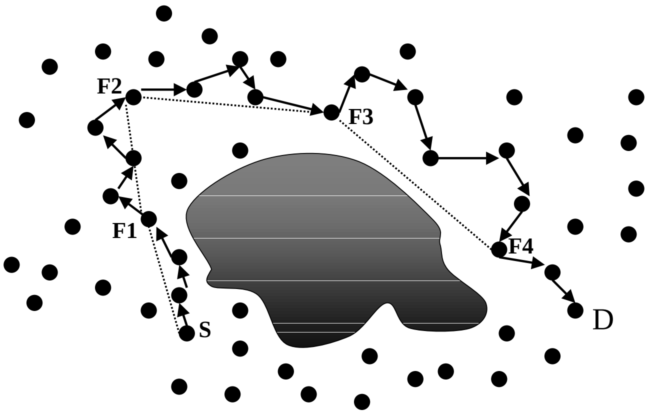

Figure 5. The figure presents how FAPDP works when source S does not have a friend that is closer to D than itself. S contacts its friend $F1$ that is farther from D in geometrical distance than S is, but such that $dist(S, F1) < max\_dist$. As in the previous example, S sends data packet to $F1$ with "F" bit set. In addition S sets the $tabu\_index$ field to 1 and thus starts the tabu mode of FAPDP. 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 $tabu\_index$ to 2. Upon reception of the packet, $F2$ checks that $tabu\_index$ 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 $F3$, $tabu\_index$ is reset to 0. This means that FAPDP is not longer in tabu mode. From $F3$ packet is forwarded to its friend $F4$ and from there to D by using the TLR protocol. The anchored path from S to D is thus a list of anchors $(LDA_{F1}, LDA_{F2}, LDA_{F3}, LDA_{F4})$

cause B is supposed to be helpful for discovering good paths or for some other reasons.

Once a potential friend has been identified, it will be evaluated by the *Friend Evaluation* component, and included in the set of friends if there is still space in set or if the node's evaluation is better than the one of some other node of the set. FM is critical in the initial phase (bootstrapping). When a node bootstraps, it does not have any information of (possible) friends. In this phase, the *Friend Selection* component will use the HELLO messages for identifying possible friends (that is, in the initial phase, will be limited to 1-2 hops neighbours). At run-time, these initial friends will disappear very likely, for being substituted by more valid friends as described above.

3.2.5. 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 FAPDP 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 and 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 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 IEEE 802.11 Medium Access Control(MAC) protocol is used; it implements the Distributed Coordination Function (DCF)[6]. In all simulations, radio range is the same for every terminode, and is equal to 250 meters. The channel capacity is 2Mbits/sec.

In order to build its local routing table, 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. To avoid synchronization of HELLO messages, a terminode jitters each HELLO message transmission by 50% of the period of a HELLO message. 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.

In this section we evaluate both components of the terminode routing, namely TLR and TRR.

4.1. Evaluation of TLR

To assess the relevance of TLR, we analyze a performance of terminode routing when TLR is used and when TLR is not used. We performed simulations in a scenario where geodesic packet forwarding towards the destination works well, and there is no need for the AGPF technique. We wanted to evaluate solely the performance of geodesic packet forwarding with TLR, against the case when TLR is not used.

Geodesic packet forwarding (GPF) uses destination location for making packet forwarding decisions. Therefore, it is important that the source knows this information accurately enough. We recall that in our proposal, once the communication has begun, terminodes are assumed to use location tracking to exchange their current locations. This is enough to assure valid location information in several situations. However, there are situations where location tracking is not able to give regular periodic location information (e.g, when GPS is temporary unavailable or when location tracking packets are lost).

In these cases, we expect TLR to perform better than when TLR is not used. The first simulation case study is when packets are forwarded towards the destination based only on the destination location. Every intermediate node forwards the packet to its neighbour, which reduces the distance to the final destination. This is performed until the destination is reached.

In the second simulation study case, TLR is used. Every terminode keeps a list of its TLR-reachable destinations. Similar to the first case, the packet is forwarded to the neighbour closer to the destination. But, when some intermediate node finds that the destination is TLR-reachable, it uses the TLR to send the packet to the destination. Note that TLR is used in two-hop neighborhood and does not need additional routing overhead compared to the case when TLR is not used. The only requirement for TLR is that all terminodes keep in their routing tables information not only about immediate neighbours, but also about their two-hop neighbours.

