ISSN: 2321-7782 (Online)
Volume 3, Issue 4, April 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Terminode Routing

Abhijeet M. Haval[1]
Assistant Professor
Computer Science and Engineering
Dhananjay Mahadik Group of Institutions
VikasWadi, Kagal, Kolhapur Maharshtra
India

Uday D. Patil[2]
Assistant Professor
Computer Science and Engineering
Dhananjay Mahadik Group of Institutions
VikasWadi, Kagal, Kolhapur Maharshtra
India

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,locals called Terminode Local Routing (TLR) and Terminode Remote Routing (TRR). TLRand 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 aprotocols 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 than10%.

I. 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 term-in ale 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 network consisting of up to several hundreds of nodes.

Each terminode Las a permanent End-system Unique Identifier (EUI), and a temporary, location-dependent address (IDA). Tie LDA h simply a triplet of geographic coordinates (longitude, latitude, altitude) obtained, for example, by means of the Global Positioning Eastern (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 terminode contents among other fields, the destination LDA and EUI and possibly source routing information, as mentioned later. Mobility management in a terminode network is based on the distributed location database, allows a terminode*A* to obtain a probable location of terminode*B (LDAB)* that A is nottracking by the previous method. Mobility management is out of scope of this paper. (see for example [7][8])

Our assumption that we can get with the mobility management the destination LDA with precision of approximately one transmission range and validity of about 10 seconds.

We also assume that multipath routing isacceptable for the transport protocol however, with the current TCP this is not acceptable since then: areproblems with managing a largeis 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) andin the vicinity of a terminode and does not use location information for making packet for warding decisions in contrast TRR is use to sent data to remote destination and uses geographic information it is the key element for achieving and reduced dependence on intermediate systems.

TRR consists of the following elements:

Anchored Geodesic Packet Forwarding (AGPF) ~n a method that allows for data to be sent to remote terminodes AGPF is solely based on locations. AGPF sent data among the anchored path . An anchored path defines a rough shape of the path from the source to destination and is given with the list of anchors. Anchors are points described by geographical coordinates and do not in general correspond to any terminodes location a good anchored path should avoid obstacles and terminode "deserts[1] from the source to the destination. Between anchors geodesic packet forwarding is performed; this is a destination.

» *Friend Assisted Path Discovery (AGPF)* is the path discovery method used to obtain anchor path . A terminode keeps a list of other terminodes that it calls friends, to which it maintains, one or several good path[S]. In FAPD. aterminode may contact itsfriends 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.

II. TERMINODE ROUTING

2.1 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 (TOR), 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&)* is obtained either using the LDA management scheme (described in [Y]) 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. AH acquired paths are maintained by a terminode by the path maintenance method. The global routing method is presented in pseudo code 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 lias 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 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(LDAd,LDAx)* 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 *mm(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.

2.2 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 pseudo code.

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 *3* 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 liiglily mobile ad-hoc network.

2.3 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.

2.3.1 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 *Fl*in a friends list to start *FAPD) S* sets "F" bit in the packet and sends a packet to Fl ; //fig.3 S user geodesic pkd. Forwarding to D
Fig (a)
Terminode*X* receives a packet for destination *D:* if(("use-TLR" bit is set)or(D is in *TLR-reachable region oi X)* X applies *TLR;* //fig. 2 (a) else if (dist(X,D) < transmission range of *X)* JTTL =*mm(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
Fig(b)

Figure 1: (a) Packet forwarding algorithm at the source,(b) Packetforwarding 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)   Fig(a)
X receives a packet for D with anchored path: P/=get anchor from a path;   *(API* is in not within a transmission range of *X)* uses geodesic packet forwarding to *API;*   *else{*   delete *API* from the path;   *P2*= get next anchor from the anchored path; *(AP2* == NULL)   use geodesic pkt. forwarding to *D;*   else use geodesic pkt. forwarding to *AP2*

Figure 2: (a) TerminodeLocal 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[1]' 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 neighbor 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.

2.3.2 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 pseudo code.

