Self Organized Routing in Wide Area Mobile Ad-Hoc Networks

Ljubica Blažević, Silvia Giordano and Jean-Yves Le Boudec
Institute for computer Communications and Applications (ICA)
Swiss Federal Institute of Technology, Lausanne

***Abstract*- This paper considers the problem of routing in a wide area mobile ad-hoc network called Terminode Network. 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 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. The use of TRR results in a scalable solution that reduces dependence on the intermediate systems, while TLR allows to reduce problems due to location inaccuracy. The paper describes TLR and TRR protocols and presents some simulation results.**

I. Introduction

We focus on the problem of routing in a large mobile ad-hoc network, that we call $terminode$ network. Nodes in this network are called $terminodes$, because they act as network nodes and terminals at the same time. 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).

The routing scheme is 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. A detailed description of TLR and TRR can be found in [4]. 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 [5]).

Many routing protocols have been proposed for mobile ad hoc networks. A recent overview of a family of protocols that do not use location information can be found in [7]. There is also a family of position-based routing algorithms. These protocols use location information in order to reduce propagation of control messages (LAR [9]), to reduce intermediate system functions or for making packet forwarding decisions (GPSR [3], GRA [6]).

In GPSR the routing is done in a greedy way by forwarding the packet closer to the destination, and when this fails, it tries to routes around the region where greedy approach failed (perimeter approach).

The rest of this paper is organized as follows: in Sections II and III we describe TLR and TRR . This is followed by Section IV that gives the description of performed simulations.

II. Terminode Local Routing (TLR)

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 distance in hops from S is at most equal to a *local radius*. The local radius is a measure, in number of hops, of the TLR-reachable area.

Every terminode proactively maintains the information (EUI, LDA) of the terminodes that are in its TLR-reachable area. TLR uses a distance vector routing protocol to send data to TLR-reachable destinations, and the only addressing information used for packet forwarding is the EUI of the destination.

In the current implementation of TLR, the *local radius* is set to two hops. 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.

III. Terminode Remote Routing (TRR)

Terminode Remote Routing (TRR) allows data to be sent to *non TLR-reachable* destinations. TRR consists of the following elements:

A. Anchored Geodesic Packet Forwarding (AGPF)

AGPF is a method to send data to remote destinations. Unlike TLR, AGPF is solely based on locations.

When S has some data for D and D is *non TLR-reachable*, S first discovers D's location (LDA_D) using a mobility management method.

Then, S sends packets by geodesic packet forwarding: the packet is sent to some neighbor X within a transmission range where the distance to D is reduced the most. At its turn, X performs the same steps: it checks whether D is TLR-reachable. If this is not the case, X sends the packet by geodesic packet forwarding. Otherwise, X uses TLR. Switching from location-based routing to TLR can reduce problems that may occur if D has moved considerably from LDA_D.

In this simplest form, geodesic packet forwarding will often not work. If there is no connectivity along the shortest line, then the method fails.

Our solution to this problem is to use *anchors*. An anchor is a point, described by geographical coordinates; it does not, in general, correspond to any terminode location. Anchors are computed by source nodes, using the path discovery methods (presented further in the paper). A source terminode adds to the packet a route vector made of a list of anchors, which is used as loose source routing information. Between anchors, geodesic packet forwarding is employed. When a

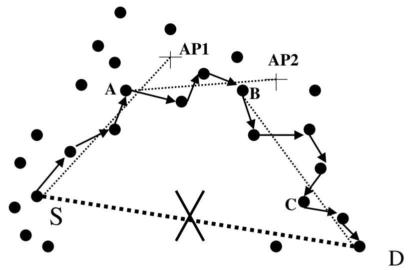

Fig. 1. The figure presents how AGPF works from S to D when there is no connectivity along the shortest line from S to D. S has a path to D given by a list of geographical locations called anchored points: $\{AP1, AP2\}$. First, geodesic packet forwarding in the direction of $AP1$ is used. After some hops the packet arrives at a terminode A which finds that it is close to $AP1$. 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.

relaying terminode receives a packet with a route vector (list of anchored points), it checks whether it is close to the first anchor in the list. 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. Figure 1 presents an example of AGPF.

B. Anchored Path Discovery

There exist two methodes to obtain anchored paths: *Friend Assisted Path Discovery (FAPD)* and *Geographical Maps-based Path Discovery (GMPD)*.

In FAPD, each terminode keeps a list of other terminodes, that it calls friends, to which it maintains one or several good path(s). With this scheme, a terminode may contact its friends in order to find an anchored path to the destination of interest. FAPD is contains two elements: Friends Assisted Path Discovery Protocol (FAPDP) and Friends Management(FM).

B.1 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).

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. "F" bit is not reset before reaching D. 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 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. We use max_dist equal to five times the transmission range of a terminode. 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. We illustrate FAPDP with an example (see Figure 2).

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.

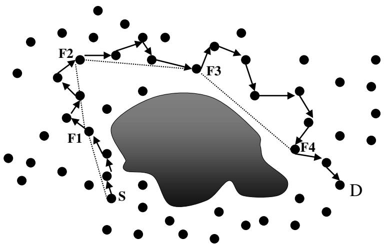

Fig. 2. The figure presents how FAPDP works when source S does not have a friend that is closer to D than itself. S contacts its friend $F1$ that is farther from D in geometrical distance than S is, but such that $dist(S, F1) < max\_dist$. In addition S sets the $tabu\_index$ field to 1 and thus starts 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. In our example, $F2$ has a friend $F3$ whose distance to D is smaller than min_dist and forwards the packet to it. 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 $(LDA_{F1}, LDA_{F2}, LDA_{F3}, LDA_{F4})$

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, because 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.