We assume that the source can not know an exact destination location all the times. 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.

In our simulations we used network of 600 terminodes. The simulation area is a rectangle of the size 5400m X 1000m. The simulated network is quite dense; in this case we verified that geodesic packet forwarding is working well. It is very rare situation where packets are stuck at some node because it has no closer neighbour to the destination. We simulated 32 CBR traffic flows. Each CBR flow sends at 2Kbps and uses 256-byte packets. Terminodes are moving according to 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 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 a maximum speed is 20m/s and the pause time is 10s. In the performed simulations, we verified that the average end-to-end of data packets in less than 1s. The two study cases (geodesic packet forwarding with TLR and without TLR) are evaluated according to packet delivery fraction. This is the ratio of the data packets delivered to the destinations to data packets generated by the CBR sources. We evaluate this metric under various location information lifetimes. We simulated five different randomly generated motion patterns. Figure 6 presents an average of packet delivery fraction for five simulation runs. This figure shows that for smaller location information lifetimes (less than 20s), the packet delivery fraction is similar with TLR and without TLR. However,

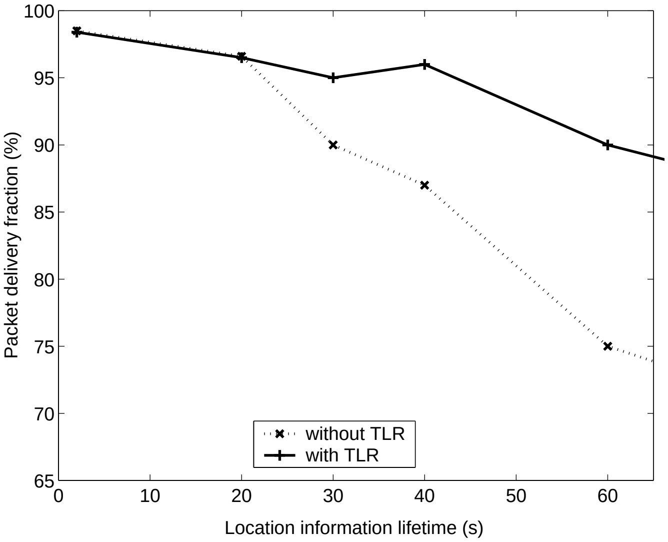

Figure 6. Demonstration of TLR utility

for higher location information lifetimes (lower precision of location information) routing with TLR gives better delivery fraction than without TLR. Therefore, we conclude that when using TLR in addition to geodesic packet forwarding, routing is more robust in the case of positional errors and inconsistent location information.

Moreover, TLR can be the only possible packet forwarding method in small ad-hoc networks, where it could be difficult to obtain location information. In this case all nodes are TLR-reachable, and geographical positioning is not needed.

4.2. Evaluation of TRR

We analyze AGPF by comparing a performance of TRR when AGPF is used and when AGPF is not used.

Before presenting the simulation and simulations results, we introduce the new mobility model called "restricted random waypoint" that is used in evaluation of TRR.

Mobility Model

In the most recent papers about mobile ad-hoc networks simulations, nodes in the simulation move according to the "random waypoint" model as described in Section 4.1.

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 for each movement a terminode 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 have 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 model.

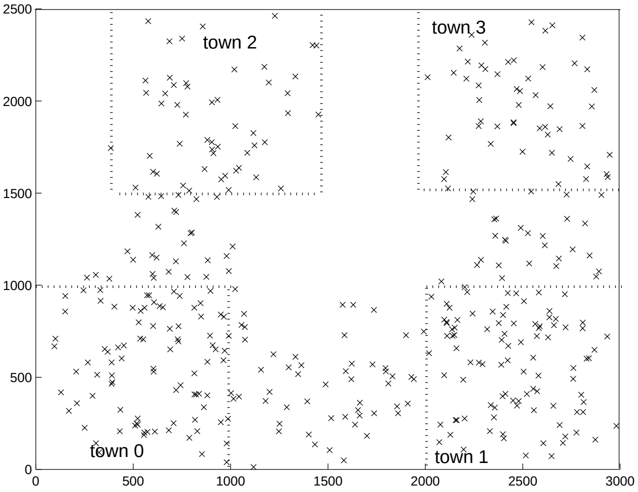

Figure 7. Model of the simulation area with four towns

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 move with the random waypoint mobility model. After a certain 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. The model of the simulated area that consists of four towns is presented in Figure 7. In our simulations, we define inside the configuration file, the pairs of towns that are connected by highways. For example in Figure 7 those pairs are (town 0, town 1), (town 0, town 2), and (town 1, town 3). This information is used by terminodes when they move from one town to another.