A source terminode adds to the packet a route vector (anchored path) made of a list of anchors, winch 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[&]. Figure 3 presents an example of AGPF.

2.3.3 Path Discovery

Friend Assisted Path Discovery (FAPD) is a path discovery method that is based on the concept of small world graphs [l8]. 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 acquaintanceship 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 well.

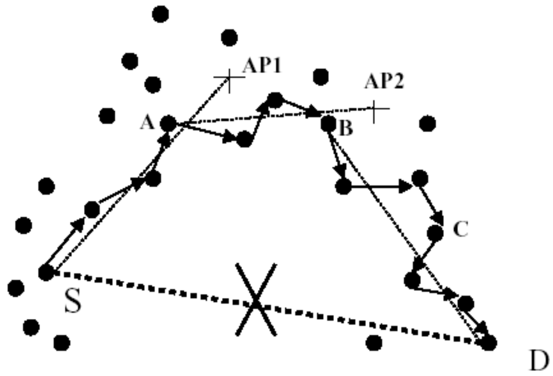

Figure 3: The figure presents how AGPF works when a terminode with *EUIs* has some data to send to a terminode with EUI_D. and there is no connectivity along the shortest line from *S* to *D. 3* has a path to *D* given by a list of geographical locations called anchors: {API. AP2}. First, geodesic packet forwarding in the direction of API is used. After some hops the packet

arrives at a terminode A which finds that API 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.

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 pseudo code.