B.2 Geographical Maps-based Path Discovery (GMPD)

In GMPD, each terminode has a summarized geographical view of the network. We believe that in a large mobile network the population is not uniformly distributed. Therefore we identify the areas with higher population density, that we call $towns$. Two towns are interconnected by all the nodes in between the two towns (we call it a $highway$). Each terminode has a knowledge of a map of towns. A map defines towns in terms of a square centered in a geographical center, and reports the existence of highways between towns. In order to find an anchored path to a destination, a node accesses its map and determines the anchored path as the list of the centers of the towns that it has to visit in order to reach the destination.

IV. Performance Evaluation

We simulated the terminode routing protocol (the TLR and TRR protocols) in GloMoSim[8]. GloMoSim is a scalable simulation environment for wireless network systems. The IEEE 802.11 Medium Access Control(MAC) protocol is used; it implements the Distributed Coordination Function (DCF)[2].

This section describes how the simulations of terminode routing are performed. A terminode periodically transmits a HELLO message to the broadcast MAC address. A HELLO message contains a list of a terminode's neighbours. 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.

Restricted Random Waypoint 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 [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. 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.

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 3. In our simulations, we define inside the configuration file, the pairs of towns that are connected by highways.

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 positions 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 position, 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; thus for every movement it selects destination within a different town.

Scenario Characteristics

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

In order to implement GMPD, 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 3 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 in 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[3]. 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.

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.

We illustrate packet forwarding when both GPF and AGPF are used, and when only GPF is used in the example presented in Figure 3. Here, S is in town 0 and D is town 3. With AGPF, S sets the anchored path to consist of one anchor: center of town 1. 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.

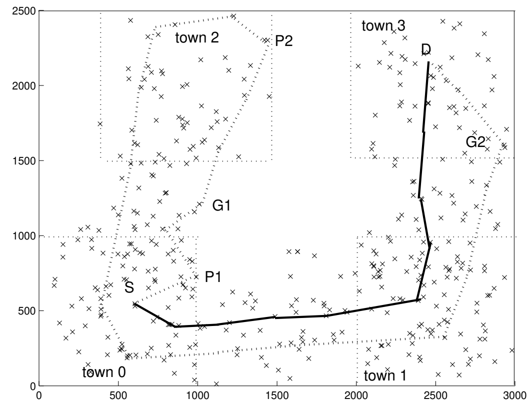

Fig. 3. Illustration of paths obtained by GPF (dotted line) and AGPF (solid line).

GPF without anchors uses a much longer path across town 2. Figure 3 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 3 clearly illustrates the case where usage of anchors gives shorter paths than when anchors are not used.

V. SIMULATION RESULTS

Implementation of TLR and TRR is done in GloMoSim. We conducted simulations of 500 terminodes forming an ad-hoc network. The size of the simulated area is 3000m x 2500m. In our experiments, radio range is the same for every terminode, and is equal to 250 meters. The channel capacity is 2Mbits/sec.

Terminodes move between 4 towns inside the simulation area according to the restricted waypoint mobility model (see Figure 3). There are 200 ordinary terminodes and 300 commuters.

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* parameters. A longer pause time means that 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. In our simulations, the pause time for commuter terminodes is equal to 1 second and *stay_in_town* is equal to 1. The *stay_in_town* parameter for ordinary terminodes is set to 2 and different degrees of mobility for ordinary terminode are obtained with different values for pause time. Traffic sources are continuous bit rate (CBR). 50 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 destina-

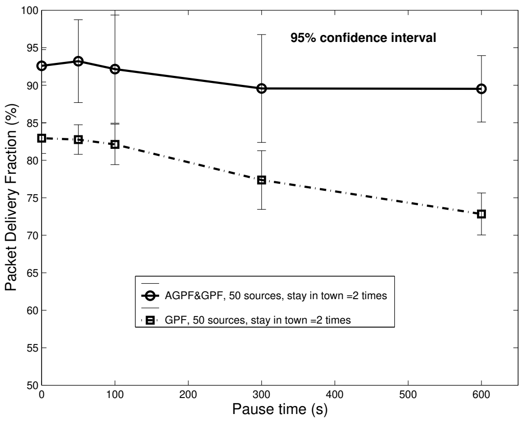

Fig. 4. Packet Delivery Fraction with 50 sources; stay_in_town parameter is 2

tion pairs are chosen from the group of ordinary terminodes.

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

Figure 4 presents the simulation result for *packet delivery fraction*. This is is the ratio between the total number of packets originated by CBR sources and the number of packets received by CBR sink at final destinations. 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 for different values of pause times.

Figure 4 demonstrates that the combination of GPF and AGPF delivers around 10 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 3). For those packets there is a higher probability to be dropped. We observe that GPF delivers more packets when mobility is higher (smaller pause times). This can be explained: with increased mobility, those source-destination pairs for which GPF gives a small fraction delivery can move to towns where GPF gives a better path. With smaller mobility, bad situations where GPF gives long complex paths last longer, and GPF does not succeed well in packet delivering.

VI. Conclusions

We focused on the problem of routing in a wide area mobile ad-hoc network called Terminode Network. Our routing scheme is a combination of two protocols called Terminode Local Routing (TLR) and Terminode Remote Routing (TRR). TRR is activated when the destination D is remote and uses location of the destination obtained either via location management or by location tracking. TLR acts when the packet gets close to the destination and uses routing tables built with hello messages. The use of TRR results in a scalable solution that reduces dependence on the intermediate systems, while TLR allows to reduce problems due to location inaccuracy.

References

[1] 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.

[2] 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.

[3] 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.

[4] L. Blazevic, S. Giordano and J.-Y. Le Boudec. Self Organized Terminode Routing. Technical report, DSC/2001/024, Swiss Federal Insitute of Techology, Lausanne.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.