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

At the beginning of the simulation, terminodes are placed at randomly chosen locations inside one of four towns. An "ordinary terminode" begins the simulation by selecting at random one destination inside the town where it is placed. Then it moves to that destination at the speed distributed uniformly between 0 and some maximum speed. Upon reaching that destination, the ordinary terminode pauses for the *pause time*, selects another destination within the same town, and proceeds as previously described. Thus, the ordinary terminode's 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 terminode selects at random a destination within a new town and move 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 terminode applies inside the new town 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. The 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 pause time that is smaller than ordinary terminodes' pause time. Then it selects at random another town (such that is connected with the current town) and the random destination inside the chosen town, and moves to this destination. It pauses in the new town for a small interval of time and then again moves to another town.

An example of a network that comes out when terminodes are moving according to the restricted random waypoint mobility model is presented in Figure 7. In such a network, not all town areas are connected with a highway (e.g, town 2 and town 3).

Scenario Characteristics

A source terminode normally tries to acquire several anchored paths to the destination of interest by means of FAPD. In our simulations, we do not implement FAPD.

In order to obtain anchored paths without using FAPD, in our simulation model based on "towns and highways", we assume that a high level geographical view of the network is available at every terminode. This means that each terminode has a knowledge of a *map* of towns. A map defines town areas and existence of highways between towns. Thus, for example, the map of towns presented in Figure 7 defines those towns that are directly connected by highways as well as towns' areas. In our simulations, a town area is a square around the town center with the given width around the town center.

When source S has some data to send to destination D, S first determines the "destination town" (DT). This is the town in which area D's location falls. If D is not inside any town area, then the destination town is the town whose center is closest to D's location. Similarly, S determines the "source town" (ST), the town where S is situated, or the closest town to S if S is on the highway.

Once S determines DT and ST, S contacts the map of towns to check if DT and ST are the same, or they are directly connected with a highway. If so, then geodesic packet forwarding (GPF) towards D has a good chance of working. Then, S does not add to the packet an achored path and S sends a packet using GPF.

Otherwise, if there is no highway from ST to DT, S finds out from the map, those town areas that a packet has to pass in order to reach D. Then S adds to the packet the anchor path. This anchored path is given by a list of centers of towns that the packet has to pass. Then S starts AGPF in order to deliver the packet.

For example, it S is in the area of town 2 and D is the area of town 3, then anchors on the anchored path are centers of town 0 and town 1. In this case, AGPF works as follows: the packet is first forwarded in the direction of the first anchor (the center of town 0). Once the packet arrives at some terminode that finds that the first anchor falls within its transmission range, the packet is then forwarded in the direction of the second anchor (the center of town 1). As before, when a packet comes to a terminode that is close to the second anchor, the packet is then forwarded in the direction of D's location.

Assuming that there are terminodes to ensure network connectivity in town areas and on highways, the packet is forwarded with AGPF mostly in the *greedy* way: packets are forwarded to terminodes that are always progressively closer to an anchor point or the destination. If however, occasionally there are regions of the network where such a greedy path does not exist (i.e, it is required that the packet moves temporarily farther away from an anchor or destination), we use the approach proposed in GPSR[8]. With this approach a packet is forwarded in *perimeter mode*: a packet traverses successively closer faces on a planar subgraph of the full network connectivity graph, until reaching a node closer to an anchor or the destination and then greedy forwarding resumes.

Given the simplicity of the network topology based on four towns, in our simulations we do not use multipath routing. We only have one path from source to destination. Thus, in our simulations, TRR uses GPF towards D if ST and DT are the same or they are connected by a highway. Otherwise, AGPF is used.

In order to assess the relevance of AGPF, we use simulations to evaluate packet forwarding in two cases: the first case corresponds to when both GPF and AGPF are used, while the second case is when only GPF towards the destination is used. The only difference between GPSR and GPF is as follows: GPSR uses the destination location for making packet forwarding decisions for the whole way until the packet arrives at the destination; with GPF, an intermediate node switches to TLR if the destination is TLR-reachable.