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. AH friends whose distance to destination is 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.

```
Flis intended receiver of a path discovery packet f'F" bit is set): S needs a path to D
else if (Flhas a path to D)
append this path infapd_auchoi-ed_pntfiand send the packet to D:
else if (packet in taboo-mode)
 {
        if(Flhas a friend F2 where dist(F2, D) <min_dist)
{exit taboo-mode; !aboo-index=Q; send the packet to F2]
 else if (taboo-index < 2 and Flhas a friend F3 such that distfFt, F3) <max_dist|
{ taboo-index++; send a packet to F3] else // taboo-index reached the maximum value
else //packet not in taboo mode
{
if (Flhas a friend F2 where dist(F2,D)<dist(Fl,D) )
send a packet to F2;
else if (Flhas a friend F3 such that dist(Fl, F3) <max_dist)
{start taboo mode; taboo_iadex^\; min_dist=dist(Fl,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 Fl that is closer to D than any other friend of *S*. *S*sends data packet to Fl according to the existing path that *S*maintains to Fl. In order that Fl 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 Fl, *S*would append anchors of this path in *fapd-anchored-path* field of the path discovery packet; *S* sends data to Fl by AGPF. Otherwise, *S* leaves this field empty and uses geodesic packet forwarding to Fl. Upon reception of the path discovery packet, Fl puts its geographical location inside the *fapd-anchored-path* field as one anchor. If Fl has an anchored path to *D*, Fl appends this path into *fapd-anchored-path* field and send the packet to *D* by AGPF. Suppose now that Fl does not have a path to *D*, but has a friend F2 whose distance to *D*is smaller than the distance from Fl to *D*. If Fl has an anchored path to F2, Fl appends it in the *fapd-anchored-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 *3* to *D*. Assuming that the path from *3* to Fl and from Fl to F2 does not contain any anchors, *fapd-anchored-path* is thus a list of anchors *(*$LDA_{F}i$ $LDA_{F}2$*)*.This list is sent from *D* to *3* by applying AGPF with the anchored path *(LDAp*2,*LDAfi)* (that is a reversed path from the one *D* received within the path discovery packet).

Once, *S*receives from *D* a packet with the acquired anchored path, *3* stores this path in its route cache. *S* can send subsequent packets to *D*, by applying AGPF with the acquired anchored path. Otherwise, if *3* has not received the anchored path during some time, or if *S* Needs more paths

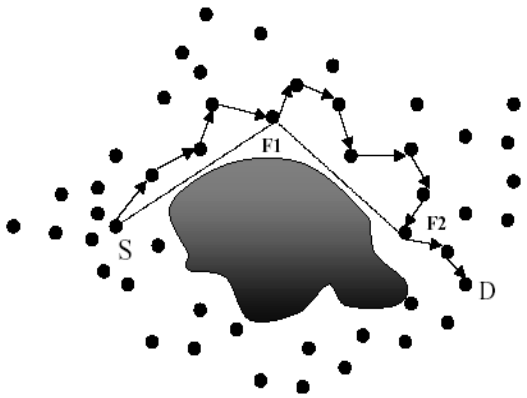

Figure 5: The figure presents how FAPD works when source *S*, has a friend *F1*that is closer to *D* than *S*. *S* sends data packet to *F1*and sets "F bit in the packet header in order to denote that this is a "path discovery packet'. Upon reception of the path discovery packet. *Fl*puts its geographical location inside the *fapd-anchored-path* field of the path discovery packet as one anchor. In this example *Fl*does not have path to *D*, but has a friend *F2* whose distance to D is smaller than the distance from *Fl*to *D*. *Fl*sends path discovery packet to *Fl*. 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 *(*$LDA_{F}1$,LDA_{F2}*)*.

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 Figure6. Here, *3* 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 *3.* 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 *Fl*that is farther from *D,* but such that *dist(S,Fl) < max-dist. S* sends the path discovery packet with *"F"* bit set to its friend *Fl.* 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. 3* 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, *Fl*finds out that it does not have a friend whose distance to *D*is smaller than *min-dist. Fl*forwards the path discovery packet to its friend *F2* that is farther from *D* but such that *dist(Fl, 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

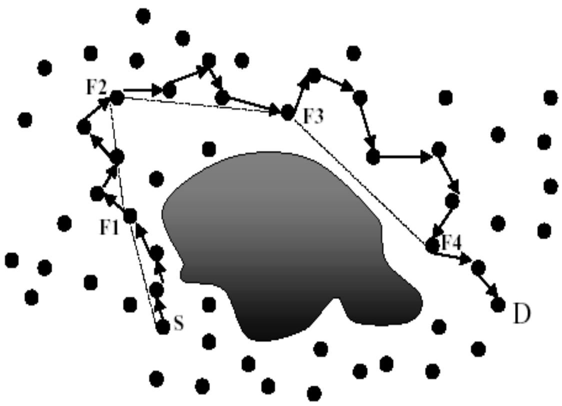

Figure 6: The figure presents how FAPD works when the source *S* does not have a friend that is closer to *D* than itself. *S* contacts its friend *F1*that is farther from *D* in geometrical distance than *S* is, but such that *dist(S,* F1) <*max-dist.* As in the previous example. *S* sends data packet to F1with " 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. Flfinds out that it does not have a friend whose distance to D is smaller than *min-dist.* Fl forwards the path discovery packet to its friend F2 where *dist(Fl,* 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. andF2 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 *mm-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 *F*2 does not have a closer friend, F2 would forward the packet to *D* by geodesic packet forwarding.

2.3.4 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.

III. Proposed System Performance Evaluation Of Terminode Routing

SIMULATION SETUP

Global Settings

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

Protocol Constants