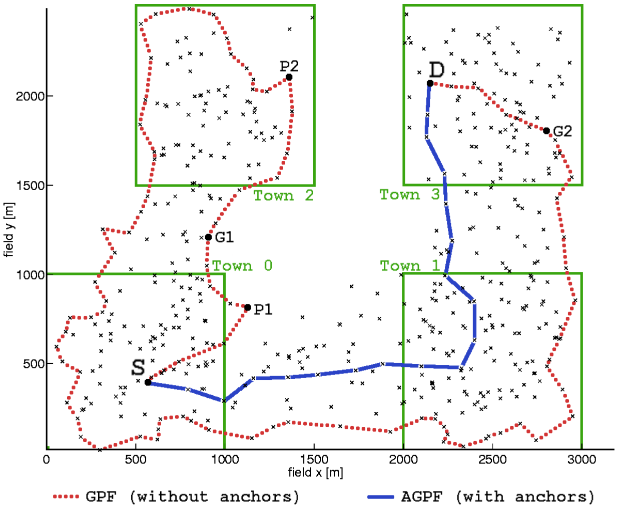

Figure 8. Figure presents the path of the packet from source S to destination D in case of two routing protocols: GPF that does not use anchors and AGPF with anchors. AGPF gives shorter path than GPF.

We illustrate packet forwarding when both GPF and AGPF are used, and when only GPF is used in the example presented in Figure 8. Here, S is in town 0 and D is town 3. In the case of AGPF, S sets the anchored path to consist of one anchor: center of town 1. AGPF 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 by using GPF. Packet forwarding is almost always in greedy mode; however, there are cases where perimeter mode is used for a very few (2-3) hops, before greedy forwarding resumes.

Geodesic packet forwarding (GPF) without anchors uses a much longer path across town 2. Figure 8 illustrates that the packet is first forwarded in the greedy mode toward D until it reaches terminode $P1$, where perimeter mode starts because $P1$ does not have neighbour closer than itself to D. The packet is thus forwarded in perimeter mode until greedy mode resumes at node $G1$ ($G1$ that is closer to D than $P1$). At $P2$, packet forwarding starts again perimeter mode. In this mode, a packet is forwarded from town 2 back to town 0, and from there through town 1 and town 3. Finally, when the packet arrives at $G2$, which is closer to D then $P2$ (where perimeter mode is started), greedy mode resumes until the packet is received by D.

Figure 8 clearly illustrates the case where usage of anchors gives shorter paths than when anchors are not used.

Simulation Results

In order to evaluate TRR, we conducted simulations of 500 terminodes forming an ad-hoc network. The size of the simulated area is 3000m x 2500m.

Terminodes move between 4 towns inside the simulation area (see Figure 7). Centers of four towns have coordinates: (550 m, 550 m), (2500 m, 500 m), (1000 m, 2000 m) and (2500 m, 2000 m) respectively. The town area is a square around the town center with the width of 500 m from the town center. There are 200 ordinary terminodes and 300 commuters.

The mobility model is the restricted waypoint mobility model. An ordinary terminode begins its journey from a random location inside the random town. As described above, it moves *stay_in_town* times inside the same town and then selects another random town to move. For each movement, a terminode takes a random speed that is uniformly distributed between 0-20m/s; before each movement, a terminode pauses for some pause time. We ran simulations with different pause times and different *stay_in_town* parameter of ordinary terminode. These parameters define different degrees of ordinary terminode mobility. A longer pause time means that ordinary terminodes are less mobile. For a fixed pause time, a larger *stay_in_town* means that a terminode is staying longer within a geographical region that corresponds to a single town. We consider different mobility rates of ordinary terminodes because this is set of nodes where all traffic sources and destinations come from. In our simulations, commuters are moving faster than ordinary terminodes. For their movements they take a random speed that is uniformly distributed between 10-20m/s. The pause time for commuter terminodes is equal to 1 second; once they reach a town, another random destination inside a different town is chosen for the subsequent movement. We checked by simulations that if terminodes move following restricted the random waypoint mobility model, there is good network connectivity (Figure 7).