We used the following configuration for the Helloing protocol. The HELLO timer is 1 second. Each entry in the routing table expires after two seconds, if it is not updated. All nodes promiscuously listen to all HELLO messages within their radio range. Nodes that have data or control packets to send should defer sending HELLO messages (up to the timer value) and piggyback the HELLO message to the data or control packet. For FAPD, the parameter values are: refresh_friends=50s, path_validity=20s, and wait_for_path=10s.

Location Management

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

Location Discovery

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

Location Tracking

Once two nodes begin to communicate, location tracking is used: data packets periodically (every 5 seconds) piggyback the local location of the sending node. If no data packet is to be sent, a node periodically sends a location reply control message with its location information. The destination location is considered stale if not refreshed for more than 10 seconds in small networks, 20s in large networks. The source then reinitiates learning of the destination location. The source does not flood the network, but uses the last known destination location to reduce the search space for the destination. Similarly to LAR1, location request is flooded only in the expected rectangular region of the destination. If available, anchored paths are used to facilitate location management operation. In this case, if the source sends data to the destination using anchored paths, the destination sends back to the source its location updates using the reversed path. Idealized no-overhead location management. For completeness, in simulations of large networks, we separate the cost of location management and location-based routing. We do this by also simulating a hypothetical no-overhead location management. In this idealized scenario, we assume that sources use location information with a lifetime of 5 seconds (which causes some location inaccuracies).

Common Simulation Parameters

Sources are constant bit rate (CBR) and send two packets of 64 bytes per second. There are 40 source destination pairs, chosen randomly (uniformly) over the set of nodes. With these values, the network is not congested because we want to measure routing protocol behavior, not the limitation of the IEEE 802.11 MAC for data packet capacity. CBR connections are started at times uniformly distributed between 400 and 500 seconds. Simulations run for 1,200 simulated seconds of simulated time. Measurements start after 300s of simulated time (small networks), or 600s (large networks). Each data point represents an average of six runs with identical traffic models, but different randomly generated mobility scenarios.

Environment /Tools required.

Platform used : WINDOWS XP Operating System.

Tools used : Network Simulator (GLOMOSIM)

Simulation parameters:

» size of the network

» node distributions (uniform and non-uniform)

» mobility level

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 CER 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-m-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 ofless 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.

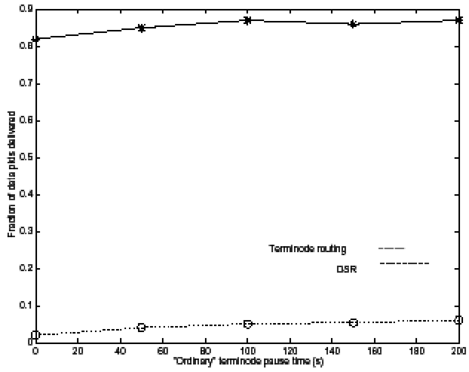

Figure 8: Packet delivery success rate of terminode routing and DSR with varying "ordinary"

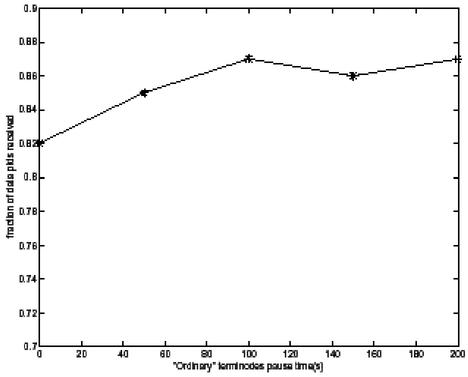

Figure 9: Packet delivery success rate of terminoderouting with varying "ordinary" terminodes' pause time. 600 nodes (100 "ordinary" and 500 "commuter" terminodes)

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 forFAPD. 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 (or 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[51] can be piggybacked in the data packet. Thus, the overhead due to FAPD control messages is not big.

IV. 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 CTRR). 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 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. PAR-SEC: 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 200O/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 {Mo-biCom'98),Dallas, Texas, USA, August 1998.
5. SrdjanCapkun, Maher Hamdi, and Jean-Pierre Hubaux. GPS-free positioning in mobile Ad-Hoc networks. Proceedings of the 34th HIGSS, January 2001.
6. TRRR 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'OO),,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'OO}, August 2000
9. J. Li, J. Jannotti, D. De Couto, D. Kaiger, and R. Morris. A Scalable Location Service for Geographic Ad Hoc Routing. Mobicom'Off, 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. Coraon. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. Proceedings of IEEE INFOCOMW, 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 Completing (MobiHOC), August 2000.
15. Elizabeth M. Rjoyer 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 imd 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. Watte. In Small Worlds, The dynamics of networks between order and randomness. Princeton University Press, 1999.

19. Ko Y. and Vaidya N. Location-aided muting in mobile ad-hoc networks. Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Compiling and Networking (MobiCom'ffS), Dallas, Texas, USA, August 1998.