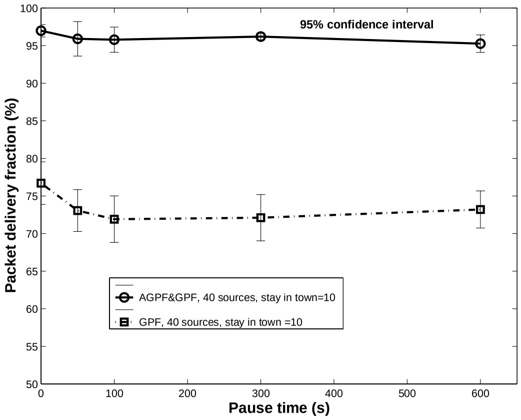

Figure 9. Packet Delivery Fraction with 40 sources; stay_in_town parameter is 10

Traffic sources are continuous bit rate (CBR). The source-destination pairs are spread randomly over the network. All CBR sources send at 2kbps, and uses 256-byte packets. All communication patterns are peer-to-peer, and CBR connections are started at times uniformly distributed between 400 and 500 seconds, and they last until the end of simulation. All simulations last for 1200 seconds. All source destination pairs are chosen from the group of ordinary terminodes.

We carried out a performance study of two routing protocols for the network in Figure 7. The first protocol is TRR with both elements: *geodesic packet forwarding (GPF)* and *anchored geodesic packet forwarding (AGPF)*. The second protocol uses only GPF.

Below we present the simulation results for *packet delivery fraction*. Each data point presents an average of at least five simulations with identical traffic models, but different randomly generated movements patterns. We evaluated the two protocols by varying mobility and traffic load levels. In order to examine the performance of the routing protocol under different degrees of congestion, we varied the number of CBR sources in the network.

The first set of experiments (Figure 9) shows packet delivery fraction when there are 40 CBR

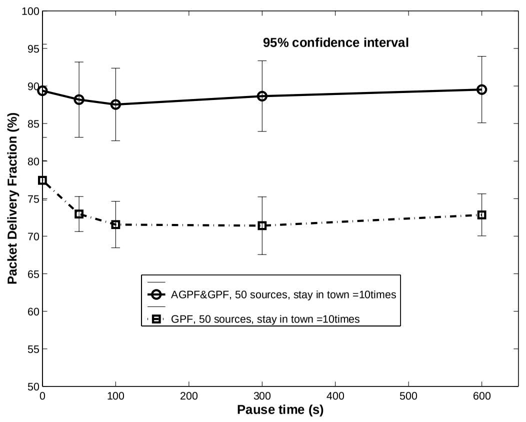

Figure 10. Packet Delivery Fraction with 50 sources; stay_in_town parameter is 10

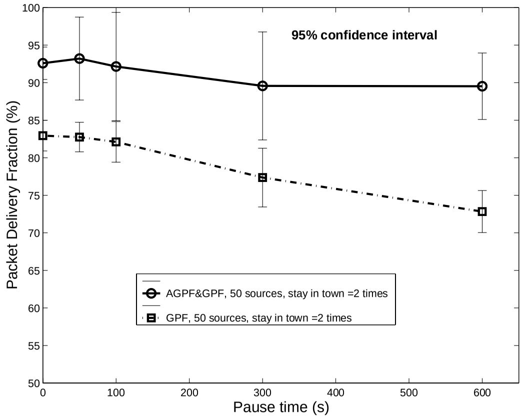

Figure 11. Packet Delivery Fraction with 50 sources; stay_in_town parameter is 2

sources. The *stay_in_town* parameter is set to 10 for ordinary terminodes (CBR sources and sinks). Different degrees of mobility are obtained for different pause times of ordinary terminodes. For higher pause times, since *stay_in_town* is high, ordinary terminodes for most of the simulation time move inside the same town area. For smaller pause times they perform moving to different town areas more frequently. Figure 9 shows that the combination of GPF and AGPF delivers about 20 percent more packets compared to the case where only GPF is used. This result is explained as follows. GPF can give complex and long paths for those source-destination pairs that are situated in towns not connected with a highway (Figure 8). For those packets there is a higher probability to be dropped. Keeping in mind that in this experiment CBR sources and destinations do not frequently change town areas, there are several flows where GPF loses most of the packets, while there AGPF have a higher success.

We have also observed end-to-end delays for the two routing protocols. It includes all possible delays: queuing at the interface queue, retransmission delays at the MAC, and the propagation and transfer times. However, we found unfair to compare average delay of packets delivered by GPF to the ones delivered by AGPF. The reason is that GPF has lower packet delivery fraction than AGPF and then average delay counts only for delivered packets. We observed that with GPF 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 consequent short delays.

A packet delivery fraction of the second set of experiments is presented in Figure 10. It differs from the previous example in that the number of CBR sources in the network is increased to 50. We observe that the combination of AGPF and GPF again delivers more packets than when only GPF is used. However, AGPF decreases its delivery fraction compared to the previous case when there are 40 CBR sources. Indeed in this experiment, we observed an increased level of congestion in the region of the network between town 0 and town 1. Since AGPF directs most of its anchored paths across this region, that result in congestion and increased packet drops. GPF performs as in the previous experiment with smaller number of sources. We explain this on the example in Figure 8. Unlike AGPF, GPF does not directly forward the packet from S in direction of town 1. Firstly, GPF forwards it to town 2 and than back to town 0 where the packet is forwarded in direction of town 1. Provided that the packet is not lost during the journey between towns 0 and 2, it contributes in the congestion along the highway between town 0 and town 1. However, in our simulations we have observed that there are many packets that are lost before taking this highway. Thus this explains why GPF is less susceptible to the increased number of CBR sources than AGPF.

Here is where multipath routing for AGPF would be beneficial. In our simple network topology based on four towns, AGPF uses only one anchored path to the destination. If, however, there were several paths over which packets can be sent, this would result in load balancing. It is left for future work to investigate mechanisms for choosing routes in the network so that data traffic is more evenly distributed in the network.

The last set of experiments is presented in Figure 11. Here there are 50 CBR sources, but ordinary terminodes are moving more frequently from one town area to another. Here *stay_in_town* parameter is set to 2. We observe that GPF delivers more packets than in the previous two experiments. This can be explained: with increased mobility, those source-destination pairs for which GPF gives a small fraction delivery in the previous two cases can move to towns where GPF gives a better path. In this way bad situations, where GPF gives long complex paths, do not last for the whole duration of the simulation. This is especially true for lower pause times. For higher pause times, again we observe GPF decreases in the packet delivery fraction.

We conclude that AGPF results in higher packet delivery fraction than GPF in all our experiments. However, we observe that the improvement of AGPF over GPF is more important when nodes stay in single town areas or close to these areas for most of the simulation time. Keeping in mind that AGPF is intended for large area mobile ad-hoc networks, we believe that this assumption would there be satisfied. We performed our simulations for a relatively small simulation area and small number of nodes. We believe that within a larger area, the benefits of good anchored paths over complex, long GPF paths will be more evident. Anchors define a rough shape of a path from the source to the destination. The source should monitor all anchored paths it is using, and react if the value of a path is deteriorated.

Finally, we discuss about *routing overhead* of the terminodes routing. Routing overhead is the total number of routing packets transmitted during the simulation. Terminodes routing generates two types of protocol packets. TLR uses HELLO messages, whereas 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 the 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, the size of the network does not have an effect on TLR overhead per node, since HELLO messages are not propagated beyond a single hop.

One possible optimization to reduce the HELLO message overhead is that nodes that have some data to forward defer the sending of HELLO messages. Then a sender piggybacks in every data packet the information it would send via a HELLO message. This is possible when the network interface is used in a promiscuous mode, and a node receives all packets from all terminodes within its transmission range. This optimization has not been implemented so far.

Since, for our simulations we have not implemented FAPD, we have not evaluated FAPD overhead. This is task left for further work.

5. Conclusions

We focused on the problem of routing in a wide area mobile ad-hoc network called Terminode Network. Routing in this network is designed with the following objectives. First, it should scale well in terms of the number of nodes and geographical coverage. Second, routing should have scalable mechanisms that cope with load balancing and 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 the 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 us to reduce problems of loops due to location inaccuracy. Anchor Geodesic Packet Forwarding (AGPF) is a component of TRR that provides paths when there are holes in terminodes distribution and the source can not reach the destination over the direct geodesic path. We performed simulations to assess relevance and performance improvements when TLR and AGPF are used. In our simulations, we introduced the topology based on four towns where nodes move between towns according to the mobility model that we called "restricted random waypoint". Our simulation results demonstrate improvements obtained with TLR and AGPF over GPSR that uses geographical information for packet forwarding.

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.

[20] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks *Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DialM'99), 1999*

[21] S. Datta, I. Stojmenovic and J. Wu. Internal node and shortcut based routing with guaranteed delivery in wireless networks *Proc. IEEE Int. Conf. on Distributed Computing and Systems (Wireless Networks and Mobile Computing Workshop), Phoenix, AR, April, 2001*

[22] R. Jain, A. Puri and R. Sengupta. Geographical Routing Using Partial Information for Wireless Ad Hoc Networks, *IEEE Personal Communications, February 2001*